Report on the 2nd Software Engineering Education Workshop (SEED 2019) co-located with APSEC 2019

Shailey Chawla
College of Science & Engineering
James Cook University
Singapore
shailey.chawla@jcu.edu.au

Bimlesh Wadhwa
School of Computing
National University of Singapore
bimlesh@nus.edu.sg

Pornsiri Muenchaisri
Department of Computer Engineering,
Chulalongkorn University
pornsiri.mu@chula.ac.th

Abstract

SEED 2019, The 2nd International workshop on Software Engineering Education (co-located with Asia-Pacific Software Engineering Conference – APSEC 2019, 2-4December, at Putrajaya, Malaysia with a theme 'Engineering Impactful Software for the Society towards IR4.0'), aims to bring together Software Engineering (SE) educators and practitioners into a dialogue to build a shared understanding of Software Engineering curriculum topics and specific issues in teaching and learning of Software Engineering with respect to the emerging topics of Artificial Intelligence, Cloud Computing, and Internet of Things (IoT) and corresponding Industry practices. SEED 2019 invited Position Papers (maximum 6 pages long) in the area of Software Engineering. The workshop accepted 5 papers and consisted of keynote talk as well as group discussion in addition to the position paper presentations.

1 Introduction

The second Software Engineering Education Workshop (SEED 2019) provided an opportunity for exploring new directions, presenting mature research, and discussing learning and teaching strategies and curriculum related issues on software engineering education.

Software development has changed a lot from basic information systems development to systems that involve Cognitive computing, Machine Learning, Blockchain, Augmented Reality, IoT, Cloud Computing and Robotics. The aim of this workshop is to bring together the practioners and educators to discuss how these novel technologies can be engineered a formal way backed by SE principles. Also, the aim is to keep the software engineering education in academia updated with the industry practices. The specific teaching and learning strategies that can be developed to equip the learners to handle industry projects are also targeted in this workshop. Software engineering discipline has taken various initiatives that have resulted the certification and licensing of software engineers in a number of countries worldwide. The workshop focus is coherent with the theme of APSEC 2019 "**Engineering Impactful Software for the Society towards IR4.0**" and will look for strategies to prepare the students by enhancing the curriculum and learning from the industry about the challenges faced in adopting new technologies.

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

The workshop call for papers was focused on following topics.

- Software Engineering Education towards IR 4.0
- Paradigms and models for incorporating novel technologies in SE
- Software Engineering education strategies
- Curriculum topics and design for software engineering
- Self-adaptive software development and teaching
- Development Tools and Environments
- Industrial case studies with lessons learnt or practical guidelines
- Empirical studies exploring project issues with new technologies
- Specific technologies practiced for engineering AI/ Blockchain/ IoT/Augmented Reality

1.1 Previous workshop

The SEED workshop started in 2015 [WGS15], where it was hosted in conjunction with International Software Engineering Conference. In the current run, it was collocated with APSEC 2019 held at Putrajaya, Malaysia.

2 Workshop Format

We prepared the workshop format such that it is interactive and fosters sharing of experiences in teaching software engineering. This was done by modelling our workshop format as a combination of paper presentation and open discussions after each paper. We also invited expert for a key note speech on software engineering education. The paper authors were guided to keep the presentations crisp and allow time for discussions and experience sharing.

The general structure of each talk was as follows:

- Keynote speech by invited guest (30 minutes)
- Paper presentation by the authors (15 minutes).
- Open discussions specific to paper context (5 minutes)
- General discussion on improving Software Engineering Education

2.1 Accepted papers

- Understanding the Decision-Making of Students in Requirements Engineering Course Projects
  Tianqing Liu, Peng Liang, Chen Yang, Zhuang Xiong, Chong Wang and Ruiyin Li
- Toward Industry Oriented Software Engineering Project Course: A Pilot Study
  Yu Liu, Ju Yan and Daqing Hou
- Teaching Object-Oriented Modeling as a Part of Programming Courses,
  Hidehiko Masuhara
- Collaborative Learning Strategies in Software Engineering Course
  Shailey Chawla
- Literature Reviews on Applying Artificial Intelligence/Machine Learning to Software Engineering Research Problems: Preliminary
  Pornsiri Muenchaisri

