DOI: http://dx.doi.org/10.1016/j.jvlc.2015.10.019

Using Project-Based-Learning in a mobile application development course—An experience report

Rita Francese [a,*], Carmine Gravino [a], Michele Risi [a], Giuseppe Scanniello [b], Genoveffa Tortora [a]

[a] *University of Salerno, Italy*
[b] *University of Basilicata, Italy*

Keywords:
Project-Based-Learning
Mobile Application Development
Android
Collaborative learning
competitive learning

ABSTRACT

In this paper, we report the experience gained in a Mobile Application Development course. We involved students in Computer Science at the University of Salerno, who in teams had to conduct a project. The goal of this project was to design and develop applications (or simply app) for Android-based devices. The adopted teaching approach was based on Project-Based-Learning and enhanced collaboration and competition. Collaboration took place among members of the same team (intra-team), while competition among different teams of students (extra-team). To allow intra-team collaboration, students used GitHub as Computer-Supported-Collaborative-Learning tool. It provided support for implicit and explicit communication among members in each team and for distributed revision control and management of software artifacts (e.g., source code and requirements models). Developed apps underwent a final public competition prized by IT managers of national and international software companies. This is how we implemented extra-team competition. IT managers expressed a positive judgment on both students' competition and developed apps. Also, students provided very good feedback on used teaching approach and support GitHub provided.

1. Introduction

Internet and mobile applications for smart devices are converging. This is manifesting an increasing interest of business users and customers in mobile devices and applications [1]. The effect is that digital enterprises are becoming mobile enterprises also because mobile devices offer a rich set of embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone and camera. Sensors and internet connectivity allow mobile applications (also simple apps, from here on) to be suitable for a variety of application domains. In this scenario, the role of mobile application developer is one of the most demanded in IT market.[1] Nevertheless, design and development of mobile applications is not an easy task [2]. This kind of developers has to master a wide range of technologies. A good computer skill is also need to deal with design and development issues. For example, developers need at least knowledge on

* Corresponding author.
E-mail addresses: francese@unisa.it (R. Francese), gravino@unisa.it (C. Gravino), mrisi@unisa.it (M. Risi), giuseppe.scanniello@unibas.it (G. Scanniello), tortora@unisa.it (G. Tortora).

[1] http://www.itcareerfinder.com/it-careers/mobile-application-developer.html

programming languages (e.g., Objective C, C++, C or Java), operating systems (e.g., Android and iOS), and so on. For these reasons, many universities decided to enrich their Computer Science program with the competencies related to the design, development, and maintenance of Android mobile applications.

In this paper, we present a teaching experience gained in a Mobile Application Development (MAD) course at the University of Salerno. The main goal was to adopt a Project-Based-Learning approach to foster teamwork and encourage students to explore new ideas in the context of mobile app market. The MAD course was organized in blended learning modality: lectures on the Android operating system were given in presence, while students' projects were remotely conducted. As for projects, students arranged in teams were required to propose applications for smart devices by considering market needs, usefulness, audience, and viability. Once the MAD lecturer had accepted students' proposals, they were asked to design and develop their proposed apps. Collaboration took place among members of the same team (intra-team). Students were asked to use GitHub as Computer-Supported-Collaborative-Learning tool since it provides support for implicit[2] and explicit communication among team members and distributed revision control and management of software artifacts (e.g., source code and requirements models). GitHub is a tool largely adopted in software development areas and recently in education as well [3]. This is why GitHub was used in the MAD course.

The lecturer and two tutors supervised the projects by fixing strict deadline and monitoring their status on GitHub. A distinguished panel of corporate IT managers were asked to judge and give a prize to the three best apps produced during this course. The adopted selection process was based on a team live presentation conducted during a public event organized at the University of Salerno. For each app, IT managers judged its originality, its estimated business value, the pleasantness of user interface, the estimated technical quality, and quality of the team live presentation. This competition among teams of students implemented the second step of our teaching approach: extra-team competition. At the end of the MAD course, we administered a questionnaire to the students. The collected responses allowed us to conduct a qualitative evaluation on student's opinion concerning their learning experience and the used supporting technologies (e.g., GitHub).

The work presented in this paper extends that we reported in [4]. With respect to this work, we provided here the following new contributions: (i) related work has been extended and improved; (ii) feedback from the lecturer perspective has been provided, and (iii) discussion of results has been improved and extended.

Paper structure: In Section 2, we discuss background. In particular, we consider the concept of Project-Based Learning and highlight possible technological solutions to support this teaching paradigm. Main issues related to mobile development for Android framework are also discussed in Section 2. Then, we present our teaching experience in Section 3, while in Section 4 we present the evaluation performed by IT managers and discuss on students' perceptions on their learning experience. Final remarks and future work conclude the paper.

2. Background

2.1. Project-Based-Learning

Project-Based-Learning (PBL) can be exploited to organize learning around projects [5]. This teaching approach considers both cooperative/collaborative learning [6,7] and constructive learning theories [8], where learners become active constructors of their knowledge. Students exploit PBL to cooperate in solving real problems and accomplishing tasks typical for world of work. It is easy to follow that PBL requires high student involvement. Producing an artifact that is of interest since others can use or view it represents a very motivating factor. It is recognized that motivation can do the difference between success and failure of a learning experience more than any other factor. In this scenario, the lecturer has a less central role and students result to be responsible for their own learning (learner-centered education [9]), while learning is the result of their interaction (learner–learner interaction) [10]. When technology supports this methodology it empowers the capability of engaging learners by providing rapid compelling interaction and feedback.

