Journal of Engineering and Applied Sciences 13 (Special Issue 12): 9308-9312, 2018
ISSN: 1816-949X

The Adoption of SWEBOK Driven Project-Based Learning to Instil Professional Software Engineer Characters in Students

Beni Suranto
Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, Yogyakarta, Indonesia

Abstract: This research enacted Project-Based Learning (PBL) strategy to improve the quality of the learning process in Software Engineering (SE) course in Department of Informatics, Universitas Islam Indonesia. PBL was adopted by combining the waterfall process model and SE professional practices suggested in the Software Engineering Body of Knowledge (SWEBOK) Version 3.0. With the application of appropriate professional practices in accordance with SWEBOK 3.0, students of SE course is expected to have the professional characters of software engineers. Two classes of SE being the objects of the study were treated differently in terms of intentions and tolerance to the team. From the results obtained, the adoption of PBL in accordance with SWEBOK 3.0 can significantly help to instill professional characters of software engineers in students. Classes treated more rigidly showed far better learning outcomes than the class treated with tolerance.

Key words: Software engineering, project-based learning, SWEBOK, learning outcomes, professional characters, tolerance

INTRODUCTION

Software Engineering (SE) course is one of the compulsory subjects in Department of Informatics in UII. In the current curriculum, SE is aimed to provide insight to students about knowledges and skills of SE processes, methods and tools and how to enact them systematically and measurably in software development. Students who have graduated from SE course are expected to have the following abilities (Teduh, 2013):

- The ability to notice software problems and what is needed to resolve it
- The ability to design software that fits to the need
- The ability to manage software development project

In particular, the SE materials are directed to form a student competence as a software developer. In general, understanding and skills acquired in the course can be utilized in other areas of interest such as the ability of system requirements analysis, system testing and project management. SE course is a part of core subjects of informatics as recommended by the Association of Indonesian Higher Education of Computer (APTIKOM).

This research was aimed to instil the characters of professional software engineers in the students taking SE course. The strategy applied was integrating practices in Software Engineering Body of Knowledge (SWEBOK) 3.0 with the Project-Based Learning (PBL) used as the learning model.

SWEBOK is an international standard that provides specifications guidance related to knowledge and practices in the profession in the field of SE (Bourque and Fairley *et al*., 2014). SWEBOK is one of standards that becomes an important part in the SE education (Ardis *et al*., 2011). SWEBOK 3.0 defines practices that relate to the character of a professional software engineer as shown in Table 1.

The guidelines in SWEBOK will be very helpful in understanding and mastering of knowledge and skills in the field of SE. In addition, SE learning based on SWEBOK will produce professional candidates who are better prepared to work in software industry (Fairley *et al*., 2014).

Project-Based Learning (PBL) can be defined as a comprehensive approach to teaching and learning that is designed to engage students in the investigation and resolution of an authentic problem. This model encourages students to be able to produce a solution to a real problem by asking questions, communicating ideas,

Table 1: Practices in SWEBOK 3.0 that relate to the character of a professional software engineer

Professionalism	Group dynamics and psychology	Communication skills
Accredition, certification and licensing	Dynamics of working in team/groups	Reading understanding and summarizing
Codes of ethics and professional conducts	Individual cognition	Writing
Nature and roles of profesional socities	Dealing with problem complexity	Team and group communication
Nature and roles of software engineering standards	Interacting with stakeholders	Presentation skills
Economic impact of software	Interacting with uncertainty and ambiguity	
Employment contracts	Dealing with multiculture environments	
Legal issues		
Documents		
tradeoff analysis		

making predictions, planning and conducting experiments, retrieving and processing data, making a prototype and making conclusions (Lee *et al*., 2014). Some of the advantages and benefits of PBL proven in previous studies were as follows:

PBL is able to teach various important strategies for students to be successful in this era. This is possible because PBL provides great motivation for students to conduct research and work collaboratively in order to complete a project that reflects their knowledge and skills. This approach is considered capable of forming students into problem solver figure who have superior communication skills (Bell, 2010).

PBL effectively increases the participation and involvement of students in the learning process in terms of behavior, cognition as well as emotions (Johnson and Delawsky, 2013).

