2018 ACM/IEEE 40th International Conference on Software Engineering: Software Engineering Education and Training

Toward Enhancing the Training of Software Engineering Students and Professionals Using Active Video Watching

Matthias Galster
University of Canterbury
Christchurch, New Zealand
mgalster@ieee.org

Antonija Mitrovic
University of Canterbury
Christchurch, New Zealand
tanja.mitrovic @canterbury.ac.nz

Matthew Gordon
University of Canterbury
Christchurch, New Zealand
mjg208@uclive.ac.nz

ABSTRACT

Soft skills (e.g., the ability to communicate, negotiate, collaborate) are essential for software engineers. However, teaching soft skills in a conventional classroom setting is time-consuming and resource-intensive. In this paper, we present and motivate the idea of adopting active video watching (AVW) for soft skills training in software engineering education. Also, we discuss how AVW can contribute to formal learning (e.g., at universities), but also to professional development (e.g., in software development organizations). Furthermore, we present AVW-Space, a web-based AVW platform. Finally, we report our experience from using AVW in a software engineering project course to teach university students presentation skills as one example of soft skills essential for practicing software engineers.

CCS CONCEPTS

• **Social and professional topics** → Computing education;
• **Software and its engineering**

KEYWORDS

Active video watching, soft skills, students, professionals

ACM Reference format:

M. Galster, A. Mitrovic, and M. Gordon. 2018. Toward Enhancing the Training of Software Engineering Students and Professionals Using Active Video Watching. In *Proceedings of 40th International Conference on Software Engineering: Software Engineering Education and Training Track, Gothenburg, Sweden, May 27-June 3 2018 (ICSE-SEET'18)*, 4 pages.
https://doi.org/10.1145/3183377.3183384

1 INTRODUCTION

Future software professionals need a rich and new mix of competences, which current professionals are often lacking [8]. These competences differ for various roles in software engineering (e.g., architects, developers, testers) and software engineering activities (e.g., requirements elicitation, test case identification, bug fixing), but also based on the "type" of skills (e.g., technical skills and non-technical soft skills). Soft skills, i.e., skills that enable a software engineer to interact with others effectively and efficiently (e.g., the ability to communicate, negotiate, collaborate, critical thinking and reasoning, societal/ethical awareness) are essential for students and professionals and highly desired by employers [9].

However, teaching soft skills is time-consuming and resource-intensive [1]. Soft skills are usually taught in the context of software project courses and group projects [13]. Effectively practicing soft skills requires a real project setting with teams and various roles, as well as coaching and frequent feedback from teachers (in university settings) or senior colleagues (in industrial settings). However, academic institutions, software organizations and teams often do not have the resources (time, financial budget, etc.) to systematically support such training.

In this paper, we put forward the idea of utilizing *active video watching* (AVW) to enhance the training of software engineers. Our vision is that AVW will not only be useful for teaching students at universities, but also for professional development of practitioners in industry. Also, we envision that AVW can help make the move from higher software engineering education in formal programs at universities to open software engineering education and training accessible for a broader audience [16].

2 ACTIVE VIDEO WATCHING

Videos are a powerful tool for learning [4], in particular for topics that require contextualization based on the learner's personal experience. Previous research shows that well-designed, assessment-focused, and easy-to-use video tutorials improve student satisfaction and learning, because they enable students to learn how and when they want [18]. However, only watching

ACM ISBN 978-1-4503-5660-2/18/05$15.00
https://doi.org/10.1145/3183377.3183384

videos is a passive activity and tends to result in a low level engagement and shallow learning [3]. A successful learning experience requires that students engage with the video content. This can be achieved by integrating interactive activities into videos [6, 11]. Such interactive activities could include commenting on videos, answering questions about videos, or discussing videos with peers.

An example system to support AVW is AVW-Space, a web-based platform which facilitates learners' engagement during video watching through interactive notetaking [15]. The platform utilizes the learners' familiarity with commenting on videos on social networks such as YouTube. It encourages engagement during AVW by providing micro-scaffolds to facilitate the commenting on videos and the reviewing of comments made by other learners. Also, it nudges learners (through notifications) who are not active enough (e.g., learners who do not comment on videos or do not review comments of other learners). AVW-Space allows educators to select and directly integrate YouTube videos to watch. YouTube was chosen because free availability of videos is a requirement for cost-efficient and continuous updates to video-based learning systems (in particular for organizations with limited training budgets).

To illustrate AVW-Space, we use the example of teaching soft skills in one of our software engineering project courses (further details provided in Section 4). In this course, we used a set of videos about presentation skills. Four videos were *tutorial* videos on how to give presentations, followed by four *example* videos with example presentations. In general, videos used in AVW-Space are selected by the educators to ensure their suitability with regards to content and pedagogical value. Note that the focus of learning is soft skills (in this example presentation skills) rather than software engineering principles and techniques. Thus, the content of videos may not always be specific to software engineering. Each video is shorter than eight minutes to ensure that learners do not lose focus.