In the context of Computer Science courses, the adoption of PBL is growing [11,12]. One of the main reason is that it enables to train students in principles, methods, and procedures under conditions similar to those characterizing development in actual software projects [13]. Software product development is the result of team effort requiring both technical- and soft-skills. These skills include the ability to communicate, to work as a team, to partition, assign and monitor the progress of the tasks, and to assume responsibility for making choices. Beside the attention for the coding activity, each project phase also needs the production of documentation satisfying determined standards [12].

A PBL approach based on collaboration and competition has been previously proposed [14]. A framework supports both individual competitive setting and collaboration by means of two interconnected modules: *competition*, which addresses individualized competitive learning by following the logic of UEFA Champions League tournament, and *intergroup competitive setting*, where groups of students compete among them. Individual learning was based on the resolution of programming exercises, while the group work was automatically evaluated.

Regueras et al. [15] experiment collaboration and competition to support learning. Students collaborate in a project consisting in the creation of a Wiki and in preparing questions to be submitted to other students, while they compete to be the quickest in answering questions.

Project-based learning and game theory tournaments have been also combined [16]. Groups of two students

[2] It is a knowledge transfer process based on communication through a shared mental or abstract model.

have to develop a software gaming tool and a set of intelligent software players. They receive the deadline to be respected. At the final day of the course, a tournament is organized by involving tools and a player for each tool. In this way, students compete against source code written by other students in a tournament environment.

2.2. Technology support for PBL

To adopt BPL in computer science courses several technological solutions have been proposed [12,17,18]. For example, Macias [18] proses a Moodle-based e-portfolio that supports PBL activities characterized by a number of deliverables and organizational resources.

The use of Wikideas and Creativity Connector tools was assessed by Ardaiz-Villanueva et al. [19]. In particular, the authors focus on the effectiveness of stimulating generation of ideas and originality by academic students accomplishing PBL activities. Differently, Zagalsky et al. [3] analyze the use of GitHub as a collaborative platform for education. Indeed, at the beginning GitHub was mainly employed to support code and project management for software development. It is currently used also in other domains that involve collaborative work, such as education, where it is exploited to manage students and their work. In particular, it is used as submission platform for hosting course material. GitHub has been also employed as supporting tool for an Agile course consisting in scrum-based PBL [12]. However, the focus of this work is on the scrum methodology.

Differently from research work presented before, our contribution consists in presenting results we gained from the adoption of GitHub in a specific course related to the teaching of emerging technologies employed for the development of apps for Android devices. We also asked students to use GitHub in all the development phases of these apps. This environment has been also employed to support communication and collaboration among team members (intra-team collaboration) and between teams and lecturer.

2.3. Developing mobile applications for Android

When developing mobile apps, one of the main challenges is to deal with multiple platforms [11]. Developers have two alternatives at their disposal. The first alternative consists in designing and developing apps by using native development tools made available by each of the major mobile platforms, such as iOS, Android, Microsoft Windows Mobile, Symbian, BlackBerry. The second alternative consists in exploiting cross-platform environments such as PhoneGap and Titanium [20]. Since different operating systems can make available different features, developers prefer to develop more versions of the same app, one for each hardware platform. As an example, the development on an app exploiting in the better way features of Android devices requires that developers master development skills related to Android operating system and associated development environment and resources. The demand for the operating system Android is immensely expanding day by day. Looking at the available resources, smartphones are equipped by sensors, such as accelerometer, gyroscope, GPS, brightness and temperature, and offer communication features, including phone calls, SMS, email, and camera functionalities.

Activities and services are the main Android components. In particular, an activity is an app component that provides a screen with which users can interact with to do something, such as dial the phone, take a picture, send an email, or view a map. Each activity is given a window in which to draw its user interface. For simplicity reason, we will refer to an activity as a GUI.

The verification of asynchronous events continuously interrupts the app execution flow. To deal with it, developers have to implement the activity logic taking into account its life-cycle. As an example, the app has to perform specific work when an activity is suspended (e.g., for the arrival of a phone call) to manage state changes, or an appropriate GUI has to be provided when a device is rotated.

A service is a background component that works for remote processes or performs either long-running operations. A user interface is not provided in case of a service. As an example, when the user has an interaction with an activity a service might fetch data over the network without blocking the interaction.

The market of Android hardware devices offers a variety of different screen sizes, processor types, custom APIs, etc. As a consequence of this fragmentation, the main challenge is to maintain similar execution performances and user experiences in all these variations. Also, testing activities become even more complicate. It might be practically impossible to test an app on all the devices and OS versions for which it has been developed to. Android is able to manage different configurations by employing non-code app resources (images, strings, layout files, etc.), which should include alternatives for each considered configuration.

GUI of Android activities can be created directly in Java or by using an XML-based layout file. Two main advantages characterize the latter approach since it allows: (i) separately managing logic and presentation (ii) maintaining different parallel layouts for difference screen sizes. Rather than setting the content as a view created in Java source code, the use of XML to specify GUI requires to set a reference to XML layout.

3. Teaching experience

In the following, we describe the teaching experience gained in a Mobile Application Development (MAD) course held at the University of Salerno for students in Computer Science. We will focus on the main aspects related to our gained experience.

Goal: The aim of the course was to increase student interest, knowledge, and practical experience in mobile development through an engaging and empowering PBL experience.

Software platform: The operating system used during the course was Android because the barriers to entry in Android remain much lower than others. Indeed, differently from iOS, developers can iterate and test their designs quicker. Furthermore, marketing costs are significantly lower. Android users grow year by year and it represents the largest overall smartphone market [1].

