2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)

Incorporating Real Projects into a Software Engineering Undergraduate Curriculum

Rafael Chanin
School Of Technology
PUCRS
rafael.chanin@pucrs.br

Jorge Melegati
Faculty of Computer Science
Free University of Bozen-Bolzano
jmelegatigoncalves@unibz.it

Afonso Sales
School Of Technology
PUCRS
afonso.sales@pucrs.br

Mariana Detoni
School Of Technology
PUCRS
mariana.detoni@acad.pucrs.br

Xiaofeng Wang
Faculty of Computer Science
Free University of Bozen-Bolzano
xiaofeng.wang@unibz.it

Rafael Prikladnicki
School Of Technology
PUCRS
rafaelp@pucrs.br

***Abstract*—Software engineering researchers and practitioners are increasingly more concerned about non-technical issues like user involvement and interaction as a way to improve software development process efficiency. This issue is also present in software engineering education. The IEEE/ACM software engineering guidelines highlights that an undergraduate course in this matter should have a real-world basis. In this paper, we present an undergraduate program that connect students with real-world projects throughout their studies. To evaluate educational results, we performed a survey with 111 students from this software engineering program. The results indicate that students in the end of this program has a much better chance of taking users' desires into consideration instead of focusing on software implementation.**

***Index Terms*—software engineering, software engineering education, real-world projects.**

I. Introduction

Software engineering is increasingly more dependent on user involvement, being more determinant to system success than schedule and budget goals achievements [1]. In a 2009 follow-up from her seminal paper, Shaw [2] speculates that, in the following ten years, problems faced by software engineers will be more "situated in complex social contexts, and delineating the problems' boundaries is increasingly difficult." This phenomenon influences how software engineering should be taught.

The 2015 version of the IEEE/ACM curriculum guidelines [3] states that a software engineering undergraduate course should have a real-world basis. It was already present in the 2004 version and, to carry that out, Lethbridge *et al.* [4] mentioned a suggested approach to distribute "discussions of process and professionalism issues throughout the curriculum". In this paper, we present an initiative in Brazil to involve students from a software engineering undergraduate program with real-world tasks. To evaluate their concern about user involvement, we performed a survey in which a scenario was presented to students. Outcomes were coded, and we performed a logistic regression to verify which characteristics determine students' responses. The results indicated that students in the end of the program have 3.7 more chance of focusing on user involvement than a student in the beginning of the program.

A. Software Engineering Experimental Agency (AGES)

The Software Engineering Experimental Agency (AGES, in Portuguese) is a laboratory and also the main track of the Software Engineering undergraduate program at PUCRS in Brazil. From the total of 3,200 hours of the program, students spend at least 480 of them in AGES. In this lab, students work in real projects interacting with real contractors, customers, users and peers from other semesters. AGES is split into four modules that take place every other semester. Each module can be perceived as a course; however half of the time is allocated in a fixed schedule allowing interactions with other students (also from other semesters) and the remaining is freely accommodated by students. Additionally, each module has a different focus: basic programming and unit testing; database, software requirements and coding; testing and software architecture; project management. Projects are selected according to a list of requirements. The most important one is that the stakeholder has to meet students twice a month. In regards to assessment, AGES modules do not have written exams and grades are based on a self evaluation, a group evaluation, technical knowledge acquired and soft skills development.

II. Research Design

In order to understand how students perceive the importance of user involvement in a software project, we ran a survey, following Wohlin *et al.*'s guidelines [5], on students from all levels in the program. The online survey consisted of three sections. The first encompasses demographics questions such as age, program semester, gender, employment status and, if employed, the organization size and role in the organization. The second section consisted of a small scenario presenting a software idea and asking students how they could contribute

2574-1934/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSE-Companion.2019.00099

to the proposed project. The last set of questions gathers their own perception on their knowledge on aspects that foster user involvement like agile methodologies and lean startup. To avoid biasing students, this section also contained questions about other eight different topics: database, coding, software maintenance, testing, software architecture, and blockchain. Students had to choose from 0 (I have never heard of this topic) to 4 (I have a lot of experience with this topic).

