# Bridging The Gap Between ABET Outcomes And Industry Expectations - A Case Study On Software Engineering Course


Saradhi Seshagiri
Technical Learning Services,
Tech Mahindra,
Hyderabad, India
E-mail: Saradhi.Seshagiri@TechMahindra.com

LNS Prakash Goteti
Technical Learning Services,
Tech Mahindra,
Hyderabad, India
E-mail: prakashgoteti@gmail.com



***Abstract* – The role of ABET accreditation system is quite significant in providing guidance towards program and course design. The program outcomes encompassing through knowledge, skill and attitude play an important role towards competency development of the students. To improve the employability quotient of these fresh graduates, the alignment of respective outcomes along with the base lined expectations of IT industry is needed. In this context Software Engineering (SE) plays an important role in the transformation journey of the graduates to become entry level developers. It also helps to bridge the gap between academia and IT industry so that these new hires are productive as soon as possible. In this paper, the expectations of an IT industry from the new hires is described in conjunction with ABET educational outcomes highlighting pedagogical aspects that enable students develop these abilities. SE course is used as vehicle and an approach is presented integrating software code of ethics (recommended by ACM – IEEE), SE principles, pedagogical aspects and assessment instruments. Subsequently experiences from our corporate learning environment are highlighted.**



***Keywords-Engineering Education; Outcome based education, Pedagogy, Bloom's taxonomy***


## I. Introduction

Software Engineering (SE) methodologies, software construction techniques and tools are critical elements for innovation and developing applications across various sectors like automobile, telecom, digital development, defense and the software industry itself. The current evolution of IT and ITeS is providing enormous opportunities to the new hires and as well as experienced developers. At the same time, the size and complexity of software is constantly increasing with rapidly changing business needs of the customers. To meet these challenges and the demands of delivering reliable and high quality software swiftly, it is important to further strengthen the engineering programs focusing on holistic approaches towards engineering education in general and software engineering education in particular.

Significant efforts are being made towards outcome based approaches in software engineering education [1]. It is important that students possess right competencies and qualities. Also the code of ethical business conduct is quite critical for smooth execution of the projects so that they become brand ambassadors of the organization. Hence imparting aspects of code of ethics [2, 3] along with necessary skills, knowledge enable the graduates improve their personality and competence. Group assignments, co – operative learning and problem based learning plays an important role to highlight individual accountability. Teams work helps in accomplishing a common goal with positive independence and group accountability [4]. Also social platforms like wiki can facilitate knowledge construction of the students [5]. The group activities can be proposed close to real life scenarios to address cognitive and social aspects of SE [6].

In India, there is significant interest among graduate students towards IT and ITeS as plenty of jobs are available. However, the major concern is being the fact that only 15% of total engineering graduates are employable [7]. This gap will provide us an opportunity to explore assessment and teaching methods with holistic approach. Also SE can be used as a vehicle to address some of the knowledge deficiencies among the students by assessing the gaps between industry expectations [8] and the abilities of these graduates.

The present paper focuses on academic and practitioners perspective and proposes an integrated approach encompassing through curriculum, pedagogy and assessments aspects along with human and ethical issues considering software engineering curriculum in mind. The reminder of the paper is organized as follows: In Section II, pedagogical and evaluation aspects are discussed. Subsequently, Section III presents industry expectations and ABET outcomes specific to engineering education, followed by section IV on holistic approaches towards SE education and section V presents few observations from our corporate learning environment.

978-1-4799-6876-3/14/$31.00 ©2014 IEEE

## II. Pedagogical Aspects

In pedagogical view point, the knowledge areas are primarily classified into factual knowledge (K1), conceptual knowledge (K2), procedural knowledge (K3) and meta–cognition (K4). The classification of knowledge areas help in identifying appropriate instruction methods that promote effective learning. A selection based approach can help instructors employ suitable learning instrument and facilitate the activity of learning. Also to promote peer learning, it can be supplemented with interesting exercises, puzzles, problems, and reading assignments that draw attention of the students. This will help them to develop problem solving, self-learning skills so that they apply these on the job. In order to enable the students achieve the outcomes curriculum design approaches [9], exploring appropriate teaching and assessment methods [10] and active engagement pedagogies play an important role [11]. The case study scenarios can be used as vehicle across various chapters to complete the steps in the development process [12]. Fig. 1 below illustrates the concept types and appropriate active engagement instruments associated with them.

