Project-based Learning for Software Engineering– An Implementation Framework

Samini Subramaniam, Fang-Fang Chua and Gaik-Yee Chan
Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia.
gychan@mmu.edu.my

***Abstract*—The ruling pedagogy for software engineering education still remains "chalk and talk" even though it has many drawbacks leading to its unproductiveness. In recent years, many researches were conducted to propose a systematic teaching and learning method to prepare students with good project management, verbal and written communication skills before facing the real working life. Particularly, teaching in this era of Internet of Things requires a good pedagogical method to ensure that the teaching and learning focus on both theory as well as experiential learning. Thus, in this paper, we propose a framework based on the project-based learning for software engineering subject that focuses on understanding common knowledge as well as the ability to develop real life product.**

***Index Terms*—Agile; Internet of Things; Project-based Learning; Scrum.**

I. INTRODUCTION

Software engineering education is vital for shaping the software development skills as well as soft skills among the computer science or information technology undergraduates. It is one of the core subjects for building the basic foundation on software practices and processes. Traditionally, the delivery mode of the subject has been mainly focusing on formal lecturing (one way) and the assessment is conducted based on formative approach i.e. tests or group projects and final exam as the summative approach. This traditional "talk and chalk" way of teaching has many drawbacks such as the retention of knowledge is not optimum as the acquisition of facts and data are mainly in the absence of real life circumstances and without much reflection, criticism, and innovation from the students' part [1]. Consequently, this traditional way of teaching has alienated itself from providing the real needs of the industry and society [1], i.e., graduates who are equipped with both theoretical and practical skills. Nevertheless, in preparing the undergraduates to face the dynamic challenges in real life project development cycle, there is a need for them to learn the subject in a more effective way.

We have adopted project-based learning in addressing to this need and proposed an implementation framework for the software engineering subject. Through project-based learning whereby the emphasis is on group work and collaboration in resolving real world projects [2], students thus learn to collaborate and communicate throughout the phases of the project and have an individual as well as group experience in participating in the overall software engineering activities.

Nowadays, software technology popularity is moving towards the trend of mobile application and Internet of Things (IoT) due to the nature and demand of close connectivity between software, devices, and all things. According to [3], the more recent trend for IoT is the light-weight plug-and-play or Cloud-based middleware. Consequently, the development methodology has to lean towards agility which could provide quick response to changing environments [4]. An example of such methodology is Agile methodology with Scrum practice to project management. Additionally, this dramatically growing demands for IoT requires us to give a second thought on how to educate the coming generation of engineers and computer scientists [5]. One of the challenges faced by the institution of higher learning is to prepare students who are equipped with both technical and non-technical skill sets as required by the industries. Teaching IoT requires a good pedagogical method to ensure that the teaching and learning (T&L) focus on both theory as well as experiential learning.

Therefore, the project-based learning implementation framework should aim to play a role in an effort of empowering software engineering towards IoT. As such, in this paper, we propose a project-based learning (PBL) implementation framework to empower software engineering towards IoT.

II. BACKGROUND AND RELATED WORKS

Project-based learning, or referred as project-oriented problem-based learning in [6,7], as project-based and problem-based learning in [1], emphasizes on collaboration and group work approach in resolving real world problems [2]. It has been applied across various ranges of disciplines, such as in Computer Science for Programming course [6], in Software Engineering subject [7] and in Engineering education [1] where it encourages initiatives, independence of thoughts and critical thinking. This has become one crucial motivation to adopt this learning approach in our implementation framework for software engineering education. According to [2], project-based learning also provides a balance between formal lectures which provides subject information and informal mentoring or socialized learning. With this adoption of approach, there is a need for a different role of the subject educators to accommodate for the assessment criteria.

