Analyzing main characteristics in Software Engineering projects

Jakeline Marcos-Abed
Computer Science Department
Tecnologico de Monterrey
México
jakeline@itesm.mx

***Abstract*—Innovative Practice. Work In Progress. One of the biggest concerns when teaching a Software Engineering course is how to make the content understood in a practical way. In this sense, what is recommended is to apply projects in class, projects that as far as possible should be related to the real world and that involve a true learning. To analyze the characteristics of these projects, a framework was elaborated based on a matrix with attributes found in different academic projects carried out over 10 years, and weighted according to certain criteria. This paper presents an analysis of the advantages and disadvantages for each kind of project, using a radar-graph to visualize the main characteristics of these. 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.**

Keywords— Software projects; software lifecycle; software engineering education.

I. INTRODUCTION

Software Engineering Fundamentals and Software Project Management courses are very important within Computer Science curricula. These courses are generally full of theory and it is recommended that they are supported by projects that relate theory and practice. The projects used are from a wide variety, like fictional cases where there is no client [1], software systems simulating the process of software development, or the use of project models that simulate a company in a way more realistic, from an organizational, process, and communication perspective [2].

In this sense, it has been seen that the more real the project, the better it is to understand the concepts and to keep students motivated. In fact, in project-based learning, the instructor has a less central role, and students take more responsibility for their own learning, which results in higher student involvement and motivation [3][4].

It is expected that the experience with these projects will provide students with an understanding of project management, working in a team, professionalism, exposure to the issues involved in developing complex projects, and contributing to the effective education for practical and industrial problems [5][6][7].

This paper describes the main characteristics found in different academic projects carried out over 10 years. The analysis of these characteristics will help us to know which projects meet more criteria and the reasons why it may be better to select a specific project.

The main contribution presented in this paper is the explicit analysis of these characteristics and the reflection it provokes to the reader, in order to help select the correct project for each entity of the academic triangle: student, professor or client (see Fig. 1).

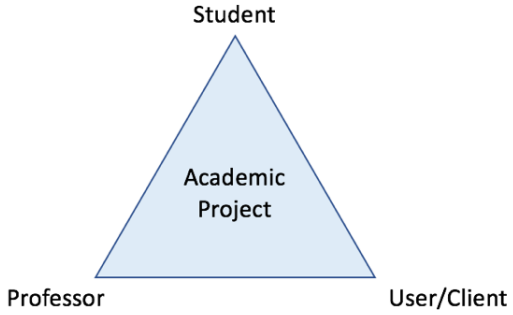

Fig. 1. Academic triangle for software projects.

II. PROJECTS EVALUATION USING A MATRIX

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 I). These characteristics were considered after a thorough analysis of advantages and disadvantages of these projects. It is worth noting that these attributes 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 was present (1) –or not (0) in that project, according to how it was developed in that moment. Following are these characteristics:

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

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 characteristic represents in a better way a real

Author would like to acknowledge financial support of Writing Lab, TecLabs, Tecnologico de Monterrey, México, in production of this work.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

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 system or warehouse system.

5. **Community service**. Does the selected project benefit the community? This is good because the final product is useful for the client and the project effort is not wasted.

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? This is directly related with some characteristics like: real user, company location, team, project and company size.

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

9. **Multidisciplinary teamwork**. Is it a multidisciplinary team? This attribute means that teams are formed by students from different courses and various disciplines. This is highly recommended because it incorporates other ways of thinking and the teams in the real life are often multidisciplinary.

10. **Multidisciplinary and multi-level teamwork**. How complex is the teamwork? Is it a multidisciplinary and multi-level team? This attribute means that teams are formed by students from different levels, various disciplines, and different courses as well (i.e. freshman, sophomore).

TABLE I. PROJECT-MATRIX WITH PROJECT ATTRIBUTES

Projects / Attributes	Education Apps	Caritas-Monterrey food bank System.	Computer Programming education Web-Apps	Monarch butterflies, Recycling games Web-Apps	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. Multilevel and Multidisciplinary	1	1	0	0	0	0	0
Total points	9	7	7	7	7	6	5

III. PROJECT ATTRIBUTES ANALYSIS

The software projects considered in this paper, range from a fictional case or case-study, to a complex real-world project. This section presents an analysis of the advantages and disadvantages for each kind of project, using a radar-graph to visualize the main characteristics of each one. The projects presented here are a few examples of the projects developed in the last 10 years.