PBL help students practice working under pressure, look for alternative solutions together with colleagues in a team and sharpen the active communication skills (Jaramillo, 2014).

PBL can motivate students to improve spirit of competition interact and collaborate in a positive way and dare to take the initiative (Hung *et al*., 2012; Burguillo, 2010).

MATERIALS AND METHODS

This research enacted the proposed PBL framework as shown in Fig. 1. This framework has three important components which are PBL as the learning method, SWEBOK as guidance for SE professional practices and waterfall process model as guidance for the project.

The practices which are mentioned in SWEBOK 3.0 and represents the characters of professional software engineer were breakdown into activities which was conducted during the learning process of SE course. These activities is shown in Table 2.

Practices listed in Table 1 were designed to be part of activities conducted by the students during SE course project. SE project was conducted in groups of each four to five students. All teams are required to complete a software development project for one semester (16 weeks). The implementation of software development projects on SE course use Waterfall process model which is a model of software development done through disjunct stages that are sequential, consisting of requirements analysis, system design, system implementation, system testing and system maintenance (Sommerville, 2011).

Waterfall process model is selected in order to facilitate the results evaluation of SE team project. In addition, waterfall is process model which is most easily understood by the students. Waterfall process model is shown in Fig. 2.

In-class lectures for SE course is still held to convey the fundamental materials in accordance with the syllabus of the course. In addition it is used to control and evaluate the implementation of the project by each team. The author as lecturer also avail the facility of Klasiber e-Learning system for materials distribution and discussions with students taking the course.

At the end of the semester, poster exhibition is held for the publication of software products that have been successfully developed by all project teams. The poster exhibition has never been done in the previous semesters and is intended to encourage all student to produce splendid software products. Detail of weekly activities in SE course adopting PBL is shown in Table 3.

This research involves two SE classes which are C class which was scheduled every Wednesday 09.30 AM-12.00 PM and F class which was scheduled every Monday at 3.30-6.00 PM. The author enacted different treatments to both classes in order to see the influence of attitude tolerance and level of practices implementation that represent the professional characters of software engineers on learning outcomes. Different treatment for C and F classes can be seen in Table 4.

Table 2: Activities conducted during the SE project

Domain	Practices	Activities
Professionalism	Documentation	Writing team journal
		Arranging Software Requirements Spesification (SRS), Software Design Document (SDD), testing plan and testing report documents
		Making poster
Group dynamics and psychology	Dynamics working in teams/groups	Team work in project implementation
	Individual cognition	Completing task based on team member
		Implementing SE techniques relevant to task given to team member
	Dealing with problem complexity	Resolving problems related to system requirement complexity
	Interacting with stakeholders	Interacting with stakeholders during the project (conducting need analysis, plan discussion, result of implementation presentation, and User Acceptance Test)
	Dealing with uncertanty and ambiguity	Overcoming stakeholder's unclear explanation on need specification
		Making sure the project runs on time
		Making sure clear information flow within the team
Communication skills	Reading, understanding, and summarizing	Reading course materials and references (articles) related to project stages
		Understanding knowledges related to project stages
		Understanding knowledges related to techniques relevant to project stages
		Summarizing informations received from stakeholders
	Writing	Writing SRS, SDD, testing plan and testing report documents
	Team and group communication	Internal discussion with the team during project completion
		Discussion with other teams through Klasiber forum or Facebook group
	Presentation skills	Presenting the result of each stage to lecturer and stakeholders
		Presenting poster

Table.3: Schedule and topics of SE course

Weeks	Topic	Deliverable
1	Course introduction Learning contract Project groups set up	Project groups set up
2	Software Process model	Project descriptions
3	Requirement analysis	
4	Requirement analysis	
5	Requirement analysis (presentation)	Software Requirement Spesification (SRS)
6	Design	
7	Design	
8	Design (presentation)	SoftwareDesignDocument(SDD)9
9	MID EXAM10	
10	Implementation	
11	Implementation	
12	Implementation	
13	Testing	Testing plan
14	Testing (presentation)	Testing report
15	Project final presentation and poster exhibition	Software, Poster
16	Final exam	

