
Ashish Sureka
IIIT Delhi, India
ashish@iiitd.ac.in

Y. Raghu Reddy
IIIT Hyderabad, India
raghu.reddy@iiit.ac.in

Masateru Tsunoda
Kindai University, Japan
tsunoda@info.kindai.ac.jp

Pornsiri Muenchaisri
Chulalongkorn University, Thailand
Pornsiri.Mu@chula.ac.th

ABSTRACT
The Software Engineering Education Workshop (SEEW) 2014 was held on 1st December 2014 at Jeju (South Korea). The workshop was co-located with the 21st Asia-Pacific Software Engineering Conference (APSEC 2014). The objective of SEEW is to create an annual discussion forum on Software Engineering (SE) Education in the Asia-Pacific region. The workshop was organized by experienced Software Engineering educators (working in Academia and Industry) from three different countries in the Asia-Pacific region: India, Japan, and Thailand. The total number of participants were 18 spread across 7 countries. The workshop consisted of 7 position paper presentations consisting of a wide range of topics and research questions. The paper presentation was followed by a group discussion which resulted in identifying important aspects and future research directions on Software Engineering Education. The workshop was a successful endeavor and the response in terms of the contributions by participants is a clear indicator and confirmation of the need of having a focused discussion forum for brainstorming on software engineering education in Asia-Pacific region.

1. MOTIVATION
The major objective of most Software Engineering Education conferences and workshops like the Conference on Software Engineering Education and Training (CSEET), International Workshop on Software Engineering Education based on Real-World Experiences (EduRex) co-located with the 34th International Conference on Software Engineering (ICSE) 2012, is to create a platform for discussing and addressing software engineering educational challenges in the various software engineering knowledge areas. The objective of Software Engineering Education Workshop (SEEW) is to create a platform where SE educators and researchers from the Asia Pacific region came together and discussed issues, challenges and opportunities in SE education.

2. WORKSHOP STRUCTURE
The workshop was group-discussion based on various topics of Software Engineering Education (including position paper presentations) and was not meant to be a mini-conference with only paper presentations. Groups were encouraged to plan and outline draft papers and furnish recommendations in their area. With the objective of creating an annual discussion forum on SE education in the Asia-Pacific region, the workshop on SE education is co-located with APSEC 2014 to be held in Jeju, Korea. Participants were selected based on the position statements. The selection was made to ensure a good mix of views, opinions and experiences rather than any quality measure of the statements. Selection was done by the workshop organizers and program committee. The workshop was restricted to a maximum 25 participants for effective discussion. The workshop program consisted of the workshop introduction by the workshop organizers followed by seven position paper presentations. Table shows the list of papers presented at the workshop. In the end, the workshop program consisted of a group discussion. The total number of participants were 18 across 7 countries.

3. POSITION PAPER ABSTRACTS
3.1 A Proposal for UI- and REST-Oriented Web Service Design for E-learning
Although UI (user interface) is important in e-learning indeed, it is difficult to design an appropriate UI on LMSs (Learning Management System) for their users. This is because the appropriateness of UI depends on its users. For example, a software tool that has a UI of many functionalities sometimes confuses beginners. Instead, a small and/or too-simplified UI tends to be too simple for many classes of users. Hence, flexible UI design is necessary in order to cope with each combination of different classes of users and software tools. It is also difficult to develop each LMS plugin without heavily depending on the LMS, which means that if a LMS becomes obsolete, plugins/extensions implemented for the LMS are required to be rewritten to be used continuously.

This paper proposes a new UI-oriented Web service design for software engineering education on LMSs (learning management systems). To improve the quality of education on LMSs, our design consists of two layers: a UI (user interface) layer and a backend layer, both are connected by using REST (Representational State Transfer). The UI layer uses some of the functionalities provided by the backend layer, and the backend layer provides all functionalities required. As a concrete example, this paper introduces our current design and implementation of a pair programming environment on LMSs based on our proposed design.

3.2 A Case Study: How to Find and Reify a Research Theme on Project Based Learning for Master’s Course Education
The article introduces a case study of a masters course education for embedded software engineering. Finding out a research theme is most important work on masters course, however, it is not easy.
The first author, who is a master’s course student, has developed an educational tool of automatic vacuum cleaner robot for national masters course education project enPtT-EMB/PEARL. Additionally, he has developed a robot by using his educational tool. These experiences have led to a research theme of dynamic rewriting programming at runtime-based on Context Oriented Programming.

