Mobile vs. Desktop Programming Projects: The Effect on Students

Mohammad Reza Azadmanesh
University of Lugano
Mohammad.Reza.Azadmanesh@usi.ch

Amanj Sherwany
University of Lugano
Amanj.Sherwany@usi.ch

Davide Eynard
University of Lugano
Davide.Eynard@usi.ch

Matej Vitasek
Easycore media Ltd.
vektor330@gmail.com

Matthias Hauswirth
University of Lugano
Matthias.Hauswirth@usi.ch

Abstract

Programming projects are a common component of computer science curricula. In this paper we investigate how the decision of using a mobile and touch versus a desktop platform affected students. We ran two sets of programming projects, one developing desktop applications, the other developing mobile apps. Several months after the conclusion of the projects, we interviewed the students about their experiences. We found that our initial expectations regarding an increased student motivation were not completely met. We discuss the specific issues we uncovered, and the lessons we learned.

Keywords Programming for mobile, computer science education, qualitative study

1. Introduction

Programming mobile apps is said to be particularly motivating for students. Based on this expectation, we redesigned our second semester programming projects, moving from the development of desktop applications to the creation of mobile apps. After the conclusion of this transition, we went back and interviewed the two groups of students: those who had participated in the original desktop projects, and those who had completed the mobile app projects. The goal of our interviews was to find out how changing our course to mobile had affected our students.

Programming for Mobile Devices. We are by no means the first who introduced mobile app development projects in an undergraduate curriculum. Mahmoud and Dyer argue that in computer science education, moving towards the development of mobile apps is useful because it reflects today's reality, and because it provides the opportunity to introduce students to different programming models. Most importantly, they claim that this practical experience in mobile app development "inspires students to be excited about learning"[8], and that this could provide a "renewed interest in pursuing a computer science major". Kurkovsky proposes mobile game development as a "motivational tool to engage students" [7]. Two reasons for this increased motivation are that (1) students can easily run their applications on their mobile devices and show them off to friends, and (2) experience in mobile development is in high demand.

Programming on Mobile Devices. Besides programming *for* mobile platforms, there is also a recent trend in programming *on* mobile devices [11], specifically with languages and IDEs focusing on end-user programming like Pocket Code [4] or TouchDevelop [10]. Programming *on* mobile devices has additional advantages, for example the low barrier of entry due to the easy availability of the programming platform. Our study, however, excludes this effect, and only investigates the difference between programming *for* mobile vs. desktop platforms, where the programming happens in an IDE on the desktop.

The remainder of this paper is structured as follows. Section 2 discusses the setting in which we conducted our study. Sections 3 and 4 present our methodology and results. Section 5 presents the lessons learned from our study. Section 6 discusses threats to the validity of our findings, and Section 7 concludes.

2. Background

We conducted this study in second-semester programming projects in the context of the three-year Bachelor program in Informatics at the University of Lugano [5].

Programming Fundamentals (PF2). During the semester of the projects, the students took PF2[1] on object-oriented programming in Java. The course follows the "Objects First with Java" textbook [1] and uses the BlueJ[2] educational IDE during the first half of the course.

Software Atelier (SA2). Concurrently to PF2, the students also took SA2. This atelier-based course complements the conceptual courses by providing students with the opportunity to practice the theory in applied projects. The first part of SA2 introduces and practices graphical user interface design. The second part consists of the programming project, which connects programming practice and GUI design by building an interactive application. This project is the subject of our study.

Programming Projects. We ran the SA2 programming projects during roughly the second half of our 14-week semester. The

[1] http://boole.inf.usi.ch/pf2-2014/

[2] http://bluej.org

PROMOTO '14, October 20, 2014, Portland, OR, USA.

http://dx.doi.org/10.1145/nnnnnnn.nnnnnnn

	Desktop	Mobile
Course	SA2 2013	SA2 2014
Teams	5	9
Team sizes	3	2-4
Team formation	self-selected	self-selected
Skills in teams	mostly homogeneous	mostly homogeneous
Topic	given	self-selected
Project duration	8 weeks	8 weeks
Language	Java	Java
Toolkit	Swing	Android
Recommended IDE	NetBeans	Eclipse + ADT
Use GUI builder	no	yes