Researchers in [8] have applied problem-based learning to software engineering group projects which allowed students to practice, apply and develop skills such as problem solving and team building. The members of the group are selected based on weak-strong selection technique which might not be realized in the real world working environment. The assessment of the subject is a 70/30 mix of group and individual assessment and the skills to be assessed include implementation skills, teamwork and leadership skills, and analytical thinking and interpersonal skills. The groups are

given complete autonomy over their software development strategies but to work with clients to elicit requirements. However, students might face challenges in making decisions in such open and unstructured learning environment and there is a need for time-to-time constant discussion and meetup for decision making.

On the other hand, [6, 7] uses the generic Project-Oriented Problem-Based Learning (POPBL) framework in T&L environment. Implementing this POPBL approach for Software Engineering (SE) courses not only exposes students to technical skills relevant to solving real world cases, but also assist in improving their soft skills such as cooperation, effective communication, critical and creative thinking, as well as efficient project management and planning. This generic framework was implemented in two SE courses, namely Programming Technique 1 (PT1) and System Analysis and Design (SAD). The project has three stages, onset, execution and closure stage before final completion. During the onset stage, the PT1 students were assigned six case studies related to current trends in mo bile application and SAD students were asked to find a suitable real-world project in a given domain. Next, during the execution stage, the SAD students were required more planning and understanding on the domain requirements where they had to interact with stakeholder involved in their project. So, SAD students have to undergo more SDLC stages rather than PT1 students. Case studies were given part-by-part to PT1 students. Finally, at the closure stage, 40 students from PT1 and 73 students from SAD participated in the survey at the end of the semester and post-mortem activities were also conducted to elicit students' feedbacks. The implementation of POPBL has provided the students with the opportunity to work in a team, gain experience working in a "real world" project and manage time more effectively. However, some students faced difficulties such as lack of time to complete the project, uncooperative stakeholder which causes delay in project delivery, having members who do not contribute to the completion of the project.

Meanwhile, [1] suggested to combine the concept of project-based learning and problem-based learning approach into the T&L of embedded system course from electronic and information engineering specialty at Chengdu University of Information Technology, China. As part of the problem-based learning, students were given some problems where they have to work in a team to find solutions to the given problem. The problem-based approach only allowed the students to master the knowledge taught in the lecture. The students were then asked to conduct experiments which are part of the project-based learning method. The experiments conducted were relevant to the courses taught in the class which allowed them to relate and apply knowledge that was taught in the class. Every student in the team had to submit a practical report explaining solutions and answering theoretical and practical questions raised by the lecturer. In this way, students had to study what had been explained in the lectures in order to do the practical report. A survey was conducted after the implementation of this approach in the embedded system course. The survey results showed that 100% of the students were satisfied with implementation POPBL in T&L, allowed them to improve their non-technical skills and the active learning method helped them to understand the course more deeply.

The above-mentioned researches [1,6,7] no doubt have provided effective teaching methods based on project-based learning. However, in this era of IoT where the development of IoTs requires the agility of the project team to handle quick response to changing environments, purely project-based learning method without encompassing agility in development and project management might not render learning to be effective as well. Therefore, our proposed project-based learning implementation framework is encompassed with Scrum practice of project management to provide the aim to play a role or perhaps the pioneer role, in an effort of empowering software engineering towards IoT.

Scrum, which has been practiced in industry as an iterative and incremental agile software development methodology, is built on the basis of *transparency*, *inspection* and *adaptation* [9]. *Transparent* in the sense that employees are working in groups to achieve the project goal in a collaborative manner and must share a common definition of "Done" for acceptance of a work product. The high frequency of artifacts *inspection*, progress updating, reporting and meeting requires members to *adapt* to dynamic changes responsively and quickly.

Scrum, therefore, could be applied into education to motivate participation among students in participating and contributing to the project with more empowerment given to each of the members. Members are better aware with their individual roles and responsibilities as well as their contribution for achieving the common project goal. Besides, they will be learning in a project working environment more effectively whereby constant discussion is needed which stimulate brain storming and knowledge sharing to solve problems. Students also tend to practice self-engagement and organizing with this adoption of software development methodology in their learning activities.

