Case Studies for Enhancing Student Engagement and Active Learning in Software V&V Education

Priyadarshan A. Manohar[1], Sushil Acharya[1], Peter Wu[2], Mary Hansen[3], Ali Ansari[4] & Walter Schilling[5]

[1] School of Engineering, Mathematics and Science (SEMS), Robert Morris University, Pittsburgh, Pennsylvania, USA

[2] School of Communication and Information Systems (SCIS), Robert Morris University, Pittsburgh, Pennsylvania, USA

[3] School of Education and Social Sciences (SESS), Robert Morris University, Pittsburgh, Pennsylvania, USA

[4] Computer Engineering Department, Virginia State University, Petersburg, USA

[5] Software Engineering Department, Milwaukee School of Engineering, Milwaukee, Wisconsin, USA

Correspondence: Priyadarshan A. Manohar, School of Engineering, Mathematics and Science (SEMS), Robert Morris University, 6001 University Boulevard, Moon Township, Pennsylvania, PA-15108, USA. Tel: 1-412-397-4027. E-mail: manohar@rmu.edu

Abstract

Two critical problems facing the software (S/W) industry today are the lack of appreciation of the full benefits that can be derived from Software Verification and Validation (V&V) and an associated problem of shortage of adequately trained V&V practitioners. To address this situation, the software V&V course curriculum at the author's institution is being improved via a National Science Foundation (NSF)-funded project. The basic objectives of this project are to enhance the quality of software education via increased student engagement and by bridging the gap between the basic principles discussed in the classroom and the complexity of real world problems. The teaching method utilized promotes higher levels of student engagement and learning through interactive, hands-on exercises, case studies and discussions. In addition, the instructional materials were purposefully designed not only for university classroom settings, but to also be deployed for on-the-job professional training in S/W industry settings, thereby helping to increase the pool of professionals with contemporary V&V knowledge and skills.

The new course curriculum enhancement described in this paper is guided by academic research and industry best practices that focus on four specific V&V focus areas: *requirements engineering*, *reviews*, *configuration management*, and *testing*. Among many educational tools that are being developed to achieve the project objectives, the work related specifically to the development of one central component, case studies, is described here. Historically, case studies have been educational tools utilized in business, law, and medicine, but are not as prominent in software engineering. The hypothesis is that case studies would be effective educational tools to introduce real-world professional practices into the classroom, which would help the students in both identifying and solving problems, and developing a perspective on applying knowledge. In this paper we describe a set of V&V related case-studies that we have drawn from industry experiences and developed as pedagogical tools. These case-studies cover several important topics in the S/W V&V domain such as software testing, legal issues in software, software consumer protection, and requirements from the customers' perspectives. We also report on the results of initial implementation of the case studies related to software testing in the classroom and show their effectiveness both in terms of satisfying Accreditation Board for Engineering and Technology (ABET) outcomes assessment and student satisfaction.

Keywords: software verification & validation, case studies, requirements engineering, reviews, configuration management, testing

1. Introduction-Case Study Based Education

Engineering education must strike a balance between the knowledge of theoretical concepts and the ability to apply the theory to solve real world problems (Bertha, 2010). It is the practical knowledge that is interesting to the student and immediately useful to the community. However, it has been realized that the practical knowledge cannot be easily taught in a class room setting as it requires both time and experience. The use of case studies is therefore important because it taps into practical knowledge and real world experiences that students are able to relate to and learn from. Among many other educational tools that have been developed to realize the learning objectives in the computing field, tools based on case studies are definitely in short supply (Towhidnejad, Hilburn, & Salamah, 2011). Traditionally, case studies have been widely used as pedagogical tools in business, law, ethics, economics, systems engineering, and medicine but they are not as common in software engineering. Case studies have many unique and distinct advantages in software engineering education including that they allow for:

- Application of knowledge or skills in a real-world setting

- Identification and clearer definition of the problem

- Participative, collaborative, inclusive, and team based approach

- Opportunities for creative brain storming, and

- Development of solutions to problems.

Therefore, the hypothesis presented in this paper is that the case studies would be effective educational tools to introduce real-world professional practices into the classroom which would help the students not only in honing their problem solving skills but also increasing self-directed learning skills and team skills. As the case studies are grounded in real world situations, they offer opportunities to develop a perspective on knowledge application, project management and project economics in addition to the domain knowledge enhancement in software engineering.

