ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 12, December 2016

Incorporating Project-based Learning to Develop and Assess a Sophomore Level Software Engineering Course

Afrin Naz

Abstract— **In this paper we talked about our five years journey to renovate a Software Engineering course for the Computer Science and Information Systems department of West Virginia University Institute of Technology by incorporating project-based learning. Our specific implementation of incorporating project-based learning into the Software Engineering course constitutes semester-long group projects, in which students go through all the steps of the software development life cycle. In the paper we also demonstrated how incorporating project-based learning into the Software Engineering course also facilitates the Accreditation Board for Engineering and Technology (ABET) assessment and evaluation process of our Computer Science program.**

Index Terms— **Project-based learning, Software Engineering, Student Outcomes, Assessment, Evaluation.**

I. Introduction

This paper presents our effort to renovate a Software Engineering course at the Department of Computer Science and Information Systems, West Virginia University Institute of Technology through incorporating project-based learning. Project-based learning employs projects closely related to real-world applications to facilitate delivering abstract concepts [8]. Our specific implementation of incorporating project-based learning into the Software Engineering course constitutes of semester-long group projects, in which students go through all the steps of developing a software. Each group starts with their own idea and then develops their software by applying the theoretical knowledge they learn from lecture contents to practical projects. Following the software development life cycle, the semester is divided into four phases: (1) communication and planning, (2) design and modeling, (3) construction, and (4) deployment. At the end of each phase, students demonstrate certain deliverables to the whole class, such as models, diagrams, and presentations. Each group has three students. While developing their software, each group serves as clients for another group's software development.

Manuscript received Dec, 2016.

***Afrin Naz**, Department of Computer Science and Information Systems, West Virginia University Institute of Technology, (e-mail: Afrin.naz@mail.wvu.edu). Montgomery, USA*

Incorporating project-based learning into the Software Engineering course also facilitates the Accreditation Board for Engineering and Technology (ABET) assessment and evaluation process of our Computer Science program. ABET is the leader in assuring quality and stimulating innovation in applied science, computing, engineering, and engineering technology education. Criterion 3 of ABET accreditation for Computing Programs specifies nine general outcomes (a-i) and two additional outcomes (j-k) that all computing baccalaureate graduates should possess. These eleven outcomes are divided into two categories: six "hard" skills (a, b, c, i, j, and k) and five "soft" skills (d, e, f, g, and h). The assessment and evaluation process of ABET requires periodic documentation and demonstration of the extent to which the student outcomes are being attained. In order to achieve and retain ABET accreditation, we also need to demonstrate the level of achievement of each student outcome and describe how the results of these processes are being utilized to effect continuous improvement of the program. In the Software Engineering course, assessments of eight outcomes (with Outcomes a, i, and j excluded) are embedded in project-based learning. Specifically, data are collected directly (for instance, using work products/deliverables presented by students) as well as indirectly (such as surveys) throughout the semester. The collected data are used to improve the course contents and project development. The assessments conducted during the past five years show significant improvements of students' "hard" skills (specification skills and design skills, particularly) and "soft" skills (presentation skills and teamwork skills, particularly).

II. MOTIVATION

The Computer Science and Information Systems Department of West Virginia University Institute of Technology obtained ABET accreditation for the first time in 2014. In order to prepare for ABET accreditation better, a sophomore course, CS 222 Introduction to Software Engineering, was selected to introduce students to ABET skills. After CS 222, students' ABET skills are further reinforced and assessed through a range of courses till the capstone series in the senior year. The renovation of CS 222 started in 2011. After examining the ABET guidelines carefully, project-based learning was decided to be the primary approach to embed ABET skills. It is worth noting that, the projects at the sophomore level are different from the projects in the capstone series at the senior level. In capstone series, projects are typically associated with the application of knowledge; whereas in the sophomore Software Engineering course, project-based learning is implemented to aid acquisition of knowledge.

