Intellectual Merit and Broader Impact: Collaborative Education toward Building a Skilled Software Verification and Validation Community

Sushil Acharya
acharya@rmu.edu
Department of Engineering

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

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

Robert Morris University
Pittsburgh, PA 15108

Abstract

Software Verification and Validation (SV&V) is proven to be an effective approach to ensure software quality. Yet it is not commonly deployed in industry practices. We started a project intent on building a sustainable community skilled in SV&V. The fundamental objective is the transformation of undergraduate education in software engineering. The project involved collaborative partners in both industry and academia. Through the discussions in focus groups, the curriculum for SV&V was vigorously reviewed, checking against the best practices in industry while identifying and prioritizing gaps. The project went on to develop new active learning tools along with outcomes assessment instruments, designed to enhance delivery and retention of knowledge in SV&V, both theoretical and practical, specifically in the areas of requirements management, software reviews, configuration management and software testing. The project resulted in 44 delivery contact hours of teaching modules using these active learning tools: case studies, class exercises and case study videos. The deliverables of the project have been shared, refined and disseminated through training workshops attended by our academic and industry partners, and are now publicly available online. The paper presents the project, and sums up on how the project achieved the goals of intellectual merit and broader impact, which are the criteria based on which the supporting agency National Science Foundation evaluated the project proposal.

Keywords: Software Engineering Education, Software Verification and Validation, SV&V, Active Learning Tools, Intellectual Merit, Broader Impact.

1. INTRODUCTION

Software Verification and Validation (SV&V) is proven to be effective in ensuring software quality and yet only rarely used in industry (Arthur, Groener, Hayhurst & Holoway, 1999; Wang, Ostroff & Hudon, 2014), we started a project intent on building a sustainable community skilled in SV&V. The goal was aimed at the direction of transforming undergraduate education in software engineering. The project involved partners in industry and academia in collaborative education. Armed with the joint partnership, we vigorously reviewed the SV&V curriculum, checking for gaps against industry best practices.

The knowledge areas listed in the IEEE/ACM (2014) Software Engineering Curriculum Guidelines encompass both theoretical and practical aspects pertinent to SV&V practices in industry. These knowledge areas are essential for undergraduate education and a subsequent professional career in software engineering. The dearth of SV&V practitioners in industry seems to indicate the ineffective pedagogy with regards to these knowledge areas (Arthur, Nance, Joines, Barton, Kang & Fishwick, 2000). We therefore set our goal to create new tools to engage the students in active learning of SV&V. Iterative refinement and re-development of the active learning tools would need the support of a collaborative partnership. Dissemination of the new pedagogy and networking to promote the deployment of the new tools aimed at building a community skilled in SV&V.

Our research proposal was awarded an NSF grant for the TUES (Transforming Undergraduate Education in STEM) program in 2013 to address the SV&V pedagogical issues. The project was funded for three years followed by the approval of one year no-cost extension. Many academic and industry partners were involved at various levels of collaboration and participation. Guided and guarded by the industry and academic partnership, we developed new SV&V learning tools. Through training workshops, we not only iteratively refined and re-developed the learning tools as well as the delivery strategies, we also further disseminated the new learning tools and broadened the partnership to implement the new teaching approach through networking. The new learning tools were first shared with the partners and are now publicly available.

This paper reports our effort in the project and present a summary outline, categorizing the new teaching tools now available. Section 2 presents the overall goal and the objectives, followed by a discussion of the partnerships and their roles involved in the project. Section 3 proceeds on to describe how the partners were organized into focus groups to critically review the existing SV&V curriculum and pedagogical approach. Section 4 explains the active learning tools and how they may engender active learning. The development methodology of the new learning tools is also described. One example from each category of the Active Learning Tools is shared briefly in our discussion. Three tables list all the Active Learning Tools from the project in their appropriate categories. Section 4 closes with discussing the appropriate delivery strategies for the new teaching approach. Section 5 describes the SV&V training workshops to refine and promote the learning tools with an even broader invitation to the partnership. Section 6 presents the two web portals to access the project deliverables – the active learning tools. Sections 7 and 8 sum up the achievements of the project to meet the evaluating criteria of NSF, namely, intellectual merit and broader impact. Section 9 presents a summary of the paper.