Context: 55 students at the University of Salerno attended MAD course. They were enrolled to the third year of the Bachelor program in Computer Science. To get demographic information about students, we asked them to fill in a pre-questionnaire when MAD course started. In particular, we included in that questionnaire statements aimed at getting information about the following points: experience on Android device use (as smart-users) and development, knowledge on Software Engineering and Web Development principles, experience on Java and Network Programming, and JUnit and Database knowledge. Each question admitted a closed answer according to a 5-point Likert scale, ranging from *Very Low* (1) to *Very High* (5). An analysis of gathered responses suggested that (see histograms reported in Fig. 1) students were generally smart-users, they did not know any knowledge on the development for Android devices, most of them thought to be good Java developer, 30 students affirmed to have a good competence in Software Engineering, 33 to be good web developer, and 14 students declared to have good network programming abilities. The latter result is probably due to the contemporaneity of a course on network programming the students have been attending. Most of students had experience in database development, while a few of them knew the JUnit framework.

Course organization: MAD course was organized in two main modules: (i) lectures on Android operating system; and (ii) a project work aimed at experiencing technologies to develop apps for smart devices accessing services available on the web. The duration of MAD course was 12 weeks. During the first part, students followed lectures in presence. The lecturer made available didactic material online on a learning platform. This part of the course covered the following topics: Android Activity lifecycle, modern interfaces, accessing to the web, threads, Android services, access to RESTful web service by JSON, accessing to native functionalities (i.e., GPS, sensors, camera, SMS, call), monetization. The second part of MAD course was conducted in distance modality and consisted in a project work. We better discuss on this part in the following.

3.1. *Organization of project work*

Teams: The students were arranged in 27 teams, each one consisting of two people, except one of three. The pedagogy of project-based learning suggests that to obtain good results groups should be composed of students with similar ability and interest in the topics being learned [21]. However, the debate on the effectiveness of homogeneous and heterogeneous groups is still open and needs further investigation [22,23]. To partially deal with issues before, team composition was based on students' preferences. We did not randomly assign team members because students had previous experience of project work gained in several

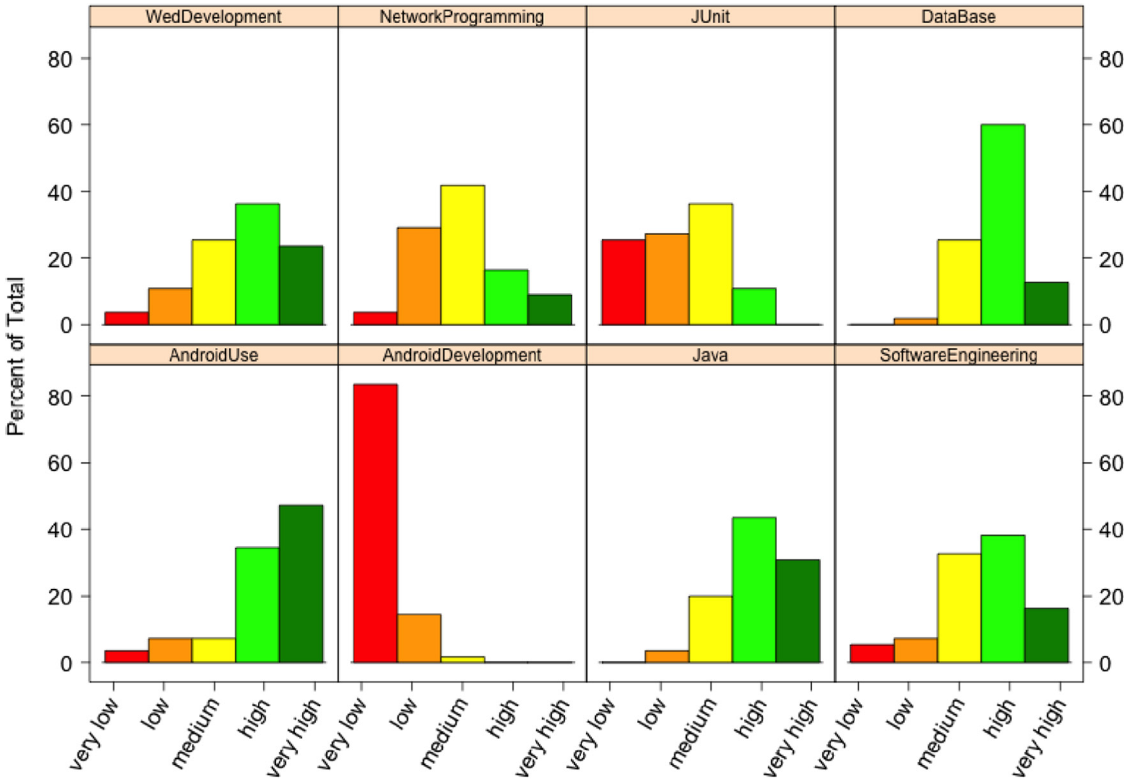

Fig. 1. Students' development competencies.

university courses and, at the last term, they knew classmates more appropriate to work with.

Software projects: Each team proposed the app to be designed and developed. We took this decision to stimulate student's entrepreneurship actions and creativity. Students had to accurately motivate their proposal, also presenting results of a market analysis performed before starting project work. Furthermore, MAD lecture asked to take into consideration the following nonfunctional requirements: an app had to interact with a remote server, through JSON. It had to exploit native device functionalities, including maps, GPS, sensors, call, and SMS. It had to handle device rotation and to use SQLite. Games were admitted if they exposed back-end functionalities, such as account management, multiuser, bonus management, and app upgrade. Each team could start app design and development after the lecture approved its project work proposal.

Lifecycle model: The lecturer suggested students to follow an incremental prototyping development lifecycle and promoted the adoption of pair programming while implementing apps. Let us recall that pair programming is a software development practice where two programmers work as a pair on the same computer. This is an agile development practice largely adopted in industrial settings [24]. As for students, it has been observed that their performances improve [25], namely they produce higher quality source code, are more confident with their work, and enjoy it more.