III. RELATED WORK

Project-based learning is well grounded in scientifically-based research and enormous prior research has generated encouraging evidences demonstrating that, project-based learning succeeded in developing students' positive attitudes towards math and science, promoting students' collaboration skills, increasing students' content knowledge, and enabling students to transfer knowledge to practical implementation [1], [11]. For instance according to Frank *et al* [3] students found that the project-based format motivated students to learn and made them feel a greater sense of responsibility for their learning. Projects are attractive in engineering courses for several reasons. Schachterle and Vinther [8] noted that projects might encourage students to learn because they can see for themselves how the information will be useful *now* rather than because they have been told that they will need this information at some ill-defined point in the future. Projects provide a natural way to incorporate important skills such as teamwork and communication into technical courses [8].

Several studies can be found in literature survey on incorporation of ABET skills in an existing engineering curriculum [5], [9], [12] Shuman *et al* [9] and Williams [12] have conducted comprehensive survey on how these skills can be taught citing a number of examples of successful and/or promising implementations. They are very positive about a number of creative ways that these skills are being learned, particularly at institutions that are turning to global and/or service learning in combination with engineering design projects to teach and reinforce outcome combinations. They also examine the difficult issue of assessing these skills. While most of these studies are encouraged by work directed at assessing these skills, but recognize that there is considerable research that remains to be done. Challenges in incorporating ethics in engineering programs have been identified by Pfatteicher *et al* [6] and Stephan [10].

IV. THE PROJECT AND THE WORK PRODUCTS

In this section along with describing the steps of project development, we will also identify our related work products. We will also demonstrate how these work products become the vehicle for our assessment process. As mentioned before, by following software development life cycle we have divided the semester into four phases: (1) communication and planning, (2) design and modeling, (3) construction, and (4) deployment. Remaining of this section is described following the same order.

A. Phase I: Communication and planning

During the communication and planning phase the students will work closely with their client group (another student group from the class). Under the supervision of the faculty the teams will go through the six steps of requirement engineering (Inception, Elicitation, Elaboration, Specification, Negotiation, and Validation) and produce related work products [7]. In the following table we provided the description of function and list of work products for each steps.

Work products for Phase I: Description of function and list of work products for each steps for Phase one

Steps	Function	List of work products
Inception	Establishing ground work	List of stakeholders, List of questions to ask the stakeholders
Elicitation	Identifying the problem and specifying requirements	List of objects, List of services, List of constraints
Elaboration	Creating use cases	One template based use-case and one UML use-case diagram
Specification	Creating the description of the system domains	Activity diagram
Negotiation	Deciding key win conditions and negotiating	Identify the key stakeholders and their "win conditions"
Validation	Determining if the requirements chosen are able to be accomplished	Validation requirements

Usually 2 weeks are allocated for Phase I. During the class time we will hold formal meetings between each client and developer groups. After going through the Inception step, during the Elicitation step the developer team will meet with their clients to explain their initial plan and collect necessary information from the clients. During the Elaboration and the Specification steps the developer team will develop "use-case", "use case diagrams" and "Activity diagrams". At the second week of Phase I, they will meet again with their clients to demonstrate their developed work products and also go through the Negotiation step. Finally the developer team will conclude Phase I by going through the Validation steps. At the end of Phase I, all required work product listed in Table 1 will be submitted to the instructor for grading.

B. Phase II: Design and Modeling

During the Design and Modeling phase the student teams will work with four different categories of design elements which are Architectural, Components, Interface, and Deployment Design [7]. For each category they will produce UML diagrams by applying CASE tools. During this Phase the developer team will create eleven different types of UML diagrams. In the following table we provided the title of these eleven diagrams along with the design element category. At the end of Phase II, all required UML diagrams listed in Table 2 will be submitted to the instructor for grading.

1. *Architectural Design:* During the Architectural Design phase the students will place their developing software into context to define the external entities (other systems, devices, people) that the software interacts with and the nature of the interaction by providing an Architectural Context Diagram. They will also identify top level components and provide the Architectural design.
2. *Component Level Design:* The teams will thoroughly describe with figures the steps they went through during the Component design. Each group needs to select one component and provide the Component design with elaborated data and algorithm. They also need to identify a low cohesive operation and apply refactoring. Finally each team needs to select one operation and provide an Activity Diagram to describe processing flow. As a bonus material they can also develop behavioral representation for a component.
3. *Interface Design:* During this phase the teams will describe briefly about Interface design and their own experience. Each team must provide at least two different schematic layouts of the user interface for their consumer. They also need to provide an example of an external interface in their system to identify information sent out or received and to provide an example of an internal interface aligned with component level design.
4. *Deployment Design:* During this last design phase the teams will identify the Deployment level design elements within the physical computing environment that will support the software and provide a diagram with Deployment level elements.