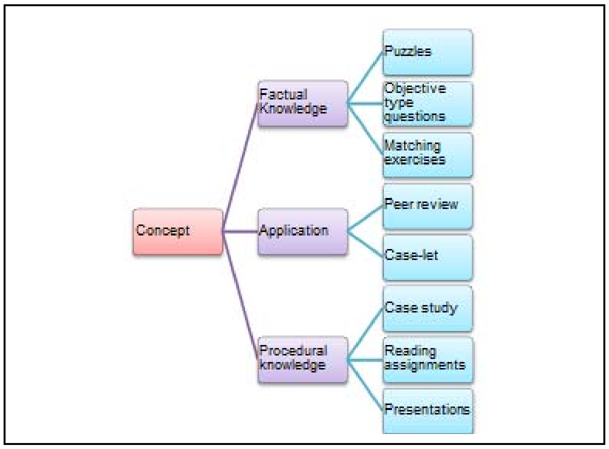


Figure 1. Framework for leveraging SE practices in learning process.

To facilitate the learning of factual knowledge areas, the participants can be given reading assignments that can be discussed in the class posing questions and deliberations. Short objective type quizzes and puzzles can be used as assessment instruments. The conceptual knowledge helps the students develop ability to apply the concept in a real life scenario, especially on the job. In this aspect, assessments can be created through enquiry based learning to promote higher order cognition [13] among the students. Similarly the procedure knowledge areas are to be covered using context based learning –preferably case study approach and problem based learning. These scenarios can be added as part of exercises section so that the students apply the concepts that they learnt to the given problem, analyze the intricacies, and construct the solution as combined solution of sub problems. The process can be assessed by using appropriate assessment instruments aligned with revised bloom's taxonomy [14]. The procedural knowledge provides an opportunity to the participants develop critical thinking skills (viz. analysis, synthesis, evaluation and creation). Additional resources [15] can also be used for cooperative learning. These can help the faculty to design the course along the breadth and depth connected with course level outcomes.

## III. Industry Expecatations And ABET

The majority of the new hires will be fulfilling the role of entry level developers. These employees perform various tasks like maintaining version controlling for the deliverables, performing unit testing on the code developed, communicating with the customers through conference call and email, learn new frameworks and tools swiftly, enhancing their knowledge by continuously upskilling themselves etc. In this aspect it is quite important that the students acquire 'learn to learn' skills as part of their engineering education. In conjunction with ABET framework [16], and the expectations from the company, the base lined competencies expected from these new hires are listed below in Table I.

TABLE I. Alignment of ABET and Industry Expectations

| ABET 3(a) -3(k) | Industry Expectations Competency Mapping |
|---|---|
| a. An ability to apply knowledge of mathematics, science, and engineering. | Problem solving skills. |
| b. An ability to design and conduct experiments, as well as to analyze and interpret data. | Ability to generate bug report. |
| c. An ability to design a system, component, or process to meet desired needs. | Ability to implement given design. Ability to build components for business layer. |
| d. An ability to function on multi-disciplinary teams. | Ability to work as team player. |
| e. An ability to identify, formulate, and solve engineering problems. | Ability to capture customer requirements. |
| f. An understanding of professional and ethical responsibility. | Demonstrate good corporate citizenship and to be brand ambassador to the organization. |
| g. An ability to communicate effectively. | Email etiquette. |
| h. The broad education necessary to understand the impact of engineering solutions in a global and societal context. | Ability to identify and contribute towards green initiatives and eco-friendlyness through corporate social responsibility initiatives. |
| i. A recognition of the need for, and an ability to engage in life-long learning. | Ability to learn a new technology or framework –based on business needs. |
| j. A knowledge of contemporary issues. | Awareness of best practices with –in and outside the organization. |
| k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. | Ability to use appropriate framework relevant to project. |

## IV. Holistic Approach TO Software Engineering Education

Developing a course to achieve the expected outcomes, involves curriculum, pedagogy or instruction method that helps the students achieve these outcomes through the course and alignment of assessment instruments with Bloom's taxonomy [17], as illustrated in Fig 2.