Case studies can be used effectively to contextualize theoretical concepts (Davis & Wilcock, 2003). It has been shown in many studies (Grant, 1997; Raju & Sanker, 1999; Sivan, Wong, Woon, & Kembler, 2001) that benefits of case studies are derived from their interactive learning strategy and the shifting of emphasis from teacher-centered to more student-centered activities. The benefits of the case study method listed by Davis and Wilcock (2003) are that the case studies:

- Allow the application of theoretical concepts to be demonstrated, thus bridging the gap between theory and practice

- Encourage effective learning

- Provide an opportunity for developing key skills such as communication, team work and problem solving

- Increase student enjoyment of the topic and hence increase their desire to learn

- Allow longer retention of the material.

The sources for case studies can be diverse (Nespoli & Lambert, 2010). For example, case studies may be developed via undergraduate student project work, co-op experiences/summer work, senior year capstone projects, graduate (ME/MS) thesis projects and professional work with industry partners. In the present work, the case studies have been drawn mainly from industrial partners, large scale government projects, and from the direct professional consulting experience of the authors.

Another important and emerging aspect of engineering education is the fully on-line engineering degree program. While there are just a few completely online undergraduate programs available at this time, more and more classes are being offered online to facilitate the education of nontraditional students such as mid-career employees and military personnel (Fisher et al., 2007). In order to ensure that the online engineering education is at least equally effective to (if not better than) traditional face-to-face education in terms academic quality, rigor and outcomes, appropriate teaching tools must be developed to suit the online teaching/learning media. In this regard, we believe case study based education is one of the superior tools to deliver an equivalent and challenging laboratory experience for online students!

In this paper the process used for developing case studies is described in Section 2, a fully developed case study in the domain of software testing is presented in Section 3, pedagogy and educational outcomes are discussed in Section 4, an example of the implementation of software testing case studies is given in Section 5, and student evaluations of active learning tools are presented in Section 6.

2. Case Study Development

The project team has used the three steps process depicted in Figure 1 in developing case studies. Once completed, the case studies are integrated in to the curriculum as in-class exercises or as homework assignments. The case study tasks may be accomplished by the students working individually or as student teams consisting of 2-4 students per team. The case study contains scenarios or real world stories along with the background material such as setting, personalities, sequence of events, relevant data, problems, and conflicts. Students involved in group work related to case studies can develop skills required for success such as group decision making, consensus building, negotiation, and tolerating differences of opinion within a diversified work place (Kauffman, Abdel-Salem, Williamson, & Considine, 2009).
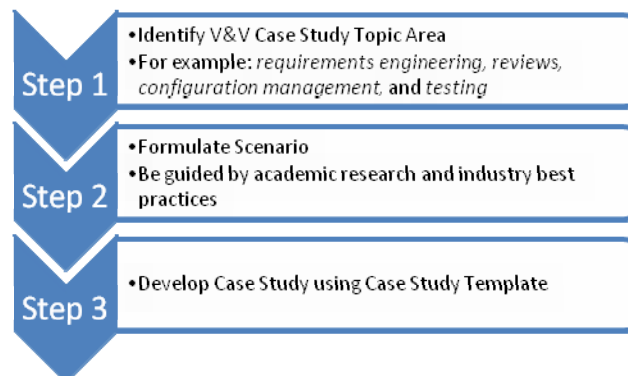
2.1 Case Study Topic

Figure 1. Case study development process

Software Testing is the Software V&V topic used in describing test case development in this paper. Testing is an investigative process in which a software system or component is evaluated against a set of predefined inputs to observe whether or not it gives the expected results. If the results are met then the user requirements are met. Testing results in demonstrating software quality and reliability and helps product development by identifying errors or missing requirements (Acharya & Manohar, 2015). Whether one is writing an individual unit test or designing a product's test plan, it is important to take a step back and think about how effective the tests are at detecting and reporting bugs in the code.

2.2 Case Study Template

A template for case study development proposed in this work is shown in Figure 2. The benefits of developing a template for a case study are twofold: (a) template provides a standardized way to document the background information, description and objectives of case studies and (b) facilitates identification of any missing information or gaps of knowledge for the students as they attempt to solve the questions based on the case study. Using a template also allows for systematic improvement of the description contained in the case study as it is delivered using revised iterations in future classes.

