Measuring Broader Impact of NSF-funded Project on Software Engineering Education

Dr. Sushil Acharya, Robert Morris University

Sushil Acharya, D.Eng. (Asian Institute of Technology) is the Assistant Provost for Research and Graduate Studies. A Professor of Software Engineering, Dr. Acharya joined Robert Morris University in Spring 2005 after serving 15 years in the Software Industry. His teaching involvement and research interest are in the area of Software Engineering education, Software Verification & Validation, Software Security, Data Mining, Neural Networks, and Enterprise Resource Planning. He also has interest in Learning Objectives based Education Material Design and Development. Dr. Acharya is a co-author of "Discrete Mathematics Applications for Information Systems Professionals" and "Case Studies in Software Verification & Validation". He is a member of Nepal Engineering Association and is also a member of ASEE and ACM. Dr. Acharya was the Principal Investigator of the 2007 HP grant for Higher Education at RMU through which he incorporated tablet PC based learning exercises in his classes. In 2013, Dr. Acharya received a National Science Foundation (NSF) grant for developing course materials through an industry-academia partnership in the area of Software Verification and Validation.

Dr. Priya Manohar P.E., Robert Morris University

Dr. Priyadarshan (Priya) Manohar Dr. Priyadarshan Manohar is a Professor of Engineering and Co-Director Research and Outreach Center (ROC) at Robert Morris University, Pittsburgh, PA. He has a Ph. D. in Materials Engineering (1998) and Graduate Diploma in Computer Science (1999) from University of Wollongong, Australia and holds Bachelor of Engineering (Metallurgical Engineering) degree from Pune University, India (1985). He has worked as a post-doctoral fellow at Carnegie Mellon University, Pittsburgh (2001 – 2003) and BHP Institute for Steel Processing and Products, Australia (1998 – 2001). Dr. Manohar held the position of Chief Materials Scientist at Modern Industries, Pittsburgh (2003 – 2004) and Assistant Manager (Metallurgy Group), Engineering Research Center, Telco, India (1985 – 1993). He has published over 70 papers in peer-reviewed journals and conferences including a 2007 Best Paper Award by the Manufacturing Division of American Society for Engineering Education (ASEE), three review papers and three book chapters. He has participated in numerous national and international conferences. He is a member of ASM International, TMS, ACerS, AIST, ASEE, and a registered Chartered Professional Engineer. Dr. Manohar's research interests include mathematical and computer modeling of materials behavior, thermo-mechanical processing of steels and other metallic materials, microstructural characterization, and structure – property relationships. He has conducted a number of technical failure investigations, consulted on various materials-related problems, and acted as an expert witness in the Court of Law. Dr. Manohar is the past chair of the Manufacturing Division of ASEE and ASM Pittsburgh Chapter.

Prof. Peter Y Wu, Robert Morris University

Peter Y. Wu is professor of Computer and Information Systems at Robert Morris University. He earned Ph.D. in Computer System Engineering from Rensselaer Polytechnic Institute. He worked for IBM Research Division, first as a post-doc research fellow and subsequently a technical staff member at the T.J. Watson Research Center. He was the chief software engineer and a founding partner of UJB Solutions, LLC, a consulting company in production planning, for two years. He previously held faculty appointments at Hong Kong Polytechnic University, and the University of Pittsburgh. His current research interests are in software engineering, geographic information systems and data analytics.

Broader Impacts of NSF Funded Project on Software Engineering Education

Abstract

The body of knowledge appropriate for undergraduate software engineering program encompasses both theoretical and practical aspects. The knowledge areas (KA) in the 2014 IEEE/ACM Software Engineering Curriculum Guidelines are imperative for undergraduate education and the subsequent professional career. However, due to the lack of active learning tools and the dearth of engaged student learning, software engineering education may not be effectively delivered, resulting in non-coverage by the instructors or non-retention by the students of the required software engineering knowledge area. A three years NSF TUES grant awarded to the authors institute in 2013 has specifically addressed these pedagogical issues. The project which involved partnerships in academia and industry developed 44 delivery contact hours of new Active Learning Tools, deployed to enhance knowledge delivery and retention in Software Verification and Validation (SV&V), specifically in these four focus areas: requirements management, software review, configuration management, and software testing.

In this paper the authors address NSF broader impacts in relation to the project. The paper describes the Active Learning Tools and briefly discusses outcome assessments. It presents student testimonials on how the Active Learning Tools effectively helped them understand knowledge areas. The mobilization of industry and academic partners, the execution of project tasks, the iterative framework used for the development, and the flipped classroom delivery strategy used for engaged learning are also discussed. The paper presents the impact achieved through effective dissemination via workshops and project websites to 31 academic implementation partner institutions, as well as scholarly publications. In conclusion the paper discusses the broader impacts of the NSF funded project on software verification and validation curriculum in undergraduate software engineering education.

1. Introduction

The body of knowledge appropriate for undergraduate software engineering program encompasses both theoretical and practical aspects. The knowledge areas (KA) listed in the 2014 IEEE/ACM Software Engineering Curriculum Guidelines [1] are imperative for undergraduate education and the subsequent professional career. However, due to the lack of active learning tools and the dearth of engaged student learning, software engineering education may not be effectively delivered, resulting in non-coverage by the instructors or non-retention by the students of the required software engineering knowledge area.

