Open Source Projects in Programming Courses

Michela Pedroni
Chair of Software Engineering
ETH Zurich
8092 Zurich, Switzerland
michela.pedroni@inf.ethz.ch

Till Bay
Chair of Software Engineering
ETH Zurich
8092 Zurich, Switzerland
bay@inf.ethz.ch

Manuel Oriol
Chair of Software Engineering
ETH Zurich
8092 Zurich, Switzerland
manuel.oriol@inf.ethz.ch

Andreas Pedroni
Institute of Neuropsychology
University of Zurich
8092 Zurich, Switzerland
pepo@access.unizh.ch

ABSTRACT

One of the main shortcomings of programming courses is the lack of practice with real-world systems. As a result, students feel unprepared for industry jobs. In parallel, open source software is accepting contributions even from inexperienced programmers and achieves software that competes both in quality and functionality with industrial systems. This article describes: first, a setting in which students were required to contribute to existing open source software; second, the evaluation of this experience using a motivation measuring technique, and third, an analysis of the efficiency and commitment of students over the time. The study shows that students are at first afraid of failing the assignment, but end up having the impression of a greater achievement. It seems also that students are inclined to keep working on the project to which they contributed after the end of the course.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Management, Experimentation, Human Factors

Keywords

Open source, Project, Assessment, Motivation, Communities

1. INTRODUCTION

We teach software engineering techniques, design patterns, and project development models in our courses. Teaching these topics to students without actually exposing them to industry grade software may not only sound preachy, but will also leave no remaining impression. Open source software allows closing this gap [7, 4] and offers great opportunities to bring real-life experience directly into the classroom. In particular, open source software can be used to emphasize the importance of high quality software design, the role of design patterns, the need of good documentation, and the relevance of social skills in a real-world environment.

Over the last years, only few instructors experimented with open source software in their classroom. Allen et. al. [1] used Dr. Java - an open source Java programming environment - to teach extreme programming techniques. Students of Fuhrman [3] freely chose an open source project which they inspected to propose corrections in the design. Fuhrman did not require students to implement their improvements, but many still did. Carrington [2] allowed students to choose from a list of open source projects that are useful to software engineering and let them inspect, report and extend the tools during his course. Except for contributions by students to Dr. Java, these experiments refrained from actively submitting changes, bug fixes, or improved designs to the open source projects.

The assignment described here combines the freedom of choosing an arbitrary open source project with the ultimate goal of students contributing to it. Allowing students to freely choose the project on which they intend to work, lets them adapt the assignment to their personal interests. Requiring students to contribute back to the open source project, increases the prestige of their work and the effort that they are willing to put in the assignment. Students experience all the tasks needed to adapt existing software: understanding it, identifying a needed improvement, designing the solution, implementing, testing and adapting to the requirements for contribution. During this work, they need to get socially involved with the other developers of the open source project, having to deal with the communication problems that occur frequently with distributed development.

This paper presents a thorough evaluation of the approach taken by using a questionnaire focusing on students' motivation [6]. The study compares the motivation of students working on an open source project to the motivation of a control group from another course working on a traditional (exercise) project.

Section 2 reports on the course setup and describes the project requirements imposed on the students. Section 3 provides a thorough evaluation based on a questionnaire from motivational psychology and shows the weaknesses and strengths of using open source software. Section 4 extends the motivational evaluation with results on students efficiency and commitment. After presenting the interpretation of the results the article concludes in section 5.

2. COURSE SETUP

This study focuses on two programming courses: an advanced Java course (the **experimental course**, roughly 70 students, master-level) and a course on concurrent object-oriented programming (the **control course**, roughly 30 students, master-level). The experimental course required groups of students (maximum 5 people) to contribute to a Java open source project of their choice. The control course relied on a given artificial project to be solved by groups of maximum 3 people. The goal of the projects was in both cases to deepen the understanding of the lectures by putting the concepts to work. Both projects spanned over 6 weeks and started at the middle of the semester. For both courses the results of the project influenced significantly the final grade of the course (respectively 40% and 65% of the final grade).

The following subsections describe each of the project assignments in more details.

2.1 Control course project

In line with the focus of the course the project required the students to build a group of command line programs, that use a given concurrency framework and demonstrated its capabilities. The second part of the project consisted in extending the framework with a rendez-vous synchronization mechanism that did not exist previously.

The exact assignment was to: (1) program the required tasks, (2) answer questions, (3) write a report including a description of the code as well as the answers to the questions.