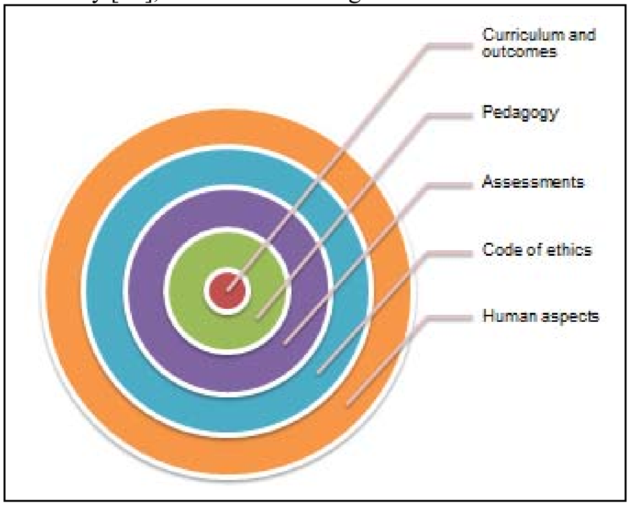


Figure 2. Framework for leveraging SE practices in learning process.

*A. Curriculum and key knowledge areas:*

Software engineering course curriculum is quite exhaustive and instructors often might find it difficult to cover the breadth and depth of the knowledge areas [9]. It would be very ambitious to address all 3(a) -3(k) in one course and hence we felt 3(b) -3(g) and 3(k) are more natural outcomes that can be addressed through course learning outcomes. The details on writing course level outcomes are self-explanatory and hence omitted here. Sectional analysis covering relevant concepts and knowledge areas which are quite important in practitioner's point of view are highlighted in Table II.

*B. Assessment instruments:*

- Reading assignments, reflective approaches help in developing cognitive aspects among students.
- Group projects focusing on 'real' software systems with open ended requirements help the students develop higher order thinking skills.
- We can inculcate collective code ownership and the faculty can introduce change requirements during the development of the project.
- First two weeks, during the semester, the students can simultaneously learn open source tools and frameworks (like Eclipse IDE and Junit) on their own as part of group engagement.

TABLE II. OUTCOMES AND SECTIONAL ANALYSIS ON SE

| Some of the ABET Outcomes that can be addressed by SE | Sectional Analysis from SE |
|---|---|
| b) An ability to design and conduct experiments, as well as to analyze and interpret data. | Software testing concepts(K1), System failure aspects(K1), bug life cycle (K2) |
| c) An ability to design a system, component, or process to meet desired needs | System modeling (K3), Evaluation of models(K1), UML design (K2), database design (K2), software reuse (K2); Coding standards (K2) |
| d) An ability to function on multi-disciplinary teams | Distributed SE (K1), configuration management (K1), software project management, concepts (K2) |
| e) An ability to identify, formulate, and solve engineering problems | Requirement elicitation (K1); traceability metrix (K2); impact analysis (K2); code refactoring (K2); Risk management concepts (K2), |
| f) An understanding of professional and ethical responsibility | Importance of SE (K1), Code of Professional software development (K2) |
| g) An ability to communicate effectively | Distrubuted environment concepts (K1) |
| k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. | Software Processes (K1); methdlologies (K2); Agile software development (K2), test case design techniques (K2); Unit testing frameworks (K2) |

- Though, enormous resources for additional reading and exploration are available within the course text, for example [17], it is important to create the need among the students to read and analyze them. This will not only improve their reading and writing abilities but promote their overall communication skills. Subsequently enable the students to become responsible for their own knowledge construction.
- Peer activities, for example, identifying some of the software engineering practices adopted in manufacturing of the space craft [18] where the instructor can evaluate them through presentations and group discussions.

*C. Code of Ethics and Human aspects :*

SE provides a context to inculcate ethical aspects in computing [19] as it provides approaches to develop the systems and evaluate the artifacts. We consider these ethical issues and associated indicators relevant from SE. These are highlighted in the Table III below. The table shows the mapping of appropriate ethical issues with SE principle, through the shaded areas, which will help the fresher's to gain the appropriate knowledge before joining any organization.