A three years National Science Foundation-Transforming Undergraduate Education in STEM (NSF-TUES) grant awarded to the authors' institute in 2013 for a project on software verification and validation specifically addressed these pedagogical issues. Project work resulted in the development of forty-four delivery contact hours of new Active Learning Tools (ALT) in the form of case studies, class exercises, and case study videos. These ALTs were developed through an academia-industry partnership and deployed to enhance knowledge delivery and retention in the field of software verification and validation, specifically in the areas of requirements management, software reviews, configuration management, and software testing.

Intellectual Merit and Broader Impacts are two merit review criteria that the National Science Foundation (NSF) expects proposers to fully address in their grant proposals. The definitions of the two criteria, as noted in the NSF Grant Proposal Guide (Ch. III Section A), are listed below [2]:

1. **Intellectual Merit:** The Intellectual Merit criterion encompasses the potential to advance knowledge
2. **Broader Impacts:** The Broader Impacts criterion encompasses the potential to benefit society and contribute to the achievement of specific, desired societal outcomes.

NSF guidance on how broader impacts can be accomplished [3]***:***

- ***Through the research itself*** (i.e., research that has potential to lead to breakthroughs in certain industries or contribute to solutions to societal problems)
- ***Through the activities that are directly related to specific research projects*** (e.g., using the research project as a training ground for students or early-career scientists)
- ***Through activities that are supported by, but are complementary to, the project*** (e.g., running an educational workshop for high school students on your research topic)

Examples of Target Outcomes for Broader Impacts Activities [3]***:***

- Full participation of women, persons with disabilities, and underrepresented minorities in STEM
- Improved STEM education and educator development at any level
- Increased public scientific literacy and public engagement with science and technology
- Improved well-being of individuals in society
- Development of a diverse, globally competitive STEM workforce
- Increased partnerships between academia, industry, and others
- Improved national security
- Increased economic competitiveness of the United States

- Enhanced infrastructure for research and education

This paper introduces the project and focuses on the broader impacts achieved by the project team in Software Engineering Education.

2. NSF –TUES Project

The goal of the NSF-TUES project was to enhance and transform a required software engineering course, ENGR3400 - Software Verification and Validation (SV&V), taught at the authors institution into a series of mini learning workshops by incorporating academic research and industry best practices through an academia-industry partnership. This endeavor was intended to boost SV&V awareness and increase skilled SV&V practitioners so as to improve product and process quality levels throughout the software development community, resulting in a larger and better skilled software V&V user community.

This project completed in August 2017: 1) Critically examined the existing SV&V course contents at the authors' institute, 2) Identified areas where improvements could be made in pedagogy, 3) Developed forty-four delivery contact hours of ALTs, 4) Tested a ALT delivery strategy, 5) Integrated and delivered new pedagogical tools in the course, 6) Performed assessments and evaluations of the effectiveness of these ALTs, and 7) Disseminated ALTs and assessment reports.

This project targeted both undergraduate students and software practitioners and:

1) Improved SV&V knowledge and skills of students & practitioners.
2) Improved SV&V teaching and learning opportunities.
3) Helped evolve a SV&V community.

A key feature of this project was the partnership of 3 academic institutions and 4 industry partners focused on developing SV&V course modules. This partnership ensured that both academic research and industry best practices were combined to provide needed learning materials for undergraduate students and practitioners. In addition, 6 academic implementing partners incorporated all or specific modules of the developed ALTs in their courses.

The PIs identified the Analysis, Design, Development, Implementation, and Evaluation (ADDIE) instruction systems design framework as the appropriate methodology to use to develop ALTs and getting both the academia and industry actively involved. Course development involved assessing the current academic offering and industry requirements, performing gap analysis to identify Knowledge Areas (KAs), developing ALTs to address inadequacies using a design framework, and assessing the course materials and delivery for further revision and improvement. Key activities included performing a gap analysis and using an iterative review process to design ALT content (Figure 1), conducting workshops, disseminating ALTs, and sharing research outcomes. An English Language Editor then served to edit final work products prior to dissemination. Using this methodology the project team developed, delivered and disseminated the ALTs.

3. Broader Impacts

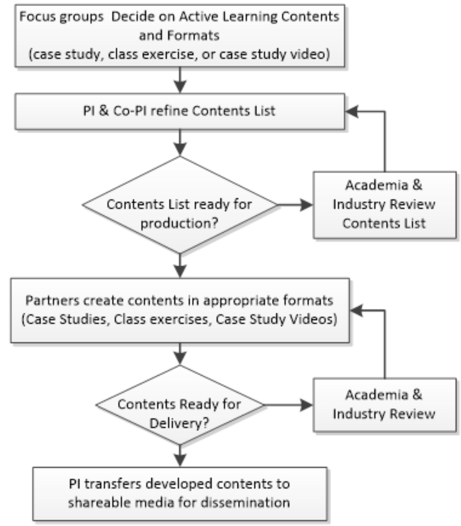

Figure 1: Review Process

The central thesis of the paper is the broader impacts achieved by the NSF-TUES project while the authors share their experience in the project. The evaluation criteria set forth by NSF for the grant proposals therefore serve as the basis for the forthcoming discussion, specifically that on broader impacts.