2. THE PROJECT AND THE COLLABORATIVE PARTNERSHIPS

The goal of our project was to enhance and transform undergraduate education in SV&V by incorporating academic research and industry best practices through collaborative partnership. The following description lays out the progressive objectives to achieve our project goal.

1. To critically review the existing SV&V course content, checking against best practices.
2. To identify gaps and priorities to indicate areas for improvement in pedagogy.
3. To design and develop new materials and active learning tools.
4. To modularize the active learning tools and integrate them into the SV&V course.
5. To develop appropriate delivery strategies for the active learning tools.
6. To evaluate the SV&V course for pedagogy and to formulate assessment instruments.
7. To disseminate the tools for deployment and feedback through networking.

Academic Partners

The project involved two categories of academic partners: development partner and implementation partner. Two institutions were development partners. They were Virginia State University and Milwaukee School of Engineering. Together with the authors' host institution, they carried out the following tasks.

- Joined in the focus groups to critically review the SV&V curriculum.
- Took part to co-develop new course modules to address the gaps in the course content identified by the focus groups.
- Performed assessment of course contents through at least two delivery cycles.

There were six implementation partners: Embry-Riddle Aeronautical University, Montana Technological University, University of Michigan at Dearborn, Virginia State University, Fairfield University, and Milwaukee School of Engineering. Together with the authors' host institution, they carried out the following tasks.

- Used the entire or parts of the courseware developed by the project in at least one course, through at least two delivery cycles.
- Performed assessment of the instruction to evaluate the course.

Industry Partners
The project involved four industry partners. They were either software companies or companies with large software development activities. Their key areas of business included banking, electrical meters, mortgage, pricing and revenue management. They were PNC Bank, Eaton Electrical Corporation, Service Link Inc. and JDA Software Group. The industry partners took part to carry out the following tasks.

- Helped in the focus groups to critically review the SV&V curriculum, checking with industry practices to identify gaps.
- Assisted in the definition and development of new course materials and tools.
- Delivered industry expert lecture sessions as guest lecturer for the SV&V course at the authors' institution.

3. CRITICAL REVIEW OF THE SV&V CURRICULUM

To critically review the SV&V curriculum, we organized the project partners into focus groups. Since strong academia-industry partnership was critical to the project, each focus group comprised of at least one industry partner and one academic partner as members. Each group was assigned one or more SV&V topics for review and discussion, and was led by the project PI and/or co-PIs. The focus groups met once every year at the authors' institution and twice a year in teleconference through various media, in addition to ad hoc virtual online meetings and discussion groups. The activities facilitated for educators and practitioners to understand one another while sharing their thoughts about the SV&V curriculum under review.

The practice of SV&V is well known in the software industry since the 90's (Pham, 1999). Listed in the knowledge areas of the standard curriculum guidelines, SV&V is an essential part of undergraduate software engineering curriculum (IEEE/ACM, 2014). It encompasses both theoretical and practical aspects of knowledge pertinent to a professional career. The knowledge areas were well defined, but the students were rarely well engaged in class. The common sentiment in the focus groups was that the application values of SV&V education were generally not made sufficiently obvious to the students. SV&V education was not effectively delivered, festered with non-coverage by the instructors or non-retention by the students of key knowledge areas.

To sharpen our focus in the review, the SV&V topics for the groups were organized into the four specific areas of software engineering, listed namely in the following:

- Requirements Management
- Software Review
- Configuration Management
- Software Testing

Instead of the traditional teacher-centric classroom, we needed new materials and tools as SV&V courseware to improve SV&V pedagogy. The new courseware should aim at engaging the students in active learning. The critical review of the focus groups therefore called for new active learning tools to cover SV&V topics in each of the four specific areas of software engineering listed above. Active learning being required for the students, the intended goals of the tools were the following:

- To incorporate both theory and practice into the SV&V topics.
- To preserve a sense of practical value in real applications when working through design and development details.
- To engage the students in interaction, with questions in class to stimulate thinking and discussion.
- To engender familiarity with industry practices and enhance understanding even when undergraduate students often lacked the experience.

The active learning tools to be developed were case studies, class exercises and case study videos. For each of the four areas of software engineering, we developed these new tools. They were intended generally for all the goals stated above, but each type of tool could also be more specific about what it aimed it. Briefly stated, the case study maintained the big picture of a real application while we might get into its details, bringing out the sense of practical value in a real application. The class exercise consisted primarily of discussion questions around a topic. But the questions were designed for stimulation as an invitation to interact. The case study video could engage the viewer in an immersive experience. The next section will discuss each of the active learning tools in further details, and briefly describe the development methodology.

