The Design and Evaluation of Class Exercises as Active Learning Tools in Software Verification and Validation

Peter Y. Wu
wu@rmu.edu
Department of Computer and Information Systems

Priyadarshan A. Manohar
manohar@rmu.edu
Department of Engineering

Sushil Acharya
acharya@rmu.edu
Department of Engineering

Robert Morris University
Pittsburgh, PA 15108, USA

Abstract

It is well known that interesting questions can stimulate thinking and invite participation. Class exercises are designed to make use of questions to engage students in active learning. In a project toward building a community skilled in software verification and validation (SV&V), we critically review and further develop course materials in the software engineering curriculum for an undergraduate course in SV&V. The project involves the joint effort with many other academic institutions and industry partners. There are four topic areas of Software Engineering in our focus: Requirements Engineering, Software Review, Configuration Management, and Testing. We see class exercises as active learning tools for the students in our flipped classroom approach. We present our design of the class exercise: in its generic components envisioning how it may be used in general, but also its use in selected examples to illustrate these components. The class exercise design includes the learning objectives to indicate how the course design meets the learning outcome objectives for ABET accreditation. We further applied a classification of the learning objectives to present the case of the class exercises as active learning tools. Our initial implementation is ready and we are in the process of implementation with partner institutions for feedback and review.

Keywords: Software Verification and Validation, SV&V, Active Learning Tool, Software Engineering.

1. INTRODUCTION

Teaching software engineering at the college level requires balance between the knowledge and hands-on experience. This is especially the case at the undergraduate level when students have just acquired proficiency in programming but are generally short on the appreciation of the practice of software development processes in industry. Funded by an NSF-TUES Grant (National Science Foundation: Transforming Undergraduate Education in STEM), our project focuses on

developing new tools for teaching software verification and validation (SV&V) at the undergraduate level. The basic objective of the project is to enhance the quality of software engineering education by increased student engagement in learning as well as bridging the gap between the theoretical knowledge discussed in the classroom and the complexity of real world problems. This endeavor will promote SV&V awareness and increase SV&V practitioners skilled in the practice. The goal is to improve product and process quality levels throughout the software development community, resulting in larger and better skilled SV&V community. Section 2 will briefly describe the background, scope and rationale of the research project.

The project is carried out through an academic-industry partnership. The entire team involves 2 academic development partners, 5 industry development partners and 12 academic implementing partners. In the project, we join with the academic and industry partners to critically review the existing course materials against current methods and best practices. We then refine and further develop the course to address the gaps and inadequacies. To engage the students in active learning, we practice a flipped or inverted classroom approach (Strayer 2012; Bishop & Verleger 2013; Frydenberg 2013) and basically use the class time for activities requiring the students to review lecture and other reading materials beforehand. During the class time, we apply active learning tools to engage the students. Section 3 will describe the specific topic areas of focus in SV&V and the tools we developed.

In this paper we present the class exercises, primarily for class discussion in concert with the other learning tools. Section 4 will describe the design of the class exercise, generically for use in different settings, identifying the components included there. We also illustrate each component of the class exercise in examples to show how the tool is used in the specific topic area of SV&V. Section 5 presents an example of a class exercise and how it is implemented in a classroom setting.

Section 6 describes the pedagogical evaluation strategy of the class exercises to facilitate support for our hypothesis that active, engaged learning will enhance student experience, interests and learning. Finally, Section 7 will present the summary conclusion of our initial implementation, and we invite IS educators to share access to these tools for review, use and feedback.

2. BACKGROUND AND RATIONALE

Software quality is not just a critical issue in the software industry, but it also becomes crucial in many aspects of today's information society at large. With software products being ubiquitous, it is a factor in privacy of information, in legal matters of liability, as well as national security. The fundamental challenge to a solution to improve software quality is in the people and processes that develop and produce the software products. Even after decades of development, the software industry continues to spend considerable time and effort to deal with the problem. Much of the improvement can be attributed to the implementation of standards and practices like SV&V. However survey still shows that SV&V is simply not adequately practiced in the software industry (Rakitin, 2013). Acharya et al (2014) reasoned that firstly, there is not enough awareness of the SV&V benefits, and secondly, the lack of practitioners who sufficiently understand the SV&V topics and processes.