In section 4 the paper first describes the deliverables of the project – the ALTs and how they may be used. The strategy of delivery in the classroom is also briefly discussed. The ALTs are now available on two public web portals. Section 5 presents the assessment of student learning with the use of the ALTs. At the authors' institute the undergraduate software engineering course was transformed both in terms of contents coverage and delivery strategy. The statistical summary from pre-test and post-test provided by an external evaluator is presented in this section, along with some student testimonials about the ALTs. Both sections 4 & 5 focus on the impacts on the students and their learning.

Section 6 follows on with discussion about the partnerships formed for the successful execution of the project: i.e. partnerships in academia as well as industry. Focus groups of the joint partnership identified best practices and industry and gaps in academic curricula. Led by the principal investigators, the joint partnership developed the ALTs – the case studies, the class exercises and the case study videos. The ALTs were disseminated in multiple workshops attended by academic partner institutions interested in implementation as well as industry partners interested in using the tools for on-the-job training. Section 6 further discusses the broader impacts in the joint community. In section 7 the authors who are the project principal investigators describe the impact the project has had on themselves.

4. Active Learning Tools (ALTs)

Active learning is "embodied in a learning environment where the teachers and students are actively engaged with the content through discussions, problem-solving, critical thinking, debate, or a host of other activities that promote interaction among learners, instructors and the material" [4]. Prince [5] defines active learning as a classroom activity that requires students to do something other than listen and take notes. Active learning is achievable by complementing lecture materials with case studies, class exercises, and case study videos. We call these materials Active Learning Tools (ALTs).

- **Case Studies:** Case studies serve as useful tools to teach applications of science and engineering principles and can be used effectively to contextualize theoretical concepts [6]. It has been shown in many studies 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 [7], [8], [9]. Case studies for this project were drawn from industry SV&V practices. Students were presented industry standard documents for review to prepare for the tasks. These tasks would 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. This project developed, implemented, and disseminated 12 case studies [10]. Each case study consists of a case study description, instruction notes, student handout, and assessment instrument.

- **Class Exercises:** Good questions raised in class invites student participation. Class exercises are designed to explicitly facilitate that. Woods and Howard [11] effectively used class exercises for information technology students to study ethical issues. Day and Foley [12] used class time exclusively for exercises, having their students prepare beforehand for class with materials provided online. Frydenberg [13] primarily used hands-on exercises to foster student understanding in data analytics. Based on the context of the class module, class exercises were 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. This project has developed, implemented, and disseminated 16 class exercises [14]. Each class exercise consists of an exercise description, instruction notes, student handout, and assessment instrument.

- **Case Study Videos:** One commonly used technique to enhance the classroom learning experience is the use of video. Videos can reinforce reading and lecture material, help to develop common knowledge, enhance the quality of discussion and overall student comprehension and accommodate students of different learning styles, increasing student motivation and teacher effectiveness [15]. Videos can aid in illustrating highly complex concepts and ideas in a short period of time, provoking meaningful discussion and analysis. Produced from the scripts first drafted by our industry partners and confirmed by the testimonies shared in focus group discussions, each case study videos provides a realistic picture for the audience to appreciate SV&V best processes in practice. These videos show how peer code review is done, and how potential tension or conflict may arise, or the tedious detailed nature of requirements elicitation. This project produced, implemented, and disseminated 4 case study videos [16]. Each case study video consists of the video, a video description, discussion questions, and an assessment instrument.

The ALTs are modularized into flexible modules of 25 delivery minutes each. Instructors have considered their various needs such as curriculum design, class time, and class size to adapt the ALTs to the situations in their institutions. Furthermore, though we recommend it, instructors do not need to practice the flipped classroom model [17], [18] or may do so partially. To summarize, the course delivery plan encourages the following:

- Use the flipped classroom model (if applicable).
- Have students work in small teams (2-3 students per team).
- Deliver tools in one or multiple 25-minute sessions.
- Use pre-test and post-test instruments to tailor course delivery.
- Evaluate student learning of the module immediately after delivery.

The complete list of disseminated ALTs with recommended delivery duration is depicted in Tables 1, 2, and 3.

Table 1: Case Studies

V&V Focus Area	Case Study Modules	Mins.
Requirements Management	Understanding User Requirements	50
	Requirements from a Customer's Perspective	250
Software Configuration Management	Continuous Integration	100
	Version Control Management System	100
Software Peer Reviews	Importance of Reviews	100
	Peer Review Tools	100
Testing	Test Case Development	50
	Performance Testing/ Load Testing	50
	Software Test Plan (STP)	100
Additional Topics	Liability for Bad Software and Support	50
	Software Legal Issues	50
	TOTAL	1000
	Contact hours (50 minute periods)	20

Table 2: Class Exercises

V&V Focus Area	Class Exercise Modules	Mins.
Requirements Management	Ambiguous Questions	25
	Business Requirements and Functional Requirements	50
	Clarifying User Requirements	50
	Needs Statement to SRS	50
	Needs Statements to User Requirements	50
	Requirement Ambiguity	50
	Stated and Implied Requirements	25
Software Configuration Management	Defect Lifecycle	50
Peer Reviews	Code Inspection	150
	Review a given SRS with Checklist	100
Testing	Cost Effective Testing Approach	50
	Test Cases for a Given Requirement	50
	Testing Tools	50
	Understanding Testing	50
Additional Topics	Deming's 14 Points on System of Profound Knowledge (SoPK)	50
	Understanding IEEE Standards	50
	TOTAL	900
	Contact hours (50 minute periods)	18