The article introduced a case-study how to find a research theme from the experience of PBL. In this case-study, we mentioned that two software developments contributed to finding a research theme. One is the development of an educational tool. Another is a robot development for ESS Robot Challenge. The first author had developed some other embedded systems on different PBL. Each size is almost 10KL. These experiences caused obtaining object-oriented design and programming capability. Through this case study and other experiences of various educations, we have recognized following points, for leading such education that masters course students find useful and novel research theme.

1. Keep single academic word in mind for long term.
2. Try and fail.
3. Discuss with many students in different fields.

To evolve our education, we are planning to encourage students community to collaborate with students between universities of different specialties.

3.3 Two Sides of the Same Coin (Software Engineering)

Software Engineering has two views: theory and application. On one side, software Engineering researchers continuously propose new approaches, theories, methodologies, and tools as a result of study and investigation in research problems. Many of research results can be used in real software development and some of them may be put on the shelf. On the other side, software engineers successfully apply software engineering processes, methods, and tools to develop software in software firms. One side needs strong knowledge in theory, while the other needs practical methods. As software engineering educators, we have to design courses and teach students for two different purposes. This proposal is summarized how we manage the two sides of the same coin (software engineering). This proposal presents the approach that is used to manage the software engineering program for both research-oriented and application-oriented at the Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand which has been offered since 2002.

With the demand in software engineer from public and private sectors, personnel with software engineering knowledge should be increasingly produced. The Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand has offered Master of Science in Software Engineering with two program plans (Plan A and Plan B) and two groups of students (Regular and Part-time) since 2002. The approach to manage the different two study plans is summarized. Plan A graduates are research-oriented while Plan B graduates are application-oriented. Some key performance indicators are presented. The strength of the program is that students have opportunity to learn and research with more than forty faculty members of the department who have expertise in various areas. In addition, software engineering faculty members have an average of 15 years of research and teaching experience. In the future, we plan to improve the approach by adopting Education 4.0 (proposed by the university), modifying the program to accommodate international students, and providing more opportunity to work with famous business and software tools.

3.4 How to Classify Personalities of Team Members on Project-Based Learning

Recently, PBL (Project-Based Learning) is sometimes applied to software engineering education. When PBL is applied, software development teams are made. For the teams, influence of the personality of each member is not negligible. The personality will affect performance of the team and it may also influence the effect of education. So, we focus on the personality and discuss how to measure and classify it. In software development companies, most software is developed by a team which consists of a project manager, system engineers, programmers, and so on. To develop software, they perform their work and communicate with each other. Hence, human side of the members is considered to affect activities of software development. Members may have adequacy of a role. For example, a person would be adequate for programmer, although he/she is not adequate for project manager. That could affect the efficiency of the development, if their role is inadequate for them. Also, if the distribution of characteristics of team members is biased, that may decrease the efficiency of communication between them. For instance, if all team members have introverted

---

<table>
<thead>
<tr>
<th>Table 1: List of Papers (Title and Authors) Presented at the Workshop</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
</tr>
<tr>
<td><strong>2</strong></td>
</tr>
<tr>
<td><strong>3</strong></td>
</tr>
<tr>
<td><strong>4</strong></td>
</tr>
<tr>
<td><strong>5</strong></td>
</tr>
<tr>
<td><strong>6</strong></td>
</tr>
<tr>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>
personality, it would affect communication between them, and it may decrease the efficiency of software development. Previous research shows the utility of roles and personality characteristics to the evaluation and formation of software development teams.

We think that likewise, on PBL of software engineering education, the influence of the human side of each member is not negligible for the teams. The personality would affect performance of the team and it may also influence the effect of education. So, we focus on the human side of software development team members, especially personality of them. First, we surveyed researches which addressed personality of members in a software development team. We focus on not only a software developer but also a software development team. So, we mainly picked up researches which treated relationships between personality of members and the performance of a team. Next, based on the existing researches, we discussed the drawbacks of them, and what is needed to analyze the relationships between personality of members [2][6][7].