4. THE ACTIVE LEARNING TOOLS (ALTs)

By active learning, we mean tools to build an environment for the teachers and the students to be actively engaged in the course content. They may interact through discussion, problem-solving, critical thinking, debate, or a host of other interactive activities. Active learning requires the student to be doing something other than listening and taking notes (Prince, 2004). In the project, we planned to achieve that by complementing the lecture materials with case studies, class exercises, and case study videos. We called these materials the Active Learning Tools (ALTs).

Case Studies

Case studies are useful tools to teach applications of science and engineering principles. They are effective to contextualize theoretical concepts (Davis & Wilcock, 2003). Many studies also showed the benefits of interactive learning strategy in case studies, shifting the emphasis from teacher-centered to more student-centered activities (Grant, 1997; Raju & Sankar, 1999; Sivan, Wong, Woon & Kembler, 2001). The case studies in our project were primarily drawn from present industry SV&V practices. Students were provided industry standard documents for review to prepare themselves for their tasks. These would involve resolution of review conflicts in the Software Requirements Specification document, or compliance to security standards, or drafting of testing plans from use cases. Our project developed, implemented, and disseminated 12 case studies (Manohar, Acharya, Wu, Hansen, Ansari & Schilling, 2015). Each case study included the case study description, instruction notes, student handout, and assessment instrument.

To briefly share one of the Case Studies, we take an example under Requirements Management. In Module RM17, the fictitious Handsome, Inc. is a company that sells men's clothing and wishes to build its first web site to sell online. While the case study provides the situation for students to solicit user requirements, the learning objective is about identifying and resolving ambiguities in the requirements statements. The supposedly real situation becomes more engaging to the students and provides the context to learn the principles behind the need for requirements to be unambiguous. More in-depth discussion of the case studies is presented in Manohar, et al (2015). Table 1 below lists the entire collection of Case Study Modules in the project, and they are all accessible at the courseware repository discussed in Section 6.

SV&V Area	Case Study	mins
Requirements Management	Understanding User Requirements	50
	Requirements from a Customer's Perspective	250
Configuration Management	Continuous Integration	100
	Version Control Management System	100
Software Reviews	Importance of Reviews	100
	Peer Review Tools	100
Software 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 (in 50 min periods)	20

Table 1. Case Study Modules

Class Exercises

Class exercises provide activity during class time to explicitly raise questions that invite student participation. 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 to prepare beforehand for class with materials provided online. Frydenberg (2013) primarily used hands-on exercises to foster student understanding in data analytics. Based on the context of the class module, class exercises may involve questions to think further into the concepts for a deeper understanding, or to apply 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 an instrument to lead to group projects. Our project developed, implemented, and disseminated 16 class exercises (Wu, Manohar, & Acharya, 2016). Each class exercise consists of the exercise description, instruction notes, student handout, and assessment instrument.

An example for discussion may be Module RM03, under Requirements Management. The learning objective is in discerning between business requirements and functional requirements. It is a communication skill too often students training in technical subjects lack. The class exercise leads the students to go through a list of requirements statements and discuss whether each one is a business requirement statement or a functional requirement statement. The students are expected to have prepared themselves studying the textbook definitions of the two different requirement statements. But even if some are not very thorough in their studying, the in class exercise tends to engage them to want to refer back to think deeper into what they have studied. A detailed evaluation of the Class Exercises is presented in Wu, Manohar and Acharya (2016). Table 2 lists all the Class Exercise Modules in the project, and they are all publicly accessible at the courseware repositories discussed in Section 6.