Focus Area :	**\<Software V&V Topic>**
Case Module Name :	*\<Case Module Name>*

Prerequisite Knowledge:	Before attempting this module you need to have knowledge of:
	  1. \<Knowledge Area 1>
	  2. …
	  n. \<Knowledge Area n>

Learning Outcomes/Objectives:	Upon completion of this module you be able to meet the following
	Outcomes/Objectives:
	  1. \<Outcome 1/Objective 1>
	  2. ...
	  n. \<Outcome n/Objective n>

Keywords:	\<list of keywords relevant to the software V&V Topic>
Scenario:	\< case study description>
Exercise:	\<description of student activity>
Instructing Notes:	\<description of how the case study is to be delivered>
Assessment Procedure:	\<description of how the student learning is assessed >

Figure 2. Case study development template

2.3 Basics Ideas to Be Introduced in Software Testing Education

Figure 3 depicts a domain-specific case study in the area of software testing. For professional software technology companies software testing is a critical aspect of reliability and quality of their business. The basic purpose of testing is to validate the testing coverage of the application being evaluated. Writing effective test cases is a skill that can be achieved by experience with and in-depth study of the application about which test cases are being written.

For example, Google (Google, 2014) suggests several important qualities such as fidelity (a test that is sensitive to defects in the code), resilience (a test that fails only when the code is defective), and precision (a test that shows the exact location of the defect in the code) that every test should try to maximize. These three qualities are often in competition or opposition with each other. It's easy to write a highly resilient test (the empty test, for example), but writing a test that is both highly resilient and has high fidelity is hard to do. Students and professionals can begin to learn how to do software testing by creating tests scripts. The development of test scripts therefore needs greater attention and a systematic procedure is suggested in Figure 3.

Focus Area :	**Software Testing**
Case Module Name :	*Industry Test Case Development*
Prerequisite Knowledge:	Before attempting this module you need to have knowledge of: Software Testing, Test Outline and Test Case Development
Learning Outcomes/Objectives:	Upon completion of this module you be able to meet the following **ABET *Criterion 3* Outcomes:** **Outcome *b*:** *An ability to design and conduct experiments, as well as to analyze and interpret data.* **Outcome *e*:** *An ability to identify, formulate and solve engineering problems* **Outcome *g*:** *Graduates have an ability to communicate effectively.* **Outcome *k*:** *Graduates have an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.*
Keywords:	Test Plan, Requirements, Formal Review, Informal Review, Reviewer, Peer Review, Recorder, Moderator, Author
Scenario:	OptSoft Inc. has been awarded a project to develop a Hospital Management System. This system has the following subsystems: • Admit-Discharge-Transfer (ADT) / Patient Registration System (PRS) • Hospital Information System (HIS) • Radiology Information System (RIS) • Picture Archiving and Communication System (PACS) • Image Acquisition Modality The Software Requirements Specification (SRS) for the System has already been completed and accepted by the customer. Your QA Manager has asked your team to come up with test cases for the PRS that will be used to test the PRS features. Everyone on the team should be able to answer the following: • What am I going to test? • What's the importance of having a formal template for test cases development? • What is the importance of preconditions in test cases development?
Exercise:	■ Develop ten test cases that are relevant to the application you are testing such as those described in Section 2.3. ■ Use the "Test Case Template" given by your instructor to write scripts for the ten test cases that you just developed (example template for writing test cases is provided in Figure 1 in this paper) ■ Identify test cases that can be automated. (Assume appropriate information when necessary.)
Instructing Notes:	This is to be delivered as a classroom/Homework assignment. Class 1 (50 minutes) 1. Ask the students to individually prepare the test cases 2. Form teams of 4 students in class 3. Ask the teams to discuss the questions given in the case study. 4. Discuss as a class (10 minutes)
Assessment Procedure:	In-class short quiz for this module may be developed and given to the students in addition to their performance in the test cases, test script writing and case review

Figure 3. Case study for software testing

3. Test Case Script Template

In this task, the students are expected to develop test cases using a given template. A sample test case script template is given in Figure 4.

The developed case studies can be integrated in the curriculum either as homework assignments or as in-class exercises. The authors have utilized the case studies in both ways and it is suggested here that the more complex case studies should be assigned as homework while relatively narrow and focused case studies may be discussed as in-class exercises. It should be noted here that an appropriate theoretical framework needs to be established via lectures to lay the foundation before case studies based on the relevant topics are given to the students. In-class

quizzes are then utilized to assess the effectiveness of student learning. Therefore students gain the necessary theoretical framework via lectures and then are able to understand the practical applications of the principles via case studies.

Student performance in case studies can be assessed based on their in-class participation and the accuracy of their response to the case study questions. A grading rubric may be appropriately developed for each case study depending upon whether it is an individual or group assignment and if it is an in-class or homework assignment.

