An Industry-Mentored Undergraduate Software Engineering Project

Azim Abdool
Department of Electrical and
Computer Engineering
The University of the West Indies
St Augustine, Trinidad
Email: azim.abdool@sta.uwi.edu

Akash Pooransingh
Department of Electrical and
Computer Engineering
The University of the West Indies
St Augustine, Trinidad
Email: akash.pooransingh@sta.uwi.edu

***Abstract*—Software design lifecycle application to a real world project is a critical skill required by the undergraduate computer engineer. Interaction with the local professional software development community is also an equally important opportunity that should be provided. This fosters growth of both technical and soft skills, and exposes the student to standards and working practices in providing quality software solutions for customers. This paper describes the structure of a group-based software development project that integrates industry mentors in the learning and assessment processes. Industry liaisons are given a forum to elucidate some of the industry's requirements of students in terms of knowledge of software design and industrial standards; students gain better understanding of some of the processes which take place in an actual industrial setting; and university curriculum gains industry relevance. The impact of this project is yet to be fully assessed. Assessment can be determined from two aspects: (1) in terms of the benefit to the student (effectiveness of learning objectives achieved, and motivation gained from project based learning and group work) and (2) the impact of providing industry led mentorship.**

I. Introduction

Software engineering forms an integral part of the computer systems curriculum for Electrical and Computer Engineering degrees [1]. Teaching software engineering concepts is a challenging task as most students perceive software engineering to be writing code [2]. This is only part of the process as it actually includes [3]: capturing the requirements; designing the system; program development and implementation (coding); unit, integration and system testing; system delivery; and acceptance testing. Most software engineering courses ensure that these concepts are effectively grasped by undergraduate students [4].

[5] highlighted the need for alternate teaching strategies for effective learning in software-based courses using flipped classroom techniques. [6] indicated that project based learning was effective in developing learning skills for software development that were not developed in the classroom environment but rather project practices in extra-curricular activity. This type of activity assisted in the development of agile method skills and practical experiences of group processes. Group based learning for software development was also implemented in [7]. Here a team-based project course focused on design and implementation phases of the software development lifecycle. [7] also noted that projects like these play a role in better preparing students for industrial work. [8] examined the need for these types of projects in developing countries.

In order to better prepare students for real life situations and expectations, it is vital that industry play a part in the learning process. Thus far, there has been very limited evidence of industry mentorship within an academic setting for student based software development projects. For example [9] utilized non-software engineering students as virtual stakeholders but did not consider industry input. In [10] the case was made to bridge the gap between academia, the skills needed by graduates in the workforce and skill shortages in industry. Here industry/academic links were examined and suggestions are made with regard to improving the current situation with the primary solution being the facilitation of student projects and internships.

This paper describes a group based software development project as a strategy for effective learning of objectives that may not be taught effectively within the classroom environment. This paper would show how the project uniquely integrates the software industry of a developing country in the learning and assessment processes. The effectiveness of industry-based mentorship is one area that has not been treated in current literature. Future work would describe the results of measuring the effectiveness of industry impact on the project.

II. Methodology

The process of software engineering involves multiple stakeholders, including the client who pays for the development of the software, an end user who is the target for using the software after development, and the developers who create the software [4]. The traditional teaching of software engineering [2] does not involve all these stakeholders. Our conceptualized project involves the local software industry in both the teaching of the software engineering (SWE) processes, as well as formative assessment of student achievement.

Within the Department of Electrical and Computer Engineering at The University of the West Indies (UWI), St Augustine campus, there are five thematic areas for teaching at the undergraduate level: communication systems, control systems, electronics systems, power systems engineering, and computer systems engineering. Each distinct area is managed by a curriculum overview group. Courses within each area run as parallel threads within a three-year BSc programme. All

978-1-4799-3922-0/14/$31.00 ©2014 IEEE

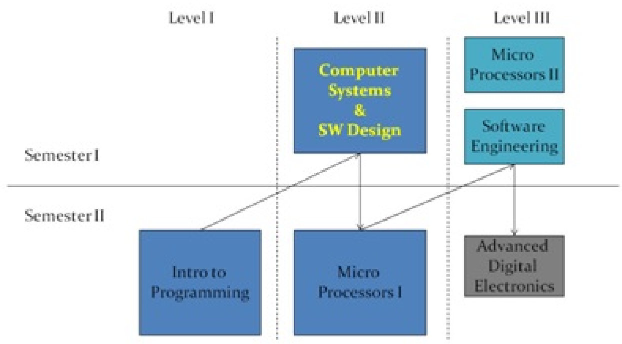

Fig. 1. Progression of CSE UG Courses

courses are mandatory during the first two years - representing the core set of competencies for all graduates with one, or more, area of specialization in the final year. The graduands of this Bachelor programme do not all necessarily end up in a primarily software-based environment, but rather one where their SWE skills must be applied to one of the other four thematic contexts.