In the **first phase** of using AVW-Space**,** learners watch and comment on videos individually (see comment section on the right in Figure 1). Learners can stop a video at any time, enter a comment and select an *aspect* to indicate the intention of the comment. For each video, educators can specify aspects which are appropriate for the type of soft skill covered in the videos. In the case of presentation skills, we specified two types of aspects: (1) The aspects specified for the four tutorial videos aimed at stimulating reflection and therefore include: "I didn't realise I wasn't doing it", "I am rather good at this", and "I did/saw this in the past". To allow learners to externalize relevant learning, there was one additional aspect, "I like this point". (2) For the four example videos, the aspects correspond to presentation skills covered in the tutorials, which include, "Visual aids", "Speech", "Structure", and "Delivery". Figure 1 shows a screenshot of AVW-Space for commenting on an example video with the four mentioned aspects that learners can select.

In the **second phase** of using AVW-Space, the educator selects comments (for each video) that will be visible to all learners. Learners can then review and rate the anonymized comments of others based on five categories: "This is useful for me", "I hadn't thought of this", "I didn't notice this", "I don't agree with this", "I like this point" (see drop-down menu in Figure 2). Also, if learners click on the time in the comment box at which a comment was made, he/she can watch the related part of the video. This allows learners to compare their own comments to those of others and facilitates further reflection on the video content.

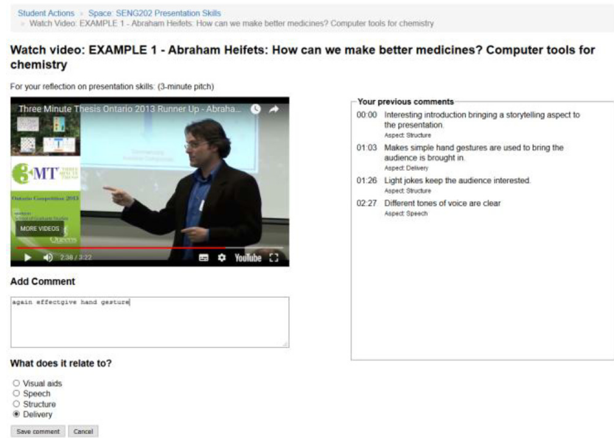

Figure 1. Phase 1 – watching and commenting on videos.

Figure 2. Phase 2 – reviewing comments of other learners.

3 AVW FOR PROFESSIONAL DEVELOPMENT

Previous research has focused on investigated methods for improving education that learners receive in formal education programs. While much work in the area of education applies to formal education models, a gap exists between traditional education and the needs of industry and industry-relevant training methods. Today's workforce in the software industry may not have the required education to perform well in current and future positions since required skills constantly evolve [12]. Also, software engineering related tasks are often done by professionals with limited education in software engineering [2], so training opportunities outside university and formal education providers are needed. This is particularly relevant since professional training programs are usually expensive and exceed the budget of small and medium-sized software companies. Also, even though massive open online courses (MOOCs) have been proposed as a solution to bring formal education benefits to professionals, they have been met with mixed results [7]. Therefore, we believe that AVW can not only enhance the learning experience in formal education, but also contribute to professional development. In fact, AVW can help universities "export" education [19] to industry. As argued by Wendt et al., software organizations seek internal professional staff development beyond traditional corporate training. Also, professionals consider small, private online courses favorably over corporate training offering and practitioners place value on the educational experience for their business and current role [19]. Therefore, AVW offerings can become part of such training. As Skevoulis argues, exported education (as university-provided services) that meets long-term needs of industry (compared to custom designed graduate programs) can be a "radical change" in the education system [17].

4 TRIAL APPLICATION

4.1 Context

To investigate the impact of AVW on student engagement and learning, we used AVW-Space in an undergraduate project course for second-year software engineering students. This is a first step before using AVW in a professional setting and professional development (see Section 3). Students in this course have already passed an introductory course to software engineering. The project includes business analysis, requirements analysis, architecture design, implementation, testing, etc. for a non-trivial software system over twelve weeks, following an iterative/incremental spiral model. Students work in teams of five to six students and deliver increasingly comprehensive versions of the design and product. Tutorials and seminars are used for weekly "stand-ups", for teaching topics relevant to the project as well as for students to present their work. This enables students to start developing communication skills as early as possible [14]. Students and teams are assessed at the end of each iteration.