<Client> Test Case Script Name			
Project ID	//the unique project identifier		
AUT Name	//the definitive name of the Application Under Test (AUT)	**AUT Version**	//the definitive version information for the Application Under Test (AUT)
Iteration ID	//the unique identifier for the iteration this test is being conducted in	**Date of Test**	//MM/DD/YYYY
Test ID	//the unique identifier for the test		
Purpose of Test	//a brief description of the purpose of the test including a reference where appropriate to the requirement that is to be tested (consider providing references to the requirements specification, design specification, user guide, operations guide and/or installation guide), as well as any dependencies from or to other Test Cases		
Test Environment	//a brief description of the environment under which the test is to be conducted (may include a description of the state of the AUT at the start of this test, details regarding the platform or operating system, as well as specific information about data used in this test)		
Test Steps	//concise, accurate and unambiguous instructions describing the precise steps the Tester must take to execute the test, including navigation through the AUT as well as any inputs and outputs		
Expected Result	//a brief and unambiguous description of the expected result for passing of a test subsequent to test execution		
Likely Problems/Bugs Revealed	//likely outcomes of testing such as feature not working, expected results not observed, missing or inaccessible features (optional field)		

Figure 4. Sample Test Case Script Development Template

Using the steps shown in Figure 1 the authors have developed and delivered the test cases studies listed in Table 1.

Table 1. List of developed and delivered test case studies

Software V&V Topic	Case Study Scenario
Requirements Management	Requirements from a Customer Perspective - Ambiguity
	Understanding User Requirements
Software	Performance Testing/Load Testing
Testing	A Software Test Plan (STP)
	Test Case Development
Software	Importance of Peer Reviews
Reviews	Peer Review Tools
Configuration Management	Continuous Integration (CI)
	Version Control Management System
Additional Topics	Liability for Bad Software and Support
	Software Legal Issues

4. Educational Outcomes Assessment

The case study based approach to teaching and learning is broad in terms of its coverage of educational outcomes and it has been suggested that it can be used to deliver all eleven "*a*" through "*k*" criteria of ABET accreditation (Towhidnejad, Hilburn, & Salamah, 2011). The flexibility of case studies coupled with the richness of data and information analysis, decision making education and conflict resolution results in strong links with ABET criteria. Kauffman et al. (Kauffman, Abdel-Salem, Williamson, & Considine, 2005) have mapped case study outcomes to the ABET criteria for engineering economy case studies. Such analysis is adopted here for case studies in software engineering as shown in Table 2.

Table 2. Case study analysis in software engineering and its relation to ABET criteria

ABET Criterion	Software Engineering Case Study Analysis
(b) An ability to design and conduct experiments as well as to analyze and interpret data	Case studies requires students to find or develop the important information and ignore data that is not relevant
(c) an ability to design a system, component or a process to meet desired needs	Case studies requires students to confront complex issues such as trade off analysis along with time, resource and risk management decisions
(d) an ability to function on multi-disciplinary teams	Case studies requires students to solve case problems, they must also learn to negotiate and understand different viewpoints prior to their decision making
(e) An ability to identify, formulate, and solve engineering problems	Case studies requires students to identify important data and ignore irrelevant data, actively look for missing data or make appropriate assumption and use mathematical / computer simulation based tools to solve engineering problems
(g) an ability to communicate effectively	Case studies requires students to make presentation of case analysis results in both oral and written formats
(h) the broad education necessary to understand the impact of engineering solutions in a societal and global context	Critical thinking required by case study analysis promotes systems thinking related to larger impact of decision alternatives
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	Case studies requires students to learn and apply contemporary engineering tools to solve case problems

Pedagogical outcomes that are relevant for software verification and validation have been identified at the author's institution based on ABET Criterion 3 outcomes assessment. The relationships between the specified ABET outcomes for this course and their correspondence with the revised Bloom's taxonomy (Google, 2014) for STEM disciplines are shown in Table 3.

Table 3. Expected pedagogical outcomes for software V&V course at author's institution

Applicable ABET Criterion 3 Learning Outcomes for Software V&V course at author's institution	Revised Bloom's Taxonomy for STEM Disciplines [11]
b. An ability to design and conduct experiments, and analyze and interpret data	I & III
c. an ability to design a system, component or a process to meet desired needs	IV, V & VI
e. An ability to identify, formulate, and solve engineering problems	II, IV & V
f. An understanding of professional and ethical responsibilities	V & VII
g. An ability to communicate effectively	III, IV & V
h. Broad education necessary to understand the impact of engineering solutions in a global and societal context	VI
i. Recognition of the need for and an ability to engage in life-long learning.	VII
j. A knowledge of contemporary issues	V & VI
k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	VI & VII

Taxa I—Pre-knowledge Conceptual Experiences: hands-on laboratory experiences via demonstrations, physical models, practical applications to demonstrate, visualize and observe basic concepts.