The computer systems engineering (CSE) theme comprises several courses which support an embedded systems for application-specific solutions focus. The managing group believes that "a CSE graduate is solidly grounded in the basic principles of computer engineering, anticipates new developments in computer engineering, and is confidently able to develop and implement computing hardware and software, where necessary, in order to solve problems" [11]. This outlook allows graduates to enter into many different industries and apply the same basic tools and techniques of the SWE process.

Figure 1 shows the progression of courses for the CSE thematic stream. All undergraduate students must complete Level I and Level II courses. Level III courses within the CSE field are optional for the student specializing in other thematic areas, but are compulsory to students choosing the CSE option only. While the Level II course, "ECNG2007 - Computer Systems and Software Design", has learning outcomes surrounding three main themes - 1. computer systems, 2. SWE, and 3. programming - this paper focuses on the teaching and assessment of SWE, through the delivery of a single group project assessment.

A. *Project Conceptualization*

Traditional theoretical approaches to teaching may lead to lack of motivation of the students and hence to possible failure in successfully achieving the objectives [12]. According to [13] peer based learning is an effective method in the development of cognitive skills. To improve students' cognitive skills in software development, a practical group-based project was selected as the method of delivering and assessing the software design learning objectives. Since the SWE design process occurs in defined stages, assessment of the course learning outcomes can be drilled down to specific objectives that can be achieved using tangible milestones.

Analysis of general learning outcomes of the Department indicated a lower coverage in the assessment of certain objectives necessary to establish base-level competency requirements of an Electrical and Computer Engineering graduate. These objectives were targeted toward the economic, social and environmental considerations from an engineering perspective. Instruction and assessment of these objectives were at the time found to not be given consideration within the CSE theme.

Based on achieving outlined objectives the following learning outcomes for the project were proposed:

Engineering Analysis

- understand SWE principles and apply in the analysis of software processes
- identify and describe software lifecycle modelling techniques and select an appropriate model for a given situation
- adopt a systems approach to the conceptual and technical design of a problem
- adapt, and manage use of, pre-existing software modules/libraries for use in small-scale and system-level software projects

SWE Process

- elicit, model and analyse stakeholder needs for both functional and non-functional requirements, to produce a standards-compliant software requirements specifications (SRS) document
- use standard design modelling techniques to appropriately model the static-dynamic interface of a software design, starting from a SRS document, and produce a standards-compliant software design document (SDD)
- produce a standards-compliant software test document (STD) that would verify a created system (unit, integration and system level for functional and non-functional requirements) and outline a procedure for validation (user acceptance)

Economic, Social, and Environmental Context

- apply management techniques, to a software design process, from initial customer conception and specifications to testing, delivering and maintaining the system, evaluating specific outcomes, and taking into context commercial, economic, social and environmental considerations

Specific Workshop and Laboratory Skills

- implement and test, using standard laboratory tools such as compilers, linkers, debuggers, target simulators and emulators, a small scale, system-level software system prototype that meets given documented specifications (SRS, SDD, and STD)

B. *Mentorship*

The learning outcomes identified ensures that the students develop competence in designing, building and testing solutions to fit a customer need. The students were required operate within the constraints of economic, social and environmental factors while conforming to industry-standard design, implementation and documentation practices. In order to provide insight from a "real" perspective, the local software development community was approached to guide the students in the form of mentorship to groups of 8-10 students. This is essential in providing the students with knowledge of real-world problems, understanding of the local market and software development trends and feedback from the local software development community.

The mentors play an integral part in the delivery of the project. A mentor, either a single point person, or a group of persons under the direction of the supervising software company, is assigned to each student group. The mentors meet with the groups at the start of the project. The project specifics and logistics are setup prior to the semester by the course staff. The problem scope is delivered to the students with mentors assuming the role of the customer in the software development process. Requirements are elicited by the students to produce a SRS, SDD and STD. Mentors can guide the students to a preferred development tool (usually used in the specific company's practices) for the coding phase, as well as in standard procedures and practices in testing and documentation. Students are expected to maintain professional interactions with the mentors throughout the project duration.