Table 3: Case Study Videos Details

V&V Focus Area	Case Video Modules	Mins.	# of Scenes
Requirements Management	Requirements Elicitation	100	5
	V&V in Scrum	50	4
Peer Reviews	Code Inspection	100	7
Testing	Testing and Security	50	5
	TOTAL	300	21
	Contact hours (50 minute periods)	6	

These ALTs are available through the project website www.rmu.edu/nsfvv (depicted in Figure 2) and ENSEMBLE, a computing portal connecting computing educators, accessible through www.computingportal.org (depicted in Figure 3). The tools and supporting documents are organized based upon SV&V topics. Folders are provided for activities related to requirements management, software reviews, configuration management, and software testing. Underneath each of these folders are folders for the three categories of ALTs: case studies, class exercises, and case study videos. There is also a folder for topical assessments. For greater and easier availability, the videos have been uploaded to YouTube. Figure 4 depicts a scene of a case study video as seen in YouTube.

Figure 2. Active Learning Tools - Project Website

5. Assessment of Student Learning

The biggest impact this project has had is in undergraduate education specifically Software Engineering students. To ensure the ALTs provided student learning, an assessment and evaluation plan was executed to assess learning outcomes. The focus groups worked with the external evaluators in developing the evaluation questionnaires and instruments for this plan.While in depth discussion including statistical analysis of project assessment and evaluation of all three project outcomes is **being presented in another paper in ASEE 2018** in this paper we provide a summary of Project Outcome 1 which states "*the Project improves knowledge and skills of students*". In-class assessment by faculty and SIR II student surveys were the tools used to assess this outcome.

ENGR3400 Software Verification and Validation is a required for students majoring in the Software Engineering track at the authors institute. Students take this course in the spring semester of their junior year, just in time for applying for summer internships. The student population in the 2014-15 class was 13, 2015-16 class was 21 and 2016-17 class was 29.

