Using the effort of academic projects for the community service

A Software Engineering Practical Approach

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

Abstract—Work in Progress. Many of our students in the Bachelor in Computer Science start working when they are sophomores. By the time they are senior students, they have already learned different computer languages and technologies and have developed skills that are very well appreciated in the labor market. Tomorrow, the best companies will hire these students with a good salary. So, why not use their knowledge for the common good today? Many academic projects require too much effort from students, and at the end of the semester all this work is thrown away.

The main achievement of this approach is to establish a win-win relationship between students from two different courses and social organizations, where students learn more, sharing their knowledge and practicing with real-life projects and team roles, and at the same time, users benefit from products developed.

Keywords—Software Engineering; Software Project Management; Software Process Model; Social Projects.

I. INTRODUCTION

Two very important courses in the Computer Science curricula [1] are the Software Engineering Fundamentals course, taught in 3rd semester, and the Software Project Management course that is offered in the 7th semester in Tecnologico de Monterrey.

In the Software Engineering Fundamentals course, students have completed Computer Science 1 and Computer Science 2, but initially they ignore important concepts about the software development life-cycle. Regarding this, the course topics are: Software Process Models, Agile Development, Requirements Engineering, Software Design, Software Architecture, Construction, Testing, Software Maintenance and Evolution.

On the other side, Software Project Management covered topics are: Project planning (Statement of Work, Work Breakdown Structure, Team organization, Project Calendar), Project Estimation, Monitoring and Control using Earned Value Analysis, Risk Management, Configuration Management and Quality Models.

For the Software Engineering Fundamentals course the textbooks [2, 3] are complemented with research papers and Software templates. Similarly, the main reference books for Software Project Management [4, 5, 6] are complemented with research papers and Software industry templates as well. All of these are very good materials, but it is necessary that learning does not remain only in theory or readings from books and papers [7].

II. WIN-WIN PROJECTS

It is true that since 2008, work has been done in social projects with a Learning and Service strategy, but since 2015, Tecnologico de Monterrey has formally incorporated Social Service as part of some academic courses, offering the student's personal talent and professional work, in benefit of the community.

As a result, social-software project development was incorporated in Software Engineering Fundamentals and Software Project Management courses, developing projects for the Department of Community Service and Social Leadership [8]. This department is a center within the Tecnologico de Monterrey, committed to the social and economic development of non-privileged groups in the country, which organizes and promotes solidarity and participation in programs and projects of social impact [6].

With this model, a client has a small project that he wishes to have, but has insufficient resources. And, on the other hand, students from both courses have to learn too much theory and it is desirable that they learn the concepts by practicing them [9, 10].

With this in mind, real-life projects were adopted as part of these courses, where students work in teams with specific roles, developing social projects from start to finish.

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 [11].

Thus, the main characteristics of these approach are:

- First, these are projects that are focused on social service, not necessarily Humanitarian Free Open Source Software (HFOSS) [12], but focused on helping a community or a client, either within or outside of the academic institution.
- Second, these projects use a teamwork relationship between the two courses, to simulate the professional working conditions, with their negotiation issues between different groups, as well as artifacts that are input and output from the teamwork, and issues related with teamwork itself.
- According to this, sophomore students can learn from senior students sharing what they learnt before or sharing other information peer-to-peer, either using social networking sites, or face-to-face [13].

III. THE CASE OF CARITAS-MONTERREY FOOD BANK

With the intention of simulating a real-life environment, students from Software Engineering Fundamentals and Software Project Management courses, worked together in a social-project for the Caritas-Monterrey food bank.

The Caritas-Monterrey food bank is an organization of the Catholic Church that serves as a link between donors and people who suffer from poverty and hunger, improving their nutrition and optimizing available resources [14].

According to the project tasks, the first step when developing social projects is to familiarize with the client domain, in this case Caritas-Monterrey food bank, having an awareness session with the intention of generate a social consciousness in students. For this reason, the food bank staff visited the classroom, and gave students a general explanation. See Fig. 1.

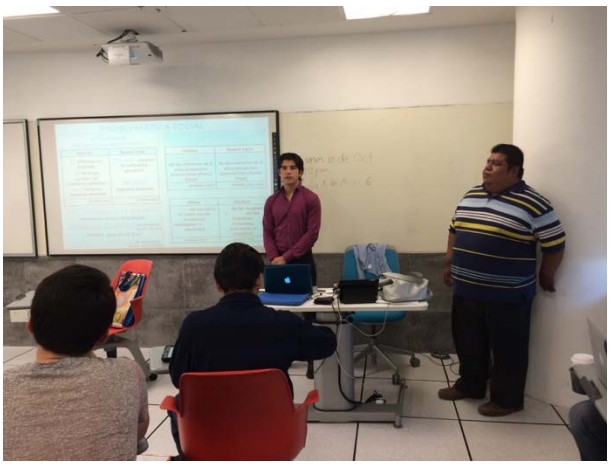

Fig. 1. Staff of Caritas-Monterrey food bank in the classroom

One or two weeks later the professor and the students went to the Caritas-Monterrey food bank center. Students from both courses made this initial visit to know about their business processes. See Fig. 2.