However, as this approach is different from the current practice of subject learning whereby autonomy is given, students might face difficulties and uncertainties in planning for the activities. Strong scaffolding, especially at the beginning of the project is recommended to allow students to gain adequate confidence throughout the study to explore independently [10]. Guidance with an implementation framework is thus needed for facilitating the group progress in completing the project activities.

III. Our Proposed Framework

The proposed project-based learning (PBL) implementation framework is designed for Software Engineering (SE) subject of Bachelor of Computer Science (BCS) for institutes of higher learning. Implementing the PBL framework in the SE subject aims to provide students with the opportunity to gain fundamental software engineering knowledge and skills. At the same time applying the knowledge and skills learnt to plan, analyze, design and implement software projects. Student shall work collaboratively in a team to deliver the project outputs. The idea of using project-based learning is most effective when students put theory into practice while the students' role changes from "learning by listening to learning by doing". In project-based learning approach, students are required to review and refine deliverables such as requirements, designs and program codes, iteratively based on regular feedback from team members and lecturers. These students' activities which revolve around a series of interactions between team members over time shall stimulate students to collaborate thus enhancing their team spirit in completing their projects.

Through project-based learning approach, students are made aware of current software engineering standards and processes. Hopefully, students are able to understand the concept of software engineering better when they are required to implement the knowledge learned in a project.

One major learning outcome, thus, is for students to be able to work in group to participate in the full cycle of software development with the most appropriate process considering the underlying technology and project duration.

Each of the projects should follow a project development and management methodology and in this case the suggested methodology should practice lean software development such as Agile with Scrum. Accordingly, the project team shall report progress made in the project and the next plan in completing the project on a short periodic basis (e.g. 2-3 hours weekly and monthly sprint). The project may evolve through several sprint cycles to eventually produce an output or deliverable. A point to note is that due to the nature of the subject which is to be conducted within 14 weeks with each week having two hours of lecture class and two hours of lab session, the actual industry practice of daily Scrum (roughly one hour daily team meeting to discuss work done and what are left to be resolved) is not being implemented in this framework. Nevertheless, weekly sprints are carried out as replacement for the daily Scrum meeting.

The following sections present our proposed PBL framework, discuss challenges and provide implementation requirements for resolution of challenges.

A. The PBL Framework

The Agile with Scrum PBL implementation framework is shown in Figure 1. In this framework, it will incur 14 weeks of short development cycle whereby for each week, there will be a 2-hours lecture session and another 2-hours lab session. During the start of the project, students are to form into groups where each group consisting of 4 or 5 members. Each group is to propose or be assigned a project title and each student is to resume one or more roles in the project. The roles are for example team leader or in this case the Scrum Master, requirements engineer, designer or architect, developer, tester, documentation and record keeper. Although one student can have multiple roles, but every member has to participate in all the planning and development activities.

Weekly evaluation starting on week two will be conducted based on written report submitted or oral presentation. The assessment has two portions (refer to Table 1 for detail breakdown of assessments), one is based on individual contributions and the other portion based on group project deliverable at every stage following a progressive reporting and assessment flow as shown in Figure 1. The stages are Project Planning and Monitoring, Requirements Engineering, Design and Architecting, Testing, Error Fixing and Reporting and the final stage of Project Completion with project post mortem. The activities in each stage are iterative and incremental in nature with review and retrospective in between stages.

At the end of the 14 weeks, each group has to produce an end product, a working standalone system, IoT or web application with functionalities that must reflect the basic features such as Create, Read, Update and Delete (CRUD) of software system together with interactivity, computation and processing.