Case Study Videos

Teachers quite often use videos to enhance the classroom learning experience. Video in general is not interactive. It may not be considered student-centric. But video, if designed right and put together well, can be extremely engaging. The media of sight and sound together with a good narrative or story line can create an immersive experience for the viewer. Students can use video to reinforce reading and studying of lecture materials, or to understand and follow instructions watching demonstration. To an entire class, watching video together can help the class to share a common basis of knowledge and that may enhance the quality of discussion and overall student comprehension. Videos can aid in illustrating highly complex concepts and ideas in a short amount of time, provoking meaningful discussion as well as analysis (Saltrick, Honey & Pasnik, 2004). In the project, we used case study videos primarily to provide a realistic way to experience SV&V best practices in industry, even personally. Produced from the scripts first drafted by our industry partners and confirmed by the testimonies shared in focus group discussions, each case study video portrayed a realistic picture for the audience to appreciate the process of SV&V best practice. For example, the video on peer code review showed also how potential tension or conflict might arise in the human interaction. When viewing the video on requirements elicitation, the viewer might gather the tedious and detailed nature of the work and feel it more personally. Figure 3 below is a scene captured from the Case Study Video on Security Inspection. Our project produced, implemented, and disseminated 4 case study videos (Acharya, Manohar & Wu, 2017). Each case study video consists of the digital video, the video description, discussion questions, and an assessment instrument.

SV&V Area	Class Exercise	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
Configuration Management	Defect Lifecycle	50
Software Reviews	Code Inspection	150
	Review a given SRS with Checklist	100
Software 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 (in 50 min periods)	18

Table 2. Class Exercise Modules

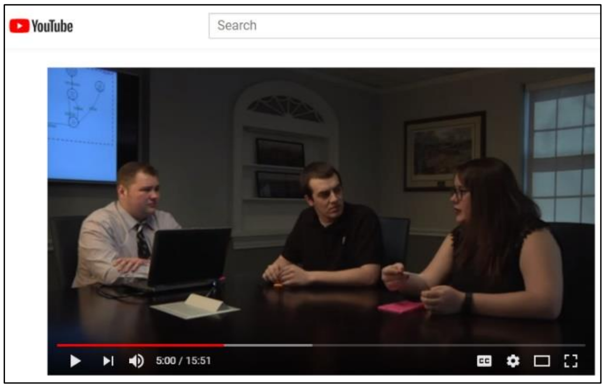

Figure 3. Security Inspection Scene

Table 4 lists all the Case Study Videos produced by the project. The videos are posted to YouTube for streaming. The hyperlinks to play the videos are accessible from the courseware repository discussed in Section 6.

SV&V Area	Case Study Video	mins	# of Scenes
Requirements Management	Requirements Elicitation	100	5
	V&V in Scrum	50	4
Software Reviews	Code Inspection	100	7
Software Testing	Testing and Security	50	5
	TOTAL	300	21
	Contact hours (in 50 min periods)	6	

Table 4. Case Study Videos

Development Methodology
The ALTs were meant to address the gaps in the SV&V curriculum identified in our critical review by the focus groups. While the authors led the development of the ALTs, we also acquired the help of the project partners to incorporate academic research and industry best practices into our effort. We started with assessing the current academic offerings as well as the industry requirements. Our gap analysis would identify the knowledge areas where the inadequacies would be addressed in the ALTs. We applied the Analysis-Design-Development-Implementation-Evaluation (ADDIE) instruction design framework to iteratively assess the course content and the delivery for further revision and improvement (Morrison, 2010). Figure 5 depicts the iterative ADDIE framework applied in our development methodology, with the key activities of review performed by our academic-industry partners in the focus groups. An English Language Editor edited the final products prior to dissemination.

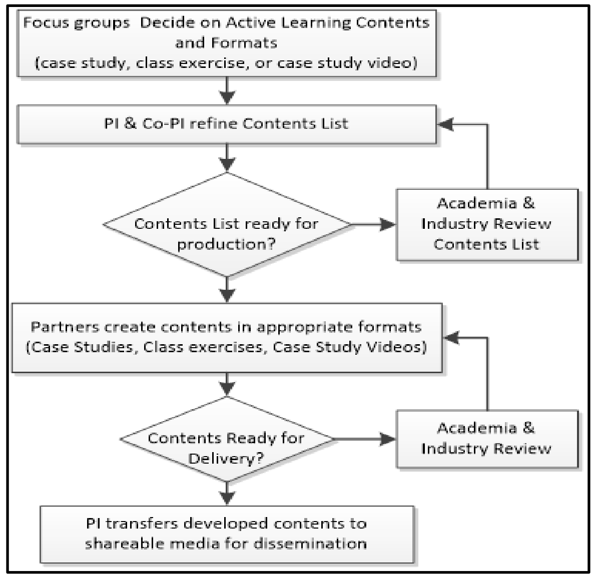

Figure 5. Applying the ADDIE framework

Using the ADDIE methodology the project team produced and disseminated the ALTs for 44 delivery contact hours of SV&V courseware. The ALTs were modularized into small modules of 25 delivery minutes each, for easy adaptability.

