Report on Problem Based Learning for Software Engineering

Sørensen, Lene Tolstrup; Falch, Morten; Skouby, Knud Erik

Published in:
Proceedings of the 2018 Workshop on PhD Software Engineering Education (SWEPHD2018)

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Report on Problem Based Learning for Software Engineering *

Lene Sørensen
Electronic Systems, CMI
Aalborg University Copenhagen
Copenhagen S, Denmark
ls@cmi.aau.dk

Morten Falch
Electronic Systems, CMI
Aalborg University Copenhagen
Copenhagen S, Denmark
falch@cmi.aau.dk

Knud Erik Skouby
Electronic Systems, CMI
Aalborg University Copenhagen
Copenhagen S, Denmark
skouby@cmi.aau.dk

ABSTRACT

Software Engineering is far more than programming and requirements specifications. In the SWEBOK, software engineering is defined through 15 different disciplines and topics also covering soft skills such as project management and inter-cooperation between teams. Problem Based Learning, PBL, is a pedagogical learning approach used amongst others at AAU. This paper discusses how PBL can support software engineering and provide soft skills called for in industry and investigates how Ph.D. students with a traditional software engineering background understand and perceive PBL. The paper is based on a case study prepared as a part of the European Erasmus Project Paths Ways to Ph.D. In this project skills in software engineering has been exchanged and developed for the target countries Russia and Jordan as a way to support their process of developing Ph.D. programs in software engineering.

CCS CONCEPTS

• Social and professional topics~Software engineering education

KEYWORDS

SWEBOK, Problem Based Learning, Software Engineering

ACM Reference format:

Lene Sørensen, Morten Falch and Knud Erik Skouby. 2018. Report on Problem Based Learning for Software Engineering . In *Proceedings of the 2018 Workshop on PhD Software Engineering Education: Challenges, Trends, and Programs (SWEPHD2018). St. Petersburg, Russia, 6 pages.*

1 Introduction

Software engineering is a discipline with a broad definition and includes numerous competence areas. Moreover software engineering has an application oriented focus, which incorporates theories and methodologies from many different disciplines within engineering, science and economics. If looking at the Guide to the Software Engineering Body of Knowledge [1], it becomes clear that software engineering as a discipline is based on more than 17 Knowledge Areas and related disciplines and that each of these represent a discipline in itself. All in all, these areas employ techniques of math, science, engineering and design and cater for strong analytical and problem-solving skills as well as communication and interpersonal skills [2].
The interdisciplinary and application oriented focus of software engineering challenges the traditional teacher centred learning methods, as hands-on experience is crucial for developing competences within software engineering. The software engineering industry demands that software engineers have skills of creativity, problem solving, analytic, cooperative, and interpersonal skills [3]. With a tighter focus on stakeholder needs, and agile development, the software engineer must be able to interrelate with stakeholders and not only able to program the software [3].

This paper discuss how these challenges can be met by the use of alternative methods of teaching with less focus on one-way teaching, which better can activate. The paper reports on the experiences with project-oriented problem-based learning (POPBL or PBL) as an inherent pedagogy within software engineering in the two-year ERASMUS+ project Joint Programs and Framework for Doctoral Education in Software Engineering, pws@phd. More precisely, the paper analyses how the Ph.D students participating in the project have found the PBL useful for their Ph.D. work. The PBL ideology has been used throughout the project and has initiated new discussions amongst the participating partners and has placed focus on the “soft skills” in discussions about new Software Engineering Ph.D. programmes in Russia and Jordan.

Problem Based Learning, PBL, has in many years been used as the primary teaching and learning form of active learning at universities such as Aalborg University. PBL is a pedagogical approach in which students work with authentic problems, self-governed group work and collaborate in problem solving and learning [4]. The skills students achieve through working and learning PBL resemble the skills, which are asked for in real software engineering work [3]. PBL is therefore often considered to be a fundamental basis for learning and for securing students’ can work in industries with real challenges.