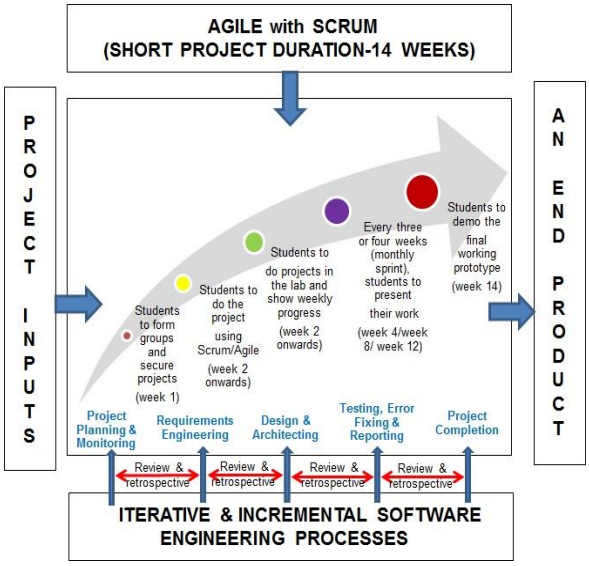

Figure 1: Our proposed PBL implementation framework

Table 1
Detail Breakdown of Assessment

Assessment (100% course work, no final examination)	Details
Project (80%)	• Weekly progress evaluation 20%, assessment on individual student based on lab works or class activities) • Final report (20%, assessment on whole group based on documentation) • Working prototype (40%, progressive assessment on functionalities of the prototype at various stages until final product)
Mid-Term Test (20%)	• Paper-based (closed book)

B. Challenges and Work Around

To implement this framework for SE subject may face some challenges. Therefore, possible work around has to be considered before implementation for optimum effective results. Some of these challenges include but not limited to, big class size; students' study culture; changing roles of instructor; difficulty of getting "real world" project, resource intensive (e.g. lab facilities) and so on. These challenges with the work around are discussed and shown in Table 2.

C. Sample Weekly Plan

As mentioned in Section B above that detailed weekly plan is required to guide the students through for successful development of the end product. An example of weekly plan for lecture class and lab activities is shown in Table 3. Due to space constraint, not all 14 weeks' schedule are shown. Also, week number and topics may not follow in the said sequence. However, all topics in the subject syllabus are covered. Some main topics under this subject not listed in the sample plan are for example, software design, design modelling, software quality, configuration management and so on.

Table 2
Challenges and Work Around

Challenges	Work Around
Big Class Size If the class size for SE subject is big, for example, more than 100 per trimester, then to use this subject as a pilot for PBL implementation would resemble a great challenge. This may lead to other challenges such as human resources, lab facilities and assignment of real world projects.	*Smaller Class Size* To have 25 students per lecture and tutorial session where each session consists of 5-6 groups of students Maximum of 2 lecture and 2 tutorial sessions to be handled by 1 lecturer. This lecturer shall be assigned only this SE subject and there will be no sharing of lecture or tutorial sessions by other subject's lecturer.
Students' Study Culture Most of the students have been nurtured in a traditional "class-room listening" rather than "self-initiated doing" way of study method. This PBL approach shall create a drastic change of role for the students. Many of them may not be able to cope with the sudden change and become more burdened in the study.	*Student Support* Provide thorough briefing to students regarding the aim, method and content of the project at the very beginning, for example, during first lecture. The level of scaffolding shall be greater at the early stage of the project with more lecturers' guidance and reducing as the project continues.
Changing Roles of Instructor The instructors would have more than one role to play, for example, in addition to the normal role as lecturer imparting knowledge to the students in class, he/she has to be a "mentor" in guiding and "assessor" in evaluating each project group from the beginning, progressively every week till the last week of the trimester.	*Choice of Staff and Training* Provide training for staff on PBL approaches. Choose the staffs already understand and have experience in PBL to kick start the pilot. Management to recognize and reward accordingly the extra time and resources contributed towards PBL by the staffs involved.
Real World Project To solve a real world problem through PBL and at the end producing a "real" product requires companies in the IT industry to become the stakeholders for the projects. It may be difficult to find such companies which are willing to offer the time and effort for this PBL implementation.	*University or Faculty Projects* There may be projects initiated at the university or faculty levels that could be implemented by PBL approach. The owners and users of these internal projects could be the stakeholders providing necessary inputs such as requirements, industry standards, processes and good practices.
Resource Intensive Each project group may use different tools, software, hardware and platform in delivering their deliverables and creating the "real" product.	*Lab Facilities* Have adequate lab facilities such as appropriate development tools, software, hardware, and network platform to facilitate the PBL implementation. This is to cater for lecture and tutorial sessions to be conducted in labs with adequate PCs and Internet access.
Bias Assessment There may be students getting free rides from other team members. They are just sleeping partners without contributing much towards the project yet getting the same marks as other group members.	*Fair Assessment* Conduct weekly class or lab activities such as peer review, oral Question and Answer (Q&A) or quiz, role playing, brainstorming and individual demonstration of task performed and get every student in the groups to participate. Marks shall be given for individual participation as well as the whole group's performance. Motivate students through competitions and reward them accordingly with prizes, for example.
Time Constraint Students may find the development schedule too tight as they may be taking more than one subject concurrently in the same trimester. Additionally, at certain stage of development, to perform certain tasks may require knowledge that has not been taught yet.	*Sequence Topics in Weekly Plan* Provide a detailed plan that lists out topics to be learned and activities to be carried for each week. Sequence the topics in such a way that students will obtain the knowledge first in the lecture session and then apply it in the next lab session.