Fig. 2. Students listening the explanation at the Caritas-Monterrey food bank

IV. THE PROJECT

The idea with this approach was simulating a real-world environment, with a real and useful output, having different team roles like Requirements engineers, Software developers, Testers, etc. Thus, students should talk with the user, and analyze the businesses processes related to the project, but only Software Engineering Fundamentals students would write and model the requirements, and Software Project Management should plan, develop, monitor and control the construction of the final product. The reason for this task division is because of the different topics they have to learn.

Although the idea of using two related courses is not new [9], the main achievement of this approach has an extra value for the community, which is the work done with the Caritas-Monterrey food bank.

From the point of view of teaching Software Engineering Fundamentals and Software Project Management courses, the Caritas-Monterrey food bank provides a rich environment and complex business process, very similar to a Supply Chain Management System of a company. For instance, Caritas-Monterrey food bank has a Warehouse that receives and stocks the goods and foods from donors, and distributes these goods and foods to the community.

Thus, with a great need to improve or develop new software and processes documentation, Caritas-Monterrey food bank represented a great opportunity for students' education, because there are many business processes to learn about, like: warehousing, inventory, order, supply and procurement, logistics and transportation, etc.

With respect to the project process, it is important to say that in this particular case the project was divided in 5 modules because of its scope. Students from each course worked together to accomplish the module requirements. Following is a representation of this approach:

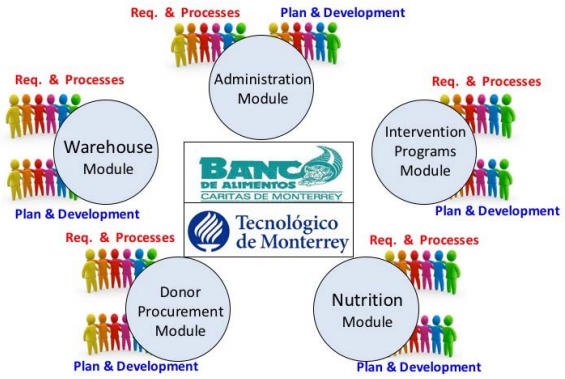

Fig. 3. Teaching Model

Teams were created in each class, linking one team with the other. Specifically, a team of Software Engineering Fundamentals would work for a team of Software Project Management. Also, the Software Project Management team would give feedback and tutoring to the Software Engineering Fundamentals students.

Specifically, the Software Engineering Fundamentals tasks would be:

- Create a Process-diagram for each key business process.
- Develop user-stories and use-case scenarios.
- Develop Software Requirements Specification (SRS) document using the IEEE format (std. 830-1998) [15].

Additionally, the tasks for Software Project Management students would be:

- Analyze the business process and user needs, along with Software Engineering Fundamentals students.
- Monitor, control and give some help to the Software Engineering Fundamentals team in developing the SRS.
- Take as input the SRS developed by Software Engineering Fundamentals team.
- According to the assigned module, develop software or build a prototype according to SRS provided by Software Engineering Fundamentals team.
- Test the software product or prototype, according to the Test Plan.

V. HOW IS THIS PROJECT EVALUATED?

For both courses the project grading criteria is 30% of the final grade. The rubric used to evaluate the project for Software Engineering Fundamentals is:

TABLE I. SOFTWARE ENGINEERING FUNDAMENTALS PROJECT

Description	Points
Statement of Work and Project scope	10
Project life-cycle selection and rationale.	15
Functional, Non-Functional Requirements	20
Process diagrams, Use-case diagrams, Use-case scenarios, Class diagrams	30
Test Plan, Test cases	10
Conclusions and Lessons Learned	10
Final presentation	5

On the other hand, the rubric used for Software Project Management project is:

TABLE II. SOFTWARE PROJECT MANAGEMENT PROJECT

Description	Points
Statement of Work & Project scope definition and Milestones	5
Software Requirements Specification	5
Work Breakdown Structure	10
Organizational Breakdown Structure - RACI chart	10
Project Estimation	20
Risk Management	10
Configuration Management	10
Quality Assurance	10
Earned value analysis (week by week)	10
Conclusions and Lessons Learned	5
Final presentation	5

It is important to mention that in addition to this practical project, there was an assessment for evaluating the topics covered throughout the semester for each course. Thus, the students learned the theory and the practice together.

