Quasi-Crowdsourcing Testing for Educational Projects *

Zhenyu Chen, Bin Luo
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
Software Institute, Nanjing University, Nanjing 210093, China
{zychen, luobin}@software.nju.edu.cn

ABSTRACT

The idea of crowdsourcing tasks in software engineering, especially software testing, has gained popularity in recent years. Crowdsourcing testing and educational projects are natural complementary. One of the challenges of crowdsourcing testing is to find a number of qualified workers with low cost. Students in software engineering are suitable candidates for crowdsourcing testing. On the other hand, practical projects play a key role in software engineering education. In order to enhance educational project outcomes and achieve industrial-strength training, we need to provide the opportunity for students to be exposed to commercial software development.

In this paper, we report a preliminary study on crowdsourcing testing for educational projects. We introduce three commercial software products as educational testing projects, which are crowdsourced by our teaching support system. We call this "Quasi-Crowdsourcing Test" (QCT) because the candidate workers are students, who have certain social relations. The investigation results are encouraging and show to be beneficial to both the students and industry in QCT projects.

Categories and Subject Descriptors

K.3 [**Computers and Education**]; D.2.9 [**Software Engineering**]: Management

General Terms

Education

Keywords

Crowdsourcing test, system testing, educational projects

1. INTRODUCTION

Educational projects are important to technology and engineering education [8]. For software engineering students, educational projects can integrate knowledge and skills from earlier studies. The projects give a chance to learn how to develop and maintain software in practice. In order to effectively train students how to accomplish software engineering tasks in the real world, it is better to use industrial software than others in educational projects [5].

Although many educators know the benefits of using industrial software, several challenges should be overcome for successful educational projects [1]. Software obtained directly from industry may not be suitable for education, because of limited time and space of teaching. The industrial software needs to be pruned and modified in order to satisfy certain education objectives. Hence, we need to motivate software companies so that they have enthusiasms to help our teaching [7]. Software used in projects should be interesting to students and allow use of modern technologies. Software under test in projects should be large, at least modest, so that students can understand the necessity of testing and the advantages and disadvantages of different testing methods. On the other hand, the functionalities of software in projects should be easily understood by students to focus on testing methods [4].

Recently, the idea of crowdsourcing software engineering tasks has gained popularity [6]. "Crowdsourcing" is to outsource a task to workers recruited from a large pool of individuals with whom one has no direct relationship [3]. Crowdsourcing is more flexible to increase or decrease the number of tasks than traditional ways. In crowdsourcing markets, the performance of workers may be poor for the tasks, even they have already passed some evaluations. It is important, but not easy, to find a number of qualified workers. Because of the workers' uneven abilities, the results submitted by workers should be inspected to assess their effectiveness. Moreover, the crowdsourcing tasks should be decomposed in a reasonable way so that workers can do it independently and then the results can be combined for the final tasks.

In this paper, we report a preliminary study on crowdsourcing testing for educational projects. Projects with commercial software are interesting to students. These projects can enhance educational outcomes and achieve industrial-strength training for students. Students in software engineering are suitable for crowdsourcing testing, because they have good knowledge of testing and have enthusiasms to try new features of software. The industry companies can organize a number of qualified workers from students with low

*This work is partially supported by the National Natural Science Foundation of China (61170067, 61373013).

ICSE '14, May 31-June 7, 2014, Hyderabad, India

cost. This is attractive to software companies. The natural complementary of crowdsourcing testing and educational projects motivates us to combine them together.

The industry partner outsources three software products under test (Baidu-Input[1], Baidu-Browser[2], and Baidu-Player[3]) to our students as educational projects. Test requirements of software products are decomposed to several independent testing tasks by industry actors. The students themselves select testing tasks following instructions in our teaching support system. The results submitted by the students will be inspected and evaluated by industry actors. We call this "quasi-crowdsourcing testing" (QCT) in this paper because the candidate testers, i.e. our students have certain social relations. We do not know what extend of the social relations affect the results of crowdsourcing testing.

To the best of our knowledge, it is the first educational report on crowdsourcing testing for students by industry partners. The investigation results are encouraging and show to be beneficial to both the students and industry. The main observations are: (1) QCT projects are more interesting to students than the traditional testing projects. (2) The educational effects of QCT projects and traditional projects are complementary. (3) More detailed guides for students are needed to improve the effectiveness of QCT projects.