2.2 Acknowledgements

Success of this workshop is because of efforts of many people that we would like to acknowledge. First, we want to give thanks to the authors and presenters of the accepted papers. Furthermore, we want to express our gratitude to the APSEC 2019 organizers; they have been most supportive and encouraging. We would like to extend our gratitude to Associate Professor Dr. Shahida Sulaiman and the workshop chair Associate Professor Dr. Wan MN Wan-Kadir, for

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

their proactive support for our workshop. A special thanks to the audience of SEED 2019 workshop for active participation in the discussions and contributing to suggestions and solutions to problems faced by academics in teaching the subject and keeping it up to date. Finally, we are glad that the people served on the program committee and supported the workshop by soliciting papers and by writing peer reviews:

Dr. Benajamin Gan, SMU, Singapore

Dr. Sangeeta Srivastava, University of Delhi, India

Dr. Eng Lieh Ouh, SMU, Singapore

Dr. Rama Bhatia, PSB Academy, Singapore

Dr. Manjot Bhatia, GGSIPU, India

Mr. Bah Tee, James Cook University, Singapore

3 Discussions

3.1 Keynote speech on 'How we teach Software Engineering Practices at National Central University Taiwan'

Presented by Deron Liang, Professor and Director, Software Research Center. National Central University, Taiwan

Prof. Deron's keynote speech was both insightful and inspirational while he shared how the 'Software Engineering Practices' course has been designed and delivered in his university. The curriculum structure involved usage of lots of tools and expects proactive and intensive efforts from the students. Week 1 starts with introduction to debugging tools followed by system and unit testing in Week 2 using Jenkins and Korat providing them exposure to various test cases and how to select them. Students are then taught multithreading coding, debugging strategies and ways to treat tough bugs in Week 3. This is followed by revision control using Git, code review and issue tracking with peers using Gitlab in week 4. The curriculum concludes with a term project which is a group work and students usually bring their own projects.

The keynote speech instantiated a series of discussions involving the timeline of the curriculum and how the students can be so made actively involved in these projects. Introduction of a wide variety of tools for testing, debugging and version control were greatly appreciated.

3.2 Understanding the Decision-Making of Students in Requirements Engineering Course Projects

Presented by Ruiyin Li

Discussion question: What kind of project is carried out by the students? What are the typical deliverables? How long is the project or what percentage of the course is the project?

According to the presenter, the projects were of small scale that students could complete within the course duration. The expectation from the students was to provide a working prototype.

Discussion question: I think that RE is a very important but neglected part of our Software Engineering curriculum. What do others think? Do they cover RE as part of their course? What % of the SE curriculum in their university is spent on topic of Requirements.

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

It was largely found out that universities cover requirements engineering in their Software Engineering curriculum. Some universities offer separate courses on RE where mostly modeling and simulation are taught. Academics realize that though we pay less emphasis on requirements engineering in the curriculum, it is given a lot of time and importance in the industry.

3.3 Toward Industry Oriented Software Engineering Project Course: A Pilot Study

Presented by Daqing Hou

Discussion question: How were students trained to take on Artificial Intelligence or machine learning projects? Are there any pre-requisite courses they need to take to identify and carry out these projects?

These students had some knowledge on the existing projects as per the curriculum. There are no set prerequisites as such and the students are not bound to necessarily take artificial intelligence or machine learning projects.

Discussion question: What kind of frameworks are used for development? How much of technical support is provided to students?

The integration of industry mentoring in the semester has proven to be very effective in this research. The students work in teams and each team has students having different roles like designer, tester, implementers or coders. The suggestion is to rotate the roles. Technically, the students were provided some support by the academic staff and the industry mentor. The students should be supported with the technical skills. One important suggestion emerging out of discussion was that curriculum should be designed in view of the capstone project.