Delivery Model
The ALTs were designed to engage the students, to impart practical knowledge into theoretical understanding. Learning still largely depends on the students' knowledge retention. The classroom delivery of the ALTs would create the setting for the students in retention activities, such as group discussion, further studying and deeper thinking in the assignments and team projects (Mishra, Hacaloglu, & Mishra, 2014). It is important to identify and incorporate the delivery strategies to meet the learning outcomes for the ALT modules.

We used a flipped classroom model (Bonwell & Eison, 1991) which allowed us to maximize utility of the class time to engage the students and incite further activities in knowledge retention. Students were expected to be prepared prior to class time and outside the classroom. There were assigned textbook readings or reviewing of lecture materials online. For effective delivery we also recommended the students to work in small teams. Overall, the flipped classroom model has proven highly effective at increasing student engagement and enhancing the preparation of students for class sessions (Day & Foley, 2006). The flipped classroom also has been shown to allow the instructor to cover more material and results in higher student performance (Mason, Shuman & Cook, 2013).

Different ALTs engaged the students in different ways. The Case Studies were explicit in the approach: each Case Study made the point to consider issues in realistic practices. Instructors presented the Case Study while guiding students into further study and discussion of the practical issues in SV&V. The Class Exercises were designed for interaction in the classroom. The instructor would bring up the question(s) and serve as a moderator to guide the discussion. The instructor might also use the Class Exercise to lead students into subsequent group or individual projects. The "Instructor Notes" component of the Class Exercise covered some of these possibilities. The Case Study Videos, by nature as multimedia, were highly engaging. The videos shared real-life perspectives of actions and their consequences. The videos by design were in sequences of scenes. For instructional purposes, we found it highly beneficial to pause the video at appropriate moments to engage the class in discussion on the spot.

To adapt to the situations in different institutions including on-the-job training in industry, we modularized the ALTs into modules of 25 delivery minutes each. Instructors may consider the various needs of curriculum design, class size and class time to adjust their delivery strategies. Although we recommended it, the flipped classroom model is not imperative. Instructors may also choose to only adopt that partially. In summary, the following are our recommendations for the delivery strategies for the ALTs.

- Use the flipped classroom model, if applicable.
- Have students work in small teams of two or three each team.
- Deliver the ALTs in sessions of one or multiple modules.
- Apply the assessment instrument to evaluate learning outcomes immediately after each session.

5. TRAINING WORKSHOPS

During the second and third years of the project, we organized two SV&V Training Workshops to disseminate and promote the use of the developed ALTs. In the one-and-a-half day workshop, we introduced the ALTs to the attendants, shared the delivery model, and chose to demonstrate several of the ALT modules followed by feedback and discussion.

We held the workshops in the authors' institution and invited not only our implementation partners but also many other institutions and industry partners to attend. The attendants were granted access to the ALTs in our repository and everyone was provided with a complete instructor's kit. We strongly encouraged consideration to implement them in their home institutions, offering post-workshop assistance to them in many ways. We gained not only much valuable feedback, but also a much larger group of implementation partners. We went into much collaborative activities with some of the partners and were much gratified when ended up seeing lasting changes in the curriculum and course contents in the partner institutions.

Institutions shared the ALTs with	
1	Auburn University, AL
2	Baldwin Wallace University, OH
3	Bowie State University, MD
4	Clarion University, PA
5	East Carolina University, NC
6	Eastern Mediterranean University, Cyprus
7	Embry-Riddle Aeronautical University, FL
8	Fairfield University, CT
9	Faulkner University, AL
10	George Mason University, VA
11	Georgia Southern University, GA
12	Grand Valley State University, MI
13	Indiana University Southeast, IA
14	Kennesaw State University, GA
15	Kentucky State University, KY
16	Kenyon College, OH
17	Milwaukee School of Engineering, WI
18	Minnesota State University, MN
19	Montana Tech, MT
20	Mount Mercy University, IA
21	North Carolina A&T State University, NC
22	Northwest University, South Africa
23	ORT Braude College, Israel
24	Rocky Mountain College, MN
25	Rose-Hulman, IN
26	SUNY Oneonta, NY
27	University of Alaska Southeast, AK
18	University of Maryland, MD
29	University of Michigan-Dearborn, MI
30	University of South Carolina Upstate, SC
31	Virginia State University, VA
32	Whitworth University, WA

Table 6. Shared ALTs with these Universities

The two training workshops were held in August, of 2015 and 2016. Other abridged versions of the workshop were also held in the following years at some other conferences. Table 6 above lists the institutions we shared the ALTs with, and 20 of these institutions attended at least one of the two training workshops. We offer the information to share the level of our effort toward meeting the broader impacts requirements of an NSF funded project.