C. Phase III: Constructions

The Construction phase includes both coding and testing. Three weeks (including Spring break) are allocated for coding and one week for testing. During the "coding" weeks the students will work under the supervision of the instructor during class period and on their own outside the class room. For testing the first iteration of the software, the student teams will conduct three different categories of testing which are Unit, Integration and Validation testing. For each category they will produce a test case and complete the test cases during the testing. For four Validation testing (two alpha and two beta), the teams are required to work with clients from different ages, genders and computing experience groups. The team will demonstrate and submit the first iteration of their developed software along with the test cases for each categories to the instructor for grading.

Work products for Phase II: Titles of design diagrams along with the design element category

Diagram Title	Design element category
Deployment diagram	Deployment
Architectural diagram	Architectural
Architectural Context Diagram (ACD) diagram	
Component design with elaborated data and algorithm diagram	Component
A diagram to demonstrate a component with elaborated data and algorithm	
A diagram to demonstrate refactoring for a low cohesive operation	
An activity diagram to demonstrate the processing flow of an operation	
A User Interface design diagram	Interface
An alternate design for the SAME User Interface diagram	
External Interface diagram	
Internal Interface diagram	

D. Phase IV: Deployment

Due to the time limitation, the students are given only one week for this phase. During this week the teams will make modifications based on test results and client (including instructor) feedback. At the last week of the semester the students will demonstrate and submit the final iteration of their developed software. No work products was associated with this phase. Student groups submit the final iteration of their project.

V. ADDRESSING STUDENT LEARNING

Our objective is not only to embed ABET outcomes in our Software Engineering class but to also evaluate these outcomes. Our main goal is to work with our students so they can attain these skills. Each group develops their software project by applying the theoretical knowledge they learn from lecture contents to practical projects. The students are graded on their actions as they go through the project development process and on the project along with the work products required to be submitted as specified by our project guidelines.

A. Addressing student learning of the "hard skills"

The term *"hard skills"* refer to those skills that engineering education has traditionally done well while inculcating its students into the world of engineering: problem solving, critical thinking, and the use of algorithms, instrumentation, and technology necessary to analyze and interpret data [11]. Out of the total eleven Student outcomes the first three (a – c) and then the last three (i – k) belong to this category. Although we have addressed all of the remaining professional or soft skills, only three hard skills b, c and k were included in our work. During the four phases we will train our students to develop Software Engineering work products (for example UML diagrams) by applying professional tools.

B. *Addressing student learning of the "soft skills"*

One important aspect of the enhanced ABET criteria (published in 2000) is "an understanding of professional responsibility." In ABET outcomes for computing, outcomes "d – h" are identified as "soft or professional" skills. In this section of our paper, we will concentrate on these "soft skills" (we put e and g together), and describe our strategies in improvement of our student's professionalism.

(a) Communication Skills: The proceedings of both the "Frontiers in Education" conferences and the annual conferences of the "American Society of Engineering Education" are replete with examples of ways to integrate communication into the core of engineering education [7]. In this class we walk through the phases of the software development cycle, with a concentration on oral, visual and writing communication skills. We believe that placing communication components to express their recently gained experience will be a good way of approaching ABET's criteria for communications in a course with significant technical content. At the end of each phase, students make presentations on their recently learned experience and demonstrate certain work products/deliverables to the whole class. In the class the students communicate with their clients (with a real world simulation) and a general audience (remaining class). The class also includes an individual project in which each student perform researchs on a current software engineering topics to prepare a formal report. Students were graded both on their development and presentation skills of these products. From our experience in this five years period, we agree with Shuman *et al* [7], by saying "Communication is the one skill that can certainly be taught and assessed".

