

Information Systems Verification and Validation: Three Perspectives: Technical, Systemic, and Philosophical (Participatory Panel) “Active Learning Tools”

Sushil ACHARYA
Engineering Department, Robert Morris University,
Moon Township, PA 15108, USA

ABSTRACT

Software quality is a crucial issue in software development. As software has become ubiquitous, software products have become critical. Software quality issues poses a problem in the software industry, as there is generally a lack of knowledge of Software Verification and Validation (V&V) benefits and a shortage of adequately trained V&V practitioners. The fundamental challenge towards a solution to improve software quality lies in the people and processes that develop and produce software. The industry desires new hires to know software development best practices so as to be able to perform from day 1. This means new hires are expected to know software processes, methods, and tools. This is where the academia needs to step in, especially those that focus on applied teaching. The academia has to develop necessary course modules and redesign their curriculum to provide graduating students the applied knowledge they need so as to be competitive in the job market.

Through a project funded by the National Science Foundation, the author's team has developed (42) delivery hours of Software V&V course modules. This development activity has embraced academia, industry partnership. These tools have been successfully disseminated to over 24 universities with many CS, IS, SE programs incorporating the tools in their existing courses and others designing new courses based on these tools. The tool is available free of cost to interested academia and industry.

Keywords: Software Verification and Validation, Academia Industry Partnership, Active Learning Tools,

1. INTRODUCTION

Software quality is a crucial issue in software development. As software has become ubiquitous, software products have become critical. With almost half of the world's population (3.4 billion) relying on the internet, it has become nearly impossible to avoid this software footprint in an average day to day life [1]. In 2015, major software glitches affected numerous companies: Bloomberg (inoperable trading terminals), Royal Bank of Scotland (payments failure), Nissan (airbag sensory detectors malfunction), Starbucks (register malfunction) and the F 35 Joint Strike Fighter (targets detection failure). Each of these resulted in financial, brand, and more important bodily damages [2]. With mission critical and high-risk applications that have human lives and resources dependent on software applications, it is imperative to not only test for, but more important to aim for zero defects. However, even after decades of development, the software industry continues to spend considerable time and

resources dealing with the quality problem. In the US alone in 2007, the cost of failed software was estimated to increase \$75 billion in re-work costs and abandoned systems [3]. We need software to be effective and efficient. Failure to develop effective and efficient software gives software practitioners and software industry bad rapport.

Software quality issues poses a problem in the software industry, as there is generally a lack of knowledge of Software Verification and Validation (V&V) benefits and a shortage of adequately trained V&V practitioners. Verification is related to building the product right and verification means building the right product. Verification deals with determining whether or not the products of a given software development phase fulfills the requirements established during the previous phase. Validation deals with system performing to meet customer's expectations under all operational conditions. Both of these are important to build quality software.

The fundamental challenge towards a solution to improve software quality lies in the people and processes that develop and produce software. Acharya et al (2014) [4] reasons that firstly, there is not enough awareness of the Software Verification & Validation (SV&V) benefits, and secondly, there are a lack of practitioners who understand the SV&V topics and processes adequately. Both the lack of awareness and personnel shortage considerably hinder significant progress in project success rates. Furthermore, Acharya et al [4] argue the root cause to be the lack of up-to-date V&V courseware. To address this situation, a SV&V course has been improved at the authors' institution through a project funded by a National Science Foundation –Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics (NSF-TUES) grant.

2. WHAT THE INDUSTRY NEEDS?

The industry desires new hires to know software development best practices so as to be able to perform from day 1. This means new hires are expected to know software processes, methods, and tools. However the dilemma is where will new hires obtain such knowledge? After the IT burst in early 2000 software companies have significantly reduced training budgets for new hires. This means the industry may not train new hires at all or provide them minimal training. This is where the academia has to step in, especially those that focus on applied teaching. The academia has to develop necessary course modules and redesign their curriculum to provide graduating students the applied knowledge they need so as to be competitive in the job market.

Also it is important to understand the software field unlike physics and chemistry is a two way street. The industry and the academia have to work hand in hand. Unlike other fields where research carried out by the academia is used by the industry in the case of software academic findings have to be shared with the industry and industry best practices have to be brought in to the classroom.

This will require strong academia-industry partnership and respect for each other. This project is carried out through such a partnership. The team involves three academic development partners, five industry development partners and twelve academic implementing partners. The development partners are responsible for course module development and the implementing partners are responsible for delivering one or more learning modules and providing student feedback so as to further refine the modules.