6. COURSEWARE REPOSITORY

We initially used the Dropbox as the central repository to share the courseware products, i.e., the ALTs. Now the ALTs are made available for public access on the web. There are two web portals. One is the project web site administered by the authors' institution (www.rmu.edu/nsfvv). The home page is depicted in Figure 7. The other is the web portal to connect computing educators administered by Ensemble, a pathway project funded by National Science Foundation for the National Science Digital Library of computing education resources (www.computingportal.org / softwareverficiationvalidation). Figure 8 depicts the web portal at Ensemble.

Figure 7. Repository home page at project web site

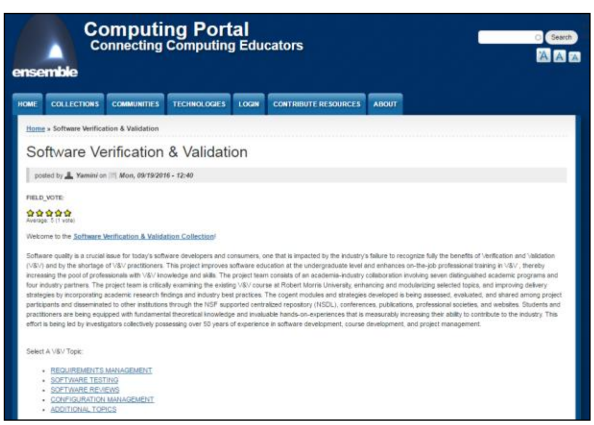

Figure 8. Repository web portal at Ensemble

At the web portals, the ALTs and the supporting documents are organized by the SV&V topic each pertains to. The topics are: Requirements Management, Software Reviews, Configuration Management, Software Testing, and an Additional Topics. Underneath each topic, there are the 3 categories of the ALTs: Case Studies, Class Exercises, Case Study Videos. The ALTs are kept there and available for download, except for the Case Study Videos. The videos are posted to YouTube for streaming, accessible via a hyperlink to play. Figure 3 above shows a scene of a Case Study Video streaming on YouTube.

7. INTELLECTUAL MERIT

The project team developed, tested, implemented and disseminated 31 ALT modules for 44 delivery contact hours over the project duration as described in sections 3, 4, 5 & 6. These ALTs can be readily incorporated in existing SE, CS, IS and CE curricula partially or in its entirety. In the case of a new course in SV&V, it was incorporated entirely. The research findings regarding the effectiveness of the ALTs have been disseminated through conferences and journal publications. To date, the research results have been presented at ASEE 2014, ASEE 2015, EDSIGCON 2015, ASEE 2016, WMSCI 2016, ASEE 2017 and ASEE 2018 annual conferences. A keynote address on Software Verification and Validation was 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. In the duration of the project from 2014 to 2018, twelve conference papers and seven journal papers were published. A book on SV&V Case Studies has been published by the Alexandria Street Press (online), and a workshop using the ALTs was conducted in EDSIGCON 2016.

8. BROADER IMPACT

Originally the project proposed to disseminate the developed ALTs to ten other institutions. As of date the ALTs have been shared among all project partners and disseminated to 30 US institutions and 3 international institutions as described in sections 5 & 6. The dissemination took place through training workshops, scholarly research publications as well as sharing of tools on the web. Two websites serve as repository for the project deliverables: the NSF-funded Ensemble repository and our own hosting institution, supported with streaming via YouTube. The ALTs are also readily usable for on-the-job training in industry. The project generated SV&V awareness and planted the growth of 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.