2. EDUCATIONAL PROJECTS

2.1 Background

The "Software Testing and Quality" is a core course for junior students in software institute [2]. The duration of the course is 12 weeks (3 hours per week). In order to teach students to understand various software testing methods and skillfully use testing methods in practice [2], we design three types of educational projects in this course.

Unit Testing (UT): Each student is required to write unit test scripts for four programs. Test scripts should cover as many statements as possible in source code. Teaching Assistants (TAs) run tests and collect the coverage information, and then evaluate their projects.

Web Testing (WT): Each student is required to write test scripts in Selenium[4] for four Web applications. Test scripts should cover as many Web elements as possible. TAs run test scripts and collect the coverage information by a tool to evaluate their projects.

Quasi-Crowdsourcing Testing (QCT): The industry partner outsources three software products to our students. Students are required to select one of software products. Students must manually test it following the test requirements and instructions provided by industry actors. Students should submit test results (texts and screenshots). The industry actors and our TAs manually inspect test results to evaluate their projects.

2.2 QCT Projects

P1: The first project is Baidu-Input on Android, which can support pinyin, stroke, handwriting, intelligent voice, and other input methods. The industry actors provide 10 functionality sets. One tester can select one functionality set and each functionality set can be selected by at most two testers. Students use different mobile phone and different versions of Android. Therefore, we require students to report compatibility problems, besides functional testing. The students are required to finish P1 in 3 days.

P2: The second project is Baidu-Browser, which is a Web browser. The industry actors provide 7 functionality sets for regression testing. One tester can select 3 functionality sets. The usability plays a key role in a successful Web browser. Hence, we require students to report usability problems, besides functional testing. The students are required to finish P2 in 5 days.

P3: The third project is Baidu-Player, which is a multimedia player. The industry actors provide 3 performance testing scenarios. One tester should cover all of these 3 test scenarios. The usability plays a key role in a successful multimedia player. Therefore, we require students to report usability problems, besides functional testing. The students are required to finish P3 in 5 days.

2.3 Score Analysis

For QCT, the industry actors and TAs evaluate the projects by inspecting test results (texts and screenshots). In order to investigate relations between different types of testing projects, we draw three scatter plots and calculate the Pearson correlation coefficients (R) in Figure 1.

Figure 1 shows the scatter plots between UT and WT, UT and QCT, WT and QCT, respectively. The results indicate that there is a correlation between UT and WT (R=0.38). There is nearly no correlation between UT and QCT (R=-0.06), WT and QCT(R=0.09). In our course, UT and WT projects require students to write test scripts. It trains students' test programming skills. These two projects are evaluated by coverage ratio. The understanding of software functionalities is not evaluated. Students' programming skills are not evaluated in QCT project. It only requires students to write tests in text and capture outputs in screenshot. QCT projects encourage students to reveal real bugs with their understanding software features.

In summary, the educational outcomes of UT and WT projects are similar. These two types of projects train students' ability of test programming for different coverage criteria. The educational outcomes of QCT projects are different from UT and WT projects. QCT projects train students' ability of understanding software products and revealing real bugs. In our experiences, some types of testing projects (such as functional testing, usability testing) are suitable for using commercial software products by crowdsourcing testing. These software products are more interesting to students than open-source software. The students are more likely to obtain a sense of achievement in QCT projects than other projects.

3. INDUSTRY FEEDBACK

QCT is a cooperation between industry and university. The basic motivation is the Win-Win cooperation for industry and university. Our experiences of QCT projects show that the industry partner can benefit in several aspects in crowdsourcing testing.

Software companies can get a number of high quality workers with low costs in crowdsourcing testing. The junior and senior students in software engineering always have some professional skills on software testing. They are suitable can-

[1]http://shurufa.baidu.com/
[2]http://liulanqi.baidu.com/
[3]http://player.baidu.com/
[4]www.seleniumhq.org/

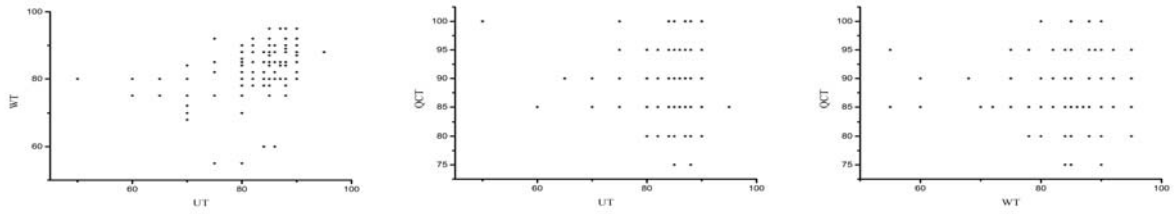

Figure 1: Score Correlations, UT and WT(R=0.38), UT and QCT(R=-0.06), WT and QCT(R=0.09)

didates for crowdsourcing testing, especially testing Internet software products. By cooperating with QCT projects, software companies can recruit a group of professional workers with very low cost.