Table 4: Different treatments for two classes

C-class	F-class
Project teams and topics were randomly choosen by lecturer	Students choose their own team and topic
In-class lectures always started with discussions, lecturer give questions for all teams, lecturer randomly choose team member to answer the question	Not every lecture started with discussion. Questions from lecturer can be answered by any team member .
For each project stage, each student must start at least 1 discussions threads and take part in at least 3 discussion threads	No obligation for online discussions (optional)
All members must be present and take part in all presentation sessions	Not all members must be present and take part inall presentation sessions
Weekly journal is compulsory	No obligation for writing weekly journal

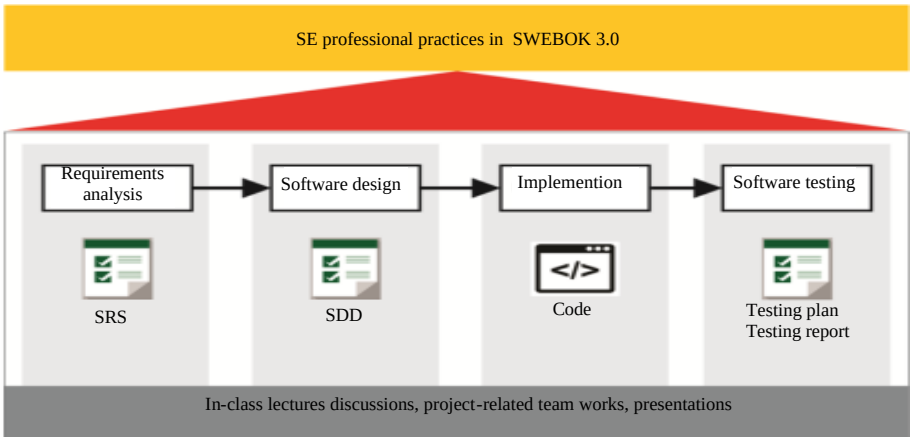

Fig. 1: SWEBOK driven PBL framework

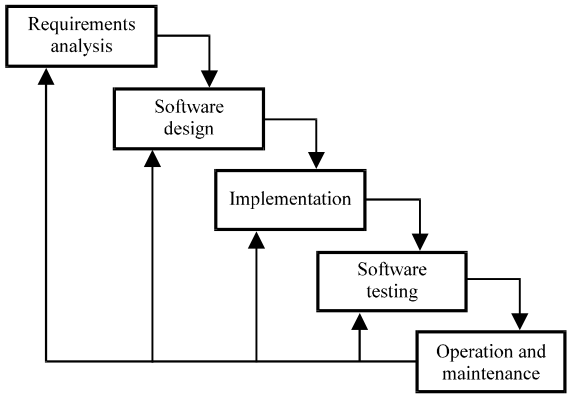

Fig. 2: Waterfall process model

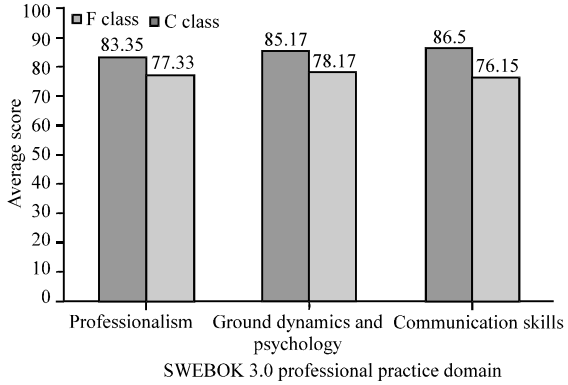

Fig. 3: Average score for SE professional practices

RESULTS AND DISCUSSION

From the results of team and individual evaluations, the adoption of PBL has been proven to effectively enhance the professional character in students as future professional software engineers. This is proven by the average score for the three SWEBOK 3.0 domains which becomes the target of PBL adoption. As shown in Fig. 3, average score for students in both classes is above 75 for all domains (professionalism, group dynamics and psychology and communication skills).