(b) Teamwork Skills: This team based project driven class gives us the opportunity to provide our students the experience of dynamics of team design work from idea development to completion. The students are held responsible for both individual and group learning and are rewarded and/or penalized based on the team's performance. Students are required to provide peer performance evaluations based on (1) positive interaction and helping others, (2) performing research and soliciting member input (3) understanding of team roles, (4) sharing responsibilities, and (5) showing good listening skills. Unlike communication skills, the teamwork skills are not only challenging to teach but also hard to measure.

(c) Ethics and Contemporary Issues: Over the last five semesters we had made major changes in relation to ethics and contemporary issues. As we all agree with Stephan [10], "True test of engineering ethics education is how graduates behave in the workplace during their careers, certainly a difficult outcome to measure a priori". Fortunately ABET criteria call for ensuring *understanding* rather than *demonstrating* that graduates are ethical. At the initial years we looked at ethical frameworks via the discussion of engineering codes of ethics and application of moral theories. Now the students also work with case studies to focus on how CS/engineering professionals perceive, articulate, and resolve ethical dilemmas in their professional life.

(d) Life Long Learning: The final "soft skill" that is addressed by ABET is the development of life long learners. Although ABET no longer use the term "Lifelong Learning" we can see that in outcome "h" as professional development. As we all can agree it is nearly impossible for engineering programs to assess this future state. However, we can help our students to become life long learners. According to Shuman *et al* [7], "One will become a proficient lifelong learner as one becomes proficient in the broad spectrum of professional skills". We also believe that as students acquire all these skills in ABET outcomes, they will, in fact, acquire the ability to do lifelong learning.

VI. Assessment and Evaluation of the Student Outcomes

We developed this class by systematically incorporating project-based learning. We identified the steps to be completed and work products to be submitted as the project is being developed over the semester. Although the students will come up with their own project idea they have to follow the class specifications step by step. As mentioned before incorporating project-based learning into this Software Engineering course also facilitates the Accreditation Board for Engineering and Technology (ABET) assessment and evaluation process of our Computer Science program. Through curriculum mapping, first we identified the student outcomes to be assessed and evaluated in this class. As we modified this course, we have developed methods of instruction delivery to offer the knowledge and skills necessary for the mastery of these outcomes. As a part of our continuous improvement effort we have also developed assessment tools and relevant performance indicators to assess and evaluate students expected competencies. According to ABET, performance indicators indicate what concrete actions the student should be able to perform as a result of participation in the program. In order to measure the student performance through our performance indicators, these indicators have been mapped to the core project and to different work products.

Following the literature we first grouped the ABET's Criterion 3 student outcomes for computing into two categories as provided in Table 3.

Table 3: ABET's Criterion 3 student outcomes for computing divided into two categories

ABET student outcomes	Category
(a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline	Hard Skills
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	

(i) An ability to use current techniques, skills, and tools necessary for computing practice.	
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices	
(k) An ability to apply design and development principles in the construction of software systems of varying complexity	
(d) An ability to function effectively on teams to accomplish a common goal	Soft Skills
(e) An understanding of professional, ethical, legal, security and social issues and responsibilities	
(f) An ability to communicate effectively with a range of audiences	
(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society	
(h) Recognition of the need for and an ability to engage in continuing professional development	

Next in Table 4, along with describing our performance indicators utilized for the outcomes, we are also identifying our instruction implementation components and assessment tools for the related skills. From the table we can see that in several cases, we have utilized the same performance indicator to address multiple student outcomes.

Table 4: Mapping of ABET outcomes with our performance indicators, course components and assessment tools

Performance Indicators	ABET outcomes	Course component and/or assessment tools
Develop a software requirements document (specification) based on the analysis of the specifications provided by a sample user for a semester long programming project.	b and f	Specification documents (Table 1)
Develop design documents in UML in a group environment based on the analysis of the specifications provided by a sample user for a semester long programming project.	c and d	Design documents (Table 2)
Perform beta-testing of application by performing client reviews, analyzing any problems each student application may have and ways of improving their work.	f	Test suit and results (Table 3)
Shares the responsibility and work load of the team	d	Peer evaluation
Provide an understanding of software project management skills in team environment.	d	Student survey
Develop a report on a software engineering topic.	f	Paper
Make group presentations on developed software products to a body of peers.	f	PPT slides sets 1, 2, 3, 4
Conduct group discussions on the social and ethical dimensions of a problem in computer science	e and g	Faculty survey on ethics
Identify the purpose of the five phases of software engineering: requirements definition, software design, software implementation, software testing, and maintenance.	c, h and k	Class project