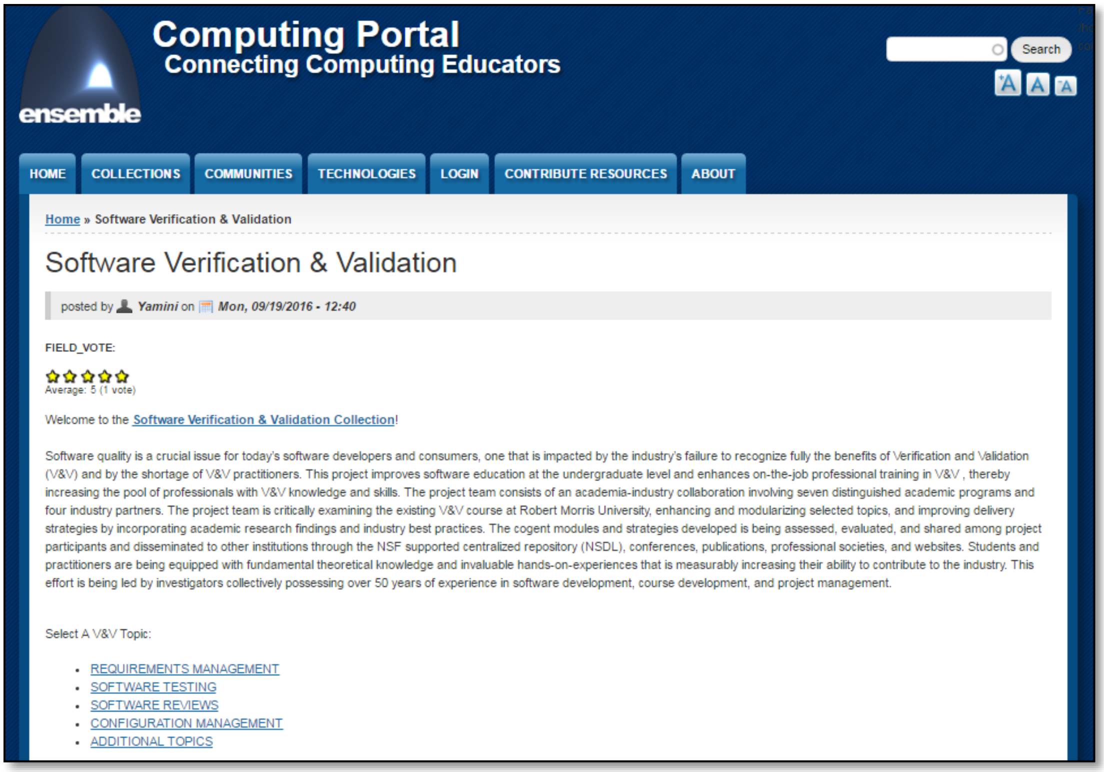

Figure 3. Active Learning Tools - ENSEMBLE

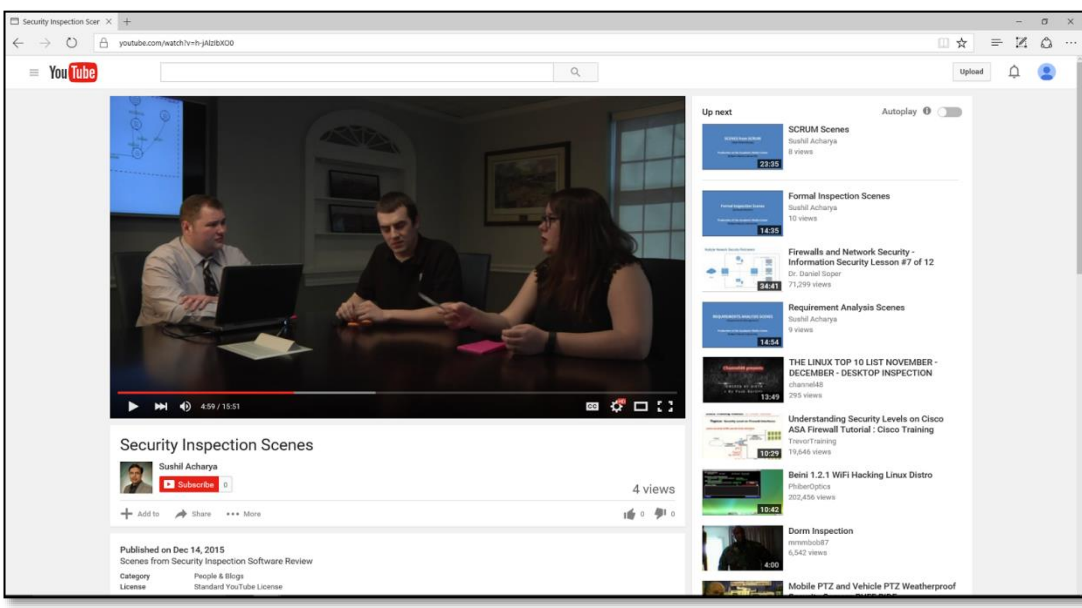

Figure 4. Active Learning Tools – Your Tube

During the project duration the external evaluator was provided with student pre- and post-tests results, student evaluations of specific pedagogical tools, student evaluation of ALTs, and Instructor evaluation.

- Pre and Post Tests were used to assess "to what extent did students learn from and find the newly developed materials to be useful". Basic psychometric analysis of the items on the pre- and post-tests was conducted, and a paired samples ***t*** test was conducted to determine significant differences in the pre- and post-test scores. The data showed improvement from pre-test to post-test. There was a mean difference of approximately one point, with the post-test scores being higher. It should be noted that the sample size for utilizing ***t*** tests and item analysis fluctuated by year, and hence the results from these analyses should keep in mind the limitation of sample size.

- The student feedback about specific pedagogical tools was obtained in the form of open ended questions. Many students provided positive comments about the activity in terms of recognizing the importance of the content, real-world application, collaborative nature of the activity, and hands-one nature of the activity. One student indicated his or her group did not have time to compete the activity, but that was the only negative response provided by the students. Some of the representative comments from the student feedback are summarized below:

  - **Class Exercise: Software Requirements Specification (SRS) Review**
    - *This was helpful in that in provided real hands-on experience doing software req's inspection. This was my very first time performing this activity and it was immensely helpful.*
    - *The activity is a good activity; we practice what we have studied in class.*
    - *This activity provided valuable experience in SRS review similar to what I have experienced in an internship in previous summers.*
    - *Small time constraint, couldn't see all major defects*.
  - **Case Study: Software Legal Issues**
    - *It was good to be reminded that the quality of software or lack of it has legal consequences.*
    - *It was an interesting story, and a good insight to how not applying the principles of SQA can have adverse effect on a company*
    - *I realized that companies can suffer dearly when a customer is not satisfied with their products. Lacking standards and not keeping the contract based on the user's need for the software.*
    - *The article was difficult to understand (I thought it was poorly written) but otherwise it was good.*
  - **Case Study Video: Scenes from Requirements Analysis**
    - *Helped good understanding skills.*
    - *Insightful exercise for understanding the process.*
    - *It did a good job at demonstrating requirement analysis while dealing with a client.*
    - *The video was professional and a real-life situation. Maybe include more group thinking.*

- A student survey tool was used to assess the effectiveness of the active learning tools and to receive feedback for future improvement. The paper surveys were carried out in-class on the last day of class. The students were asked to rate the value of the in-class activities in enhancing their knowledge and learning process experience. 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, class exercises, case study 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. 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.

- One of the authors has been teaching the SV&V course since 2005 and is required to perform an ABET Criterion 3 outcomes assessment. In the spring 2013 term, when the new pedagogical tools were not yet available, there was a weakness associated with the ABET 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. When appropriate Case Studies were developed in 2014 and implemented in 2015, 2016, and 2017, the student performance related to outcome “***e***” rose to the excellent range (>=90%). In 2014-89%, 2015 – 90%, 2016 – 83%, and 2017 – 90%. In 2016 35% of the students were international students resulting in the lower score. English not being their first language led to the lower score. This presented evidence that the Case Study based teaching method is more effective in supporting student learning and increasing the ability to identify, formulate, and solve engineering problems.