3. ACTIVE LEARNING

Effective teaching requires effective teaching tools. Engineering education must strike a balance between the knowledge of theoretical concepts and developing an ability to apply the theory to solve real world problems. Such a balance between theory and practice requires careful handling of two types of knowledge called *episteme* by Aristotle (meaning theoretical knowledge) and *phronesis* referring to practical knowledge [5]. It is the practical knowledge that is interesting to the student and immediately useful to the community. However, it has been realized that the practical knowledge cannot be easily taught in a class room setting as it requires lots of time and experience. In engineering education, student-centered lectures have been the predominant model of teaching. However, this may not be the most effective method for imparting knowledge in all disciplines, as students may not be able to retain and apply knowledge they have gained to the extent that is required in their professional careers.

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” [6]. Prince [7] defines active learning as a classroom activity that requires students to do something other than listen and take notes. The author suggests that active learning is achievable by complementing lecture materials with teaching tools in the form of case studies, class exercises and case study videos and by using the flipped classroom delivery model to conduct classes.

4. ACTIVE LEARNING TOOLS

Active learning/teaching tools complement lectures and make class delivery more interesting to the learners. More importantly such tools effectively assist the student in retaining knowledge. Active learning teaching tools such as case studies, class exercises, and case study videos have been utilized in a variety of teaching disciplines, including Biology, Medicine, Law, and Business. It is proposed in this work that these interactive pedagogical tools are especially important for software education as well, specifically for software verification and validation courses, where graduates are expected to develop

software that meets rigorous quality standards both in functional and application domains. The software V&V professionals must interact with other software professionals such as developers, and with customers for requirements elicitation. They must then contribute to develop a project proposal that can subsequently be turned into a legal binding contract. The accuracy and reliability of the software product are usually the technical performance measures in which customers are keenly interested. Therefore, case-study based education and videos of software development scenarios are expected to enrich and enhance undergraduate education in software V&V.

Through this project eighteen (18) delivery hours of case studies, sixteen (16) delivery hours of exercises, and six (6) delivery hours of case study videos totaling forty two (42) delivery hours of Software V&V course modules have been developed. These tools have been successfully disseminated to over 24 universities with many CS, IS, SE programs incorporating the tools in their existing courses and others designing new courses based on these tools.

An iterative development methodology depicted in Figure 1 was used to ensure the modules reflected both academic research and industry best practices. The content development process began with a meeting of the focus groups at the author’s institution. The groups drafted a list of active learning content topics and delivery formats. The list was reviewed by the Principal Investigator (PI) and co-PIs and shared with the partners for further review. The finalized list was then used to guide the development process. In this methodology, an industry partner or academic partner led the development effort through a collaborative effort. Once the contents were ready for review, they were shared with focus group members and subsequently with all partners. The finalized contents were then transferred to a shareable media where they became available for delivery, further reviews, and dissemination. For ease of adaptation, each active learning tool is of 25 minutes duration, with some active learning tools having multiple parts delivered in multiple sessions.

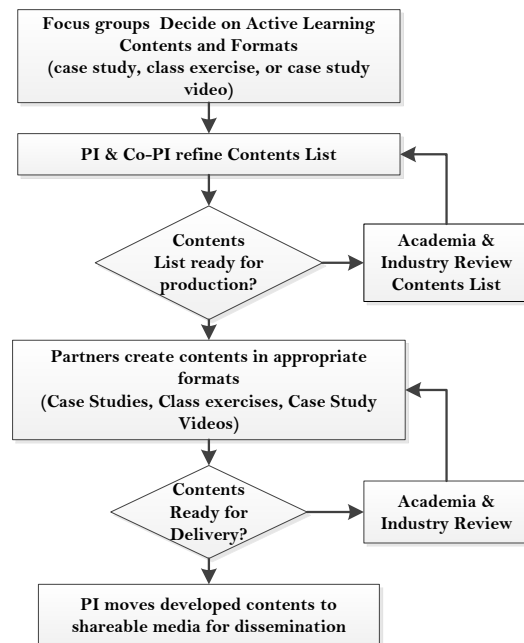


Figure 1: Development Methodology

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

- Requirements Engineering
- Software Testing
- Software Reviews
- Configuration Management

We identified these as the critical areas in the software development 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 developed 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. A more extensive coverage of the study cases developed is being disseminated in another publication [8]. In one case study requirements ambiguity is addressed. Students are asked to design a website for a brick and mortar company. These requirements are used to design a home page and a transaction page by the original authors of the requirements and another team. The difference in the designs is studied to understand why ambiguity free requirements are desired.