The paper is organized as follows: Section 2 outlines the analytical framework for the paper. In section 3, the pws@phd project is

The 2018 Workshop on PhD Software Engineering Education: Challenges, Trends, and Programs, September 17th, 2018, St. Petersburg, Russia

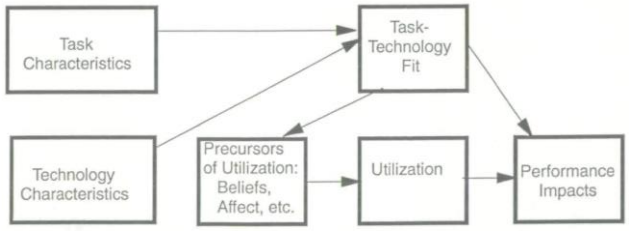
described in more detail to understand the foundation of the reporting. Section 4 describes the SWEBOK and how the skills achieved from Problem Based Learning are linked. The teaching and learning perspectives of the PBL approach and how they have been used in the project are described in section 5. Additionally, section 5 includes an overview of a survey which has been given to students in the project after taking classes in PBL. Section 6 discusses the learnings from the project and the conclusion.

2 Analytical Framework

Various versions of the technology adoption model, TAM [5]), have been applied for studies of teaching technologies such as e-learning [6], Massive Open Online Courses (MOOC) [7], and Learning Management Systems (LMS). PBL does not presuppose the use of any specific technology (although LMS systems are widely used). Still PBL can be seen as a kind of teaching technology, and thus TAM will be an appropriate framework for analysing the adoption of PBL. In this paper we are dealing with the use of PBL within a specific study discipline (Software engineering), we suggest to use the Technology Task Fit model (TTF) [8], which also has been applied for studies on the adoption of MOOCs [7] and studies of LMS [9].

According to this framework, the main factors affecting performance of a particular technology are technology characteristics and task characteristics, how these fit together and how the technology is utilized (see Figure 1).

Figure 1 Link from Technology to performance



Source: [8]

The technology-task fit framework, as well as many other TAM models are created with the purpose of preparing a quantitative analysis of the statistical relationship among the various factors based on input from questionnaires. This approach will not be applied in this case. Even though quantitative data have been collected, they are not suitable for this kind of analysis. We will therefore in this paper focus on a qualitative analysis.

A qualitative analysis has been made of data collected through open-ended questions delivered in a survey. The participants (students) were present in a school room while providing the input to the survey which secured that all of them delivered the survey. The survey results have afterwards been analysed for perspectives and themes which have been raised by the students and interpreted. This process is referred in [10], p. 184-185) to be a common way of analysing qualitative data.

3 PWS@PHD – the Project

In order to enhance PhD training in software engineering the European Erasmus+ K2 project, Joint Programs and Framework for Doctoral Education in Software Engineering, pws@phd was initiated in 2015. It is a 3-year project where 11 partners from Finland, UK, Germany, Russia, Jordan and Denmark collaborate and exchange knowledge and expertise between institutions (fase.it.lut.fi). The purpose of the project is to develop guidelines for Russia and Jordan with respect to setting up their own Ph.D. programs in Software Engineering, and through intensive schools in the project to illustrate content and areas of expertise to include in the national Ph.D. programmes. The project has conducted seven two-weeks PhD schools each focusing on one particular topic within software engineering. With the exception of the school, which focused problem-based teaching methods, all topics were defined by the SWEBOK.

The project has created a network of universities that all have expertise in various areas of Software Engineering. Russia and Jordan are so-called target countries for the project. The target countries have a limited number of degree programs in Software Engineering especially at the Ph.D. level, where no specific programmes are available.

The purpose of the project is therefore twofold: To make guidelines for the target countries with respect to internationalisation and creation of Software Engineering Programmes for PhDs; and to exchange knowledge and information on key areas within software engineering which can motivate content for the proposed Ph.D. Programmes.

The exchange of knowledge and information takes place through the 7 intensive schools held by partner universities. The 7 intensive schools each represent knowledge areas and disciplines related to key areas in Software Engineering and for a typical knowledge and skills that a Ph.D. student within Software Engineering would need to acquire. The schools are organized through 2 weeks of intense course for Ph.D. students from the partner universities. The students can select which of the schools they would like to attend and therefore do not need to attend them all.

The topics for the seven intensive schools were: Research methods; Problem Based Learning; Advanced Software Engineering Methods and Tools; Math and Computing Foundation's; Software Engineering Models and Modelling; Human Computer Interaction and Business and Economic Viewpoints. Details can be found at fase.it.lut.fi.

During the project period the researchers involved in the project have conducted a number workshops and seminars to produce guidelines for the target countries with respect to how they should set-up Ph.D. programmes in Software Engineering and what the content could be. However, the actual implementation of the guidelines is highly dependent on the political and organizational set-up in the target countries, which is outside the scope of the project.

