Poster: A matrix for analyzing projects in Software Engineering courses

Jakeline Marcos-Abed.
Computer Science Department.
Tecnológico de Monterrey.
Monterrey, N.L., México, 64849
jakeline@itesm.mx

ABSTRACT

[1] The careful selection of a project for Software Engineering courses is important from the point of view of the student, the teacher and the project user or client -if available. But, how do you determine whether to develop a complex fictional case, or a simple real-world project with a considerable amount of learning about the software lifecycle? How do you determine whether or not to develop a social project aimed at the community?

In order to analyze project characteristics a framework was elaborated, based in a matrix with attributes found in different academic projects carried out over 10 years, and weighted according to certain criteria.

It is expected that the result of this analysis will help professors choose the most appropriate software project to develop in class, depending on the characteristics of each project.

CCS CONCEPTS

Social and professional topics~Software engineering education

KEYWORDS

Software projects; software lifecycle; software engineering education.

1 INTRODUCTION

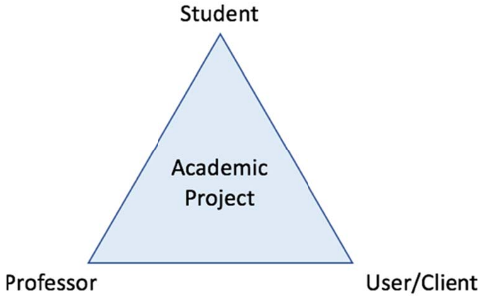
Two very important courses in the Computer Science curricula [1] are the Software Engineering Fundamentals course, taught in the third semester, and the Software Project Management course that is offered in the seventh semester in the analyzed university.

Within the computer science curricula, Software Engineering Fundamentals and Project Management courses are full of theory and concepts that students must learn [1]. But it is necessary that learning does not remain only in theory or readings from books and papers [2].

Project-based courses are highly valued in software engineering education. They allow students to experience some of the challenges they will face in their future work [3]. In addition, teamwork in projects is very important in order to learn different and very valuable communication and leadership skills [4].

With this kind of projects, students should be aware of the differences with real life, in terms of team size, requirements for compatibility with legacy systems, quality requirements, to name a few [5]. Furthermore, it is expected that the experience with these projects will provide students an understanding of project management, working in a team, professionalism, and exposure to the issues involved in developing complex projects [2]. As a consequence, we expect that students may turn out to be carriers of innovation when they join the business world [5].

2 PROJECT SELECTION



The software projects developed in a class range from a fictional case or case-study, to a complex real-world project. In the end, there is a wide variety of software projects, and each one has benefits for one or more entities of the triangle: student, professor, client. (see Fig. 1).

Figure 1: Academic triangle for software projects.

3 PROJECT MATRIX DESIGN

With the intention of analyzing academic project attributes, a matrix was created with the main characteristics of these projects, developed by students within the last 10 years for Software Engineering Fundamentals and Software Project Management courses (see Table 1). It is worth noting, that these characteristics

[1] Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195053

are inherent to each kind of project, and they were evaluated with a binary mode: 1 or 0. This value shows if the corresponding characteristic is present (1) –or not (0) in that project, according to how it was developed in that moment. Following is a list of these characteristics and the project-matrix:

1. **Understanding concepts**. Does the project provide sufficient understanding of software development lifecycle?

2. **Feasible scope**. Is the selected project feasible in one semester according to the project scope?

3. **Real user**. Will the software product be developed for a real user? This attribute represents a real-life project, with timing constraints and a product that will be useful for a client.

4. **Business processes**. Does the selected project involve a collateral learning of business processes? For instance, learning concepts of supply chain management or warehouse systems.

5. **Community service**. Does the selected project benefit the community?

6. **Replicability**. Is it a replicable project? Or is it just for one time? Some projects cannot be developed every semester.