Taxa II—Basic Conceptual Knowledge: learning, understanding, memorizing basic engineering concepts, definitions, terms, symbols, theories, laws and equations.

Taxa III—Applied Conceptual Knowledge: solving simple concept-based problems and conducting related laboratory experiments.

Taxa IV—Procedural Knowledge: working knowledge of solving multi-concept engineering problems.

Taxa V—Advanced Knowledge and Analytical Skills: inter-domain and open-ended problem solving skills.

Taxa VI—Project-based Knowledge: creative, conceptual, analytical, design, manufacturing and management skills.

Taxa VII—Professional Engineering Knowledge and Practices: life-long learning experiences, skills and practices.

It is clear from the information presented in Tables 2 and 3 that it is possible to evaluate student learning outcomes *b*, *c*, *e*, *f*, *g*, *h* and *k* using the case study based educational tools.

5. Implementation of the Case Study Method

One of the authors has been delivering a S/W V&V course since 2005 and is required to perform an ABET Criterion 3 outcomes assessment. Figure 5 depicts a graphical display of the class assessment performed in Spring 2013 when the case study approach was not incorporated as a pedagogical approach. The Spring 2013 class had seven software engineering junior level students (all males) and all of them were considered for this study. This chart presents percentages of students scoring 80% or better on a variety of assessment tasks.
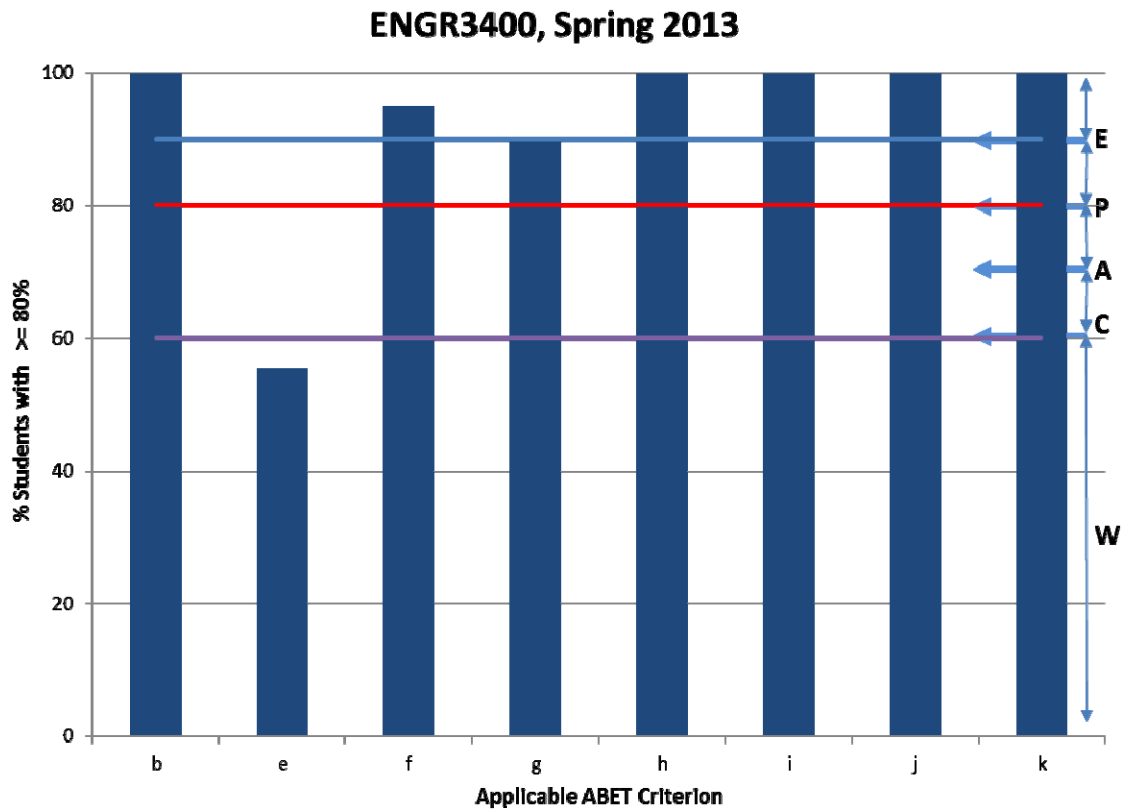
ENGR3400, Spring 2013

Figure 5. Student outcomes assessment with respect to the specified ABET criteria in Spring 13 term-case studies were not available for this class. (E-Excellent, P-Proficient, A-Adequate, C-Concern and W-Weakness)

The student performance in each assessment task was measured and regrouped in terms of ABET outcomes to calculate percentage of students that scored within certain levels of assessment vector as detailed in Table 4 given below.