TABLE III. ETHICAL ISSUES AND SE PRINCIPLES

| SE Principles | Ethical Issues | | | | | | |
|---|---|---|---|---|---|---|---|
| | a | b | c | d | e | f | g |
| Phased life-cycle plan | ■ | | | | | | |
| Continuous validation | | | ■ | | | | |
| Disciplined process control | ■ | | | | | | |
| Clear accountability for results | | | | | ■ | | |
| Use better and competent people | | | | | ■ | | ■ |
| Commitment to improve the process | | ■ | | | | ■ | |
| Team work | ■ | | | | | | ■ |

a) Quality of life b) Use of power, c) risks and reliability d) property rights e) Privacy f) equity and access g) honest and deception

Specific to human aspects, we refer individual and team level contribution and performance. Collective code ownership plays an important role while developing the software. To inculcate these, instructor can evaluate based on individual's contribution on teams work and decision making in addition to quality of the deliverables.

## V. Our Experiences in Training Fresh Graduates

The majority of the new hires will be fulfilling the role of entry level developers. The real life case studies from projects enable them to work in teams and also come out with the collaborative efforts towards solving the scopes. We provide scenarios to each pair, where two participants will take collective code ownership for the scope given. Each pair will be evaluated based on the deliverables. However, they will be discussing technical aspects in the class through peer –learning session. This will promote self-learning and problem solving skills. Also the participants will be evaluated as a pair. However, the participants prepare individual work breakdown structure. This approach of team work has also shown its impact on the individual performance.

We have considered 312 participants trained on Java platform across six batches during one year; compared individual batch performance. The duration of each batch is around three months. The software developed by the participants is evaluated by developers working in projects. The broad level outcomes are:

- Ability to understand the scope and develop use case modeling.
- Ability to develop reusable components using design patterns and layered approach.
- Ability to perform unit testing.
- Ability to develop web application using Java Database connectivity, servlets and JSP technologies.

Following are our observations through this approach:

- Participants tend to focus on why and how to apply a skill or concept, not just remembering and recalling facts.
- Collective code ownership and discipline to adhere to deadlines.
- Participants discuss in the class some of the areas of ambiguity and make assumptions towards decision making.
- Participants reflect upon their own learning process.
- Some of the participants are not proactive in proposing, justifying a solution, evaluating alternatives; Also we find few cases where communication skills to be improved so that the developers express their views. Such individual behavior impacts the individual performance and thus affecting the overall performance of the batch, as evident by batches 1 and 5 in Fig. 3

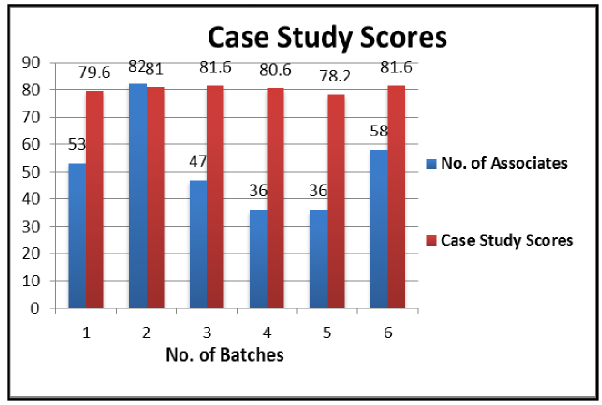


Figure 3. Case study result comparision over a span of year for 6 batches.

- We captured couple of experiences from the participants which are as below:
  "*I liked the case studies very much in the entire program. It helps a lot to improve the practical knowledge and handling errors. Doing the case study, it gave a handful of experience.*"
  "*Case study helped me know how to develop an application in the real time scenario.*"

## Conclusions

In this paper we present base-lined expectations of our company from fresh engineering graduates and discussed in conjunction with ABET criteria. The gaps are highlighted in terms of hard aspects (curriculum outcomes, pedagogy, and assessments) and soft aspects (code of ethics and human aspects). Systematic approach is described towards course design to promote holistic learning among the student community. The pedagogical aspects towards active engagement are presented based on observations from our learning environment. Software engineering course is considered as vehicle to illustrate the approach. Subsequently, some of our experiences from entry level program are highlighted with relevance to this work. We feel such approaches help towards identifying and bridging the gap between abilities of the graduates and expectations from the industry. The approach and the observations described in the work might help academic community to equip the students to become employable.

## Acknowledgment


Authors are grateful to Mr. Murthy AS, CTO, Tech Mahindra for the financial support. Authors are also thankful to Mr. Nimish Gupta, IBU Head, TLS for several useful discussions.