  This author also observed positive change in student behavior. Student attendance improved as students were more interested in coming to class to do hands on exercises. While working in smaller groups to work on ALTs students were eager to speak out amongst their peers. Something they would be reluctant to do in a traditional classroom setting. Towards the end of the semester even introvert students had the confidence to raise hands to answer questions.

6. Partnerships in the Project

As Principal Investigators, the authors led the project which involved partnerships in academia and industry. This section discusses the broader impacts on the community as a result of academia industry joint project activities. In addition the authors’ own learning experience is then discussed in Section 7.

6.1. Academic Institutions

The project comprised of two categories of academic partners: *Development Academic Partners* and *Implementation Academic Partners*. These academic partners offered one or more bachelor degrees in Software Engineering, Computer Science, Computer Engineering, and/or Electrical Engineering. These partners also shared strong desire to strengthen their programs. Mobilizing partners was a time consuming and tedious task. Getting academic partners to agree that SV&V is important for software quality and should be delivered to software engineering students took multiple email exchanges, phone calls, and assurance. The composition of the partners and their tasks are described below:

- ***Development Academic Partner***

Two institutions Virginia State University (a HBCU partner) and Milwaukee School of Engineering supported this project as a '**Development Academic Partner**' and carried out the following tasks:

  - Worked with assigned focus groups (described later) to critically review current course.
  - As a focus group member co-developed six hours of course modules to address identified gaps in a content area familiar to the University program and its local industry partner(s).
  - Assessed course contents through at least two delivery cycles.

- ***Implementation Academic Partners***

Six institutions Embry-Riddle Aeronautical University, Montana Tech, University of Michigan, Virginia State University, Fairfield University, and Milwaukee School of Engineering supported this project as an '**Implementation Academic Partner**' and carried out the following tasks:

  - Used entire or partial courseware developed by this project in at least one course through at least two delivery cycles.
  - Evaluated the course(s) and assess the instruction at several levels.
  - Attended a Software Verification and Validation workshop

The project initially proposed to disseminate the developed ALTs to 10 institutions. However by the end of the project duration the ALTs were disseminated to 28 US and 3 international institutions. These institutions are listed in Table 4. The dissemination took place through two workshops, the project website, the NSF-funded Ensemble repository, conferences/publications, and YouTube. The project generated SV&V awareness and created competent SV&V practitioners. Beyond the enhanced SV&V course itself, this project contributed to the development of a SV&V community spanning industry and academia.

Twenty of the institutions listed attended one or both of two Software Verification and Validation workshops held at the authors institute in August 2015 and 2016. In these day-and-a-half workshops participants were introduced to the tools and recommended delivery strategy and were given access to the ALT repository. This strategy eventually led to the formation of a larger implementing group. The instructional strategy focused on student learning. A complete instructor's kit and need-based mentoring sessions (via WebEx, Skype, email, and phone) were provided for instructor mentoring and to encourage other institutions to use and adapt these ALTs for their needs. Broader impact was made by making the ALTs available to a large number of software engineering students through these academic partners. These implementation strategies has brought about a lasting change in the curriculum and course content of some partner institutions and serves as a model of change for other universities and colleges.

Table 4: Universities with whom ALTs were shared

Auburn University, AL	Milwaukee School of Engineering, WI
Baldwin Wallace University, OH	Minnesota State University, MN
Bowie State University, MD	Montana Tech, MT
Clarion University, PA	Mount Mercy University, IA
East Carolina University,	North Carolina A&T State University, NC
Eastern Mediterranean University (Cyprus)	Northwest University, South Africa
Embry-Riddle Aeronautical University, FL	ORT Braude College, Israel
Fairfield University, CT	Rocky Mountain College, MN
Faulkner University, AL	Rose-Hulman Institute of Technology, IN
George Mason University, VA	SUNY Oneonta, NY
Georgia Southern University, GA	University of Alaska Southeast, AK
Grand Valley State University, MI	University of Maryland, MD
Indiana University Southeast, IA	University of Michigan-Dearborn, MI
Kennesaw State University, GA	University of South Carolina Upstate, SC
Kentucky State University, KY	Virginia State University, VA
Kenyon College, OH	Whitworth University, WA

6.2. Industry Partners

The project also had four industry partners that were either large software companies or companies with large software development activities. Eaton Electrical Corporation, ServiceLink, PNC, and JDA Software Group supported this project as Industry Partners. Their areas of expertise are respectively in electrical meters, mortgage, banking, and intelligent pricing and revenue management. These partners carried out the following tasks:

- Critically reviewed and identified knowledge gaps in V&V courseware.
- Assisted in developing course modules.
- Delivered expert lecture sessions to undergraduate students at PI Institute.

In the 3rd and final year ANSYS joined the project team and contributed to reviewing the ALTs. Broader impact was achieved in two fronts. First, by making ALTs available to industry partners employers have been able to train/retrain their employees. Industry partners have informed us that most of the case study videos and some of the case studies have been used to train/retrain their employees. Second the established relationship has assisted with internship and jobs for students from the authors' institute, participation in program visitor boards, and continuing work on new initiatives like new grant proposals and curriculum revisions.