Table 1. The two Cases: Desktop vs. Mobile Projects

ID	Case	Duration	Prior PL Experience
D1	desktop	45	Python, Java
D2	desktop	21	Python
D3	desktop	43	Python, Java
D4	desktop	28	Java
M1	mobile	24	Python
M2	mobile	43	Python
M3	mobile	14	Python
M4	mobile	46	Java, C++, Matlab,Python
M5	mobile	34	Python
M6	mobile	19	Pyhton

Table 2. Participants

project is assigned 3 ECTS credits, which corresponds to roughly 75 working hours for each student. Thus, students are expected to spend about 11 hours on the project each week. Given that the atelier courses aim to foster teamwork, we designed the SA2 projects as team projects. Table 1 summarizes the two cases, the use of *desktop* vs. *mobile* projects, which we discuss further below.

Desktop Projects. The desktop projects conducted in 2013 required the development of an interactive application representing a "world" in which particle effects could be built and observed. Students would implement a simple physics simulation with particle emitters, walls, and forces. It required mastering a variety of GUI controls (e.g., windows, buttons, sliders) and concepts (drag-and-drop, pan/zoom interface). The students formed 5 groups, each consisting of 3 students. The groups were mostly homogeneous in terms of the students' skills. All groups started with the same requirements. We recommended that they use NetBeans as their IDE. In PF2 they had been using the educational BlueJ IDE, so the move to the full-fledged NetBeans was a significant change. We asked them to use Swing as the GUI toolkit, which is the GUI toolkit introduced in PF2. They programmatically constructed the GUI and did not use interactive GUI design tools. At the time the projects started, PF2 had not yet covered GUI development. Thus, we encouraged them to first tackle the non-GUI aspects of their application (the "model"), and develop the user interface after they learned about Swing in PF2. The two TAs met with each group once a week, to provide feedback on the contributions so far, and to help plan the subsequent week.

Mobile Projects. In 2014 we changed to projects targeting a mobile platform. Specifically, the applications had to run on Android. Moreover, we gave students the freedom to propose their own projects. Our intention was to provide students with a more attractive platform while allowing them to use the language they learned in PF2 (Java), and to provide them with more choice [6], two changes that we expected to increase their motivation. We came to the decision to use Android in a meeting together with the students, despite some students not owning Android phones. A total of 27 students were enrolled in SA2. They formed 9 groups of 2 to 4 people. The groups were mostly homogeneous in terms of skills. The 9 proposed projects ranged from text messaging applications to 3D car racing games.

The projects lasted 8 weeks, but were preceded by a month where students learned the basics of Android development (IDE installation and configuration, the UI components, the step-by-step development of a simple Hangman game) within SA2. At week 2 of their project, students had to show they had solved the main issue related to their project (e.g. for an NFC-based video game, they showed they could pass messages using NFC). At week 4, the alpha versions of the applications were shown in class. This allowed students to have early feedback on their apps from their peers. At week 8, a final presentation of the projects took place. Students had to show how they had dealt with the three main facets of their projects (Java development, UI, and Android), and describe their own products.

3. Methodology

Given our specific context and the small number of students available for our the study, we do not aim at *quantifying* the advantages or disadvantages of moving programming projects to mobile platforms. Our goal is different: we want to better *understand* how our move to mobile affected the students. We thus performed a *qualitative* study based on semi-structured interviews [2].

Sampling. We took a convenience sample of students taking part in the two instances of the courses, by asking for volunteers and selecting students based on their availability. We contacted and interviewed the students in August 2014, about two months after the conclusion of the mobile projects, and 14 months after conclusion of the desktop projects. We conducted interviews with 10 students who took the course. Table 2 lists the 10 participants, showing the duration of their interviews and their programming experience before the start of SA2. Most of them knew Python, because they learned it in the first semester.