Discussion Question: Going forward, how do you see scaling it? Would you engage with more industry advisors? What if you have more students? We sometimes have 200-300 students in a software engineering class.

The key reason of the success of this model was close collaboration with industry advisor. The time and commitment required by industry advisor is intensive and with larger cohorts many such advisors will need to be associated with the course. This proposal is not scalable because in this case personalized attention to groups was given but it has been motivating and engaging for the students.

3.4 Teaching Object-Oriented Modeling as a Part of Programming Courses

Presented by Hidehiko Masuhara

Discussion Question: Do students learn and practice same programming language in Programming course I & II? If not, what are the 2 languages followed in programming courses. A repository of problems is used to provide a pool of programming projects to the students.

Different languages like Scala, Javascript and C are taught to the students at various levels of the course.

Discussion question: Could the choice of first programming language ease student learning of object oriented design concepts ?

First the basic programming constructs are important for the students. For the object oriented concepts, the students learn about modeling in UML, mostly class and object diagrams are used to understand interaction among different objects. Design patterns can be taught to students that can be very useful in their capstone projects.

Discussion question: Is University staying with OO Design in SE course or is there any initiative to change to say Python or Javascript?

Currently students are taught object oriented design in software engineering course. Javascript is taught already in the university as it lays foundation for many other technologies on web development and mobile computing.

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

3.5 Collaborative Learning Strategies in Software Engineering Course

Presented by Shailey Chawla

Discussion question: Was a pre-experiment perception survey done to get some comparisons with the post- survey data on attitude towards social engagement.

The experiment will involve pre and post surveys to capture the difference in attitude towards social engagement.

Discussion question: How big will be the control group and how will it be studied, to do a comparison between collaborative and non-collaborative learning?

The control group has 100 students. There are 10 tutorial sessions, when the study will be conducted. The studetns in different tutorial sessions will be divided into pairs, individually or in groups of 3 students. The students will be provided simple exercises at beginning of curriculum and complex exercises towards the end, to capture difference between complexity of tasks.

3.6 Literature Reviews on Applying Artificial Intelligence/Machine Learning to Software Engineering Research Problems: Preliminary

Presented by Pornsiri Muenchaisri

Discussion Question: How can artificial intelligence and machine learning contribute to or from software engineering field.

The discussion question instigated many responses from audience stating that relation can be both ways, AI/ML for solving Software engineering problems and Software Engneering for solving AI/ML problems in an organized manner. It was also suggested that take the whole SE processe to learning with propagation neural networks by improving all processes after learning and adjusting. Another insightful finding was that though in academics we stress upon incorporating newer methodologies to students, but in reality, they use the same old methods once they join industry. What he means is that although we try to improve SE curriculum with new knowledge and strategies, but they may not use the new methods in practices.

Discussion question: How to redesign the SE curriculum so that the student are ready for the industry.

Inclusion of AI and ML to enhance the SE process or using SE for AI and ML projects is the key concern. A related research discussed in the review focusses on requirements elicitation and modeling but not so much on validation and management. Under the scope of this study, more coding-related problems are studied and none is reported in testing.

4 Workshop Summary

The SEED workshop at APSEC 2019 received an overwhelming and active response from the audience. There were about 15 members in the audience attending the workshop. The workshop focus is on software engineering education to help bridge the gap between academia and industry and seeks participation from both academia and industry. At the end of the workshop there were open discussion questions as well on various topics like specific language or tools important in SE courses, motivation of students and initiatives for topics like SE for AI, Cybersecurity, IoT applications in universities. We intend to continue the workshop in future conferences and be at the forefront of research in software engineering education to prepare better IT professionals.

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

References

[BS14] Zhu Bin and Zhang Shiming, "Experiment teaching reform for software testing course based on CDIO," 2014 9th International Conference on Computer Science & Education, Vancouver, BC, 2014, pp. 488-491.