2.2 Open source project

At the beginning of the course, students were informed about the grading scheme and that they would have to form groups for the project. Students received the project description only at the middle of the semester. At this occasion, lecturers and students discussed the nature of the community work that open source projects involve. The discussion showed that very few of the seventy students had actively participated or even contributed to an open source project. Obviously, all of them had already used open source software.

The project description included an initial selection of popular and active open source projects (see Table 1).

Each description of the proposed projects included references to the relevant web pages, wikis, mailing lists, and URLs of the bug tracking systems. It also included an informal evaluation of the projects' organization such as specifics about the development process (e.g. was it bug-driven or planned). This information provided a starting point when looking for possible contributions to the project.

Besides this selection of projects students could also look for other Java open source projects and assess their suitability for contributions. As Table 1 shows, half of the students chose one of the proposed projects and the other half selected one of their own.

Table 1: Open source projects

Projects Suggested		Anteater: http://aft.sourceforge.net
Projects Suggested		ArgoUML: http://argouml.tigris.org
Projects Suggested		JEiffel: http://se.ethz.ch/projects/benno_baumgartner
Projects Suggested		JSR's: http://www.jcp.org
Projects Suggested		Open Office: http://www.openoffice.org
Projects Suggested		Tomcat: http://tomcat.apache.org
Projects Suggested		XmlIO: http://www.bifrost.org/xmlio/index.shtml
Projects Suggested	Projects Chosen	Azureus: http://azureus.sourceforge.net
Projects Suggested	Projects Chosen	Eclipse: http://www.eclipse.org
Projects Suggested	Projects Chosen	GPSylon: http://www.tegmento.org/gpsylon
Projects Suggested	Projects Chosen	JUnit: http://www.junit.org/index.htm
Projects Suggested	Projects Chosen	Maven: http://maven.apache.org
Other Projects	Projects Chosen	CaCMS: http://cacms.sf.net
Other Projects	Projects Chosen	Columba: http://www.columbamail.org
Other Projects	Projects Chosen	FreeGuide TV: http://freeguide-tv.sourceforge.net
Other Projects	Projects Chosen	Gham: http://www.hattrickitalia.org/gham
Other Projects	Projects Chosen	Hunt for Gold: http://huntforgold.sourceforge.net
Other Projects	Projects Chosen	JackSum: http://www.jonelo.de/java/jacksum
Other Projects	Projects Chosen	Jython: http://sourceforge.net/projects/jython
Other Projects	Projects Chosen	Tapestry: http://tapestry.apache.org
Other Projects	Projects Chosen	WTflash: http://sourceforge.net/projects/wtflash

The assignment was to: (1) get an overview of the project, (2) identify the parts to which they would contribute code, (3) contribute, (4) write a report recalling their experience.

2.3 Outcomes

The students generally spent a lot of time and produced quality software. Their contributions include the world map plug-in integrated in the latest versions of Azureus, various bugfixes for the Eclipse Maven plug-in, CSV-exporting Tapestry components, bugfixes and extension of the GPL Hattrick Manager, bugfixes of JUnit, bugfixes and major improvements of the Columba Mail Client, extensions of GPSylon, bugfixes of the FreeGuide TV, a GUI for JackSum, extensions of WTFlash, extensions to the game "Hunt for Gold" and finally extensions to CaCMS to include WebDav support. Only to mention the projects that produced code that is shipping in the products. The course's wiki page[1] provides much more details.

3. EVALUATION OF MOTIVATION

3.1 Background

Students who are motivated by a task are more likely to succeed in solving it. According to Pintrich, students' motivation is strongly correlated with their academic success [5]. In general, students are more motivated if [5]:

1. they believe that they are able to solve the task at hand.

2. they feel in control of their learning.

3. they are personally or situationally interested in the task.[2]

[1]http://wiki.se.inf.ethz.ch/tjp_06/index.php/Project_page

[2]Personal interest describes a disposition of an individual to be attracted to a particular activity or topic while situational interest describes a state of an individual where the interest results from the task itself.

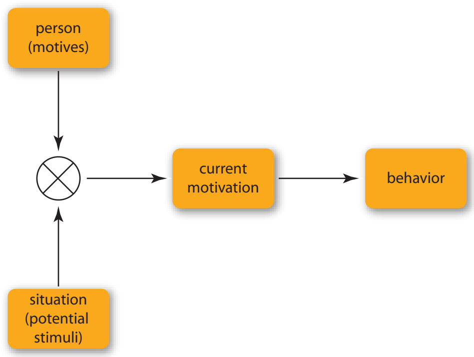

Figure 1: Basic model of classical motivational psychology [6]

4. they believe what they are doing is valuable and useful for themselves.