3.5 Process Mining Software Repositories in an Educational Setting

A graduate or an undergraduate level Software Engineering course generally consists of a team-based semester long project and emphasizes on both technical and managerial skills. Software Engineering is a practice-oriented and applied discipline and hence there is an emphasis on hands-on development, process, usage of tools in addition to theory and basic concepts. Course Instructor can easily assess and provide feedback on product and deliverables produced. Providing feedback on process and team-work is not straightforward. In this position paper, I present an approach and a broad framework for mining the process data (process mining) from software repositories archiving data generated as a result of constructing software by student teams in an educational setting. I present an application of mining several software repositories such as team wiki (used during requirement engineering), version control system (development and maintenance), and issue tracking system (corrective and adaptive maintenance), peer code review system and mailing lists in the context of a graduate or an undergraduate Software Engineering course. I propose and present effective visualizations, metrics and algorithms to provide an insight into practices and procedures followed during various phases of a software development life-cycle. The visualizations and metrics can be classified as learning analytics and provide a multi-faceted view to the instructor serving as a feedback tool on development process and quality by students. I will talk about how the event logs produced by software repositories can be used to derive useful insights and actionable information for the course instructor. The insights are: degree of individual contributions in a team, quality of commit messages, intensity and consistency of commit activities, bug fixing process trend and quality, component and developer entropy, process compliance and verification. I present the result of an empirical analysis on a software repository dataset consisting of 19 teams of 5 members each and discuss challenges, limitations and recommendations [4][5][6][8].

3.6 My Experiences in Teaching Software Engineering Related Courses

I have been teaching various software engineering related courses at several universities in China, UK, and Japan over the last 30 years, including Principles of Compiler, Introduction to Software Engineering, Software Specification Techniques, Program Design, Formal Semantics of Programming Languages, Software Verification and Validation, and Java Programming Language. In this presentation, I will talk about my various experiences and techniques in teaching those courses to different kinds of students.

3.7 Introducing Software Product Engineering in Undergraduate Computer Science Curriculum

Traditional engineering disciplines like electrical engineering, mechanical engineering, metallurgical engineering are considered as disciplines and tend to focus on the design of artifacts. The current Computer Science Engineering curriculum in India looks at Software Engineering as a course rather than a discipline. The focus the course is on teaching software processes rather than software product design which is imperative if it’s to be considered as an engineering discipline. This paper argues the need for introducing software product engineering as an additional subject in computer science and engineering undergraduate curriculum. An undergraduate computer science and engineering program should consist of a two-course software engineering sequence: (1) software process and project management and (2) Software product engineering. Software Process and project management focuses on software product delivery with process conformance which is more inline with Industrial Engineering and Management. The Software product engineering should focus on teaching design fundamentals, various parts of design (UI design, Logical design, protocol design, algorithmic design, database design, etc.), design qualities, and the various non-functional attributes/constraints of design.

4. PROGRAM COMMITTEE

1. Ashish Sureka (Department of Computer Science, IIIT-Delhi, India)
2. Y. Raghur Reddy (Software Engineering Research Center, IIIT-Hyderabad, India)
3. Hoh Peter In (Department of Computer Science and Engineering, Korea University, Korea)
4. Masateru Tsunoda (Department of Informatics, Kindai University, Osaka, Japan)
5. Pornsiri Muenchaisri (Department of Computer Engineering, Chulalongkorn University, Thailand)
6. Rakesh Shukla (Infosys Labs, India)
7. Rahul Purandare (Department of Computer Science, IIIT-Delhi, India)
8. Masataka Nagura (Hitachi Limited, Japan)
9. Koji Toda (Fukuoka Institute of Technology, Japan)

5. GROUP DISCUSSIONS

The workshop was concluded by a small group discussion on current significant challenges for Software Engineering Educators. In addition to the discussion on position papers, the participants discussed new best practiced for Software Engineering Education and Training as well as innovative curriculum or course formats. The participants had an extensive experience in Open-Source Software (OSS) development and emphasized the need to teach students to participate in OSS projects. Participants shared their experiences and views on involving students in OSS projects. Involvement in OSS projects will provide real-world experience to students and will also provide an understanding of the issues found in large, complex software projects. Cooperation in education between industry and academia was also another point of discussion. The participants proposed that Industry-Academia collaboration is needed to close the gap between industry software engineering needs and academic software engineering education.
6. ORGANIZERS