7. Principal Investigators

The principal investigators led the project team to develop, test, and implement 31 ALT modules over the project duration. This experience has broadened their scope of knowledge in industry best practices. Case studies are most certainly typical examples. Class exercises also drill down on relevant understanding regarding the practical issues. The case study videos had the scripts verified with industry partners, but in making the videos, the spotlight often also illuminated on the subtleties of human interaction affecting the proper implementation of software processes in industry.

The ALTs can be incorporated in existing SE, CS, IS, and CE curriculum partially or in its entirety (in the case of a new course, which some institutions are attempting). Delivering the course with the ALTs has also been a good learning experience for the principal investigators. Although the flipped classroom approach is not new, the practice still requires adaptation by the teacher as well as the students. That adaptation includes the effective use of the assessment instruments which come with the ALTs. The recommended flipped classroom delivery strategy of the ALTs is discussed in another paper [19].

Primarily the project work products and research findings have been disseminated through various venues and have been and/or are being used in multiple institutions. To date, conference papers have been presented at ASEE 2014, ASEE 2015, EDSIGCON 2015, ASEE 2016, WMSCI 2016, and ASEE 2017 annual conferences. A keynote address on Software Verification and Validation was also delivered in WMSCI 2016. In 2016 this project was presented at the NSF Showcase at SIGCSE 2016 and in the Envisioning the Future of Undergraduate STEM Education: Research and Practice symposium organized by AAAS in 2016. During the project duration nine conference papers (2014 -1, 2015-3, 2016-3, 2017-2) and seven journal papers (2014 -1, 2015-1, 2016-2, 2017-2, 2018-1) were published. A book on SV&V Case Studies has been published by Alexandria Street Press (online). A workshop was conducted in EDSIGCON 2016.

The joint effort of the partnership greatly expanded the academic and industry network for the principal investigators. The focus groups and the workshops brought people from different institutions and corporations to work closely together. As a result PIs have published multiple conference papers [10, 19, 20, 21, 22, 23], and journal papers [24, 25, 26, 27, 28, 29] with academic partners. In addition the PIs are working on projects of mutual interest with the academia and industry. In 2017 joint grant proposals valued at $1.5 million have been submitted.

8. Summary

A three years NSF TUES grant awarded to the authors institute in 2013 has specifically addressed the need for ALTs and has addressed the NSF broader impact merit criteria. The development/delivery of ALT has transformed the undergraduate software engineering course both in terms of contents coverage and delivery strategy at the authors institute as well as partner institutions. The partnerships between the academia and the industry has been instrumental in developing, and disseminating forty-four delivery contact hours of ALTs – the case studies, the class exercises and the case study videos. The ALTs have been disseminated through multiple workshops attended by academic partner institutions interested in incorporating the tools into their curriculum as well as industry partners interested in using the tools for on-the-job training. Faculty members from multiple institutions have incorporated the ALTs into their curriculum, many have delivered the ALTs using the flipped classroom strategy, and students have provided positive feedback on the contents, the delivery and knowledge retention. For the principal investigators the experience gained has broadened the scope of knowledge in industry best practices. In addition to scholarly publications the project has resulted in academia-academia or academia-industry partnerships that are jointly involved in projects of mutual interest.

And finally below is an email received from a student in his own words. "*When I brought in my Software V&V textbook, the developers thought I was trying to learn testing and documentation practices, not that I had already taken a semester-long class. My boss had never seen that as a required college course before. I've been complimented on my ability to communicate many times when having round table discussions with the development team and the non-technical internal teams as I'm seen as sort of a middle-man in that process. All of those activities you made us do in class really helped me reach where I am now, not just V&V though that is the most applicable to my job. When I came to Robert Morris University I wanted to learn how to program without realizing how much really went in to developing a complete software product. It's now my first job, a passion, and I hope a job for a long time that I can continue to grow in.*" The student started his first job as Team Lead of a new SQA team and the credit goes to the SV&V course he took at the authors' institution. The NSF TUES grant awarded to the authors has made an impact on software V&V education.

Acknowledgement
The authors acknowledge the support of NSF through a grant entitled "Collaborative Education: Building a Skilled V&V Community", NSF-TUES Award # 1245036.

References

[1]. Software Engineering 2014 IEEE/ACM, Software Engineering 2004, *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, a Volume of the Computing Curricula Series, February 23, 2015. Joint Task force on Computing Curricula IEEE Computer Society and Association of Computing Machinery
[2]. The National Science Foundation, "Proposal and Award Policies and Procedures Guide – Part 1 Grant Proposal Guide" https://www.nsf.gov/pubs/policydocs/pappguide/nsf15001/gpg_print.pdf, pp III-2.
[3]. Latest NSF Guidelines on Broader Impacts, Broader Impacts Resource Center, Penn State Eberly College of Science Outreach Office. [Web Page], Retrieved March 12, 2018 from URL ***https://broaderimpacts.psu.edu/nsf-guidelines-and-trends/***
[4]. Promoting Active Learning, [Web Page], Retrieved January 21, 2017 from URL https://utah.instructure.com/courses/148446/pages/active-learning
[5]. Prince, M. (2004). "Does active learning work? A review of the research." *Journal of engineering education* 93.3: 223–231.
[6]. Davis, C., & Wilcock, E. (2003). *Teaching materials using case studies*. UK Center for Materials Education.
[7]. 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
[8]. Raju, P.K. & Sanker, C.S. (1999). "Teaching real-world issues through case studies," *Journal of Engineering Education, 88*(4), pp. 501–508.
[9]. 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
[10]. Manohar, P., Acharya, S., Wu, P., Hansen, M., Ansari, A. & Schilling, W., (2015) *Case Studies for Enhancing Student Engagement and Active Learning in Software V&V*

Education Journal of Education and Learning, ISSN# 1927-5250, Vol. 4, No. 4., Dec. 2015, pp. 39 – 52.