9. SUMMARY

We reported on the effort of our NSF funded project in the TUES program to transform undergraduate education in STEM. Motivated by the scarcity of SV&V practice in the software industry even when it was proven to be effective to ensure software quality. We brought in the support of collaborative partnership of academia as well as industry. The partnership formed focus groups to critically review the existing SV&V curriculum. We then proceeded to develop new Active Learning Tools (ALTs) for a new teaching approach for SV&V. The focus groups helped to refine the ALTs through iterative re-development. The ALTs introduced a new pedagogy. Through training workshops to share and promote the ALTs, we also gained feedback to improve them and our delivery strategies as well. Since we invited more to join the workshops, we broadened the partnership to deploy the new ALTs. In some cases we began to observe lasting changes in their SV&V curriculum and course contents. The continued dissemination of the new pedagogy we hope will result in more intellectual merit, broader impact and build a sustained community skilled in SV&V.

10. ACKNOWLEDGEMENTS

The project was supported by National Science Foundation through the NSF-TUES Award, Grant# 1245036. The authors gratefully acknowledge this support. We also want to express our deepest appreciation to all our project partners, in academia and in industry, for their enthusiastic support and collaboration. It was a most rewarding learning experience for us.

11. REFERENCES

Acharya, S., P.A. Manohar, & P. Wu (2017) Using Case Study Videos as an Effective Active Learning Tool to Teach Software Development Best Practices, *Journal of Systemics, Cybernetics & Informatics 15*(3), 70-75. ISSN#1690-4524.

Arthur, J.D., M.K. Groener, K.J. Hayhurst, & C.M. Holoway (1999). Evaluating the Effectiveness of Independent Verification and Validation. *IEEE Computer 32*(10), 79-83.

Arthur, J.D., R.E. Nance, J.A. Joines, R.R. Barton, K. Kang, & P.A. Fishwick (2000). Verification and Validation without Independence: a Recipe for Failure. Proceedings of the 2000 Winter Simulation Conference, pp.859-865.

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

Davis, C. & E. Wilcock (2003). *Teaching Materials Using Case Studies*. UK Centre for Materials Education.

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

Frydenberg, M. (2013). Flipping Excel. *Information Systems Education Journal 11*(1) 63–73. ISSN#1545-679X.

Grant, R. (1997). A Claim for The Case Method in The Teaching of Geography. *Journal of Geography in Higher Education 21*(2), 171–185.

IEEE/ACM (2014). Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *IEEE/ACM Software Engineering 2014.* The Computing Curricula Series, by the Joint Task Force on Computing Curricula of IEEE Computer Society and Association of Computing Machinery. Feb 23, 2015.

Manohar, P.A., S. Acharya, P. Wu, M. Hansen, A. Ansari, & W. Schilling (2015). Case Studies for Enhancing Student Engagement and Active Learning in Software V&V Education. *Journal of Education and Learning 4*(4), 39-52. ISSN#1927-5250.

Mason, G.S., T.R. Shuman, & K.E. Cook (2013). Comparing the Effectiveness of an Inverted Classroom to a Traditional Classroom in an Upper-Division Engineering Course. *IEEE Transactions on Education 56*(4), 430,435.

Mishra, D., T. Hacaloglu, & A. Mishra (2014). Teaching Software Verification and Validation Course: A Case Study. *International Journal of Engineering Education 30*(6A), 1476-1485.

Morrison, G.R. (2010). *Designing Effective Instruction* (6th edition). John Wiley & Sons.

Pham, H. (1999). Software Reliability. John Wiley & Sons, Inc. p. 567. ISBN#981308384.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education, 93*(3), 223–231.

Raju, P.K., & C.S. Sanker. (1999). Teaching Real-World Issues Through Case Studies. *Journal of Engineering Education 88*(4), pp. 501–508.

Saltrick, S., M. Honey, & S. Pasnik (2004). Television Goes to School: The Impact of Video on Student Learning in Formal Education. Center for Children & Technology, Education Department, Corporation for Public Broadcasting.

Sivan, A., L.R. Wong, C. Woon, & D. Kembler. (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.

Wang, C-W., J.S. Ostroff, & S. Hudon (2014). Precise Documentation and Validation of Requirements. *Formal Techniques for Safety-Critical Systems*. (Artho & Olveczky, eds). Springer. ISBN#9783319054162.

Woods, D., & E. Howard (2014). An Active Learning Activity for an IT Ethics Course. *Information Systems Education Journal 12*(1) 73–77. ISSN#1545-679X.

Wu, P., P.A. Manohar, & S. Acharya (2016). The Design and Evaluation of Class Exercises for Active Learning in Software Verification and Validation. *Information Systems Education Journal 14*(4), 4–12. ISSN#1545-679X.