The goal of the project is to motivate the target countries to create Ph.D. students in software engineering with an understanding of the international education programmes and with skills that can create international possibilities and interest in them, as well as local industry relevance and possibilities. Local industries in Russia and Jordan call for specialized Software Engineering experts with international skill and knowledge.

A total of approximately 180 students has attended the 7 schools from the partner countries. Some of these (around 10) have attended more than one school.

4 SWEBOK and PBL

The skills that are achieved by Problem Based Learning is applicable as skills in Software Engineering.

4.1 About the SWEBOK

The Software Engineering and Systems Engineering Vocabulary hosted by IEEE offers two different definitions: (1) "systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software" (2) "application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software". (http://www.computer.org/sevocab). The latter is the definition referred to in the widely recognized SWEBOK. The major difference between the two definitions is the focus on a quantifiable approach. It should however be noted that SWEBOK includes qualitative as well as quantitative methods.

The SWEBOK [11] is a body of knowledge which describes generally accepted knowledge about software engineering. Software engineering is presented within 15 knowledge areas and basic concepts. A total of 150 reviewers representing 33 countries have worked on the book (ibid). It is published within the IEEE Computer Society. The purpose of the SWEBOK is (Bourquet and Fairley, 2014):

1. To promote a consistent view of software engineering worldwide
2. To specify the scope of, and clarify the place of software engineering with respect to other disciplines such as computer science, project management, computer engineering, and mathematics
3. To characterize the contents of the software engineering discipline
4. To provide a topical access to the Software Engineering Body of Knowledge
5. To provide a foundation for curriculum development and for individual certification and licensing material.

The 15 knowledge areas about software are (Bourquet and Fairley, 2014):

1. Requirements
2. Design
3. Construction
4. Testing
5. Maintenance
6. Configuration Management
7. Engineering Management
8. Engineering Process
9. Engineering Models and Methods
10. Quality
11. Engineering Professional Practice
12. Engineering Economics
13. Computing Foundations
14. Mathematical Foundations
15. Engineering Foundations.

The pws@phd project covers requirements (1), design (2), construction (3), testing (4), models and methods (9), economics and mathematical foundations (14) as well as engineering management 7). For details about the content and definition of the knowledge areas, see [11].

Generally, the SWEBOK covers multiple disciplines and knowledge areas and many of these are related to: managing the software process or parts of this process; involving and understanding stakeholders of the software and the software process; as well as cooperating between different disciplines. The essence of these activities is creativity, cooperation, problem solving and team work – all features of the PBL approach. More formal, the SWEBOK includes the following non-technical skills [2]: Professionalism (ethics, economics impact of software, legal issues etc.); Group Dynamics and Psychology (group dynamics, individual cognition, dealing with problem complexity, interacting with stakeholders etc.); and Communication Skills (writing, team and group communication, presentation skills).

4.2 About Problem Based Learning

Problem based learning was first introduced at McMaster University in Canada back in 1969 [12]. Here teaching of medical students were presented to problem based cases, which were used as a point of departure for their learning. In the 1970's different variations of problem based learning was introduced in a number of other countries including Netherlands, Australia, Sweden and Denmark.

In this period two new universities were established in Denmark: Roskilde University (RUC) in 1972 and Aalborg University (AAU) in 1974. Here PBL was introduced as the main pedagogical model for teaching right from the beginning. The model applied was a further development of the case based PBL model introduced at

McMaster University, as the key learning activity was project work carried out in groups. This variation of PBL can be denoted Project Oriented Problem Based Learning. At Aalborg University the basic principles of the PBLmodel are [4]:

- Project organization creates the framework of problem-based learning
- Courses support the project work
- Cooperation is a driving force in problem-based project work
- The problem-based project work of the groups must be exemplary
- Students are responsible for their own learning achievements

Each semester, the students must do a new project in collaboration with other students. They should spend half time working on project work and half time attending courses. A common theme for the projects is defined in the curriculum, but it is the task of the students to decide on their own topic, formulate a problem formulation for the project, and argue how this can be linked to the semester goals. Working with software engineering this often means a combination of many different skills is necessary to carry out the project.

5 PBL and SE

Problem Based Learning is a key for working in interdisciplinary teams, finding new solutions and being creative. Essentially, these elements are sought when working with software engineering both in industry and at university level.