[Chr16] H. B. Christensen, "Teaching devops and cloud computing using a cognitive apprenticeship and story-telling approach," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. New York, NY, USA: ACM, 2016, pp. 174–179.

[CL19] Cico, Orges, and Letizia Jaccheri. "Industry trends in software engineering education: a systematic mapping study." In *Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings*, pp. 292-293. IEEE Press, 2019.

[CRAA18] Chakraborty, Partha, Rifat Shahriyar, Anindya Iqbal, and Amiangshu Bosu. "Understanding the software development practices of blockchain projects: A survey." In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 28. ACM, 2018.

[Dev17] Nitish M. Devadiga, Software Engineering Education : converging with the Startup Industry, 2017 IEEE CSEE&T, GA, 2017, pp. 192-196.

[GPBP18] Professional Standards in Engineering Education and Industry 4.0. In *2018 IEEE International Conference" Quality Management, Transport and Information Security, Information Technologies"(IT&QM&IS)*(pp. 638-642). IEEE.

[HERM08] Heidi J. C. Ellis, Ralph A. Morelli , Support for Educating Software Engineers Through Humanitarian Open Source Projects, Trinity College Hartford, CT / Gregory W. Hislop Drexel University Philadelphia, PA Hislop@drexel.edu ; CSEETW'08

[Jwo15] Jung-Sing Jwo ; Teaching Software Engineering Through Comics and Micro-Movie Design Projects , The 10th International Conference on Computer Science & Education (ICCSE 2015), July 22-24, 2015. Cambridge University, UK

[MMJCALC18] Mardis, Marcia A., Jinxuan Ma, Faye R. Jones, Chandrahasa R. Ambavarapu, Heather M. Kelleher, Laura I. Spears, and Charles R. McClure. "Assessing alignment between information technology educational opportunities, professional requirements, and industry demands." *Education and Information Technologies* 23, no. 4 (2018): 1547-1584.

[MRC17] M. Missiroli, D. Russo and P. Ciancarini, "Cooperative Thinking, or: Computational Thinking Meets Agile," 2017 CSEE&T, GA, 2017, pp. 187-191

[SACRHE19] Amershi, Saleema, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. "Software engineering for machine learning: a case study." In Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, pp. 291-300. IEEE Press, 2019.

[SCT17] K. Steppe, S. Chin and W. W. Tuck, "Improving the Teaching of Software Design with Automated Modelling of Syntactic Dependencies," 2017 IEEE CSEE&T, GA, 2017, pp. 144-151.

[VGTCF19] Garousi, Vahid, Gorkem Giray, Eray Tuzun, Cagatay Catal, and Michael Felderer. "Closing the gap between software engineering education and industrial needs." *IEEE Software* (2019).

[Wey17] Weyns, Danny. "Software engineering of self-adaptive systems: an organised tour and future challenges." *Chapter in Handbook of Software Engineering* (2017).

In: S. Chawla, B. Wadhwa, P. Muenchaisri (eds.): Proceedings of the 2nd Software Engineering Education Workshop (SEED 2019), Putrajaya, Malaysia, 02-Dec-2019, published at http://ceur-ws.org

[WGS15] Wadhwa, Bimlesh, Bharat Gera, and Ashish Sureka. "A report on software engineering education workshop (seed) colocated with ISEC 2015." In *Proceedings of the 8th India Software Engineering Conference*, pp. 206-207. ACM, 2015.

[XCFT17] Larrucea, Xabier, Annie Combelles, John Favaro, and Kunal Taneja. "Software engineering for the internet of things." *IEEE Software* 34, no. 1 (2017): 24-28.

[YLHC15] Jeong Yang, Young Lee, David Hicks, and Kai Chang, “Enhancing Object-Oriented Programming Education using Static and Dynamic Visualization,” 2015 IEEE Frontiers in Education (FIE): Launching a New Vision in Education Engineering, pp. 806-810, 2015.