III. Results

We gathered 111 responses across all levels of the program. In order to analyze responses, three of the authors read and labeled each answer individually and classified them into one of these categories (codes): **personal opinion** (e. g. "It is a bad idea"), **development** (e. g. "I would gather the requirements in order to start coding"), and **user involvement** (e.g. "I would talk to dog owners to verify whether this is a problem to them"). Then, they compared the results and agreed upon a single code for each student answer. This process was done in order to minimize misinterpretation and to improve the validity of the classification. Since the point was to focus on the software engineering aspects of the project, and not at the idea *per se*, we decided not to take answers classified as "personal opinion" into account. We understood that these students did not understand the purpose of the question. The final set consisted of 73 valid records.

Then we applied logistic regression to verify if these variables could determine if the students would involve users during the development process. Given the sample size, it was necessary to diminish the number of possible combinations. In order to do that, we created two different categories for each variable: semester (*first to fourth* or *fifth to eighth*); *employed* or *not employed*; *job position* (developer or intern); *agile knowledge* (little/no experience or some/a lot of experience); and *lean startup knowledge* (little/no experience or some/a lot of experience).

After running the model with all variables against our codes with a 95% confidence level, using the SPSS[1] software, the only variable that presented a statistically significant result was the semester students were in as shown in Table I.

TABLE I
Logistic Regression Results

Variable	B	Standard error	Significance
Age	-0.212	0.560	0.706
Semester	**1.329**	**0.659**	**0.044**
Employed?	0.372	0.807	0.645
Job position	-0.039	0.342	0.909
Agile Knowledge	0.300	0.539	0.579
Lean Startup Knowledge	-0.743	0.926	0.423
Constant	-1.099	1.707	0.520

Since we have found a statistically significant result for the variable "Semester", we can calculate the odds ration based on the B coefficient: e^B = 3.775. This means that students from the 5th, 6th, 7th, and 8th semester present 3.775 higher chance of focusing on **user involvement** than those students from the beginning of the program.

It is interesting to notice that all other predictor variables did not present statistical correlation with the outcomes. One may think that experience or age could influence the outcome, but that did not happen in our model. The results found are very interesting and well connected with the intention of this software engineering program.

IV. Conclusion

This paper presented a software engineering undergraduate program that gives students the opportunity to work in real-world projects throughout the whole course. Through a survey run across all student levels of this program, our results showed that students in the end of the program have a bigger probability of taking users' desires into account. Such result substantiate claims from the literature that distributing touch-points to the industry throughout the curriculum could be a way to make courses more linked to the market. Even though these are just preliminary results, there is at least an indication that further research can be performed in order to verify the effect of incorporating real projects into an educational environment.

Besides that, the lack of statistical support that the agile knowledge methodologies increases the probability of concern about user involvement is a warning signal. Given that user feedback is key for these methodologies, future work could be done to check if students and practitioners are really capturing these main concepts. This matter is beyond the scope of this paper.

Acknowledgment

This work is partially funded by FAPERGS (17/2551-0001/205-4). The authors would like to thank the students who participated in this study.

References

[1] M. Bano, D. Zowghi, and F. da Rimini, "User satisfaction and system success: an empirical exploration of user involvement in software development," *Empirical Soft. Eng.*, vol. 22, no. 5, pp. 2339–2372, 2017.

[2] M. Shaw, "Continuing prospects for an engineering discipline of software," *IEEE Software*, vol. 26, no. 6, pp. 64–67, 2009.

[3] IEEE/ACM Joint Task Force on Computing Curricula, "Software engineering 2014, curriculum guidelines for undergraduate degree programs in software engineering," 2 2015. [Online]. Available: http://securriculum.org

[4] T. C. Lethbridge, J. Diaz-Herrera, R. J. LeBlanc, and J. B. Thompson, "Improving software practice through education: Challenges and future trends," *Future of Soft. Eng. (FOSE 07)*, vol. 2, no. 87, pp. 12–28, 2007.

[5] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[1] Version 20. https://www.ibm.com/analytics/spss-statistics-software