A. Assessment and Evaluation of the "hard skills"

The main challenge in the assessment and evaluation of the "hard skills" was identification of work products to be utilized to confirm the attainment of these skills. Over the years we included new work products and removed some. All the students have to grade the students from other groups on the quality of these work products. However only the grade provided by the instructor was counted towards class grades. In Fig 1 we demonstrated the student group's grades in attainment of Student Outcome b for four years (2013 – 2016). From this figure we can see the wide range between the grades of different groups have significantly reduced after 2013 showing more consistent pattern in student learning.

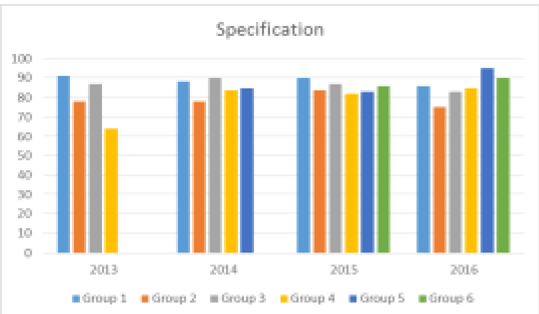

Figure 1: The student groups grades in attainment of Student Outcome b for four years

In Fig 2 we demonstrated the student groups grades in attainment of Student Outcome c for four years (2013 – 2016).

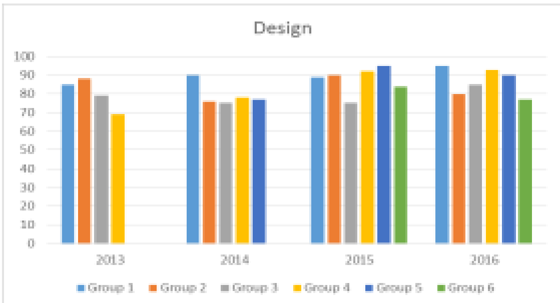

Figure 2: The student groups grades in attainment of Student Outcome c for four years

B. Addressing student learning of the "soft skills"

Our experience in assessment and evaluation of the "soft skills" varied with different skill sets. Whereas it went very smoothly for the Communication Skills, it was not the case for the others.

(a) Communication Skills: As mentioned before the assessment and evaluation of the communication skill went very smooth from the very beginning. In Fig 3 we demonstrated the student group's grades in Phase III presentation for four years (2013 – 2016).

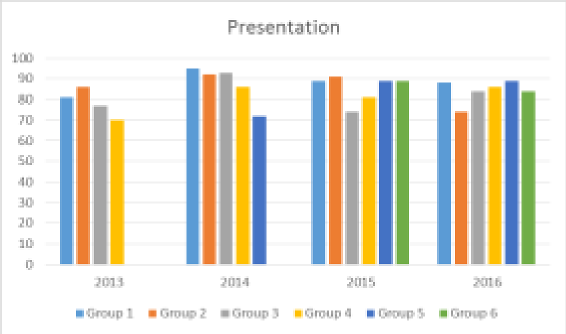

Figure 3: The student groups grades the student group's grades in Phase III presentation for four years

(b) Teamwork Skills: Unlike the "hard Skills" and communication skills, our experience with teamwork skill is quite challenging. Our data collected during 2012 showed decrease in teamwork skills as we went through the semester. After careful observation we have identified two main reasons. First reason was absence of adequate time to work together. In order to resolve this problem, starting from next semester, dedicated time slots were included during the class time to allow the students to work together. The second reason was absence of good leadership. Hence, along with teaching course contents, the instructor worked more closely as a team member with the students to improve team work skills. In Fig 4, we demonstrate the grades from peer evaluation in teamwork skills during spring semester of 2014.