5.1 About Soft skills in Software Engineering

Within the project pws@phd, PBL has been taught at a school in Copenhagen in 2016. At this school, Ph.D. students from partner countries were taught about the principles of PBL and they tried to work with PBL principles in order gain a better understanding of their own research work in Software Engineering. Furthermore, industrial representatives presented their views on important skills for Ph.D. students going into industry. One of the presenters from Microsoft, Denmark, pointed to team work and the development of interpersonal, creative and problem-solving skills as the most important skills of a software engineer. This is backed up by [13] who has described soft skills required as a software engineer in Microsoft. He points to skills such as: change management (being able to work with different methods and procedures amongst others); self-development (keeping up to date in knowledge and education); composure (stress management); problem solving skills (relates to bug fixing, general problems, analytical and methodological procedures); other soft skills (such as interpersonal communication skills, being able to communicate with all).

In [3] the authors have made an analysis of 500 advertisements for IT positions with a focus on identified the soft skills mentioned in the ads. The outcome is that communication and interpersonal skills are in high demand as well as analytical and problem-solving skills. The authors think that there are too little emphasis in ads about soft skills.

Sedelmaier and Landes (2014) developed a Software Engineering Body of Skills (SWEBOS). Based on the hierarchy made in the SWEBOK, [2] has developed a similar hierarchy divided into the following competence areas: Competences for the professional collaboration with others; Communicative competences,; Competences for structuring one's own way of working; Personal Competences; Capability to understand complex processes, systems and relationships; Capability to apply one's individual knowledge and skills to concrete and novel situations; and additional competences such as accepting responsibility etc.

5.2 PBL what do students think?

At the intense school in Copenhagen in 2016, 25 Ph.D. students from the pws@phd project participated in an intense school with focus on problem-based learning and the link to software engineering. The school was organized by Aalborg University and took place during 2 weeks in August, 2016. A survey was distributed to the students during the intense school and after to understand which skills they achieved. The survey was formed as a collection of closed questions using a Likert scale and an option to write-in. Additionally, some questions were completely open for write in for the students. The students answered to the questionnaires during the school and on paper, which secured that all students participated. (however differing answers for each question since some students did not answer to all questions)

During the school, the students were experiencing a combination of different teaching and learning forms: presentations about PBL from university staff; presentations from industrial representatives who came to talk about important skills to achieve as an engineer to work in industry; exercises and group work (exercising the PBL elements) on understanding software engineering and on producing abstracts and performing reviews as a process linked to be an academic and do disseminations.

The PhD students particularly liked the presentations and the possibility to talk with industrial representatives. 17 out of 21 of the Ph.D. students were very satisfied or extremely satisfied with these visits and wanted to have more time and to get more insight into the companies.

The PhD students liked the group work where they themselves should try out the principles of PBL in a small group work over a couple of days. Asked about how they liked the group work 16 out of 20 students were very satisfied or extremely satisfied with the group work and though it was a useful experience and that could add to insights on alternative perspectives on how the group participants thought about the topics. On top of this also some PhD students experienced that the dynamics of the group work was not so easy and that good group dynamics were extremely important for the performance and learning.

The students were asked directly on which three elements from the first week, they found most interesting. The elements identified were: Group work (13 students wrote that); abstract exercise (11 students); company visits (9 students) – additionally, the students mentioned that they liked the networking, teamwork, the learning process, communication with other PhD students.

In the second week, the students were asked to work with the concept of wicked problems [14], which basically explains that not all questions are well-defined and can be solved by mathematical solutions but wicked problems need more work and are related to the principles of PBL. The students were happy about the exercises on wicked problems, where 8 of the 19 students answering were satisfied or extremely satisfied.

When asking the students about the skills they thought they acquired during the school, they mentioned the following perspectives:

- Communication skills
- Thinking outside the box
- Time management
- Knowledge sharing
- Presentation training
- Working in groups
- Discussions
- Understanding of problem solving and problem formulation
- Better understanding of PhD in other countries
- Understanding wicked problems
- Relevance for PhD students to enhancing skills regarding thesis work
- Reacting calm to criticism
- Communication
- Solving problems through game strategy.

These skills link very well to the topics which were covered and exercised during the intense school.

5.3 Overall Themes

Analysing the data and the number of answers received in each category, there can be identified the following overall themes relating to the survey answers and to which skills the students think they have acquired during the Problem-Based Learning school:

- Communication skills
- Team and group working skills
- Language skills
- Understanding wicked problems
- Presentation skills
- Problem Solving
- Creativity and critical thinking

These themes were covered more or less as topics and experience working with the Problem-based Learning principles. Since the intense school only lasted 2 weeks, the students could not acquire skills in project planning or in understanding of practical, complex problems.