The research project therefore aims at the root cause in the lack of SV&V courseware for effective education in academia and on-the-job training in industry. We create focus groups comprising of academic and software industry partners to critically review our existing course materials in joint effort, to identify the gaps and inadequacies when checked against current methods and best practices. Then we refine the lecture materials and develop active learning tools for teaching SV&V. We modularize the teaching materials and tools into small deliverables of 25 minutes duration and in generic formats for adaptation to various settings. The developed modules are easily integrated into software courses and can also be adapted by the industry for on-the-job training. The project's goal includes the committed support from academia and industry to sustain growth and further development for a skilled SV&V community.

Our hypothesis is that the class exercises would be effective learning tools for students since they facilitate student learning by doing and subsequently applying what they learn to solve problems in the real world. These focused exercises would enhance the understanding of the underlying theoretical concepts presented in class (and in preparatory reading) and provide a context for their application. Several class exercises developed in this work will allow the students to gain insight into the entire lifecycle of software testing process including planning, designing, implementing, recording, reporting and managing aspects of the process.

3. TOPICS AND LEARNING TOOLS

The course enhancement effort is guided by the following four specific SV&V topic areas.

- Requirements Engineering,
- Software Testing,
- Software Reviews, and
- Configuration Management.

We identified these as the critical areas in the software engineering process and areas of importance in the industry as well. The SV&V course modules are therefore based on each of these topics.

For each of these SV&V topic areas, we develop active learning tools to be used in the course modules. These learning tools include the following:

Case Studies
Case studies are drawn from industry SV&V practices. Students are presented industry standard documents for review to prepare for the tasks. These tasks may be resolution of review conflicts in the Software Requirements Specification (SRS) document, or compliance to security standards, or drafting of testing plans from use cases, but certainly are not limited to these. A more extensive coverage of the study cases developed is being disseminated in other publications (Manohar et al. 2015).

Case Study Videos
Often produced from the scripts first drafted by our industry partners and confirmed by the testimonies shared in focus group discussions, case study videos provide a realistic picture for the audience to appreciate many SV&V processes in practice. These may show how peer code review is done, and how potential tension or conflict may arise, or the tedious detailed nature of requirements solicitation.

Class Exercises
Based on the context of the class module, class exercises are designed for the class time to explicitly raise questions to invite student participation. It may be questions to think further into the concepts for a deeper understanding, or practice using their knowledge with hands-on practice for problem solving. There are many ways of using class exercises. For a small class, the teacher may simply use the exercise to engage the students in discussion and practice. For larger classes, the students can form small groups to use the class exercise as instrument leading to group projects.

Woods and Howard (2014) effectively used class exercises for Information Technology students to study ethical issues. Day and Foley (2006) used class time exclusively for exercises, having their students prepare themselves with materials provided online. Bishop and Verleger (2013) presented a comprehensive survey of the research in different ways of using class exercises, often referred to as the "flipped" classroom.

The research project is on-going, but the initial implementation of the learning modules is ready for sharing and review. The following sections will describe in further details the class exercises for the SV&V course.

4. COMPONENTS OF A CLASS EXERCISE

Each class exercise consists of the following components:

a) Exercise Description,
b) Instruction Notes,
c) Student Handout,
d) Assessment Instrument.

The Exercise Description provides the general information about the exercise. That includes the module name, the focus topic area, any prerequisite knowledge, ABET learning outcomes, keywords, expected delivery duration, and a single sentence description of what the student is supposed to do in the exercise. Figure 1 illustrates the Exercise Description in a template, filled out as Class Exercise for distinguishing between business requirements and functional requirements.

The Instruction Notes describes for the teacher how to deliver the exercise in class. It may serve as a guide card for the teacher about this exercise, but it includes materials the teacher may need to use for the exercise, such as a slide set presentation. It may also serve as a check list, a reminder about what to do, such as administering the assessment instrument at the conclusion of the class. Figure 2 illustrates that for the exercise in Requirements Engineering.

Student Handout includes everything that the students need to participate in the class exercise. Quite often it is the work sheet for the students. But it may also include other documents or artifacts for review, or tools for use. The instructor will need to prepare sufficient number for the students to use or share. In this example of Business Requirements versus Functional

Requirements exercise, the Student Handout is a work sheet, illustrated in Figure 3.

Exercise Description

Module Number: RM03

Focus Area	Requirements Engineering
Exercise Module Name	**Business Requirements versus Functional Requirements**
Prerequisite Knowledge	Before attempting this exercise module, the student should have knowledge of these terms: (a) Requirements (b) Business Requirements (c) Functional Requirements
Learning Outcomes	Upon completion of this module, the student should be able to meet the following ABET Criterion 3 outcomes. • #7- ability to communicate effectively. • #11 – ability to use the techniques, skills, and tools necessary for engineering practice.
Keywords	Requirements, Business, Functional
Duration	25 minutes x1
Exercise	For each requirement statement, identify whether it is a functional requirement or a business requirement.