Table 4. Descriptors of ABET outcomes assessment vector

% of students with at least 80% or better score in assessment tasks	Descriptor of the Resulting Proficiency Status
90%-100%	Excellent (E)
80%-89%	Proficient (P)
70%-79%	Adequate (A)
60%-69%	Concern (C)
< 60%	Weakness (W)

It is seen from Figure 5 that there was a weakness associated with learning outcome "*e*" (an ability to identify, formulate, and solve engineering problems), where less than 60% of the students scored better than 80% on the assessment tasks, causing ABET outcome "*e*" to be identified as a "weakness". One of the main reasons for the lower outcome percentage is because the student performance data was obtained through exams, which may not be the best suited tools for assessing outcome "*e*". It is shown in Table 1 that case study based education can be used to enhance outcome "*e*".

From the 2013 ABET Outcome Assessment report this instructor realized a need for more applied, higher level learning tools. In the 2015 delivery of the S/W V&V course case study approach was incorporated as a pedagogical approach and relevant outcomes assessment was performed. The Spring 2015 class had twelve software

engineering junior level students (all males) and all of them were considered for this study. This time the student performance data for outcome "*e*" was obtained assessing student performance in case study related tasks. The results of this evaluation are presented in Figure 6.

It can be seen clearly that the student performance related to outcome "*e*" is now in the excellent range (>= 90%) as compared to being an area of concern (< 60%) in Spring 13. This presents clear evidence that the case study based teaching method is more effective in delivering an ability to identify, formulate, and solve engineering problems to the students. Therefore case study based educational tools will be progressively developed, adopted and delivered in several other aspects of the software V&V area such as legal issues in software, software consumer protection, and requirements from the customers' perspectives. The results of those implementations will be reported later as more data become available.
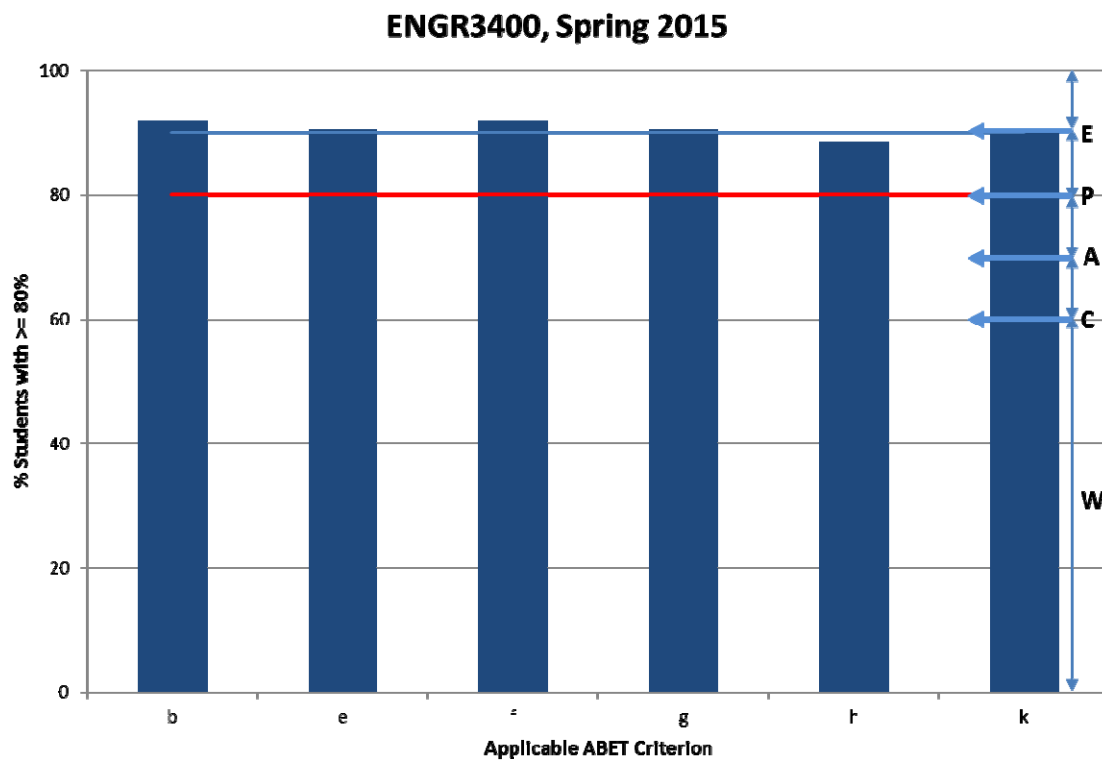
Figure 6. Student outcomes assessment with respect to the specified ABET criteria in Spring 15 term-case studies were delivered in the class