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

[12]. Day, J.A. and Foley, J.D. (2006) Evaluating a web lecture intervention in a human–computer interaction course. IEEE Transactions on Education, 49(4):420–431, 2006.

[13]. Frydenberg, M. (2013). "Flipping Excel," *Information Systems Education Journal, 11*(1) pp.63–73. http://isedj.org/2013-11/ ISSN: 1545-679X

[14]. Wu, P., Manohar, P.A., & Acharya, S. (2016). "The design and evaluation of class exercises for active learning in software verification and validation." *Information Systems and Education Journal*, *14*(4), 4–12.

[15]. Corporation for Public Broadcasting. (2004). Television Goes to School: The Impact of Video on Student Learning in Formal Education.

[16]. Acharya, S., Manohar, P., & Wu, P. (2016) *Using Case Study Videos as an Effective Active Learning Tool to Teach Software Development Best Practices* Journal of Systemics, Cybernetics and Informatics, ISSN# 1690-4524, Vol. 15, No. 3, September 2017, pp 70-75.

[17]. Bonwell, C. C., & Eison, J. A. (1991). *Active Learning; Creating Excitement in the Classroom*. ASHE-ERIC Higher Education Report No. 1, Washington, D.C.: School of Education and Human Development, The George Washington University. Retrieved from http://files.eric.ed.gov/fulltext/ED336049.pdf

[18]. Mason, G.S.; Shuman, T.R.; Cook, K.E. (2013), "Comparing the Effectiveness of an Inverted Classroom to a Traditional Classroom in an Upper-Division Engineering Course, *"IEEE Transactions on Education"*, vol.56, no.4, pp.430,435, Nov. 2013.

[19]. Bruce Maxim, Sushil Acharya, Stein Brunvand, & Marouane Kessentini (2017), "WIP: Introducing Active Learning in a Software Engineering Course", 2017 ASEE Annual Conference in Columbus, OH.

[20]. Sushil Acharya, Priya Manohar, Peter Wu, Ali Ansari, & Walt Schilling (2014) "Collaborative Education: Building a Skilled Software Verification and Validation User Community" 2014 ASEE Annual Conference & Exposition - Computers in Education Division, Indianapolis

[21]. Sushil Acharya, Priya Manohar, Peter Wu, Ali Ansari, & Walt Schilling (2015), "Integrated Active Learning Tools for Enhanced Pedagogy in a Software Engineering Course", 2015 ASEE Annual Conference in Seattle, OR. – (Poster)

[22]. Priya Manohar, Sushil Acharya, Peter Wu, Ali Ansari, & Walt Schilling, (2015) "Case Study Based Educational Tools for Teaching Software V&V course at undergraduate level." 2015 ASEE Annual Conference in Seattle, OR.

[23]. Sushil Acharya, Priya Manohar, Peter Wu, & Walt Schilling (2016), "Enhancing Verification and Validation Education using Active Learning Tools Developed through an Academia-Industry Partnership ", 2016 ASEE Annual Conference in New Orleans, LA.

[24]. Sushil Acharya, Priya Manohar, Peter Wu, Walter Schilling, Ali Ansari (2015), "Collaborative Education: Building a Skilled Software Verification and Validation User Community", Computers in Education Journal, ISSN# 0736-8607, Vol. 5, No. 4, Oct. - Dec. 2014, pp. 26 - 35.

[25]. Priya Manohar, Sushil Acharya, , Peter Wu, Mary Hansen, Ali Ansari, & Walt Schilling, (2015) "Case Studies for Enhancing Student Engagement and Active Learning in Software

V&V Education" Journal of Education and Learning, ISSN# 1927-5250, Vol. 4, No. 4., Dec. 2015, pp. 39 – 52.

[26]. Sushil Acharya, Priya Manohar, Peter Wu, Walt Schilling, & Ali Ansari "Integrated Active Learning Tools for Enhanced Pedagogy in a Software Engineering Course (2016)", Computers in Education Journal, ISSN# 0736-8607, Vol 7, No. 2, Apr. - Jun. 2016, pp. 17 - 28.

[27]. Sushil Acharya, Priya Manohar, Peter Wu, & Walt Schilling, "Using Academia-Industry Partnerships to Enhance Software Verification & Validation Education via Active Learning Tools (2017)" Journal of Education and Learning, ISSN# 1927-5250, Vol. 6, No. 2., Jun. 2017, pp. 69 – 84.

[28]. Sushil Acharya, Priya Manohar, Peter Wu, Bruce Maxim, & Mary Hansen, "Design, Development and Delivery of Active Learning Tools in Software Verification & Validation Education (2017)" Journal of Education and Learning, ISSN# 1927-5250, Vol. 7, No. 1, 2018, pp 13 - 28.