• 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. A more extensive coverage of the class exercises developed is being disseminated in another publication [9]. In one exercise students are required to go through a formal inspection meeting incorporating review best practices. Students understand the importance of preparation time, respect for team members, respect for time, and the importance of the task at hand.

• Case Study Videos

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 best 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 elicitation. In one video students are able to view scenes dramatizing the requirements elicitation process and understanding the dos and don'ts in requirements gathering.

5. DISSEMINATION OF COURSE MODULES

The developed course modules have been delivered multiple times in the author's institution. Implementing partners have

also delivered one or more module and provided student feedback. The assessment and evaluation of these tools have been helpful in fine tuning the tools. The course modules is available for free for both the academia and the industry.

The active learning tools 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 tools related to Configuration Management, Requirements Management, Software Reviews, and Software Testing. Underneath each of these folders are folders for the active learning tools: Case Studies, Class Exercises, Case Study Videos, and Topical Assessments. For greater availability, the videos have been uploaded to YouTube.



Figure 2: Active learning tools in project website

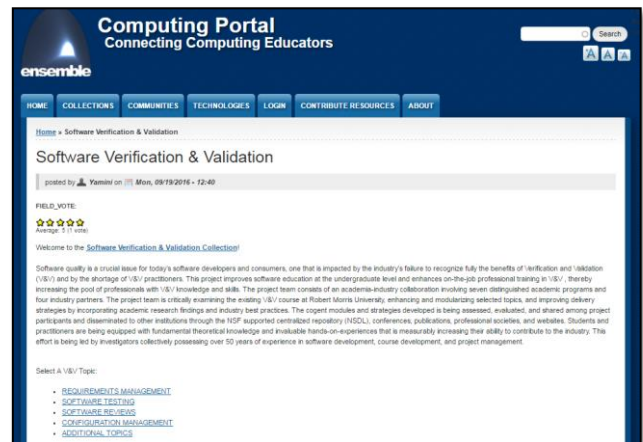


Figure 3: Active learning tools in ENSEMBLE

6. CONCLUSION

Software quality is a crucial issue in software development. Lack of SV&V awareness and the shortage of SV&V practitioners poses multitude of problems in the software industry. New hires need to be versed in software development best practices to be competitive in the job market. An academia industry partnership is crucial in developing course modules that engages students in industry best practices Through a vibrant academia-industry partnership and academic research, a project funded by a NSF-TUES grant has developed, delivered,

and disseminated 42 delivery hours of active learning tools which include Case Studies, Class Exercises, and Video Case Studies in specific SV&V topics viz. *requirements engineering*, *configuration management*, *software reviews*, and *software testing*. These tools can be accessed freely by interested software instructor or practitioner through the project websites.

7. ACKNOWLEDGEMENT

The author acknowledges the support of NSF through a grant entitled “Collaborative Education: Building a Skilled V&V Community”, NSF-TUES Award # 1245036. The author also acknowledges the support of project co-PIs Priyadarshan Manohar and Peter Wu, both from the author’s institution.

8. REFERENCES

- [1] Internet World Stats – Usage and Population Statistics, [Web Page], Retrieved January 15, 2016 from URL <http://www.internetworldstats.com/stats.htm>
- [2] Jee, C., Top software failures 2015/2016: Amazon, RBS, Starbucks - the worst software glitches this year, [Web Page], Retrieved January 15, 2016 from URL <http://www.computerworlduk.com/galleries/infrastructure/top-10-software-failures-of-2014-3599618/>
- [3] Michaels, P., Calculating the cost of failed software projects, [Web Page], Retrieved January 15, 2016 from URL <http://www.computerweekly.com/search/query?q=Calculating+the+cost+of+failed+software+projects>
- [4] 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.
- [5] Bertha C.: “How to teach engineering ethics course with case studies”, proc. ASEE Annual Conference, 2010.
- [6] Retrieved February 20, 2016 from URL <https://utah.instructure.com/courses/148446/pages/active-learning>
- [7] Prince, M., “Does Active Learning Work? A Review of the Research,” Journal of Engineering Education, Vol. 93, 2004, pp. 223-231.
- [8] Manohar, P., Acharya S., Wu P.Y., Hansen M, Ansari A.A. & Schilling Jr W.W. (2015), Case Studies for Enhancing Student Engagement and Active Learning in Software V&V Education, Journal of Education and Learning, Vol 4, No. 4, December 2015, Page, 39-52.
- [9] Wu P., Manohar, P., & Acharya S., Class Exercises for Active Learning in Software Verification and Validation, 2015 Proceedings of the Conference on Information Systems and Computing Education Wilmington, North Carolina USA.