A. Children's shelter website development

The purpose of this community service project was to plan and develop a website for a children's shelter. The user/client was the shelter's director. The assigned team consisted of 3 students of the same course, so the teamwork and team communication was easy.

This project had a community service attribute, with a real user and a highly feasible scope. Students learned software engineering concepts and they liked to work for a children's shelter using their work. This project cannot be replicable, so the following semester a different project was needed, and since it was a real user there was a dependency on him for certain activities, making it difficult to monitor and control. (see Fig. 2).

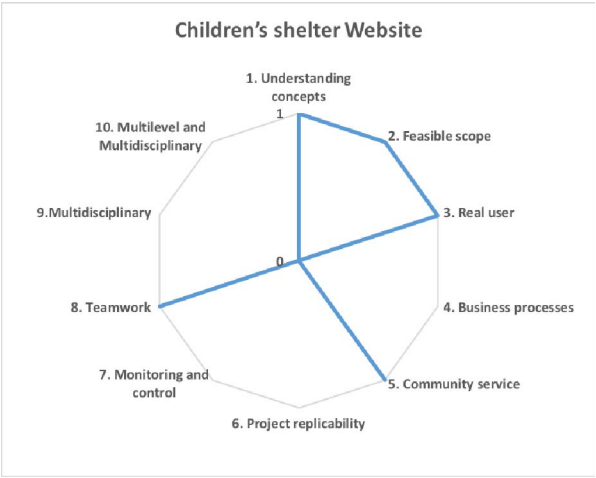

Fig. 2. Project attributes for children's shelter website.

B. Case-study: Warehouse system.

The purpose of this case-study project was to plan and develop a functional prototype for a warehouse system. There was no real user/client, so the project effort had no useful product and this was discarded at the end of the semester [5]. The teamwork and team communication was easy because the assigned team consisted of 3 students.

Also, this project was very easy for the professor to monitor and control, because it was a fictional case and it had a highly feasible scope [6]. It can easily be replicated and students learned software engineering concepts and some business processes related with warehouse systems. (see Fig. 3).

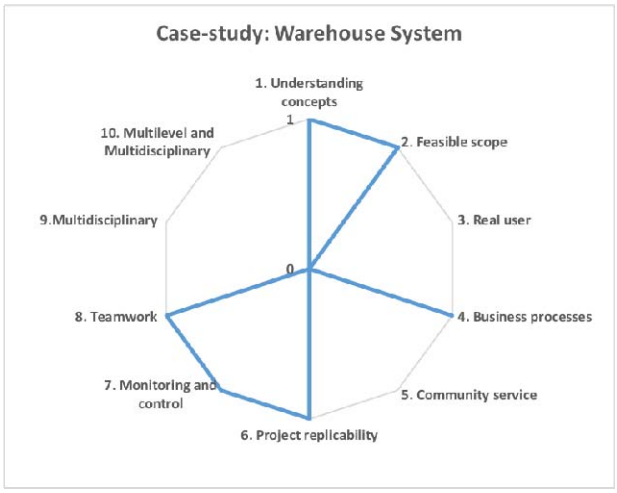

Fig. 3. Project attributes for case-study warehouse system.

C. *University's nutrition department. Nutrition App.*

The purpose of this community service project was to plan and develop an application to help users to follow a nutrition plan. The user/client were two nutrition specialists from university's nutrition department. The assigned team consisted of 3 students of the same group.

This project had a community service attribute, with a real "in-house" user and feasible scope. Students learned software engineering concepts, and they worked well as part of a small team. The project cannot be replicable, so the following semester a different client is required. There was no collateral learning about business processes (see Fig. 4).

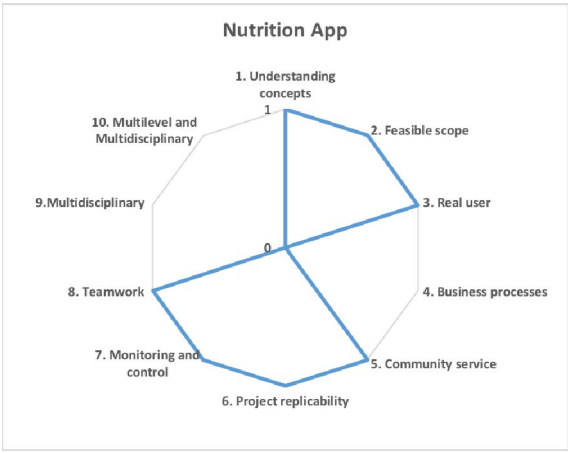

Fig. 4. Project attributes for nutrition application.