Figure 1. Exercise Description

Instruction Notes

Module Number: RM03

Focus Area	Requirements Engineering
Exercise Module Name	**Business Requirements versus Functional Requirements**
Instruction	1. Print the exercise sheet and ask the students to: a. Identify the functional requirements and business requirements in the list. b. (15 minutes) 2. Use the accompanying slide set to discuss the results in class.
Assessment	Ask the students to take the survey by the assessment instrument.

Figure 2. Instruction Notes

The Assessment Instrument is a simple survey primarily for indirect assessment of student learning outcome, and also for student feedback. It is designed for generic use in every exercise, to be completed quickly at the conclusion of the class exercise. Figure 4 below is the assessment instrument for the exercise.

Software Verification & Validation Exercise

Module Number: RM03

Consider each of the following requirements statement, and write down **BR** for Business Requirement or **FR** for Functional Requirement.

BR/FR		Business Requirements vs Functional Requirements
	1	The solution will automatically validate patients against the Hospital Management System. _____
	2	The solution will enable doctors to record patient diagnosis. _____
	3	The solution will enable lab reports to be automatically sent to the respective doctor. _____
	4	We need to implement a web and mobile based employee tracking system that tracks medical staff and increases efficiency by means of monitoring medical staff activity, absenteeism and productivity. _____
	5	The system shall display the longitude and latitude of the medical staff through GPS.
	6	The system shall display the position of the medical staff on Google map.
	7	The system shall allow medical supervisors to send notifications to their medical staff.
	8	We need to establish an online patient portal.
	9	The portal should list our hospital services.
	10	The system shall be able to register a patient using the following fields: Name (20 characters max), Address (200 characters max), and Social Security Number (9 characters long).

Figure 3. Student Handout

Software Verification & Validation
Exercise Module Assessment

Module Number: RM03

Please fill out the survey.

		Yes	No
1	My team and I understood the purpose of this activity.	☐	☐
2	My team and I could complete the activity with the directions provided.	☐	☐
3	At least one member of my team was uncertain of how to carry out the steps of the activity.	☐	☐
4	The activity used a real-world application.	☐	☐
5	I could imagine carrying out this activity as part of my job.	☐	☐
6	I communicated verbally in a small group while completing this activity.	☐	☐
7	I communicated verbally in a large group while completing this activity.	☐	☐
8	I / We provided written communication as part of this activity.	☐	☐
9	I / We made a formal presentation as part of this activity.	☐	☐
10	I thought critically about the content while completing this activity.	☐	☐

Provide your overall thought about this activity.

Figure 4. Assessment Instrument

Our initial implementation at the time of this writing has completed 16 exercises, of total expected delivery time at 800 minutes. Table 1 below lists the module names each with its time duration, categorized in the four SV&V focus areas, and the category of Additional Topics in extra.

focus area	CLASS EXERCISE	min
Requirements Engineering	Ambiguous Questions	25
	Business Requirements and Functional Requirements	25
	Clarifying User Requirements	25
	Needs statements to SRS	25
	Needs statements to User Requirements	25
	Requirements Ambiguity	50
	Stated and Implied Requirements	25
Software Testing	Cost Effective Testing Approach	50
	Test Cases for a Given Requirement	50
	Testing Tools	75
	Understanding Testing	75
Software Reviews	Code Inspection	175
	SRS Review	50
Configuration Management	Defect Lifecycle	25
Additional Topics	Deming's 14 Points on System of Profound Knowledge (SoPK)	50
	Understanding IEEE Standards	50
	TOTAL	**800**

Table 1. Class Exercises

We have used one class exercise module in Requirements Engineering to illustrate the use of the class exercise components. Most of the class exercises listed here have been reviewed in focus groups. In August, 2015, a one-and-a-half-day workshop was organized for project partners and other invited participants to jointly learn and review all the learning tools we have developed: Case Studies, Case Study Videos, and Class Exercises. Eleven academic institutions and industry partners attended this workshop where we shared these tolls and discussed delivery strategies. Since the project is funded by the National Science Foundation, these course materials will be made available at the National Science Digital Library's (NSDL) national portal.