Briefing. Immediately before each interview we informed the students that this interview was for a study to understand our teaching approach, without explicitly mentioning that we were studying the difference between mobile and desktop projects. Students then signed a consent form. The form stated that they could abort the interview at any time, asked them for permission to record the interview, and informed them that we would keep the recordings confidential and that we would anonymize their interviews.

Interviews. We prepared an "interview guide" containing an ordered list of the broad questions we wanted to cover. We recorded the audio of each interview with a cell phone. The interviews lasted between 14 and 46 minutes.

Field notes. We did not write field notes [9] *during* the interviews, so as not to affect our ability to fully understand the interviewee and to ask relevant probing and follow-up questions. Instead, we recorded the interviews and we later listened to them, taking field notes at that point.

Comparison. We partitioned the participants into two cases, based on whether they took part in the mobile or the desktop version of the project. We then analyzed the coded notes to determine commonalities and differences between the mobile and the desktop group [3].

4. Results

In this section we discuss the results of our study, structured according to the themes that came up in the interviews.

4.1 Motivation

Our initial assumption, and the reason for moving to Android, was that the use of a mobile platform will increase the students' motivation. To gain insight into the motivation of the students, we asked questions regarding cognitive, affective, and behavioral aspects of motivation [12]. For the cognitive aspect, we asked them about the details of their projects to find out if they recall the details. For the affective aspect, we asked about how happy they were about the project, the parts that were most interesting for them, and the parts they were directly involved in. For the behavioral motivation, we asked them about the amount of time they spent on the project and how they solved the major problems they had.

We found that motivation is not just a simple scalar measure. The interviews showed that students' motivation varied throughout the course of the project, and that this change may be different for the mobile and desktop cases.

At the start. We found some evidence that students in the mobile case were particularly excited at the start of the project. M4 said that "At the beginning, we had a good motivation, because we had a big idea", and M2 said that it "It was interesting to know that we were doing something that we could play on the phone, my friend could play it on his phone". We did not find such positive statements from students in the desktop case, and we found a statement that showed the opposite. In particular, D1 said that "Before I started, I was not so motivated, because as an application, it was nothing innovative. It was not like first time ever someone had done something like this and then it got more interesting". While this statement is primarily about D1's lack of interest in the specific application, D1 might have considered mobile application more innovative.

Getting started. We found evidence that the motivation in the mobile projects dropped during the first phase of the project. For example, M5 said "Maybe it was a bit too steep up the hill in the beginning, because way too many things came together, because you can't start focusing on one aspect" and M4 said that "In the middle, [we started to worry about] whether we can finish on time, maybe not!" We found only one statement about motivational problems in initial phase of the desktop projects: D4 said "I didn't want to do [it] in the beginning because it was too complex". We believe that the additional complexity of mobile app development, specifically the use of an IDE and APIs that are different from the ones used in the programming course, may have been responsible for the issues at the beginning of the mobile projects.

Steady state development. We found several statements of students in mobile projects that indicated that their motivation in the middle of the project was low. M1 said "I didn't like work in Eclipse, I like working with Bluej, because when we were working with Java, it was Ok, but when we were working in Eclipse with Android application, [...] it was a terrible experience." M2 said "The emulator didn't always do what we wanted, it crashed, that was the most frustrating part, when you start the emulator and it just crashes and you don't even know why." M5 told of similar experiences: "With one simple problem, I spent about three hours only to lately discover that it was not a problem at all and it was something wrong with my emulator, something wrong with my platform [that] I have no idea! [...] If you have these constant problem, it just drags you down and makes you demotivated." M6 said "It [the emulator] was really slow, really really slow". These comments all are related to issues with the mobile development environment, and especially the Android emulator. If all students had had access to Android devices, they would have been less dependent on the emulator, which would have mitigated at least the crashes. The interviews of the desktop project participants showed no such negative statements referring to the IDE, and the comments were more positive. For example, D1 described "It was a learning experience [...] we had this [course] in parallel, we had the part with the programming course in Java and at the same time we had this on-hands experience ... "

Rush to the finish line. We did not find any negative comments with respect to the last phase of the project. While this often is the most intense phase, it also is the phase when pieces finally fall into place, and the goal comes within reach. As M4 stated: "At first I started slow, but towards the end I had a lot of development really fast".