Deadlines: Project works admitted more deliverables. The lecturer informed students about deadlines for the presentation of these deliverables. The project proposal was the first deliverable, while the second was the Requirement Analysis Document (RAD). Subsequent deliverables are referred to: GUI prototype, mobile app prototype, and the final version of the app that had to include also software components to be run on server side.

Documentation: Students had to document their work and the produced documentation had to be compliant with project templates the lecturer proposed. In particular, a Project Proposal has to present the idea underlying the project work, motivation, a summary study of market, also considering the apps available on the main app stores, and the novelty of app to be developed. On the other hand, a RAD had to better detail requirements, providing software models (actors, use case diagrams, class diagrams, sequence diagrams, navigation diagrams, and screen mock-up). Teams were also asked to produce test cases to be used on the final version of the app (included developed server side software components).

3.2. Using GitHub

Student projects were managed through GitHub, a distributed revision control and source code management system [26]. It currently hosts over one million source code repositories, and has 340,000 registered contributors. Each repository on GitHub has a dedicated project page. It hosts source code files, commit history, open issues, and other data associated with a project [27]. Some studies (e.g., Zagalsky et al. [3]) have highlighted that GitHub can be also used as a powerful learning management tool, differently used by various educators even in similar environments (e.g., technical background) and with similar requirements (e.g., class size, course type).

Project work evolution was monitored by MAD lecturer and tutors, namely, by supervising that each team correctly performed its work to be able to participate in an App Challenge,[3] namely a public competition where students were asked to participate. Students participation was encouraged, but not mandatory. The lecturer created a GitHub account for each team, downloading on it project templates students had to use in their project work. To this aim, the lecturer uses the GitHub mechanism for milestones, typical of many project systems. In GitHub a new milestone simply has a title, description, and a date. GitHub also provides a graph view that summarizes project activities. Operating in this way, the lecturer had a high-level view of students' activities during the development of apps.

To handle communication in a single centralized place, each team used GitHub communication features rather than emails. This also allowed to have communication messages visible to all the member of each team. Using labels the lecturer as well as the team members were also able to create issues for discussion. Team members can set-up email notifications when people comment or tag them in an issue. GitHub communication features favor awareness, which has a very relevant value of activity information for small teams [28]. Indeed, notifying members of actions on shared artifacts can help them to maintain mental models of others activities [29]. Furthermore, it can avoid potential coordination conflicts [30]. In fact, GitHub was also used to enable the communication among the lecturer and the students. in each project work team members could notify the lecturer when a deliverable was completed and ready to undergo a review. Then, the lecturer could accept it or notify changes to be performed by adding a checklist-based revisions to team GitHub accounts.

It is also important to highlight that the transparency of GitHub. That is, a student involved in a project was able to look at the documentation of students in both his/her team and other teams. Students could also look at how other students coded, what they paid attention to, and how they solved problems. Thus, the availability of this information can enable to learn better ways to code and possibly improve students' knowledge [27]. GitHub transparency also favored competition since a team can monitor the state of other teams and is possibly stimulated to perform in a better way tasks of his/her project work.

4. Project work evaluation

We assessed the team work and the produced apps by considering the following evaluation levels:

[3] http://www.zerottonove.it/unisa-grande-successo-per-la-prima-edizione-di-app-challenge/

- *Lecturer*: MAD lecturer and tutors continuously monitored artifacts produced by each team. In particular, she used GitHub to monitor the progress of teams and to assess whether they respected fixed deadlines. The communication among the lecturer and the students took place also in presence if needed. For example, students had to show three versions of their app (as mentioned before) and in this case revision meetings were planed and conducted in presence. A wrap-up meeting was also conducted before the App Challenge. The goal was to prepare students to that competition.
- *App Challenge*: The main goal of this App Challenge was to stimulate students in engaging in the team work, as well as to have excitement throughout MAD course. It is worth recalling that participation in App Challenge was on voluntary base. In our case, all the students enrolled to the MAD course participated. During the competition, students gave a demonstration (the imposed time limit was eight-minute) to a panel formed by external IT managers of national and international companies, whose business included the development of apps for smart devices. Each team of students had to show that their app meets the market needs, explain which technologies they selected and why, discuss their choice on the User Interface, and present a live app demonstration. The first three projects received a prize from the jury composed of ten IT managers. The first prize consisted in two iPads, the remaining were external hard disks. We asked managers to fill a grid scored from 1 to 10 concerning the following aspects: originality, business value, User Interface, technical quality, and team presentation of project. Prizes were assigned on the basis of IT managers' evaluation. One of the main goal related to the organization of our App Challenge was to assess students' apps also from a professional perspective. App Challenge results did not affected students' final mark in MAD course.
- *Students*: We were also interested in collecting some feedback from students about their perception on using GitHub, developing apps for smart devices equipped with an Android operating system, and their participation in the App Challenge. To this end, we asked students to fill in the questionnaire reported in Table 1.
- *Software and project metrics*: We collected both software and project metrics [31]. A metric is a quantitative measure of a degree to which a software system or a process possesses some property. We collected software and project metrics for two main reasons: (i) to assess team productivity and work and (ii) to study the value of these metrics to estimate the effort needed to develop mobile apps. We will focus here only on the first point. The interested reader can find details on the second point in [32]. We collected the following metrics from students' projects and produced software artifacts:
  - *Requirements*: The number of functional requirements individuated in the RAD.
  - *Checkouts*: The number of local working copy taken from the repository by the team members. It provides an indicative idea of how much the team members are active and how their work is distributed.
  - *Time*: The time students were active on each development phase. It has been obtained by analyzing activity logs of GitHub.
  - *User interfaces*: The number of graphical components composing an user interface of an app. In particular, we considered the number of XML file describing Android activity user interfaces.
  - *LOC*: The number of lines of code, source code comment excluded.
  - *Cyclomatic complexity*: It is a measure that estimates the complexity of a program. It measures the amount of decision logic in source code. It is a measure of how is structured a program. A high Cyclomatic Complexity denotes a bad structure and high risk of errors.
  - *Depth inheritance tree*: It measures the software complexity of an inheritance hierarchy. It is the length of the longest path from a given class to its root class. Some studies have shown that higher DIT (Depth Inheritance Tree) rate corresponds to larger error density and lower quality [31]. The smaller the DIT, the more abstract and simpler the class would become, but decreases the class reusability. While the more a class inherits, the more difficult to understood the design is.