5. AN EXAMPLE CLASS EXERCISE

In this section, we use an example to show how class exercise can illustrate the problems that may arise in SV&V. Refer to Table 1 for the class exercises in the various focus areas. We take Ambiguous Questions as a class exercise example, about the gathering of requirements for an information system development project. Requirements are of two types: functional requirements and non-functional requirements (Suri & Gassert 2005). Functional requirements relate to the actions that a software product must carry out to satisfy the fundamental reasons for its existence. Non-functional requirements are the desirable properties or qualities that the software product must have for customer satisfaction. These are the characteristics pertaining to making the software product fast, usable, portable, reliable, attractive, and the like. However, it is in this area that a lot of ambiguity can arise. In class, the students are taught the terms that potentially may cause ambiguity and confusion in the requirements gathering process. The following lists some of these terms.

- acceptable, adequate.
- high-quality, state-of-the art performance.
- to the extent practicable.
- efficient.
- use-friendly.
- simple, easy, flexible.
- robust.
- seamless.
- optimal, maximal, minimal, reasonable.
- including but not limited to.
- and so on.

The class exercise will then test the students' understanding of these ambiguous terms in a requirements elicitation and gathering process. The students are given a few statements and they are asked to identify the ambiguous term or terms that sound ambiguous and discuss how they may rephrase the requirements that will clarify the meaning, making it unambiguous. The statements are:

1. For a web-based system it is required that loading of all webpages must be completed within a reasonable amount of time.

2. Access right to data is limited to the individuals with managerial rank but those with access rights may also grant access rights to others.

3. Even though the (stock) market is open during business hours, access to stock prices should be available 24/7, supporting client access to the market at the client's time locale.

4. A user should be able to customize the system behavior to cater to his/her own needs. Yet the system should provide a default case for everyone.

5. Every book is identified by the ISBN in the catalog. When a member of the library takes a book out on loan, the system must also identify which copy of the book was loaned

out, so that the member will be responsible for any damage to that specific copy of the book upon its return.

The instructor will then lead the discussion with questions to drill down the ambiguous terms identified by the students. Questions such as: Are the requirements unambiguous for the developers? Are they clear enough for the customer? Who are the customers? Do we charge the customer more if the customer cannot provide us more details? Are the requirement statements testable – so that one can demonstration satisfaction of these requirements? When should we stop discussing requirements, and become ready to do design and development? These questions can be discussed in detail.

6. PEDAGOGICAL EVALUATION

Since the class exercises are designed with specific learning objectives, we mapped the objectives to the learning outcomes derived by ABET (Accreditation Board for Engineering and Technology, Inc.) for engineering program accreditation (ABET-EAC 2014). While experienced teachers may intuitively know that good class exercises presented in an interesting way will invite student engagement into active learning, we proceeded to analyze how the ABET outcomes correspond to the levels in Bloom's taxonomy of knowledge and learning (Bloom et al 1956), to present the case of the class exercises as active learning tools, in our case, specifically for SV&V in the undergraduate curriculum. Table 2 below lists the eleven pedagogical outcomes derived by ABET pertaining to the accreditation of undergraduate engineering curriculum.

We examined the specific learning objectives of the class exercises we developed for the project. For each exercise, we identified the outcomes specifically addressed by the learning objectives as well as the other outcomes the exercise may involve but not specifically address. Figure 5 presents our results: S indicates an outcome specifically addressed, and indicates an outcome that may be involved in the exercise.

The 11 Learning Outcomes by ABET-EAC	
1	An ability to apply knowledge of mathematics, science and engineering.
2	An ability to design and conduct experiments, as well as to analyze and interpret data.
3	An ability to design a system, component or process to meet desired needs.
4	An ability to function on multidisciplinary teams.
5	An ability to identify, formulate and solve engineering problems.
6	An understanding of professional and ethical responsibilities.
7	An ability to communicate effectively.
8	A broad education necessary to understand the impact of engineering solutions in a global and societal context.
9	Recognition of the need for, and an ability to engage in lifelong learning.
10	The knowledge of contemporary issues.
11	An ability to use the techniques, skills and modern engineering tools necessary for engineering practice.