4.2 Satisfaction

In our interviews we asked the students about their overall satisfaction with the course.

Students in mobile as well as desktop projects generally expressed satisfaction, as exemplified by D2, who said "I think that was one of the best semester I had since I'm in this university", and D4 said "At the end of the semester, I was like I know how to do anything, you can give me anything to do in Java and I can do." However, there is a significant risk of bias, given that they knew their interviews were going to be analyzed by the teaching team. Nevertheless, we found some statements that differ between the desktop and the mobile case.

When asked what they would think of using mobile instead of desktop applications, D4 said "For me is better to stay like to desktop application, because it is only the second semester and for me learning Java or C++ and I think that also for Android application you need a bit more of experience".

M3 said "I think Android was a great choice, however I feel it funny that I can now design interface for mobile but not for desktop, I feel that should be the other way around. I should do learn how, the other one [Desktop] seems more basic knowledge than Android. So, I don't know if that was the right order." This shows that the student perceived the project as not just an attractive target platform, but also as a platform that requires additional knowledge.

4.3 Learning

The educational purpose of SA2 is for students to learn about GUI design and implementation, to practice programming, and to practice team work.

Most of the students in the mobile and desktop projects mentioned these points. For example, M3 said "Well, a lot of Java specs, I've learned how mobile applications work, which are extremely useful for me right now. The graphical user interfaces, that is the most practical experience we had".

And without us bringing up that topic, most students mentioned that they learned about the difficulty of team work.

A difference we found between the mobile and desktop cases is exemplified by M5, who said "It is the combination, everything comes together, because I guess from the user side you don't really understand, OK you know yes there is the user interface, and then there is some coding, but what it is really, but when you get to do it you understand that. They all need different things actually, it is not just coding it is different part, the Java part, which does a certain job, then it comes to Android and they have to interact, but you can't forget that there is also user interface that have to interact with these two, so it was really nice to see, nice and not nice because now we know the magic behind it, we know how the trick works". This indicates that in the mobile projects, students had to deal with aspects like heterogeneity and integration, something that is less prevalent in the desktop projects.

4.4 Preferences

Our idea that mobile platforms are more attractive for programming projects was based on the belief that students generally like mobile platforms. We thus asked students about their general use of mobile apps, and what they thought about mobile platforms.

To our surprise, despite the fact that all participants were informatics students, two students did not actually own any smartphone. D3 said "If I would have one smarthpone, I would have used the smartphone".

M5, who has a smartphone said "I don't like to be on the phone that much, and I'm almost always on the computer, so it [the computer] is more convenient, and I just get around easier." M5 is not alone: D1, D4, M3, and M6 all expressed similar feelings. Their arguments against smartphones included the small screen size, limited performance and battery life, and feature limitations in mobile apps.

5. Lessons Learned

We now discuss the key lessons we learned from our findings.

Closely integrate projects with programming course. In our programming course, we used a different IDE, a different set of APIs, and a different deployment platform. This required that students in the mobile projects had to acquire additional skills that students in the desktop projects did not need. The desktop teams thus were able to focus more on programming, while the mobile teams spent more time on learning how to integrate, configure, and build, and on learning an additional set of APIs.

Ensure device availability. Our context required some students to run their apps on an emulator, because they did not own any Android device. Our school provides all students with a laptop, so phones were actually less prevalent than notebooks. This may not be the case in general, but for us it caused the problems with the emulator to become more prevalent.

Consider all aspects of motivation. Our findings confirmed that there are multiple dimensions to motivation [12]. The motivational boost coming from the attractiveness—at least to some students—of mobile app development might be counter-balanced by negative effects on motivation due to a steeper learning curve and complications in working with the mobile platform.

6. Threats to Validity

Our study has a limited focus: we compare two concrete instances of a project course, in a specific undergraduate curriculum. The programming language in both projects is Java. The desktop applications are based on the Swing GUI toolkit, while the mobile platform is Android. While we do not claim that our findings can be generalized to other contexts, we do believe that our methodology as well as our findings can be a useful starting point for studies in different contexts.