Our choice in selecting these metrics was mostly based on their simplicity in collecting and because they are well known and widely adopted in software engineering and in software industry (e.g. [31]).

Together with OO and traditional size code metrics, we also measured method calls in mobile apps. Method calls classified as both internal method and API (Application Programming Interface) calls. Internal method calls are invocations to methods the original developer implemented in the app, while API calls represent invocations to methods that Android provides. The number of API calls represents a good estimate of the use that an app makes of native functionalities.

5. Results

All teams participated in our App Challenge and released their app on time. As for the kind of developed apps, 11 apps were video games, including graphics retro-based games, or based on word guessing or math ability, 13 apps supported productivity (e.g. apps providing information on a City Hall, supporting people management or personal training), while the remaining apps implemented social media, e.g. for sharing their own travel diary or meeting people.

5.1. App challenge

The scores of App Challenge jury are graphically summarized in Fig. 2. Descriptive statistics are also reported in Table 2. Scores could range from 10 to 500. Thus, a mean score 331 with only two apps that scored less than 300 revealed a good opinion of the IT managers on the students' apps. The app considered the best scored 405. The app was a very captivating game. Developers were also

Table 1
The perception questionnaire.

ID	Question
P1	Managing my project with GitHub was easy
P2	Using the Software Configuration Management features offered by GitHub (e.g., commit, check-out) was easy
P3	Using the communication features offered by GitHub (e.g., notification, tagging) was easy
P4	I think that the app I developed is complex
P5	Basing on your experience, the development of mobile app is easy with respect to traditional desktop applications
P6	Basing on your experience, the development of mobile app is easy with respect to traditional web applications
P7	My experience in the development of mobile app during this course was involving
P8	The final competition was a stimulus for improving the quality of my work with respect to a traditional exam
P9	After this course, my Android development competences are: 1) very low; 2) low; 3) medium; 4) high; 5) very high.

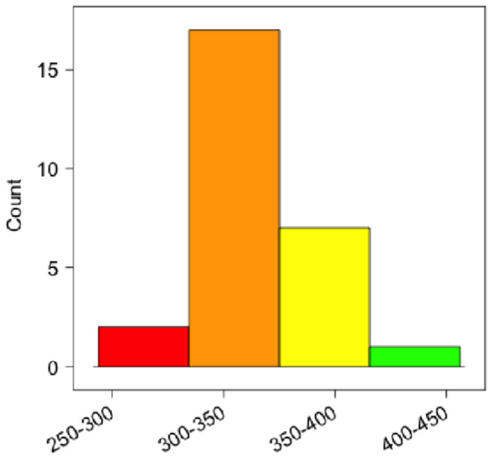

Fig. 2. The scores attributed by the jury.

Table 2
Jury evaluation statistics.

Min	Max	Median	Mean	St. Dev.
282	405	328	331	27.17

very able to present and motivate their app, also creating a suggestive video of their proposal shown during the challenge. The User Interface was very simple and fascinating. The technical complexity was lower because the game mainly worked on the mobile device, except for the server-side score management.

5.2. Student perception

Concerning the opinions the students had on their Android development competencies, they perceived a notable improvement at the end of MAD course (see Fig. 3). In particular, this figure depicts histograms related the perceptions before and after the course, collected by a pre-course questionnaire and perception questionnaires

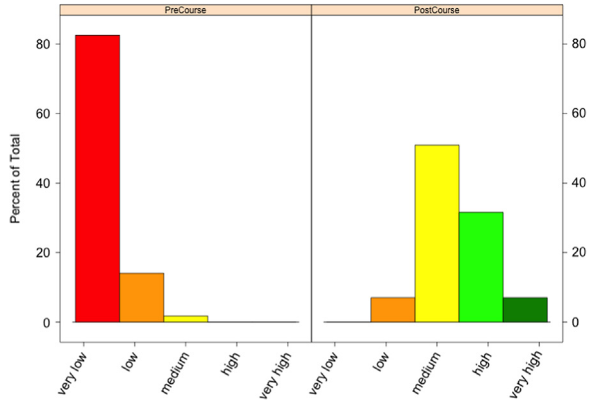

Fig. 3. The student perceptions on their Android development competencies pre- and post-course.

(question P9) administered to students before and after MAD course, respectively.

Responses to our perception questionnaire are graphically summarized in Fig. 4. In particular, the greater part of students asserted that GitHub eases the management of projects, 31 expressed a positive judgement (question P1). 36 students positively judged the configuration management support offered by GitHub (P2), while 33 expressed a positive judgement on its communication feature (P3).() 30 students judged complex the apps they developed (P4). 36 students considered easier the development mobile apps with respect to desktop applications (P5), while most of students considered easier develop web apps with respect to mobile apps (P6). A high number of students (50) perceived the course involving (P7) and the final competition was very appreciate by 55 students (P8).