C. Leveraging ICT

A Learning Management Systems (LMS) tool was used to maintain active communication amongst participants and a level of transparency and accountability. LMS tools are a necessary technology in supporting the educational needs in the modern context [14]. LMS provide a computer-driven infrastructure which supports the delivery and management of educational course material. It allows a central point for administration, collaboration and communication between course stakeholders and acts as a single point of reference for all learning content.

LMS are used on a spectrum from web-enabled, to web-enhanced/web-blended, to web-delivered courses. The UWI Campus is serviced by a branded version of Moodle [15], an open-source web-based LMS, called myeLearning. Course "shells" within myeLearning are automatically generated from the pre-existing campus student records database, Banner, and teachers and students automatically assigned. myeLearning incorporates several activities, such as forums, and wikis, that can be used for collaboration and discourse amongst the course participants [16].

Within the ECNG2007 course shell, there are several activities which are utilized. One such activity, is a forum. The forum tool is used to create "group logs". Each of the student groups are required to keep a track of all group work on this forum. All group activities, such as internal group and industry-liaison meeting minutes, draft submission artifacts for group and course administrator feedback.

Since forum messages are dated, this allows course administrators to keep track of student progress. At the beginning of the course, students are presented with a few theoretical models of the software lifecycle process [4]. This gives them an idea of the major stages involved in the creation of a software product. On initiation of the project, it is expected that intermediate deliverables of these standard lifecycle models be adhered. By adopting a lifecycle model at the beginning of the project, the students implicitly adopt a timeline for intermediate deliverables, since it gives event sequencing.

The "group log" forum therefore plays several roles with respect to time management of the project. It allows for course facilitators to asynchronously help set a plan of work at the start of the project. It serves as a tool for intervention, if the students do not deliver work at intermediate points. At the end of the course, it allows for reflection as to whether the particular lifecycle strategy used was effective, and also informs the time management section of the marking rubric for each group.

D. Project Assessment

At the end of the project, mentors are invited to the students' presentations, and asked to evaluate and comment on the students' progress. Mentors evaluate the groups according to the interactions made with the group during the project. The course staff themselves are responsible for marking the output reports, being informed by the group demonstrations of the systems which have been built. The group's final mark is also determined by a self assessment which must be completed by the group as a whole. During the presentation, a representative, usually the group leader, from each of the other groups also has a peer assessment to make. Within each group, each member is peer assessed to determine an individual mark. Each of these evaluations are done using developed rubrics provided by the course staff which speak to each of the project learning outcomes.

These evaluations are fused using a mixture of norm and criterion based referencing. The marking of the documents by the course staff is done using a purely criterion based method. However, in order to ensure standardization of marks, a norm-based referencing method is used to normalize the marks between groups such that the average mark for each group approaches the average determined by the instructors, and the remaining rubrics vary the standard deviation/variance of each of the individual members using a statistical normal distribution.

III. Measuring Impact

A. Impact of Projects

This project served to introduce the engineering student to the field of SWE. Students were able to take a problem posed by society (industry mentor acting as a customer) and systematically complete the software lifecycle under their guidance. This guidance was essential in providing the students with relevant and appropriate knowledge in software development that would not necessarily be best learnt in a solely classroom-driven environment. The problem spaces identified for this project were geared toward community and national development. Students were able to complete projects that affected their own community and were able to provide a product for use within that specific target area. This paper does not examine the impact of providing these types of community engagement and solutions.

Since the inception of these projects several positive points have been noted in terms of student interest and development in the field. Students leave the project with an added sense of achievement and some students continue along the software development field. Students have since secured internships with industry mentors, as well as entered local, regional and international competitions. A few software development companies have since been formed with one company offering mentorship to students of the course.

B. Measuring Impact of Industry Mentorship

A method for measuring the impact of the industry mentors is required to justify the approach of this paper. Continued work focuses on utilizing surveyed feedback from the stakeholders to measure effectiveness as follows:

- Software Employers - integration of graduate students in the workforce
- Graduates - preparation for the world of work
- Current Students - effectiveness of mentorship and relation to achieving learning outcomes and in personal motivation
- Industry Mentors - communication with students and improvement to the process

IV. Conclusion

This paper provides the structure of an industry mentored software development project offered to the Level II students at the Department of Electrical and Computer Engineering at the University of the West Indies, St Augustine. The project introduces students to a small-scale software engineering design process which carries an industry-proposed system from an initial conceptual model to a finished product. Industry mentors provided direction on the processes involved in software design. Feedback from mentors is essential in terms of knowledge of software design and industrial standards. This project is uniquely assessed using a mixed peer, teacher and industry mentor marking that hinges on a mixture of norm and criterion-referencing. ICT, in the form of a course LMS, is also leveraged in order to effectively relay student progress, and feedback from multiple stakeholders.