Table 2. The Learning Outcomes by ABET-EAC

Class Exercise	ABET-EAC Learning Outcomes									
	1	2	3	4	5	6	7	8	9	10	11
Ambiguous Questions				S			S				
Business vs Functional Requirements				S	I		S				
Clarifying User Requirements	S			S			S				
Needs Statements to SRS	S			S			S				I
Needs Statements to User Requirements	I			S	I	I	S	I			
Requirements Ambiguity				S			S				
Stated and Implied Requirements	I			S		I	S	I			
Cost Effective Testing Approach	I	S	I	I	I	I	S				
Test Cases for a Given Requirement	I	S		I		I	S				
Testing Tools	I	S									S
Understanding Testing	I	S	S		I		S				
Code Inspection				S		S	S				S
SRS Review	S			S		S	S				S
Defect Lifecycle					S		S				S
Deming's 14 points on SoPK							S				
Understanding IEEE standards							S				S

Figure 5. Class Exercises mapped to ABET-EAC Learning outcomes

To make the case for these exercises to be active learning tools, we adopted the revised Bloom's taxonomy for STEM (Science, Technology, Engineering and Mathematics) disciplines proposed by Girgis (2010). Table 3 lists the seven levels derived from the revised Bloom's taxonomy. We adopted the term "taxa" for each level as proposed, with the description wording specific for STEM education.

Taxa	Description
I	**Pre-Knowledge Conceptual Experiences** hands-on laboratory experiences via demonstrations, physical models, practical applications to demonstrate, visualize and observe basic concepts.
II	**Basic Conceptual Knowledge** learning, understanding, memorizing basic engineering concepts, definitions, terms, symbols, theories, laws and equations.
III	**Applied Conceptual Knowledge** solving simple concept-based problems and conducting related laboratory experiments.
IV	**Procedural Knowledge** working knowledge of solving multi-concept engineering problems.
V	**Advanced Knowledge and Analytical Skills** inter-domain and open-ended problem solving skills.
VI	**Project-based Knowledge** creative, conceptual, analytical, design, manufacturing and management skills.
VII	**Project-based Knowledge** creative, conceptual, analytical, design, manufacturing and management skills.

Table 3. Engineering Knowledge Taxonomy

We classified the ABET-EAC learning outcomes by the proposed taxonomy to present the mapping in Figure 6 below to indicate the expected "taxa" levels each outcome focuses on.

Taxa	Description	1	2	3	4	5	6	7	8	9	10	11
I	Pre-knowledge Conceptual Experience	X	X									
II	Basic Conceptual Knowledge	X	X			X			X			
III	Applied Conceptual Knowledge	X	X		X			X	X			
IV	Procedural Knowledge			X	X	X		X				
V	Adv Knowledge and Analytical Skills			X	X	X	X	X		X	X	
VI	Project-based Knowledge			X					X		X	X
VII	Professional Knowledge and Practices						X			X		X

Figure 6. ABET Outcomes classified in taxonomy

By the ABET-EAC learning outcomes each of the class exercise specifically addresses, we then determine the "taxa" levels each class exercise focuses on. Table 4 is an extension of Figure 5, to include the classification results, showing that the class exercises correspond to all the levels in the knowledge taxonomy.

The class exercises, along with other active learning tools for the project, have been readily shared since Fall 2015 to begin implementation and delivery at the partnering institutions.

CLASS EXERCISE	ABET-EAC outcomes by learning objectives	(Taxa) Knowledge Levels
Ambiguous Questions	4,7	III,IV,V
Business and Functional Requirements	4,7	III,IV,V
Clarifying User Requirements	1,4,7	I,II,III,IV,V
Needs statements to SRS	1,4,7	I,II,III,IV,V
Needs statements to User Requirements	4,7	III,IV,V
Requirements Ambiguity	4,7	III,IV,V
Stated and Implied Requirements	4,7	III,IV,V
Cost Effective Testing Approach	2,7	I,II,III,IV,V
Test Cases for a Given Requirement	2,7	I,II,III,IV,V
Testing Tools	2,11	I,II,III,VI, VII
Understanding Testing	2,3,7	I,II,III,IV, V,VI
Code Inspection	4,6,7,11	III,IV,V,VI, VII
SRS Review	1,4,6,7,11	I,II,III,IV, V,VI,VII
Defect Lifecycle	5,7,11	III,IV,V,VI, VII
Deming's 14 Points on SoPK	7	III,IV,V
Understanding IEEE Standards	7,11	III,IV,V,VI, VII