5.3. Project metrics

Metric values of all projects are summarized by means of boxplots in Fig. 7. In general, the produced apps do not have a large number of functional requirements: the projects were characterized by median value equal to 8. The number of checkouts is not elevated (54 on average), probably because students worked in pair programming, on the same PC, and then they executed checkout at the end of their work day. The time took to develop the app on average was 59 working days. The time to accomplish analysis phase should also be added (about one month). The User Interfaces produced for each app were 33 on average. Average LOC value was 3609. Cyclomatic Complexity was on average 1.83, which denotes a good modular structure of source code (low risk of errors for values less than 10). DIT was 4 on average. This means that the classes are not much reused.

Table 3 reports the descriptive statistics for API (Application Programming Interface), internal method calls and the total number of calls. Internal method calls are invocations to methods the original developer implemented in the app, while API calls represent invocations to methods that Android provides. Half of the apps made more than 726 API calls, that is they made a large use of the functionalities offered by the Android operating system.

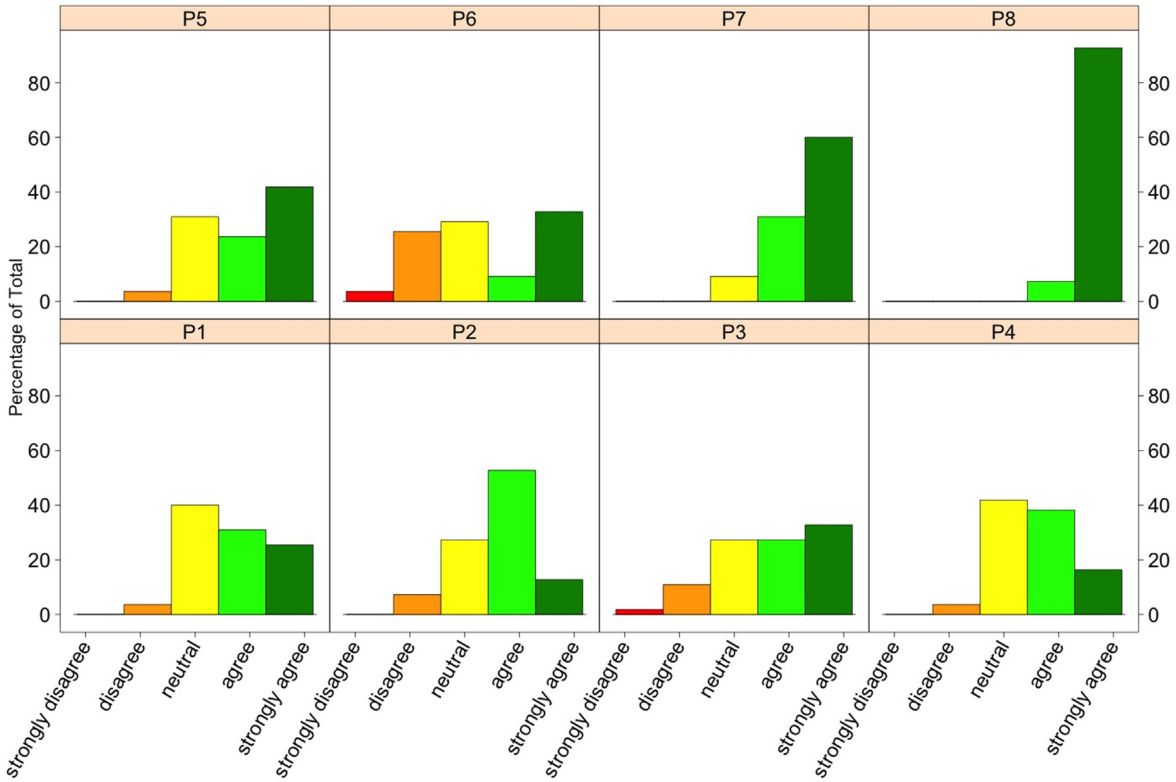

Fig. 4. The perception questionnaire results.

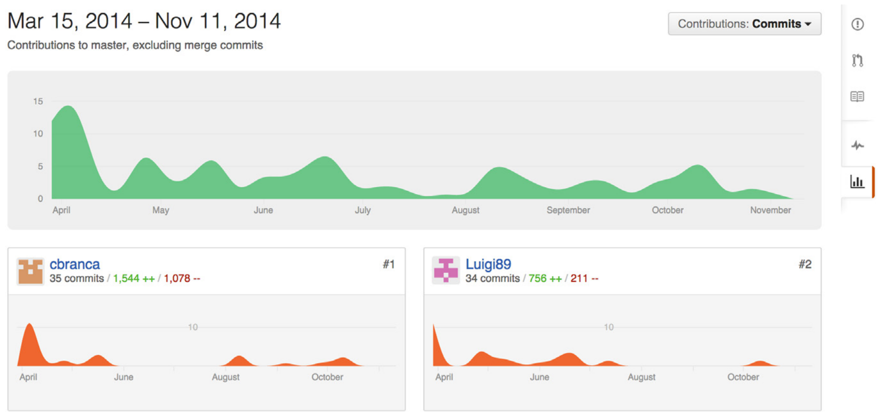

Fig. 5. The commit distribution of a project.

5.4. *Lecturer's perspective*

The lecturer can express a good judgment on GitHub and its support. This tool has been easily adopted by students because its integration in the Eclipse platform. Students well knew this platform because adopted in several programming course at the University of Salerno. The lecturer also appreciated graphical reports offered by GitHub. She was able to monitor teams and members activities, graphically summarized by the feature *Graphs* implemented in GitHub. Fig. 5 shows the distribution of commits a team and its two members executed during MAD course. This graph does not include commit with zero changes. It is also interesting to observe when this activity occurred, specifically in which day of week and time of day, as shown in Fig. 6. From these reports, for example, we can deduce that for the considered project, the developer on the left in Fig. 6 between the two team members performed the greatest number of changes and that the team work mainly occurred between 10:00–12:00, and 17:00–19.00 in all the day of the week.