The impact of this project is yet to be fully assessed. Assessment can be determined from two aspects: (1) in terms of the benefit to the student (effectiveness of learning objectives achieved, and motivation gained from project based learning and group work) and (2) the impact of providing industry led mentorship. Further steps are to be taken to assess the effectiveness of this approach with the use of formative type assessments on students and industry mentors.

Acknowledgment

The authors would like to thank the UWI Department of Electrical and Computer Engineering postgraduate research support group headed under Dr Cathy-Ann Radix, and the ECNG2007 support team, headed under Mr Aniel Maharajh. The authors would also like to thank all of the industry mentors who have contributed their time and effort in contributing toward this project since inception.

References

[1] T.B. Hilburn and D.J. Bagert, *A software engineering curriculum model*, Frontiers in Education Conference, 1999. FIE '99. 29th Annual ,vol.1, pp.12A4/6,12A411, 10-13 Nov, 1999 doi: 10.1109/FIE.1999.839269

[2] M. Sastry and K. Mallalieu and A. Abdool, *Teaching Software Engineering as an Undergraduate Elective*, Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009), 2009.

[3] D. Suri and C. Taylor, *Introduction of software engineering to freshman: Challenges and rewards*, Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual, pp.F1J-11,F1J-16, 10-13 Oct, 2007 doi: 10.1109/FIE.2007.4418066

[4] S. L. Pfleeger and J. Atlee, *Software Engineering: Theory and Practice*, 4th ed, Pearson Prentice Hall, 2009

[5] A. Amresh and A.R. Carberry and J. Femiani, *Evaluating the effectiveness of flipped classrooms for teaching CS1*, Frontiers in Education Conference, 2013 IEEE, pp.733,735, 23-26 Oct. 2013 doi: 10.1109/FIE.2013.6684923

[6] K. Matsuo and S. Anzawa, *Work in progress Project practices of agile software development for undergraduate students*, Frontiers in Education Conference (FIE), 2010 IEEE, pp.S2D-1,S2D-2, 27-30 Oct. 2010 doi: 10.1109/FIE.2010.5673161

[7] S. Jarzabek and Pin-Kwang Eng, *Teaching an Advanced Design, Team-Oriented Software Project Course*, Software Engineering Education & Training, 18th Conference on, pp.223,230, 18-20 April 2005 doi: 10.1109/CSEET.2005.34

[8] R. Osman, *Teaching Software Engineering in Developing Countries: A Position Paper*, Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual, pp.648,653, 16-20 July 2012 doi: 10.1109/COMPSAC.2012.91

[9] G. Gabrysiak and H. Giese and A. Seibel and S. Neumann, *Teaching requirements engineering with virtual stakeholders without software engineering knowledge*, Requirements Engineering Education and Training (REET), 2010 5th International Workshop on, pp.36,45, 28-28 Sept. 2010 doi: 10.1109/REET.2010.5633109

[10] J.B. Thompson, *Why Better Industrial/Academic Links Are Needed If There is to Be an Effective Software Engineering Workforce*, Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on, pp.105,112, 9-12 March 2010 doi: 10.1109/CSEET.2010.16

[11] Department of Electrical and Computer Engineering *Computer Systems Engineering Group Page*, Retrieved March 9, 2013, from http://sta.uwi.edu/eng/electrical/thematicgroups.asp

[12] T. Jenkins, *On the difficulty of learning to program*, Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 4, pp. 53-58, 2002

[13] D. R. Newman and B. Webb, and C. Cochrane, *A content analysis method to measure critical thinking in face-to-face and computer supported group learning*, Interpersonal Computing and Technology, 3(2), 56-77, 1995

[14] W.R. Watson and S. Lee and C.M. Reigeluth, *Learning Managemnt Systems: An Overview and Roadmap of the Systematic Application of Computers in Education*, F. M. Neto, & F. V. Brasileiro, *Advances in Computer-Supported Learning*, Hershey: Information Science Publishing, pp. 66-96, 2007

[15] Moodle, *Moodle*, Retrieved from Moodle: https://moodle.org/about/, June 29, 2007

[16] M. Dougiamas and P.C. Taylor, *Interpretive analysis of an internet-based course constructed*, 2nd conference of herdsa (the higher education research and development society of australasia), pp. 7-10, 2001