Table 4. Classed Exercises mapped to ABET-EAC outcomes and Knowledge Taxonomy

7. SUMMARY

In a project aimed at developing a sustained community skilled in Software Verification and Validation (SV&V) for the software industry, we have embarked on critically reviewing and re-developing an undergraduate SV&V course in the software engineering curriculum. Apart from refining the lecture materials, we are using active learning tools in the flipped classroom approach. These active learning tools include Case Studies, Case Study Videos, and Class Exercises. In this paper, we reported the details of the Class Exercise in our design. The Exercise consists of

four generic components: Exercise Description, Instruction Notes, Student Handout and Assessment Instrument. We illustrated the Class Exercise in one module on distinguishing between Business Requirements and Functional Requirements. The finished Class Exercises in our initial implementation are also reported. To evaluate them, we further mapped the learning objectives of the Class Exercises to the expected learning outcomes for ABET accreditation of an undergraduate engineering program. We also analyzed them based on the classification by Bloom's taxonomy of knowledge adapted to software engineering. The finished Class Exercises in our initial implementation are also reported. The Class Exercises, along with other learning tools developed in the project are being reviewed and will subsequently be shared publicly in the NSDL portal.

8. ACKNOWLEDGEMENT

The authors acknowledge the support of NSF through an NSF-TUES grant entitled "Collaborative Education: Building a Skilled V&V Community", Award # 1245036. The input of industrial partners for this project including Eaton Electrical Corporation, PNC Bank, ANSYS, ServiceLink, and JDA Software Group is gratefully acknowledged. The support of the academic development partners Virginia State University (an HBCU), and Milwaukee School of Engineering is highly appreciated. In addition input from academic implementation partners including Embry-Riddle Aeronautical University, Montana Technical University, University of Michigan-Dearborn, Fairfield University, Auburn University, East Carolina University, Kennesaw State University, Bowie State University, and Clarion University is appreciated.

9. REFERENCES

ABET-EAC. (2014). Criteria for Accrediting Engineering Programs (2014-2015 Cycle). Accreditation Board for Engineering and Technology (ABET) Engineering Accreditation Commission (EAC), published March 2014. http://www.abet.org/wp-content/uploads/2015/04/E001-14-15-EAC-Criteria.pdf

Acharya, Sushil, Manohar P., Schilling Jr W.W., Ansari A.A., & Wu P.Y. (2014) Collaborative Education: Building a Skilled Software Verification and Validation User Community. 121st ASEE Annual Conference & Exposition, June 2014, Indianapolis, IN.

Bertha, C. (2010). How to teach engineering ethics course with case studies. Proceedings of the 117th ASEE Annual Conference and Exposition, Louisville, KY.

Bishop, J.L. & Verleger M.A. (2013). The Flipped Classroom: A Survey of the Research. ASEE 120th Annual Conference and Exposition, Atlanta, GA.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. David McKay Company, New York.

Day, J.A. & Foley J.D. (2006) Evaluating a Web Lecture Intervention in a Human-Computer Interaction Course. IEEE Transactions on Education 49(4):420-431, 2006.

Frydenberg, Mark. (2013). Flipping Excel. Information Systems Education Journal, 11(1), pp.63-73. http://isedj.org/2013-11/ ISSN: 1545-679X.

Girgis, M. (2010) A New Engineering Taxonomy for Assessing Conceptual and Problem-Solving Competencies. 117th ASEE Annual Conference & Exposition, Louisville, KY.

Lopez, Gustavo, Cocozza F., Martinez A. and Jenkins M. (2015). Design and Implementation of a Software Testing Training Course. 122nd ASEE Annual Conference & Exposition, June 2015, Seattle, WA.

Manohar, Priya, Acharya S., Wu P.Y., Ansari A.A. & Schilling Jr W.W. (2015). Case Study Based Educational Tools for Teaching Software V&V Course at Undergraduate Level. 122nd ASEE Annual Conference & Exposition, June 2015, Seattle, WA.

Rakitin, Steven R. (2000). Software Verification and Validation for Practitioners and Managers by Steve R. Rakitin, 2nd Edition, Artech House. ISBN: 1-58053-296-9.

Strayer, Jeremy F. (2012). How Learning in an Inverted Classroom Influences Cooperation, Innovation and Task Orientation. Learning Environments Research 15(2):171-193.

Suri, Deepti and Gassert J. (2005). Gathering project Requirements: A Collaborative and Interdisciplinary Experience. 112th ASEE

Annual Conference & Exposition, June 2005, Portland, OR.

Woods, D. & Howard, E. (2014) An Active Learning Activity for an IT Ethics Course. Information System Education Journal 12(1):73-77. http://isedj.org/2014-12/ISSN:1545-679X.