Generally, there is a call for soft skills within software development. As already mentioned, Ahmed et al (2012) have analysed more than 500 advertisements for IT positions and have looked at which soft skills are demanded. Communication skills is the absolute highest demand in all adds. After this follows interpersonal skills, analytic and problem-solving skills and team players.

[2] has developed a Software Engineering Body of Skills (SWEBOS). In this they have identified and categorised the soft skills that relates to the non-technical side of the SWEBOK. As part of having competencies for professional cooperation, the software developer needs to have competences in cooperating with others, communication, presenting skills, self-reflection and respect for others, to mention a few (Ibid). As part of the competences for structuring own's own way of working there is mentioned analytical thinking and problem structuring skills (ibid). and as part of the competences in solving problems, there is mentioned that the software engineer must have create potential, and must be capable to evaluating different approaches etc. (ibid). The SWEBOS (ibid) include

6 Conclusion

This paper has focused on understanding how Ph.D. students who generally come from a traditional Software Engineering background can understand the need for also acquiring soft skills for use in future works. From the empirical study carried out, it is clear that the students during the intense course in Problem Based Learning have acquired skills within a number of relevant soft skills categories.

The paper shows that Problem Based Learning train students' skills in important soft skills relating to software development. The soft skills are important for the students to become a full member of organisations where software development is being exercised.

Based on the experiences from the project pws@phd, it can be concluded that such a project has potential and significance to exchange knowledge and skills for the future software engineers. There is a high likelihood that the software engineering Ph.D.s who have participated in schools during the project period, has a higher chance to understand the requirements and needs for becoming internationally compliant software engineers by acquiring early understanding also on soft skills.

ACKNOWLEDGMENTS

The authors of the paper appreciate students' participation in the survey. The paper is made as part of an Erasmus+ K2 grant.

REFERENCES

[1] P. Bourque and R.E. Fairley (Eds.). 2014. *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, www.swebok.org.

[2] Y. Sedelmaier and D. Landes. 2014. Software Engineering Body of Skills (SWEBOS), 3-5 April 2014, Military Museum and Cultural Center, Harbiye, Istanbul, Turkey, IEEE Global Engineering Education Conference (EDUCON).

[3] F. Ahmed, L.F. Capretz and P. Campbell. 2012. Evaluating the Demand for Soft Skills in Software Development. *IT Pro*, January/February 2012, Published b the IEEE Computer Society.

[4] I. Askehave, H. L. Prehn, J. Pedersen and M.T. PEdersen. 2015. PBL Problem Based Learning. Aalborg University, www.aau.dk.

[5] F.D. Davis, 1993. User Acceptance of Information Technology: System Characteristics, User Perceptions and Behavioural Impacts. *International Journal of Man-Machine Studies*, 38(3): 475-487.

[6] Ertl, B., Winkler, K., & Mandle, H. (2007). E'Learing: Trends and Future Development. In Advances in computer-supported learning (pp. 122-144). IGI Global.

[7] B. Wu, & X. Chen, 2017. Continuance intention to use MOOCs: Integrating the technology acceptance model (TAM) and task technology fit (TTF) model. *Computers in Human Behavior*, 67, 221-232.

[8] D. Goodhue. 1995. Understanding User Evaluations of Information Systems. *Management Science*, 41(2), 1827-1844

[9] T.J. McGill and J.E. Klobas. 2008. A Task-Technology Fit View of Learning Management System Impact. Computers and Education 52(2), 496-508.

[10] J.W. Creswell, 2009. Research Design. Qualitative, Quantitative and Mixed Methods Approaches. Sage Publishers, 3rd Edition.

[11] P. Bourque and R.E. Fairley (Eds.). 2014. *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, www.swebok.org.

[12] J.E. Holgaard, T. Ryberg, N. Stegeager, D Stentoft and A.O. Thomassen. 2014. *Problembaseret Læring og Projektarbejde ved De videregående Uddannelser*. Samfundslitteratur. IN DANISH.

[13] M. Orsted. 2000. Software Development Engineer in Microsoft. A Subjective View of Soft Skills Required. ICSE, 2000 Limerick, Ireland.

[14] J.C. Caillus. 2008. Strategy as a Wicked Problem. *Harvard Business Reviews*, May 2008.