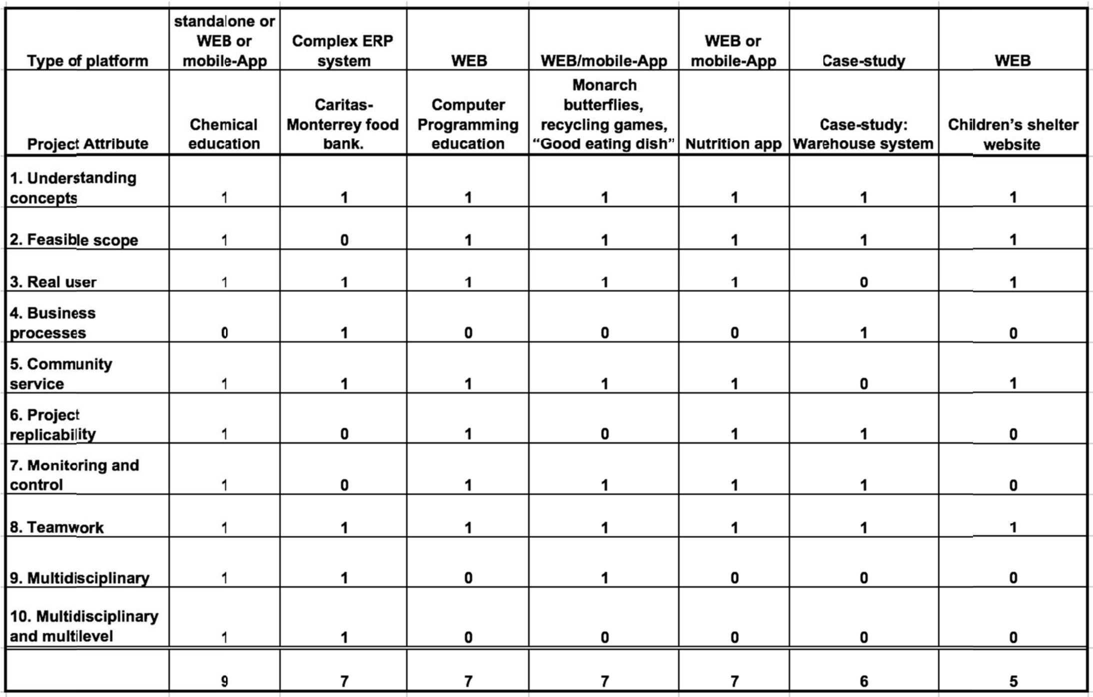
7. **Monitoring and control**. Is it easy for the professor and students to monitor and control the project?

8. **Teamwork**. Does the project promote teamwork? This is good for encouraging communication and leadership skills.

9. **Multidisciplinary teamwork**. This attribute means that teams are from different courses and disciplines.

10. **Multidisciplinary and multi-level teamwork**. This attribute means that teams are from different courses, different disciplines, and different levels as well.

Table 1: Project-matrix with project attributes.



Type of platform	standalone or WEB or mobile-App	Complex ERP system	WEB	WEB/mobile-App	WEB or mobile-App	Case-study	WEB
Project Attribute	Chemical education	Caritas-Monterrey food bank.	Computer Programming education	Monarch butterflies, recycling games, "Good eating dish"	Nutrition app	Case-study: Warehouse system	Children's shelter website
1. Understanding concepts	1	1	1	1	1	1	1
2. Feasible scope	1	0	1	1	1	1	1
3. Real user	1	1	1	1	1	0	1
4. Business processes	0	1	0	0	0	1	0
5. Community service	1	1	1	1	1	0	1
6. Project replicability	1	0	1	0	1	1	0
7. Monitoring and control	1	0	1	1	1	1	0
8. Teamwork	1	1	1	1	1	1	1
9. Multidisciplinary	1	1	0	1	0	0	0
10. Multidisciplinary and multilevel	1	1	0	0	0	0	0
	9	7	7	7	7	6	5

4 CONCLUSIONS

There is a wide variety of software projects to use in Software Engineering courses, and each one has benefits for one or more entities of the triangle: student, professor, client.

Each of the projects shown in this matrix has its advantages and disadvantages. For example, a case study is easy to control by the professor [6], however the product is discarded at the end of the course. On the other hand, it would be beneficial to use the effort of these projects for the common good [2]. However, a project as complex as that of Caritas-Monterrey food bank is difficult to complete in one semester.

As seen in table 1, the development of small applications for teaching Chemistry -or any subject, has shown to be a highly achievable project, with a good degree of difficulty and with a useful product for members of the community.

REFERENCES

[1] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM, New York, NY, USA

[2] Marcos-Abed, Jakeline. (2016). Using the effort of academic projects for the Community Service: A Software Engineering practical approach. 1-4. 10.1109/FIE.2016.7757498.

[3] Bosnic, Ivana & Cavrak, Igor & Orlić, Marin & Zagar, Mario. (2013). Picking the right project: Assigning student teams in a GSD course. Software Engineering Education Conference, Proceedings. 149-158. 10.1109/CSEET.2013.6595246.

[4] Shuto, Masashi & Washizaki, Hironori & Kakehi, Katsuhiko & Fukazawa, Yoshiaki & Yamato, Shoso & Okubo, Masashi. (2016). Learning Effectiveness of Team Discussions in Various Software Engineering Education Courses. 227-231. 10.1109/CSEET.2016.31.

[5] Ghezzi, Carlo & Mandrioli, Dino. (2005). The challenges of software engineering education. 637- 638. 10.1109/ICSE.2005.1553624.

[6] Frailey, Dennis. (2016). Teaching Project Planning with No Project. 37-45. 10.1109/CSEET.2016.25.