Some students had some problems in the project start-up phase. Specifically, this was true for those teams that begun to locally work on projects and, successively, tried to publish it on GitHub. In addition, some students disagreed on the use of an open repository because they wanted avoid the diffusion of the source code and the design and analysis artifacts of their apps. Unauthorized people could use public these artifacts to commercialize students' apps.

The final recommendation from the lecturer is: use GitHub because it is flexible for distributed revision control and management of software artifacts and because it provides effective support for Project-Based-Learning and enhanced collaboration.

6. Conclusion and future work

In this paper, we present and discuss a teaching experience carried out during a Mobile Application Development course with Bachelor Students at the University of Salerno. Students were arranged in teams and were asked to implement mobile apps for Android devices and cooperate to implement them by exploiting GitHub. At the end of the course, students participated in a competition to establish the best developed app. A jury composed by IT professional managers assessed the apps in this competition.

Outcomes from our experience can be considered positive: all the students delivered project artifacts on time, with a good level of quality and completeness with respect to the established requirements (as gathered values for software metrics also suggest). The possible motivations could be related to the following aspects: first, all the students were enthusiastic in developing apps for smart devices; secondly, their activity was monitored thanks to the use of a GitHub which enabled continuous monitoring of the team work in all the phases of the development process, starting from the project proposal. Last but not least, let the students present their work to IT managers belonging to top IT companies further motivated students. Indeed, by examining the project activity of the teams, when they knew of the company involvement their production notably increased. The lecturer and the tutors continuously motivated the students, also providing suggestions on the way they had to communicate. The App Challenge was successful because it allowed the best students to be placed or to increase their familiarity with the work market. For example, a TLC company involved in this competition hosted the winning students for a stage because they demonstrated to be young talent with a strong innovation ability. Many other students were required by the other companies involved in the App Challenge. Overall, all the companies manifested a positive judgment on the competition and on the work the students did. In fact, many of these companies asked to be informed and involved in future similar initiatives. The reason of this success could be that the adopted learning approach stimulated both intra-team collaboration and extra-team competition. Future work can be directed in the evaluation of the relevance of the combination of these two approaches on learning.

As future work, we also plan to fully involve the IT professional managers in the next edition of the MAD course. In particular, we would involve them as the role of coach. Future work will be also devoted to introduce in the next year course cloud platforms for implementing the back-end of the apps for smart devices.

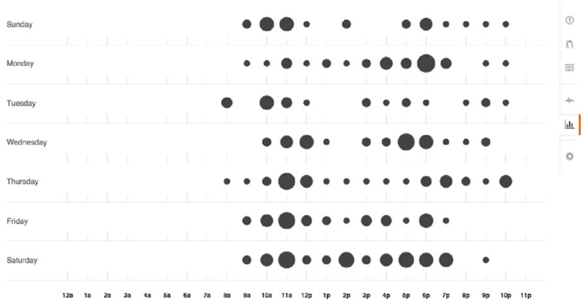

Fig. 6. Work distribution by day of week and time of day.

Table 3
Descriptive statistics on the API and internal method calls.

Method call	Min	Max	Median	Mean	St. Dev.
API	36	11,449	726	1398	2236
INTERNAL	9	1325	180	331	400

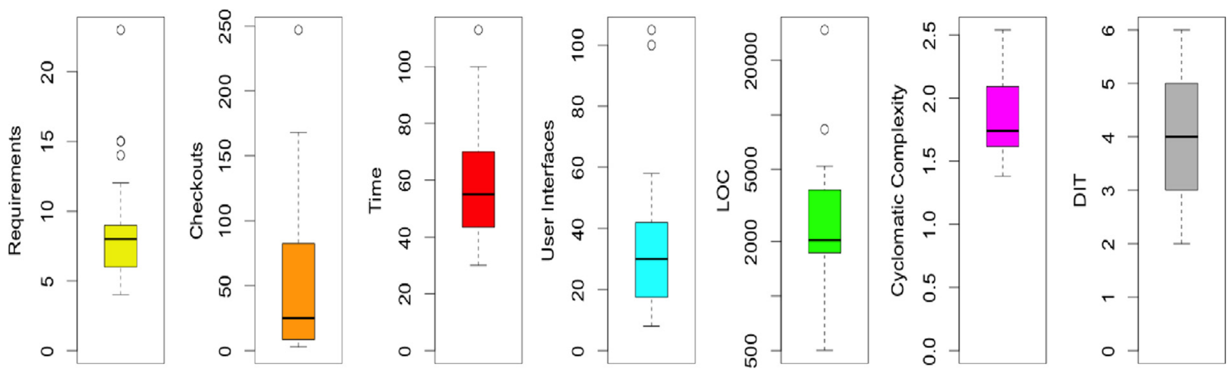

Fig. 7. The boxplots of the project metrics.

Acknowledgements

We thank all the students to this course and the IT managers, who judged the apps these students developed.