Table 1: Summary of Industry Feedback

Project	P1	P2	P3	A	B
# Tester	20	55	54	13	14
# F-test	274	231	252	316	574
# V-test	186	47	62	113	114
% V-test	67.88%	20.35%	24.60%	35.76%	19.86%

Table 1 summarizes the feedbacks from industry. "# Tester" indicates the number of testers. Testers will remark a test as "failing" if the test output is not consistent with the expected output. "# F-test" indicates the number of failed tests marked by testers. In practice, testers may make some false positives in crowdsourcing testing. The industry actors will manually inspect test outputs to decide whether they are valuable. "# V-test" and "% V-test" indicate the number and the percentage of valuable tests, respectively. In our experiences, the industry actors may ignore some failed tests for some non-technical reasons. For example, students may reveal some bugs, which have been revealed in the previous testing and the developers cannot fix them. Hence, TAs inspect test outputs again and discuss with the industry actors for fair project evaluation, as final results shown in Table 1.

In order to compare the performances of student testers, the industry partner provides two groups of testers recruited from their open crowdsourcing platform[5]. Group "A" are the senior testers and group "B" are the junior testers based on their past performances in crowdsourcing testing. As Table 1 shows, our students can achieve the group "A" in all projects. The industry actors feedback that the test results submitted by our students are more detailed, professional and useful than the results by other testers. The original scores of P1 are very low. The industry actors ignore many usability bugs and some functional bugs that cannot be fixed in time. TAs discuss with the industry actors again to review the bugs again for fair evaluation.

There are other benefits for industry partners. Software companies can get higher quality graduates from education institutions. Students with complete knowledge and hands-on practice with techniques can improve the practice of industrial software engineering. The industry actors can also learn about some state-of-the-art testing methods to improve their techniques. Software companies have the opportunity to transfer and use our research results in the future cooperation. For example, the industry partner has set up a research project to use our test prioritization techniques [9] to reduce the costs of manual output inspection.

[5]http://test.baidu.com

4. STUDENT FEEDBACK

In order to study the educational outcomes of QCT projects, we conducted a survey on the students after the course. This is an anonymous online survey to understand the students' true feelings. As a result, 43 students did the online survey. Due to the limited pages, we select three questions with answers, which are most related to this paper, to explain our experiences of QCT projects.

4.1 Student Survey

Q1: Are QCT projects more interesting than UT and WT projects?

No.	Answer	# Answer	% Answer
A	More	30	69.77%
B	Less	2	4.65%
C	Almost the same	11	25.58%
D	Hard to say	0	0.00%

Software testing projects are easier to be boring for students than other software engineering projects. Hence, it is important to design some interesting projects to keep the students' attention. The survey results show that 69.77% students think that QCT projects are more interesting than UT and WT projects. An important reason is that the software products in QCT projects are popular in Internet. Students can reveal real bugs and improve the real software products for a sense of achievement. There are fewer complaints of QCT projects than UT and WT projects from students in the teaching support system. The students finished QCT projects with high quality, although we limited 3-5 days for the projects. As a result, the students get higher scores as shown in Figure 1.

Q2: Do you learn more in QCT projects than UT and WT projects?

No.	Answer	# Answer	% Answer
A	More	6	13.95%
B	Less	19	44.19%
C	Almost the same	15	34.88%
D	Hard to say	13	6.98%

44.19% students say "less" and 34.88% students say "almost the same" for this question. The result seems negative, but it is reasonable. UT and WT projects focus on the students' skills of programming test scripts and building automatic test frameworks. This is a so-called "hard"

ability for students. QCT projects focus on the students' capability of understanding software functionalities and test requirements. This is a so-called "soft" ability for students. However, we believe that both two types of abilities are important to software testing, as well as software engineering. That is, QCT projects could be complementary to UT and WT projects for software engineering education.

Q3: Which ones are your motivations for QCT projects?