D. *University's environmental quality center.*

The purpose of this community service project was to plan and develop applications for: monitoring monarch butterflies, recycling games and "Good eating dish" game. The user/client was a professor from university's environmental quality center. The assigned team consisted of 3 students of the same course, but different study areas.

This project had a community service attribute, with a real "in-house" user and feasible scope. Students learned software engineering concepts, and they worked well as part of a multidisciplinary team. Students from Technology area had interaction with students from Digital Arts area.

This project cannot be replicable, so the following semester a different client was required. There was no collateral learning about other disciplines (see Fig. 5).

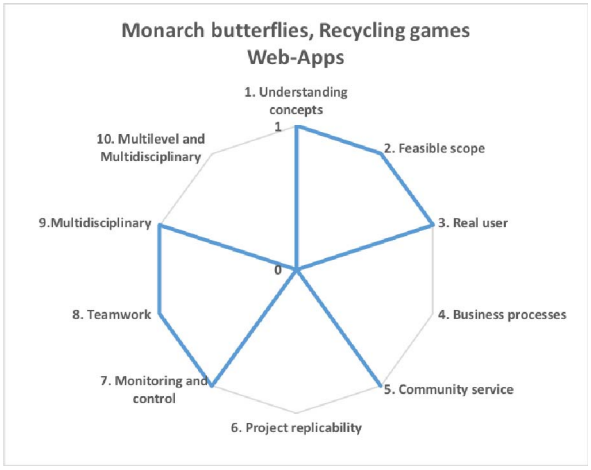

Fig. 5. Project attributes for University's environmental quality center.

E. *Computer programming education apps.*

The purpose of this community service project was to plan and develop applications for computer programming education apps. The user/client was a professor from university's Computer Science department. The assigned team consisted of 3 students of the same course.

This project had a community service attribute, with a real "in-house" user and feasible scope. Students learned software engineering concepts, and they worked well as part of a small team. This project can be replicated to develop different apps for computer education (see Fig. 6).

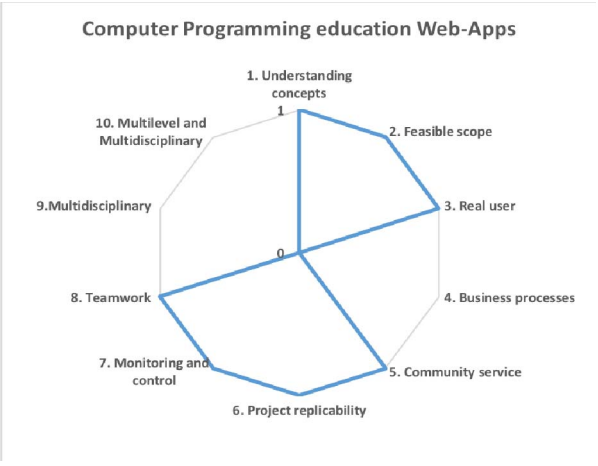

Fig. 6. Project attributes for Computer programming education apps.

F. Caritas-Monterrey food bank.

The purpose of this community service project was to plan, analyze and model a supply chain management system for a food bank. The user/client was a manager from Caritas-Monterrey food bank. The assigned team consisted of 3 students from Software Engineering Fundamentals course, plus 3 students from Software Project Management course.

This project had a community service attribute, with a real user. Students learned software engineering concepts and several business processes related with a food bank, which is very similar to a supply chain management system. Also, students work as part of a multidisciplinary and multilevel team, and they learned about the interaction with other areas [5]. With this type of projects, students realize the importance of cooperation, coordination and collaboration with their team. In the end of the course they report a satisfaction on the basis of helping an institution like Caritas food bank [5].

This project was very complex, because the user was far from our university. The monitoring and control was complicated because it was a complex system and the multidisciplinary-multilevel teamwork add difficulty level to the project (see Fig. 7).