As shown in Fig. 3, the difference in treatments, related to the intentions of the activity and tolerance to the project teams for C and F classes, significantly affect the results achieved in each class. It can be concluded that rigidity with multiple obligations "enforcing" students to be active and professional is able to effectively encourage students to demonstrate an attitude that represents the characters of professional software engineers.

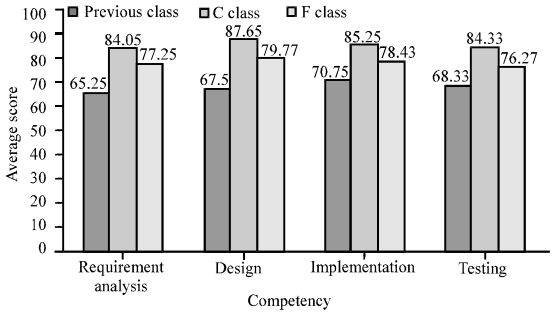

Fig. 4: Average score for SE competencies

Furthermore, considering the understanding achievements of the material according to the syllabus of the course, the adoption of PBL was also shown to have helped students achieve competency standards as the average score obtained by students in both classes for all material is above 75. This result is much better compare to the previous SE classes as shown in Fig. 4.

CONCLUSION

From the research result, here are some conlusions drawn: Adoption of PBL in SE course learning process significantly helps students to be able to do professional practices as suggested in SWEBOK 3.0. The implementation of practices suggested in SWEBOK 3.0 in SE project can effectively help to instill the characters of professional software engineers in students. Students treated rigidly in the project and class activities are able to behave more professionaly and achieve better learning outcome.

ACKNOWLEDGEMENT

This research was supported by the Board of Academic Development teaching grant at Universitas Islam Indonesia.

REFERENCES

Ardis, M., P. Bourque, T. Hilburn, K. Lasfer and S. Lucero *et al*., 2011. Advancing software engineering professional education. IEEE. Software, 28: 58-63.

Bell, S., 2010. Project-based learning for the 21st century: Skills go the future. Clear. House, 83: 39-43.

Bourque, P. and R.E. Fairley, 2014. Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. 3rd Edn., IEEE Computer Society Press, Los Alamitos, California, ISBN:9780769551661, Pages: 348.

Burguillo, J.C., 2010. Using game theory and competition based learning to stimulate student motivation and performance. Comput. Educ., 55: 566-575.

Fairley, R.E.D., P. Bourque and J. Keppler, 2014. The impact of SWEBOK Version 3 on software engineering education and training. Proceedings of the 2014 IEEE 27th International Conference on Software Engineering Education and Training (CSEE&T), April 23-25, 2014, IEEE, California, USA., ISBN: 978-1-4799-4970-0, pp: 192-200.

Hung, C.M., G.J. Hwang and I. Huang, 2012. A project-based digital storytelling approach for improving students' learning motivation, problem-solving competence and learning achievement. J. Educ. Technol. Soc., 15: 368-379.

Jaramillo, C.M.Z., 2014. Teaching software development by means of a classroom game: The software development game. Developments Bus. Simul. Experiential Learn., Vol. 36,

Johnson, C.S. and S. Delawsky, 2013. Project-based learning and student engagement. Acad. Res. Int., 4: 560-570.

Lee, J.S., S. Blackwell, J. Drake and K.A. Moran, 2014. Taking a leap of faith: Redefining teaching and learning in higher education through project-based learning. Interdiscip. J. Prob. Based Learn., 8: 2-17.

Palacios, R.C., T.E. Caro, G.A. Crespo and G.J.M. Berbis, 2012. Identifying Technical Competences of IT Professionals: The Case of Software Engineers. In: Professional Advancements and Management Trends in the IT Sector, Ricardo, C.P. (Ed.). IGI Global, Madrid, Spain, pp: 1-14.

Sommerville, I., 2011. Software Engineering. 9th Edn., Pearson Inc., USA., ISBN: 9780137053469, Pages: 773.

Teduh, D., 2013. Course outline software engineering. Ph.D Thesis, Islamic University of Indonesia, Yogyakarta, Indonesia.