5. they have social and/or academic goals that they pursue.

By comparing the above motivational influences for the open source project approach and the traditional project approach, one can expect the open source project to increase personal and situational interest (3) and to give them the impression that the task is more valuable and useful (4) than the traditional projects'. One can also expect that the traditional project approach is better at providing assurance of success (1) and impression of control (2).

To assess the validity of using an open-source project against using a traditional project, a good indicator is to compare the motivation associated to both projects. The Questionnaire on Current Motivation (QCM) [6] assesses the current motivation of students working on a specific task and therefore is well suited for such purpose. The QCM is based on the "classical" model of motivational psychology [6] (see Figure 1) that states that personal and situational factors influence the current motivation which in turn influences behavior i.e. learning. It also benefits from a whole theoretical psychology framework that allows to easily compare and aggregate questions.

The QCM itself uses 18 items (questions) that measure four factors of current motivation: *anxiety* (fear of failure), *probability of success*, *interest*, and *challenge*. The full questionnaire is available online.[3] The QCM items formulate statements for which students assign 1 to 7 points depending on how much they agree with the statements (1: totally disagree, 7: totally agree). As an example:

I enjoy problem solving tasks like the ones that emerge in the project work. ☐ *1* ☐ *2* ☐ *3* ☐ *4* ☐ *5* ☐ *6* ☐ *7*

3.2 Results

To assess the motivational implications of the open source project assignment, the students of the Java programming course filled in the QCM twice: once just after receiving the project description, and the second time at the end of the project. The control course had a similar setting. In the rest of the section we use the following abbreviations:

QCM_{Et1}: questionnaires of the experimental group at the beginning of the open source project

QCM_{Et2}: questionnaires of the experimental group at the end of the open source project

QCM_{Ct1}: questionnaires of the control group at the beginning of the project

QCM_{Ct2}: questionnaires of the control group at the end of the project

The responses QCM_{Et1}, QCM_{Et2}, QCM_{Ct1}, and QCM_{Ct2} can be used to compare the two groups in two dimensions:

- Motivational differences between the experimental and the control group (comparing QCM_{Et1} to QCM_{Ct1} and QCM_{Et2} to QCM_{Ct2}).

- Motivational changes over time for both of the groups (comparing QCM_{Et1} to QCM_{Et2} and QCM_{Ct1} to QCM_{Ct2}).

Differences at the beginning. The 18 items of the QCM were combined into measures for each of the four dimensions. Based on this data, the analysis uses T-Tests[4] for independent sets to obtain the factors that differ significantly between the two groups for each of the two points in time (means are gathered in Figure 2).

The comparison shows that QCM_{Et1} and QCM_{Ct1} differ significantly for the factors *probability of success* and *anxiety* (see (∗a) respectively (∗b) in Figure 2). The interpretation of such a result is that when students begin to work on an assignment requiring to contribute to a real-world open source project, they feel more uncertain about their success and therefore their fear of failure is higher than for a traditional small "toy" project. More surprisingly, students are at this point in time not significantly more interested or more challenged by the open source project than by the traditional project.

Differences at the end. The comparison of QCM_{Et2} and QCM_{Ct2} shows that the factors *probability of success* and *anxiety* do not differ significantly between the experimental and the control group any more. Interestingly enough, the level of confidence is identical in both groups. The factor *interest* changed (with $p = 0.065$): the interest of the experimental group in their project grew while the interest of the control group diminished. The interpretation is that the fascination of working on an open source project settles only in after the first hurdle of basic understanding and involvement. Working on a traditional project is interesting in the initial design phase but looses fascination over time.

Another interesting outcome can be detected by comparing individual statements. First, the statement *"If I succeed with the project, I will feel a little proud of my proficiency"* was significantly higher at the end of the projects for the experimental group than for the control group. Second, the statement *"I would also work on a project like that in my free time"* also produced a significantly higher result for the experimental group. This is consistent with the following

[3] See http://se.ethz.ch/people/pedronim/qcm.pdf, for the sake of the reviewing process, the questionnaire is included in the Appendix.

[4] T-Tests allow to determine if the means of data differ significantly. In general, this is assumed to be the case if the calculated value $p < 0.05$.

intuitive interpretation. On one hand, contributing to a real-life project is very likely to make students proud and may even make them wish to continue contributing after the mandatory work is completed. On the other hand, having completed a project specifically designed for a course is not that rewarding.