No.	Answer	# Answer	% Answer
A	Score	33	76.74%
B	Reward	6	13.95%
C	Achievement	22	51.16%
D	Interesting	22	51.16%
E	Others	2	4.65%

It is not surprising that scores are important to students. Students with excellent performances will get a shopping card as a reward. However, only 13.95% students are motivated by rewards. The value of reward seems too weak for students. It is challenging to attract students for crowdsourcing testing after this course. We will study how to price crowdsourcing testing in the future.

4.2 Benefits to Students

QCT projects are more interesting to students and are complementary to other testing projects. Moreover, we believe that QCT projects can also provide some additional benefits to the students.

Students can get deep insights on specific industrial problems in QCT projects, because software products under test are carried out with some industrial goals. QCT projects can give students a better opportunity to assess themselves for their knowledge about testing methods than other projects can do. Moreover, students get acquainted with practical problems that are encountered by practitioners during their work. It may be unknown to the students prior to facing those problems themselves. Students can get some experiences of revealing real bugs, which are different from injected bugs in other projects.

Students can participate in state-of-the-practice projects. Three software products in QCT projects are popular in Internet. The first project requires students to install Baidu-Input in their mobile phones. Some issues on mobile application testing should be considered in this project. The Web browser and multimedia player are also widely used in Internet. The students are required to use auxiliary tools to collect performance information. Students are exposed to industrial projects that may be more cutting-edge than those usually taught in other projects.

5. LESSONS AND DISCUSSION

It is the first attempt to introduce crowdsourcing testing into our course as educational projects. The QCT projects can be of benefit to both industry and students. However, The flip side of the coin is the costs that are incurred by the students and industry actors.

In some cases, students have spent time learning something that will be of no benefit to them in industrial projects. In order to reduce this risk, we require the industry actors to provide the test requirements of software products before distributing QCT projects. We will analyze and discuss the technique requirements for these projects. As a result, we select three software products, Baidu-Input, Baidu-Browser, Baidu-Player, which were widely used by our students, for QCT projects. Since the students are easy to understand the primary functionalities of software, it can reduce the evaluation cost of techniques and give some confidence in their effectiveness. However, this may reduce the potential chances to learn new and cutting-edge techniques that might provide them with a much-desired skill.

There are some risks of evaluating students' projects based on his or her performance by industry actors. The value orientation of industry actors is different from that of educators. It is well known that there are no standard criteria to some software testing tasks, such as usability testing. On the other hand, the types and difficulties of software products in QCT projects are mixed. For example, a software product is mature and fully tested. Only a few of bugs remained in software. It is difficult to reveal bugs and then form "valuable" results for industry actors. Students in this project will be risky to get a low score. A positive point is that the results of QCT projects are evaluated in industrial environments rather than educational environments.

It is important to evaluate projects fairly in software engineering education. The students use texts and screenshots to describe the results of functional testing, compatibility testing, usability testing, and performance testing. It is difficult, or even impossible, to evaluate test results automatically by existing techniques. The test results should be inspected manually by industry actors and TAs. For industry, only a part of failed test results will be inspected to reduce costs, by using some prioritization techniques [9]. For education, all test results should be inspected to evaluate the performances of students. This will increase the costs of industry actors. However, the total costs of testing these software products are reduced, because the industry partner does not need to spent time in organizing qualified workers in crowdsourcing testing.

6. REFERENCES

[1] A. Bertolino. Software testing research: Achievements, challenges, dreams. FOSE'07, pages 85–103, 2007.

[2] Z. Chen, J. Zhang, and B. Luo. Teaching software testing methods based on diversity principles. CSEE&T'11, pages 391–395, 2011.

[3] J. Howe. The rise of crowdsourcing. *Wired*, 14(9), 2006.

[4] C. Kaner and S. Padmanabhan. Practice and transfer of learning in the teaching of software testing. CSEE&T'07, pages 157–166, 2007.

[5] T. Lethbridge, J. L. Díaz-Herrera, R. J. LeBlanc, and J. B. Thompson. Improving software practice through education: Challenges and future trends. FOSE'07, pages 12–28, 2007.

[6] F. Pastore, L. Mariani, and G. Fraser. Crowdoracles: Can the crowd solve the oracle problem? ICST'13, pages 342–351, 2013.

[7] T. J. Reichlmay. Collaborating with industry: strategies for an undergraduate software engineering program. SSEE'06, pages 13–16, 2006.

[8] L. van der Duim, J. Andersson, and M. Sinnema. Good practices for educational software engineering projects. ICSE'07, pages 698–707, 2007.

[9] S. Yan and et al. A dynamic test cluster sampling strategy by leveraging execution spectra information. ICST'10, pages 147–154, 2010.