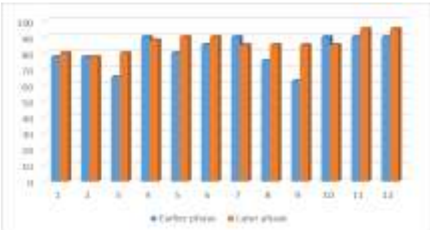

Figure 4: The grades from peer evaluation in teamwork skills during spring semester of 2014

As shown in Fig 4 we still see inconsistency in this category which is based on students grading only. Unfortunately most of the time the students are randomly selecting an option. We are considering including teachers grading in this category. In the future, the instructor will individually interview each student and ask questions both on their own experience in project development and also on their partners input.

(c) Ethics and Contemporary Issues: As mentioned before, over the last five semesters we had made major changes in relation to ethics and contemporary issues. In Table 9 we are presenting our faculty rubrics for ethics. We followed ABET's guideline to design the "Faculty Rubric" used for student outcome 'e' for ethics. We are considering to spend more time in use of appropriate tools in making ethical decisions.

(d) Life Long Learning: Finally the performance indicator for student outcomes 'h' is calculated using the semester long student group project (Average of Phases I, II and III) as the instructor grades students based on their overall experience as professional development.

VII. Conclusions

In this paper we talked about our five years journey to renovate a Software Engineering course for the Computer Science and Information System department of West Virginia University Institute of Technology by incorporating project-based learning. Project-based learning can be implemented in a variety of ways. Key features that define project-based learning include starting with a problem to be solved, which then forms the basis of the project work, and the fact that projects result in an end product such as a report, presentation, or physical artifact [2]. In this innovative Software Engineering class we utilize a semester long group project which will take our students through the steps of software development life cycle. The student groups will come up with their own project idea and will build their own software project over the semester by applying the Software Engineering process, methods and tools. Incorporating project-based learning into the Software Engineering course also facilitates the Accreditation Board for Engineering and Technology (ABET) assessment and evaluation process of our Computer Science program.

Acknowledgment

The author would like to thank the other faculty members of the department.

REFERENCES

[1] R.A., Atadero, K.E.Rambo-Hernandez, and M.M. Balgopal. "Using social cognitive career theory to assess student outcomes of group design projects in statics," Journal of Engineering Education, 104(1), 55-73, 2014.
[2] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial and A. Palincsar, "Sustaining project-based learning: Sustaining the doing, supporting the learning," Journal of Educational Psychologist, *26*(3&4), 369-398, 1991.
[3] T. Frank, M., I. Lavy, and D. Elata, "Implementing the project-based learning approach in an academic engineering course. International Journal of Technology and Design Education, 13(3), 273-288, 2003.
[4] http://www.abet.org
[5] A. Naz. and M. Casto, "Best of Two Worlds in a Software Engineering Class, Student Outcomes of Accreditation Board of Engineering and Technology (ABET) and Information Literacy Standards of Association of College and Research Libraries (ACRL)", *Frontiers in Education* annual conference, October, Oklahoma City, OK, 2013.
[6] S.K.A. Pfatteicher, "Teaching vs. Preaching: EC2000 and the Engineering Ethics Dilemma," *Journal of Engineering Education*, 90 (1), 137–142, 2001.
[7] R. S., Pressman, Software Engineering A Practitioner's Approach. McGraw-Hill Higher Education, 2010.
[8] L. Schachterle and O. Vinther, "Introduction: The role of projects in engineering Education," European Journal of Engineering Education, 21(2), 115-120, 1996.
[9] Shuman, L. J., M. Besterfield-Sacre, and J. McGourty. (2005). The ABET "professional skills" - can they be taught? Can they be assessed?. *Journal of Engineering Education*, 94 (1), 41-55.
[10] K. D. Stephan, "Can Engineering Ethics Be Taught?," IEEE Technology and Society Magazine, 5- 8, 2014.
[11] J.W. Thomas, " A Review of Research on Project-Based Learning", San Rafael, CA: Autodesk Foundation, 2000.

[12] M. S. Williams, "Trends in engineering education: using ABET's program outcomes as a framework for change," *International Journal of Engineering Education*, 94 (1), 41-52, 2004

Afrin Naz completed her PhD in 2007. Her main research area is Engineering Education. She has launched many workshops for teachers. She is an IDEAL ABET scholar.