Table 3
Weekly Plan with Topics and Class/Lab Activities

Week	Topics and Class/Lab Activities
Week x	**Topics** Agile Methodology Project Management Concepts (Scrum) Project Planning, Scheduling and Control **Class/Lab Activities** Role playing as Scrum Master, requirements engineer, architect, developer, tester and so on. Discussion or brainstorming on project planning, scheduling and tasks assignment <u>Assessment:</u> 1% out of 20% for individual student participation in class activity. 1% out of 20% for group documentation (project plan).
Week y	**Topics** Eliciting Requirements with Various Elicitation Techniques. **Class/Lab Activities** Eliciting requirements through interviews. One group to be the requirement engineers (interviewers) and the other group as stakeholders, then swap over the roles. <u>Assessment:</u> 1.5% out of 20% for individual participation in interview activity. 1.5 % out of 20% for group documentation (revised project plan with a list of functional requirements)
Week z	**Topics** Software Testing and Debugging. White Box and Black Box Testing and Techniques. **Class/Lab Activities** Designing test cases for functional testing of the project. Each group member to orally present the test case for functional testing for peer review. Revise the project plan to include the designed test cases. <u>Assessment:</u> 2% out of 20% for individual participation in designing test cases and peer review. 2 % out of 20% for group documentation (Revised project plan to include test cases for manual functional testing.)

Referring to Table 3 and take week x as an example, the topics covered in lecture are Agile methodology and project management with Scrum concept, project planning, scheduling and control. The class activities should be related to these topics such as role playing as Scrum Master, requirements engineer, designer, tester and so on. Another activity that can be conducted is to have a group discussion or brainstorming on project planning, scheduling and task assignments for the project during class or lab session. Students' performance shall be graded based on each individual's participation in the role-playing activity and as a group in creating the project plan.

IV. Characteristics of Our Framework

Our proposed PBL framework encompasses the characteristics such as "learning by doing", "resolving real world problem", "role of instructor as mentor", "interdisciplinary", "group work" and "an end product" [2]. Additionally, bundle with work around, weekly plan, agile or lean method of development with Scrum nature to project management, the framework exhibits flexibility of tailoring T&L towards changing environments. Under our framework, students have the option to propose their own suitable projects, but these projects have to be evaluated for feasibility, suitability and approved by the lecturers before commencing. This is in-line with the findings from [10] that when students are driven by intrinsic motivation, for

example, they select the project they like, then exceptional results could be achieved.

There is no single best teaching method, however, the selection of teaching method has to consider the background knowledge of the students and the nature (practical or theoretical) of the subject matter [10]. Therefore, we introduce blended learning strategy in our proposed framework. Class and lab activities are blended with the traditional way of class room lecturing with lecturers imparting knowledge to the students on one hand (theoretical), and on the other hands, students participating in activities such as peer review, role playing, hands-on using software tools, self-organized group discussion or brainstorming, oral presentation, and so on to gain the actual experience (practical).