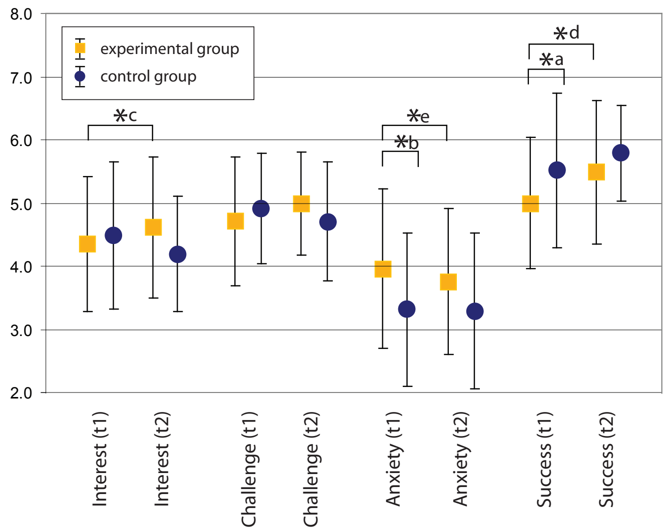

Figure 2: Mean and standard deviation of the four factors. (*) denotes significant differences ($p < 0.05$).

Comparison over time for both groups. The second part of the evaluation was done using T-Tests for paired sets of data to obtain significant changes over time. These tests show that for the experimental group the values for the factors *interest*, *anxiety*, and *probability of success* significantly improved (see (∗c), (∗d), and (∗e) in Figure 2). It seems that students working on the open source project first underestimate their capabilities and then gain confidence. This results in a significant increase of the *probability of success* and a reduction of the *anxiety*. The increase of *interest* is probably due to students beginning to understand the challenging and interesting sides of their programming project while working on the open source project. For the control group no significant change occurred during the project.

3.3 Outcome

The open source project results in a somewhat more unstable situation. Students start with a higher level of fear of failure and a lower level of probability of success, while gradually gaining more confidence and finally showing a deeper interest in the subject. In particular, students feel more proud of completing the open source project and are more likely to deepen their knowledge by continuing to work on it after they finished the official part of the work. This conclusion verifies the assumptions from section 3.1 which stated that students working on the open source project value their work more, but are less confident in their capabilities and control. With the present data it is not possible to declare one of the approaches definitely better than another, but using open projects is as good as using traditional projects and helps students build self-esteem.

4. COMPLEMENTARY ITEMS

The previous section showed how an open source project impacts on students' motivation. To estimate students' activity, learning effect, and commitment additional items complement the QCM. These questionnaires were distributed to the students of the experimental course at three occasions during the course period - at the beginning, in the middle, and at the end.

From over thirty additional items, three are detailed here. The first item assesses *activity*. This item addresses how much time is spent on the open source project. The second item - *learning effect* - quantifies the students' perception on the knowledge learned because of the open source project. The third item is the *commitment* that results from the project work. This third item was evaluated only at the middle and at the end of the course. The results of these complementary three items help to identify improvements for a future course.

4.1 Results

The item addressing the *activity* was: "I am putting much effort into the project". Students assigned 1 to 5 points capturing how much they agree with this statement (1: totally disagree, 5: totally agree). Figure 3 shows that the mean effort invested in the open source project drops towards the middle of the project's lifespan.

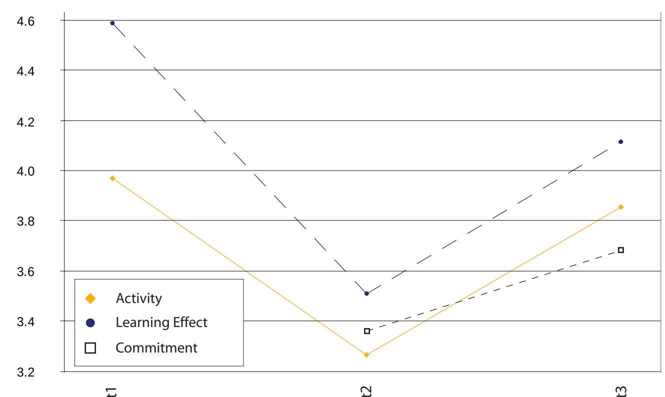

Figure 3: Means of *activity*, *learning effect* and *commitment*

The statement "I am learning much by doing the project" assessed the *learning effect* that students attribute to the open source project. As for the *activity*, the *learning effect* felt decreases in the middle of the course and rises again towards the end (see Figure 3).