VI. RESULTS AND CONCLUSIONS

Working with these kind of projects is not easy, because it is a complex business process environment and must be considered that the Caritas-Monterrey food bank center is far from the university, so it is difficult to get there. Besides, because it is a real project, there are real users that have too many activities, not only this project.

However, students could deal with these obstacles and work as a team to produce prototypes of modules and the related documentation. Therefore, students learned in a practical way by creating different artifacts, for example:

- Software Requirements Specification. Using software industry standard templates, and UML to model business processes, diagrams and use-case scenarios.
- Software development plan. Including the SOW, WBS, RACI charts, Project Estimation, Configuration Management, Risk Management, Quality Management and Lessons Learned.

From the user perspective, a software prototype and the complete process documentation was finished, presented and delivered to the food bank. This documentation and prototype will be used as a basis for the software construction. Because it is a large scope project, a software house will help them for building some modules. Meanwhile they will keep using the legacy system they are using so far.

Finally, in all the projects for both classes, the students were asked to include in the final document a section for conclusions and lessons learned. Particularly with this teaching model, when lessons learned were analyzed at the end of the project, student's comments were found very interesting. Here is a summary:

TABLE III. STUDENTS' COMMENTS IN LESSONS LEARNED

Students' Comments	Total
The importance of understanding the business process and user needs.	18
The importance of communication among stakeholders.	16
Collaborative work is essential for a successful project.	14
The importance of a good process for documenting and modeling requirements.	11
"I really like to work with a real project as the Caritas-Foodbank".	9
Software development goes beyond developing code.	8
The importance of teamwork organization, besides order and punctuality at meetings.	6

If this table is compared with the typical issues of software development industry [10], the key issues are the following: understanding the customer's domain and requirements, working in a team, organizing the division of work, and coping with time pressure and hard deadlines.

With the approach explained in this paper, students' comments can be observed and how the process of development was assessed, highlighting the fact that understanding the business process and user needs, in addition to effective communication and teamwork are some of the key factors to a successful project. This is not news to many who have worked in the software industry, but it is for the students who often give more importance to the project programming than the process itself.

In summary, according to the author experience with this kind of projects, students not only learned the theory behind Software Engineering, but also they learned a lot about the development process, and business processes, emphasizing the idea that programming is not the one and only important thing. Of course, in addition to this is the fact that they can use academic projects' effort with a distinctive focus on contributing to the common good, which is an idea that we promote.

As a future work, this approach will be evaluated, specifically: the product and the impact of these social projects; the software artifacts generated in this process and the collaborative learning process.

ACKNOWLEDGMENT

Special thanks to Caritas-Monterrey food bank staff for their excellent support provided in accomplish this project, and to the Department of Community Service and Social Leadership from Tecnologico de Monterrey.

REFERENCES

[1] Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. (2013).

[2] Pressman, R. (1982). Software engineering: A practitioner's approach. New York: McGraw-Hill.

[3] Sommerville, I. (2011). *Software engineering* (9. ed., International ed.). Boston: Pearson

[4] Cotterell, M., & Hughes, B. (1995). *Software project management*. London: International Thomson Computer P.

[5] Jalote, P. (2002). *Software project management in practice*. Boston: Addison-Wesley.

[6] Marchewka, J. (2003). Information technology project management: Providing measurable organizational value. Hoboken, NJ: Wiley.

[7] Jalote, P. (2008). A concise introduction to software engineering. London: Springer.

[8] IDESS. Retrieved January 5, 2016, from http://www.itesm.mx/wps/portal/idess

[9] Bavota, G., Lucia, A., Fasano, F., Oliveto, R., & Zottoli, C. (n.d.). Teaching software engineering and software project management: An integrated and practical approach. *2012 34th International Conference on Software Engineering (ICSE)*.

[10] Dahiya, D. (2010). Teaching software engineering. SIGSOFT Softw. Eng. Notes ACM SIGSOFT Software Engineering Notes, 1-1.

[11] Ellis, H. J., Jackson, S., Burdge, D., Postner, L., Hislop, G. W., & Diggs, J. (2014). Learning within a professional environment. *Proceedings of the 15th Annual Conference on Information Technology Education - SIGITE '14*. doi:10.1145/2656450.2656468

[12] The Humanitarian FOSS Project. (n.d.). Retrieved June 29, 2016, from http://hfoss.org/

[13] Davidson, C. N., & Goldberg, D. T. (2010). *The future of thinking: Learning institutions in a digital age*. Cambridge, MA: MIT Press.

[14] Cáritas Monterrey. Retrieved January 8, 2016, from http://www.caritas.org.mx/programas/alimentacion.html

[15] IEEE. Retrieved January 7, 2016, from http://www.ieee.org/index.ht