Interviews conducted by [6, 7, 11] with students after they experienced project-based learning provide proven facts that PBL indeed could boost team spirit in an open learning environment. This indicates that students with different attitudes and personalities could work together in a team resembling working in a real working environment. Besides acquiring technical skills, other soft skills such as time management (e.g. students learn to prioritize tasks), communication skills (oral and written) and so on are among the most valuable skills for their future jobs.

Effective evaluations and measurement strategies are certainly needed to ensure that students are deriving maximum benefits from PBL and that the proposed PBL framework is being carried out in the most effective way. For this reason, a software engineering subject will be selected to pilot test if the implementation of PBL improves the performance of the students in acquiring and understanding the concepts of the chosen subject. One of the assessment methods is to compare the final results of students who have undergone PBL with the results of students that were taught using traditional method. By doing this, the success of PBL implementation can be ascertained and the proposed PBL framework can be improved gradually. On the other hand, qualitative analyses will be useful in obtaining individual reflections on PBL, which can be used to evaluate the effectiveness of PBL. Interviews and surveys can be done to ask the students to reflect their experiences on PBL and the results of the analyses can be used to further improve the efficacy of the proposed PBL framework.

V. Conclusion and Future Work

As the traditional curriculum with "chalk and talk" pedagogy has many drawbacks, it is crucial to adopt a project-based learning method to help students to understand the underlying knowledge as well as to develop their non-technical skills before embarking their steps into the industry especially in this era of IoTs. In this paper, we propose an Agile with Scrum project-based learning implementation framework that would be implemented in the near future into the software engineering curriculum. This framework is expected to help software engineering students to experience the management of a project in a more effective way as well as improving their technical and non-technical skills. Besides this, an effective evaluation strategy (i.e. qualitative analysis) will be adopted to measure the effectiveness of the proposed PBL framework and the outcome of the evaluation will be used to improve the proposed framework.

References

[1] Z. Chen, "Active Learning through combination of project-based learning with problem-based learning in engineering education," in *Proc. of the International Conference on Social Science and Higher Education (ICSSHE 2015)*, 2015, pp. 68-71.

[2] N. Harmer, *Project-Based Learning-Literature Review*. Plymouth University, 2014, pp. 1-33.

[3] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q.Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1-20, Feb. 2017.

[4] E. John, L. Kalle, and S. Keng, "Agile modeling, agile software development, and extreme programming: the state of research," *Journal of Database Management*, vol. 16, no. 4, pp.88-100, Oct.-Dec. 2005.

[5] K. Gerd, A. K. Bandara, N. Smith, M. Richards, and M. Petre, "Educating the internet-of-things generation," *Computer*, vol. 46, no. 2, pp. 53-61, 2013.

[6] N. Ibrahim, and S. A. Halim, "Implementation of project-oriented problem-based learning (POPBL) in introduction to programming Course," in *Proc. of the 4th International Research Symposium on Problem-Based Learning (IRSPBL)*, 2013, pp. 279-288.

[7] N. Ibrahim, and S. A. Halim, "Generic framework design of project-oriented problem-based learning (POPBL) for software engineering Courses," in *Proc. of the 8th Malaysian Software Engineering Conference (MySEC)*, 2014, pp. 359-364.

[8] D. Delaney, and G. G. Mitchell, "PBL applied to software engineering group projects," in *Proc. of the International Conference on Information and Communication in Education*, 2002, pp. 1093-1098.

[9] K. Schwaber, and J. Sutherland, "The definitive guide to scrum: the rules of the game," in *The Scrum Guide™*, 2016, pp. 1-17.

[10] R. Pucher, and M. Lehner, "Project-based learning in computer science-A review of more than 500 projects," *Procedia-Social and Behavirol Sciences*, vol. 29, pp. 1561-1566, 2011.

[11] A. Breiter, G. Fey, and R. Drecshler, "Project-based learning in student teams in computer science education," *SER.:ELEC. ENERG.*, vol. 18, no. 2, pp.165-180, Aug. 2005.