6.1 Ashish Sureka
Ashish Sureka is an Adjunct Professor at Indraprastha Institute of Information Technology, Delhi (IIIT-D). His current research interests are in the area of Mining Software Repositories, Software Analytics, and Social Media Analytics. He graduated with an MS and PhD degree in Computer Science from North Carolina State University (NCSU) in May 2002 and May 2005 respectively. He has worked at IBM Research Labs in USA, Siemens Research Lab (India) and was a Senior Research Associate at the RD Unit of Infosys Technologies Limited before joining IIIT-D in July 2009. He has received research grants from Department of Information Technology (DIT, Government of India), Confederation of Indian Industry (CII) and Department of Science and Technology (DST, Government of India). He has published several research papers in international conferences and journals, graduated several PhD and MTech students, organized workshops co-located with conferences, and received best paper awards. He was selected for ACM India Eminent Speaker Program. He holds seven granted US patents.

6.2 Y. Raghu Reddy
Y. Raghu Reddy works as an Associate Professor at International Institute of Information Technology, Hyderabad. His Primary research interests are Software Engineering and Usability Engineering. He has published a number of peer-reviewed publications in these areas. He has authored a book titled Aspect Oriented Software Development: An Approach to Composing UML Design Models. He is a part of several international conference committees and editorial boards. He has worked on projects with Government of India as well as companies like CA Technologies, QWest Telecom, etc. He is currently the Head of the Software Engineering Research Center at IIIT-Hyderabad and the CA technologies Endowed Faculty Chair at IIIT-Hyderabad. Previously, he was also the IIIT-Head for CA Technologies-IIIT Hyderabad innovation Center established at CA Technologies, Hyderabad. He is an ACM India Eminent Speaker.

6.3 Masateru Tsunoda
Masateru Tsunoda is a Lecturer at Department of Informatics, Faculty of Science and Engineering, Kindai University. Before that he was an Assistant Professor at Faculty of Information Sciences and Arts, Toyo University. He obtained his DE degree from Information Science, Nara Institute of Science and Technology. He has published several research papers in reputed Software Engineering conferences and journals.

6.4 Pornsiri Muenchaisri
Pornsiri Muenchaisri is an Associate Professor at the Software Engineering Lab, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Thailand. Her research interests are in the area of Software engineering, Object-oriented analysis, design, and programming, Software metrics, Software improvement and Software evolution. She received her Ph.D. in Computer science from Oregon State University, U.S.A., 1998 (with a scholarship from Ministry of Science and Technology).

7. CONCLUSION
In this report, we present a summary and a few reflections on the SEEW 2014 (Software Engineering Education Workshop). The main objectives of the workshop were to bring together educators, researchers and practitioners from academia and industry to discuss and advance current the state of SEE (Software Engineering Education) and to facilitate and promote group discussions on various topics of SEE. The total number of participants were 18 across 7 countries. The workshop consisted of 7 position paper presentations consisting of a wide range of topics and research questions. The paper presentation was followed by a group discussion which resulted in identifying important aspects and future research directions on Software Engineering Education. The response in terms of the contributions by attendees is a clear indication of the need of having a focused discussion forum for brainstorming on the topics of SEE in Asia-Pacific region. The participants believe that while ICSE (International Conference on Software Engineering) which is the flagship conference of Software Engineering provides a discussion forum to discuss Software Engineering Education, a regular track (annual event) or workshop on Software Engineering Education is recommend with APSEC.

8. ACKNOWLEDGEMENTS
The workshop organizers and authors of this report would like to extend their thanks to the individuals that presented at the workshop and all the workshop participants for their contribution in the discussions. A special thanks to the APSEC 2014 workshop chair Prof. Keijiro Araki (Kyushu University, Japan) and Prof. Seok-won Lee (Ajou University, Korea) and the APEC 2014 General Chair Prof. Sungdeok (Steve) Cha (Korea University, Korea) for providing all the support needed to make the event a success.

9. REFERENCES