References

[1] Gartner, Top 10 Mobile Technologies and Capabilities for 2015 and 2016. https://www.gartner.com/doc/2665315, 2014.
[2] A.D. Lucia, R. Francese, M. Risi, G. Tortora, Generating applications directly on the mobile device: an empirical evaluation, in: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI'12, Capri Island, Naples, Italy, May 22–25, 2012, pp. 640–647.
[3] A. Zagalsky, J. Feliciano, M.-A. Storey, Y. Zhao, W. Wang, The emergence of GitHub as a collaborative platform for education, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW), ACM, New York, NY, USA, 2015, pp. 1906–1917.
[4] R. Francese, C. Gravino, M. Risi, G. Scanniello, G. Tortora, On the experience of using Git-Hub in the context of an academic course for the development of apps for smart devices, in: Proceedings of the International Conference on Distributed Multimedia Systems, 2015, pp. 292–299.
[5] P.C. Blumenfeld, E. Soloway, R.W. Marx, J.S. Krajcik, M. Guzdial, A. Palincsar, Motivating project-based learning: sustaining the doing, supporting the learning, Educ. Psychol. 26 (3–4) (1991) 369–398.
[6] P. Dillenbourg, What do you mean by collaborative learning? Collab.-Learn.: Cognit. Comput. Approaches (1999) 1–19.
[7] G. Scanniello, U. Erra, Distributed modeling of use case diagrams with a method based on think-pair-square: results from two controlled experiments, J. Vis. Lang. Comput. 25 (4) (2014) 494–517.
[8] J.R. Savery, T.M. Duffy, Problem based learning: an instructional model and its constructivist framework, Educ. Technol. 35 (5) (1995) 31–38.
[9] D.A. Norman, J.C. Spohrer, Learner-centered education, Commun. ACM 39 (4) (1996) 24–27.
[10] J. Laffey, T. Tupper, D. Musser, J. Wedman, A computer-mediated support system for project-based learning, Educ. Technol. Res. Dev. 46 (1) (1998) 73–86.
[11] M.E. Joorabchi, A. Mesbah, P. Kruchten, Real challenges in mobile app development, in: Proceedings ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), ACM Press, New York, NY, USA,2013, pp. 15–24.
[12] S. Kizaki, Y. Tahara, A. Ohsuga, Software development PBL focusing on communication using scrum, in: Proceedeings of the IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAAI), 2014, pp. 662–669.
[13] S. Yadav, J. Xiahou, Integrated project based learning in software engineering education, in: Proceedings of the International Conference on Educational and Network Technology (ICENT), 2010, pp. 34–36.
[14] B. Silva, P. Girão, A study and a proposal of a collaborative and competitive learning methodology, in: Education Engineering (EDUCON), IEEE, Los Alamitos, CA, USA, April 2010, pp. 1011–1018.
[15] L.M. Regueras, E. Verdú, M.J. Verdú, J.P. De Castro, Design of a competitive and collaborative learning strategy in a communication networks course, IEEE Trans. Educ. 54 (2) (2011) 302–307.
[16] J.C. Burguillo, Using game theory and competition-based learning to stimulate student motivation and performance, Comput. Educ. 55 (2) (2010) 566–575.
[17] Q.C. Lang, Developing an asynchronous computer mediated communication tool for project-based learning, in: Proceedings of the 4th International Conference on Distance Learning and Education (ICDLE), 2010, pp. 222–225.
[18] J. Macıas, Enhancing project-based learning in software engineering lab teaching through an e-portfolio approach, IEEE Trans. Educ. 55 (4) (2012) 502–507.
[19] O. Ardaiz-Villanueva, X. Nicuesa-Chacón, O. Brene-Artazcoz, M.L. S. de Acedo Lizarraga, M.T.S. de Acedo Baquedano, Evaluation of computer tools for idea generation and team formation in project-based learning, Comput. Educ. 56 (3) (2011) 700–711.
[20] R. Francese, M. Risi, G. Tortora, G. Scanniello, Supporting the development of multi-platform mobile applications, in: Proceedings of the 15th IEEE International Symposium on Web Systems Evolution (WSE), 2013, pp. 87–90.
[21] Y.-T. Lin, Y.-M. Huang, S.-C. Cheng, An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization, Comput. Educ. 55 (4) (2010) 1483–1493.
[22] C.A. Bowers, J.A. Pharmer, E. Salas, When member homogeneity is needed in work teams: a meta-analysis, Small Group Res. 31 (3) (2000) 305–327.
[23] S. Hooper, M.J. Hannafin, The effects of group composition on achievement, interaction, and learning efficiency during computer-based cooperative instruction, Educ. Technol. Res. Dev. 39 (3) (1991) 27–40.
[24] L. Williams, R. Kessler, Pair Programming Illuminated, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, (2002).
[25] C. McDowell, L. Werner, H.E. Bullock, J. Fernald, Pair programming improves student retention, confidence, and program quality, Commun. ACM 49 (8) (2006) 90–95.
[26] GitHub. ⟨https://github.com⟩.
[27] L. Dabbish, C. Stuart, J. Tsay, J. Herbsleb, Social coding in GitHub: transparency and collaboration in an open software repository, in: Proceedings of the ACM International Conference on Computer Supported Cooperative Work (CSCW), ACM, New York, NY, USA, 2012, pp. 1277–1286.
[28] P. Dourish, V. Bellotti, Awareness and coordination in shared workspaces, in: Proceedings of the ACM International Conference on Computer-supported Cooperative Work (CSCW), ACM, New York, NY, USA, 1992, pp. 107–114.
[29] T. Gross, C. Stary, A. Totter, User-centered awareness in computer-supported cooperative work-systems: structured embedding of findings from social sciences, Int. J. Hum.-Comput. Interact. 18 (3) (2005) 323–360.
[30] A. Sarma, Z. Noroozi, A. van der Hoek, Palantir: raising awareness among Configuration management workspaces, in: Proceedings of the 25th International Conference on Software Engineering (ICSE), 2003, pp. 444–454.
[31] V.R. Basili, L.C. Briand, W.L. Melo, A validation of object-oriented design metrics as quality indicators, IEEE Trans. Softw. Eng. 22 (10) (1996) 751–761.
[32] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, On the use of requirements measures to predict software project and product measures in the context of android mobile apps: a preliminary study, in: Proceedings of the Euromicro Conference on Software Engineering and Advanced Applications, 2015, pp. 357–364.