## References


[1] Michael J. Lutz, J, Fernando Naveda and James R,. Vallino, "Under graduate Software engineering", Comm of ACM, 75, 2014, pp. 52 -58; John H. Benamati, Zafer D. Ozemir, and H. Jeff Smith. "Aligning Undergraduate IS Curricula with Industry needs", *Communications of ACM*, 53(Mar. 2010), pp. 152–156. DOI= http://doi.acm.org/ 10.1145/1666420.1666458; ; Samuel Greengard. ACM and India 2010. *Communications of ACM*, 53, 1(Jan. 2010), 31. DOI= http://doi.acm.org/10.1145/1629175.1645395

[2] Chuck Huff, C. Dianne Martin, "Computing consequences: A Framework for Teaching Ethical Computing" Comm of ACM, 12, 1995, pp. 75 -84

[3] Kenneth C. Launden, "Ethical Concepts and Information Technology, Comm. Of ACM, 38, 1995, pp. 33 -39

[4] Karl A. Smith, Sheri D. Sheppard, David W. Johnson, and Roger T. Johnson, "Pedagogies of Engagement: Classroom – Based Practices", J. of Engg Education, 2005, pp. 1- 15.

[5] Marija Cubric, "Wiki –based Process Framework for Blended Learning", Proceedings of ACM conference on WikiSym, Oct. 21 -23, Montreal, Canada, 2007, pp 11 -22.

[6] Orit Hazzan, "Cognitive and Social Aspects of Software Engineering: A Course Framework", ACM conference on ITiSE, 30th June -2nd July, Thessaloniki, Greece, 2003

[7] Kirti Garg, Vasudeva Varma, "People Issues Relating to Software Engineering Education and Training in India*"* ( ISEC, 08 February 19-22, 2008, Hyderabad, India)

[8] Alex Radermacher and Gursimran Walia, "Gaps between Industry expectations and the Abilities of the Graduates", ACM conference on SIGSCE -13, 6th -9th March, 2013, Denver, Colorado, USA.

[9] Allen B Tucker, "New directions in the introductory computer science curriculum", proceedings of SIGSCE, pp. 11 -15, Phoenix, Arizona, USA, 1994.

[10] CH Ushakiran, Prakash Goteti, "Imparting Technical Skills through Outcome Based Curriculum and Case Study Assessments" Innovation and Technology in Education, MITE 2013, 20th -22 December, Jaipur, India2013, pp 321 -324 doi:10.1109/MITE.2013.6756358;

[11] Karl A. Smith, Sheri D. Sheppard, David W. Johnson, and Roger T. Johnson, "Pedagogies of Engagement: Classroom – Based Practices", J. Engineering Education, 94, 2005, pp. 87-101.

[12] Evelyn Stiller, Cathie LeBlanc, "Effective Software Engineering Pedagogy" , J. of Computing sciences in colleges, 17, pp. 124-134, 2002

[13] GV Madhuri, Kantamreddy, and Prakash Goteti, "Promoting Higher order thinking Skills using Problem Based learning" European J. of Engineering Education, 37, pp. 117 -123, 2012.

[14] L. W. Anderson, D.R. Krathwohl, P.W. Airasian, K..A. Cruikshank, et al. "A taxonomy for learning and teaching and assessing: A revision of Bloom's taxonomy of educational objectives", Longman, Inc: AW, 2001.

[15] Available at: http://vlssit.iitkgp.ernet.in/isad/isad/1/references/

[16] Richard M. Felder and Rebecca Brent, 2004. "The ABC's of Engineering Education: ABET, Bloom's Taxonomy", Cooperative learning, and so on, In Proceedings of the 2004 American Society for Engineering Education Annual Conference and Exposition, session 1375, 2004.

[17] Ian Sommerville, "Software Engineering", 9th Edn, AW, 2010

[18] (Accessed on October, 12th, 2014); Available at: http://en.wikipedia.org/wiki/Mars_Orbiter_Mission

[19] Chuck Huff and Almut Furchert, "Computing Ethics: Towards a Pedagogy of Ethical Practice",Comm. Of ACM, 57, 2014, pp. 25 -27; Also, Chuck Huff and C. Dainne Martin, "Computing Consequences: A framework for Teaching Ethical Computing", Comm. Of ACM, 38, 1995, pp. 75 -84.