6. Student Evaluation of Active Learning Tools

A student survey tool was developed to assess the effectiveness of the active learning tools developed here and receive feedback for future improvement. Appropriate Institutional Review Board (IRB) clearance was obtained for the survey tools employed in this study. Ten software engineering junior level students (all males) participated in this study. The survey was carried out in-class on the last day of class using paper survey. Instructor was not present in the room and the survey was anonymous. The students were asked to rate the value of the in-class activities in enhancing their knowledge and learning process experience (Table 5). The feedback was given on a Likert scale of 0 (don't know), 1 (not useful), 2 (somewhat useful), 3 (moderately useful) and 4 (extremely useful). The results showed that 90% or more of the students who just completed the class found all of the active learning tools (case studies, exercises, videos) to be moderately or extremely useful. Related specifically to case studies, 100% of the students found the case studies to be moderately or extremely useful. On the other hand, the students found less utility in the written homework assignments and textbook readings.

Another question in the survey asked students about the types of activities they do in the class, including paying attention to lectures, engaging in small group or class discussions, completing real-world applications, thinking critically, reviewing research, or utilizing professional standards to some degree (Table 6). In this class that

deployed active learning tools, the majority of the students (> 70%) responded that they completed real-world applications and felt accountable to classmates in full class discussions. Their communication skills were also utilized to a greater extent in these activities, thus increasing the educational value of active learning tools.

Table 5. Descriptive results from the general evaluation of course instructional activities

Rate the value of the assignments/activities completed in this class. Address the activities as a whole, rather than focusing on one single instance.											
Activity	NA/ Don't Know (0)		Not Useful (1)		Somewhat Useful (2)		Moderately Useful (3)		Extremely Useful (4)		Descriptive Stats
	n	%	n	%	n	%	n	%	n	%	M	SD
1. Lecture					3	30.0	6	60.0	1	10.0	2.8	0.63
2. Exercises							3	30.0	7	70.0	3.7	0.48
3. Case Studies							6	60.0	4	40.0	3.4	0.52
4. Video Case Study					1	10.0	2	20.0	7	70.0	3.6	0.70

Table 6. Descriptive results from the general evaluation of course instructional activities-Part 1

Address how often you do each of the following in a TYPICAL DAY IN THIS CLASS:											
	NA/ Don't Know (0)		Not at all (1)		To a small degree (2)		To a moderate degree (3)		To a large degree (4)		Descriptive Statistics
Activity:	n	%	n	%	n	%	n	%	n	%	M	SD
1. Pay attention during at least 90% of the class session					1	10.0	4	40.0	5	50.0	3.4	0.70
2. Engage with classmates in small group discussion about course content	1	10.0	1	10.0	2	20.0	6	60.0			3.3	1.06
3. Complete real-world applications of course content					1	10.0	8	80.0	1	10.0	3.0	0.47
4. Think critically about course content					1	10.0	4	40.0	5	50.0	3.4	0.70
5. Engage with classmates in full class discussion about course content							2	20.0	8	80.0	3.8	0.42
6. Check texts or other communication via a handheld device			6	60.0	4	40.0					1.4	0.52
7. Feel accountable for my contribution to class.					1	10.0	8	80.0	1	10.0	3.0	0.47
8. Review research in the field.					5	50.0	3	30.0	2	20.0	2.7	0.82
9. Utilize/refer to professional standards.							4	40.0	6	60.0	3.6	0.52

Finally, the vast majority of the students (>= 90%) reported positive student behavior in this class as compared to other classes in their major in terms of the time spent in learning, subject interest, and understanding (Table 7).

Table 7. Descriptive results from the general evaluation of course instructional activities-Part 2