The item to measure the *commitment* of students was included only twice: at the middle and at the end of the course. It stated "I feel responsible to make the project a success." and the resulting averages show an increase towards the end of the course (see figure 3). Additional comments made by students in the questionnaires support this observation: "*It was a lot of work, but very cool to have taken part in an open source project.*"

The evaluation for the three additional items used a variance analysis with repeated measures to find out whether the averages between points in time differ significantly. This was the case for all of the presented items.

4.2 Outcome

At first sight, the decrease of *learning effect* and *activity* in the middle of the course might be surprising. But it is important to see the course and the project in the context of the other activities the students take part in. In fact, students were busy with midterm exams for other courses at this point in time. To overcome the problem of multiplied pressures, instructors need to consider both the project phases and the other obligations students have at university. In particular, during the time consuming phase of design and implementation lectures could be reduced or transformed into interactive lab sessions.

5. CONCLUSIONS AND FUTURE WORK

The paper has two main contributions. Firstly, we show that having students collaborate on open source projects within the frame of a course is interesting and has advantages over using an artificial project. Secondly, using a study backed up by research in psychology enables a scientifically sound evaluation of the approach.

This article described our first attempt at using open source projects within a course. It showed that students were obviously afraid at first but felt more proud of their achievement in the end. The next iterations of the course will integrate this result and find solutions to cope with students' fear. A first approach is to give them a much more detailed standard operating procedure to get started with their projects. Such an approach could include the following steps: (1) use the open source software as a tool, (2) explore the code in search for programming patterns, (3) identify weak points or adequate extensions that result in a contribution, and (4) design, implement and deliver the code. As a second measure, we plan to have several groups of students work on the same project (but not the same subject) so that they constitute a community within the open source community. This would then show the projects in a more friendly and social way than currently.

6. ACKNOWLEDGMENTS

We thank Barbara Moser and Jessica Schnelle for their contributions to this work. Thanks also to the students of *Techniques of Java Programming* for participating in the experiment and filling in the questionnaires. We also thank Bertrand Meyer, Piotr Nienaltowski, and Volkan Arslan for their full cooperation and the students of *Concurrent Object-Oriented Programming* for acting as a control group.

7. REFERENCES

[1] E. Allen, R. Cartwright, and C. Reis. Production programming in the classroom. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 89–93, New York, NY, USA, 2003. ACM Press.

[2] D. Carrington and S.-K. Kim. Teaching software design with open source software. *Frontiers in Education*, 3(33):S1C– 9–14, November 2003.

[3] C. P. Fuhrman. Appreciation of software design concerns via open-source tools and projects. In *10th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, at 20th European Conference on Object Oriented Programming (ECOOP)*, Nantes, FR, July 2006.

[4] K. J. O'Hara and J. S. Kay. Open source software and computer science education. *J. Comput. Small Coll.*, 18(3):1–7, 2003.

[5] P. R. Pintrich. A motivational science perspective on the role of student motivation in learning and teaching contexts. *Journal of Educational Psychology*, 95(4):667–686, 2003.

[6] F. Rheinberg, R. Vollmeyer, and B. D. Burns. QCM: A questionnaire to assess current motivation in learning situations. *Diagnostica*, 47:57–66, 2001.

[7] M. J. Wolf, K. Bowyer, D. Gotterbarn, and K. Miller. Open source software: intellectual challenges to the status quo. In *SIGCSE '02: Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, pages 317–318, New York, NY, USA, 2002. ACM Press.

APPENDIX

QCM questions, adapted from [6] (appearing here for purpose of anonymity):

(I) I enjoy problem solving tasks like the ones that emerge in the project work.
(S) I think I can tackle the difficulties of the tasks involved in the project assignment.
(S) Probably, I will fail solving the project assignment.
(I) What I like about the project assignment is the role of the researcher that discovers connections.
(A) I feel pressure having to perform well solving the project assignment.
(C) This project assignment is a real challenge for me.
(I) After reading the instructions, the project assignment seemed very interesting to me.
(C) I am very curious how well I will do in this project.
(A) I'm a bit afraid of being embarrassed by my performance in the project.
(C) I am determined to work very hard for the project.
(I) I enjoy doing the project, I would not need any gratification.
(A) Failing the project assignment would embarrass me.
(S) I believe everyone can succeed in doing the project.
(S) I believe I won't succeed in the project assignment.
(C) If I succeed with the project, I will feel a little proud of my proficiency.
(A) If I think about the project, I am a bit worried.
(I) I would also work on a project like that in my free time.
(A) The requirements of the project work paralyze me.

(C): Challenge
(I): Interest
(S): Probability of success
(A): Anxiety