Students used AVW-Space in week four of the course to learn about presentation skills as one component of communication skills of a professional software engineer. At this stage, students had not received any formal training on presentation skills. However, students had already gained some experience in giving presentations from weekly stand-ups to the class. Students used AVW-Space in their own time and had one week to complete all videos. AVW-Space collected data about how active students were while watching videos. After students had finished watching all videos, they completed the NASA-TLX questionnaire [10] and the TAM questionnaire [5]. This allowed us to get insights into the perceived usefulness of AVW and AVW-Space.

4.2 Student Engagement

Thirty-four students used AVW-Space, but only 18 wrote comments. We refer to those 18 students as *constructive* students, and to the remaining 16 students as *passive*. Constructive students watched more videos than passive students (6.94 versus 6.31 out of eight videos on average), but this difference is not statistically significant. Some students watched videos multiple times (resulting in a total average of 8.28 videos watched by constructive students versus 7.13 videos watched by passive students). Constructive students made on average 11.17 comments (stdev: 15.33; min: 1; max: 63).

4.3 Student Learning

Right before and after using AVW-Space, we asked students to write down statements related to (i) structure, (ii) delivery and speech, and (iii) visual aids of presentations. These statements were captured in a separate questionnaire outside of AVW-Space. Students had one minute to write down their thoughts and whatever came to their mind for each category. Statements from students were then coded and marked using an ontology for presentations developed previously [15]. The scores represent the number of domain concepts (from the ontology) which students mentioned in their statements. No additional evaluation of the quality of comments was performed. Table 1 shows average knowledge scores and standard deviations.

Table 1: Knowledge Scores Before/After Video Watching

	Passive	Constructive
Before active video watching	10.69 (4.35)	10.17 (5.26)
After active video watching	9.14 (4.02)	11.86 (5.02)

As Table 1 shows, there was no significant difference in the scores of passive and constructive students before and after AVW. This could be due to the small number of students involved this comparison. The scores of the constructive students improved slightly, but the difference is not significant. However, a Wilkinson's signed rank test showed that the scores of passive students decreased significantly from before to after active video watching ($W = 0$, $p = .027$). This could be due to the fact that we relied on self-reporting of students: Passive students invested less effort in describing concepts and practices for presentations in their statements. After having done this already before AVW, they may have felt that they were performing a redundant task. This could also be confirmed by the fact that passive students did not comment on any of the videos, so the engagement with the videos was low already during AVW. Therefore, their lower willingness to further reflect on presentation skills after AVW is not surprising. These findings are preliminary and more studies are needed. Previous studies outside software engineering showed that AVW-Space can increase conceptual knowledge related to presentation skills of undergraduate/post-graduate students [15].

4.4 Student Feedback

The TAM questions were based on a scale from 1 (highest) to 7 (lowest). We aggregated the answers to questions "Using AVW would enhance my effectiveness when developing transferrable skills", "I would find AVW useful in my studies/job", "If I am provided the opportunity, I would continue to use AVW for informal learning", "Using AVW would enable me to improve my transferable skills quickly" and "Using AVW would improve my performance considering the development of transferable skills" into one aggregated score for TAM-usefulness. The average score for constructive students was 4.18 (min: 2.6; max: 7). For passive students the average score was 4 (min: 2.8; max: 5.8). These numbers indicate that overall students found AVW useful. From student feedback we learned that the structured way in which comments about videos were required enforced constructive comments. Yet, students felt that a more active discussion about videos and comments would enhance their learning experience.

The NASA-TLX questionnaire captured how demanding AVW was, how much effort students invested, how frustrated they felt and how well they thought they performed their task (on a scale from 1 [lowest] to 20 [highest]). The average score for mental demand was 7.21 (min: 1: max: 20), 6.21 for effort (min: 1; max: 20), 6.56 for frustration (min: 1; max: 20) and 8.42 for performance, i.e., how well students thought they wrote and reviewed comments (min: 1; max: 20). These numbers indicate that students did not feel overwhelmed by the workload required by AVW. In textual comments, students explained their scores and why they felt frustrated or not frustrated. Frustrations were mostly caused by the design of AVW-Space (e.g., some students were not sure if comments were meant as notes for others or for themselves) rather than by AVW as such.

5 CONCLUSIONS

The growing importance of soft skill in software development calls for new methods to educate future and current software professionals. We presented active video watching as a novel approach towards integrating the training of soft skills into academic and industry education. Initial trials showed promising results. In addition to benefits discussed in this paper, AVW provides benefits of online education, such as teaching independent of time or location, sharing of teaching resources amongst educators, educating independent and self-guided students, and the opportunity to balance pedagogical needs and resource constraints. However, while AVW increases engagement, it requires effort to select suitable videos and prepare related material.

Our future work, in addition to conducting more studies to evaluate the efficiency of AVW, includes investigating several questions: (1) What (soft) skills are appropriate for learning through AVW (and what are criteria for video-based learning in software engineering])? (2) What are criteria for selecting appropriate videos for video-based learning in software engineering? (3) What other "scaffolding" is required for effective and efficient AVW in software engineering (e.g., topic-specific aspects for videos; interactive activities beyond commenting, reviewing of comments, nudging in case of inactivity and other motivators for students to comment)?; (4) How does the learning experience using active video watching depend on background experiences, motivation and attitudes of learners?