Answer each of the following regarding ENGR 3400 compared to other recent courses in your major:												
Question	Much Less (1)		Slightly Less (2)		The Same (3)		Slightly More (4)		Much More (5)		Descriptive Statistics	
How does:	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**M**	**SD**
1. the **amount of time** that you spent on this course OUTSIDE OF CLASS MEETINGS compare to the **amount of time** spent working on other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?	1	10.0	4	40.0	4	40.0	1	10.0			3.5	0.85
2. the **amount you learned** in this course compare to the **amount you learned** in other undergraduate courses **IN YOUR MAJOR in the last two semesters**?			1	10.0	5	50.0	4	40.0			4.3	0.67
3. your **interest in the types of instructional activities utilized in this course** compare to your interest in the types of instructional activities used in other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?			3	30.0	3	30.0	4	40.0			4.1	0.88
4. your **interest in content of this course** compare to your interest in the content of the other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?	1	10.0	3	30.0	4	40.0	2	20.0			3.7	0.95
5. your **understanding of content of this course** compare to your understanding of the content of the other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?			4	40.0	4	40.0	2	20.0			3.8	0.79
6. the **amount of real-world application problems in this course** compare to the amount of real-world applications of the other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?	1	10.0	1	10.0	3	30.0	5	50.0			4.2	1.03
7. the **amount of verbal communication required in this course** compare to the amount of verbal communication required in the other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?			1	10.0	1	10.0	8	80.0			4.7	0.67
8. your **satisfaction of this course** compare to your satisfaction in the other undergraduate courses **IN YOUR MAJOR in the last 2**			2	20.0	5	50.0	3	30.0			4.1	0.74

semesters?									
9. the **amount of time** that you spent on this course OUTSIDE OF CLASS MEETINGS compare to the **amount of time** spent working on other undergraduate courses **IN YOUR MAJOR in the last 2 semesters**?	1	10.0	4	40.0	4	40.0	1	10.0	3.5　0.85

7. Summary and Future Work

To develop case studies, the needed data, information and scenarios have been drawn from industry partners and from professional consulting work of the authors. The template for developing case studies has been created and presented in this paper. Case notes and instructors' supplementary materials have been prepared. The case studies developed here cover many areas in the software verification and validation knowledge domain including software testing, legal issues in software, software consumer protection, and requirements from the customers' perspectives. Applicable student learning outcomes for the software V&V course have been determined, and their relationship to ABET criteria and revised Bloom's taxonomy for STEM disciplines has been mapped. The effectiveness of case study based educational tools has been determined based on this evaluation context. Using baseline data from the Spring 2013 semester when case studies were not available, the potential impact on increased student learning and engagement based on the incorporation of the new educational tools described in this paper, has been seen. The newly developed case studies were introduced to the students in Spring 2015 term. The effectiveness of these teaching/learning tools was assessed, and there was evidence that student learning in the areas of problem identification, formulation and solving is significantly improved with the use of case studies. The development of more case studies in different areas of S/W V&V has been undertaken and those case studies will be delivered in the near future.

Acknowledgements

References

Acharya, S., & Manohar, P. (2015). *Cases in Software Verification and Validation*. Alexander Street Press.

Bertha, C. (2010). *How to teach engineering ethics course with case studies*. ASEE Annual Conference, Louisville, Kentucky.

Davis, C., & Wilcock, E. (2003). *Teaching materials using case studies*. UK Center for Materials Education.

Fisher, F., Hadim, H., Esche, S., Ubell, R., & Chassapis, C. (2007). *Feasibility of a fully online undergraduate mechanical engineering degree for non-traditional learners*. ASEE Annual Conference, Honolulu, Hawaii, USA.

Girgis, M. (2010). *A new engineering taxonomy for assessing conceptual and problem-solving competencies*. ASEE Annual Conference, Louisville, Kentucky.

Google. (2014). *Effective Testing*. Retrieved from http://googletesting.blogspot.com

Grant, R. (1997). A claim for the case method in the teaching of geography. *Journal of Geography in Higher Education*, *21*(2), 171-185. http://dx.doi.org/10.1080/03098269708725423

Kauffman, P., Abdel-Salem, T., Williamson, K., & Considine, C. (2005). *Privatization initiatives: A source for engineering economy case studies*. ASEE Annual Conference, Portland, Oregon.

Nespoli, O., & Lambert, S. (2010). *Engineering design case studies: Effective and sustainable development methods*. ASEE Annual Conference, Louisville, Kentucky.

Raju, P. K., & Sanker, C. S. (1999). Teaching real-world issues through case studies. *Journal of Engineering Education*, *88*(4), 501-508. http://dx.doi.org/10.1002/j.2168-9830.1999.tb00479.x

Sivan, A., Wong, L. R., Woon, C., & Kembler, D. (2001). An implementation of active learning and its effects on the quality of student learning. *Journal of Innovations in Education and Training International*, *37*(4), 381-389. http://dx.doi.org/10.1080/135580000750052991

Towhidnejad, M., Hilburn, T. B., & Salamah, S. (2011). *Reporting on the use of a software development case study in computing curricula*. ASEE Annual Conference, Vancouver, Canada.

Copyrights