Besides the platform (mobile vs. desktop) several other factors varied between the two courses. Most importantly, different students participated in the projects, different TAs mentored the projects, and students were able to choose their project topic in the mobile case but were required to work on a pre-determined topic in the desktop case. This was one reason for doing a qualitative study: instead of trying to quantify the overall effects on students, our interviews allowed us to dig deeper and try to find the reasons for those effects.

We believe that our bias is limited in this study, because we used both mobile and desktop platforms, and we are not directly invested in either the mobile or the desktop platforms we used.

This is a qualitative study. Its goal is to inform the discourse on the advantages and disadvantages of targeting mobile platforms in undergraduate programming projects. It is not supposed to directly evaluate whether mobile platforms are more effective, and it is not supposed to *quantify* the advantages or disadvantages of either type of platform. The number of students in our projects would probably be too small, and the context of our study too specific, to draw meaningful generalizable quantitative conclusions. However, our qualitative results hopefully provide a basis for follow-up studies that measure the various aspects we identified here, and that investigate the use of mobile projects in other contexts.

7. Conclusions

In this study we investigated how the decision of using Android as the target platform affects students in a second semester Bachelor programming project.

Students enumerated several reasons that initially made Android attractive to them, including: being usable by their friends, earning money, being trendy, and being accessible all the time. This confirmed the expectation set by previous reports [7, 8]. However, while students had a strong motivation for starting their first experience on Andorid, our interviews uncovered several issues. Instructors deciding to move to mobile projects should consider those issues, to avoid the risk that their move negatively affects the outcome of their course.

References

[1] D. J. Barnes and M. Kölling. *Objects First with Java*. Prentice Hall / Pearson Education.

[2] A. Bryman. Interviewing in qualitative research. In *Social Research Methods*, chapter 15. Oxford University Press, 1st edition, 2001. ISBN 978-0198742043.

[3] K. M. Eisenhardt. Building theories from case study research. *The Academy of Management Review*, 14(4), October 1989.

[4] A. Harzl, P. Neidhoefer, V. Rock, M. Schafzahl, and W. Slany. A scratch-like visual programming system for microsoft windows phone 8. In *PROMOTO' 13*, 2013.

[5] M. Jazayeri. The education of a software engineer. In *Proceedings of the 19th IEEE International Conference on Automated Software Engineering*, ASE '04, pages .18–xxvii, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2131-2. . URL `http://dx.doi.org/10.1109/ASE.2004.68`.

[6] A. Kohn. *Punished by Rewards*. Houghton Mifflin, Boston, MA, USA, 1999. ISBN 0-618-00181-6.

[7] S. Kurkovsky. Engaging students through mobile game development. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 44–48, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-183-5. . URL `http://doi.acm.org/10.1145/1508865.1508881`.

[8] Q. H. Mahmoud and A. Dyer. Mobile devices in an introductory programming course. *Computer*, 41(6):108–107, June 2008. ISSN 0018-9162. . URL `http://dx.doi.org/10.1109/MC.2008.200`.

[9] C. B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Trans. Softw. Eng.*, 25(4):557–572, July 1999. ISSN 0098-5589. . URL `http://dx.doi.org/10.1109/32.799955`.

[10] N. Tillmann, M. Moskal, J. de Halleux, and M. Fahndrich. Touchdevelop: Programming cloud-connected mobile devices via touchscreen. In *Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, ONWARD '11, pages 49–60, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0941-7. . URL `http://doi.acm.org/10.1145/2048237.2048245`.

[11] N. Tillmann, M. Moskal, J. de Halleux, M. Fahndrich, J. Bishop, A. Samuel, and T. Xie. The future of teaching programming is on mobile devices. In *Proc. 17th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2012)*, July 2012. URL `http://research.microsoft.com/apps/pubs/default.aspx?id=164416`.

[12] M. Tour-Tillery and A. Fishbach. How to measure motivation: A guide for the experimental social psychologist. *Social and Personality Psychology Compass*, 8(7):328–341, 2014. ISSN 1751-9004. . URL `http://dx.doi.org/10.1111/spc3.12110`.