ACKNOWLEDGMENTS

This work was supported by EU-FP7-ICT-257184 ImREAL, Ako Aotearoa and a teaching development grant at University of Canterbury. We thank Vania Dimitrova, Lydia Lau, Amali Weerasinghe, Alicja Piotrkowicz and Jay Holland for their contribution to AVW-Space.

REFERENCES

[1] Suzanne Anthony and Benjamin Garner. 2016. Teaching Soft Skills to Business Students: An Analysis of Multiple Pedagogical Methods. *Business and Professional Communication Quarterly*, 79, 3: 360-370.
[2] Pablo Antonino, Andreas Morgenstern and Thomas Kuhn. 2016. Embedded-Software Architects: It's not Only about the Software. *IEEE Software*, 33, 6: 56-62.
[3] Michelene Chi and Ruth Wylie. 2014. The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 49, 4: 219-243.
[4] Michael Cronin and Karen Cronin. 1992. Recent Empirical Studies of the Pedagogical Effects of Interactive Video Instruction in "Soft Skill" Areas. *Journal of Computing in Higher Education*, 3, 2: 53-85.
[5] Fred Davis. 1989. Perceived Usefulness, Perceived Ease of Use, and Other User Acceptance of Information Technology. *MIS Quarterly*, 13, 3: 319-340.
[6] Sergio Antonio de Freitas, Wander Silva and George Marsicano. 2016. Using an Active Learning Environment to Increase Students' Engagement. In *Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*. Austin, TX, 2016, pp. 232-236.
[7] Michele Dodson, Karat Kitburi and Zane Berge. 2015. Possibilities for MOOCs in Corporate Training and Development. *Performance Improvement*, 54, 10: 14-21.
[8] Christof Ebert and Steve Counsell. 2017. Toward Software Technology 2050. *IEEE Software*, 34, 4: 82-88.
[9] Murat Erder and Pierre Pureur. 2017. What Type of People are Software Architects? *IEEE Software*, 34, 4: 20-22.
[10] Sandra Hart. 2006. NASA-Task Load Index (NASA-TLX); 20 Years Later. In *Proceedings of the 50th Annual Meeting of the Human Factors and Ergonomics Society*. San Francisco, CA, 2006, pp. 904-908.
[11] Geza Kovacs. 2016. Effects on In-video Quizzes on MOOC Lecture Viewing. In *Proceedings of the Third ACM Conference on Learning @ Scale (L@S)*. Edinburgh, Scottland, 2016, pp. 31-40.
[12] Raghuraman Krishnamurthy. 2017. Breezing my Way as a Solution Architect: A Retrospective on Skill Development and Use. *IEEE Software*, 34, 3: 9-13.
[13] Maira Marques, Sergio Ochoa, Maria Cecilia Bastarrica and Francisco Gutierrez. 2017. Enhancing the Student Learning Experience in Software Engineering Project Courses. *IEEE Transactions on Education*, in press.
[14] Jorge Maturana, Gonzalo Tampier, Guillaume Serandour and Richard Luco Salman. 2014. Developing Teamwork Skills in First and Second Year Engineering Students. *International Journal of Engineering Education*, 30, 5: 1225-1233.
[15] Antonija Mitrovic, Vania Dimitrova, Lydia Lau, Amali Weerasinghe and Moffat Mathews. 2017. Supporting Constructive Video-Based Learning: Requirements Elicitation from Exploratory Studies. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*. Wuhan, China, 2017, pp. 224-237.
[16] Jose Luis Martin Nunez, Edmundo Tovar Caro and Jose Ramon Hilera Gonzalez. 2017. From Higher Education to Open Education: Challenges in the Transformation of an Online Traditional Course. *IEEE Transactions on Education*, 60, 2: 134-142.
[17] Sotiris Skevoulis. 2009. Engineering a Successful Partnership Between Academia and the Financial Industry: A Software Engineering Program for IT Professionals. In *Proceedings of the IEEE 24th International Conference on Software Engineering Education and Training (CSEET)*. Honolulu, HI, 2009, pp. 343-350.
[18] Jason Wells, Robert Mathie Barry and Aaron Spence. 2012. Using Video Tutorials as a Carrot-and-Stick Approach to Learning. *IEEE Transactions on Education*, 55, 4: 453-458.
[19] Kevin D. Wendt, Ken Reily and Mats Heimdahl. 2016. First Steps towards Exporting Education: Software Engineering Education Delivered Online to Professionals. In *Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*. Austin, TX, 2016, pp. 241-245.