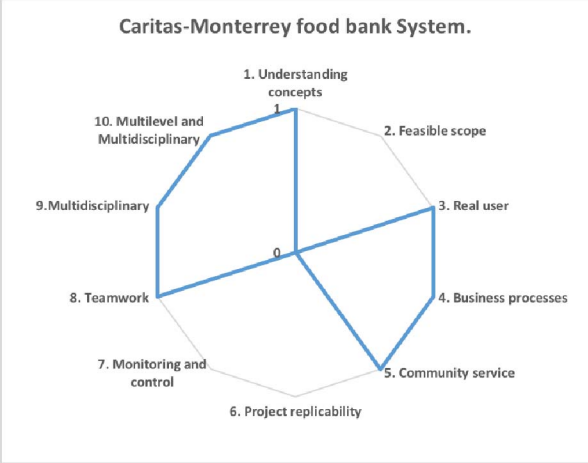

Fig. 7. Project attributes for Caritas-Monterrey food bank.

G. Chemistry education projects.

The purpose of this community service project was to plan and develop Chemistry education applications.

The user/client was a professor from university's department. The assigned team consisted of 3 students from Software Engineering Fundamentals course, plus 3 students from Software Project Management course, plus 3 students from Chemistry course.

This project had a community service attribute, with a real "in-house" user and feasible scope. Students learned software engineering concepts, and they practiced teamwork communication and leadership skills. Students from Information Technology area have interaction with students from the Chemical Engineering area. Also, this is a multilevel project because students from different courses work together as part of a team. Thus, Software Engineering Fundamentals students worked with the requirements documentation, while students from the Project Management course worked with the implementation of the project and Chemistry students worked with the project contents (see Fig. 8).

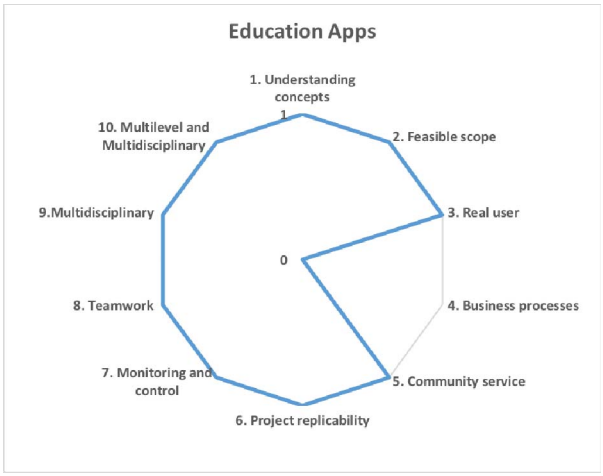

Fig. 8. Project attributes for Education Apps.

IV. CONCLUSIONS

There is a wide variety of software projects to use in Software Engineering courses. The important aspect is that each one has benefits for one or more entities of the triangle: student, professor, client (see Fig. 1).

It is important to mention that in addition to all this projects, there was an assessment for evaluating the topics covered throughout the semester for each course. No difference was seen in these test's results during the analyzed periods so the students learned well the theory and the practice together. But it was observed that the motivation was very high when the students had a useful product for a client.

It could be said that 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 [1], 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 [5]. However, a project as complex as that of Caritas-Monterrey food bank is considerably 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.

Finally, all of the projects presented here were good enough to promote software engineering learning. The projects attributes proposed could be used by professors to evaluate which kind of project they want to use and the implications of choosing one project over another.

ACKNOWLEDGMENT

Author would like to acknowledge financial support of Writing Lab, TecLabs, Tecnologico de Monterrey, Mexico, in production of this work.

REFERENCES

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

[2] Broman, D., Sandahl, K., & Baker, M. A. (2012). The Company Approach to Software Engineering Project Courses. IEEE Transactions on Education, 55(4), 445-452. doi:10.1109/te.2012.2187208.

[3] Macias, J. A. (2012). Enhancing Project-Based Learning in Software Engineering Lab Teaching Through an E-Portfolio Approach. IEEE Transactions on Education, 55(4), 502-507. doi:10.1109/te.2012.2191787.

[4] Johanyak, Z. C. (2016). Real-World Software Projects as Tools for the Improvement of Student Motivation and University-Industry Collaboration. 2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA). doi:10.1109/icimsa.2016.7504028.

[5] 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.

[6] 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.

[7] 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.

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