Towards Building a Collaborative Learning Environment in Software Testing Classes

Maria-Camelia Chisăliță-Crețu[1]

(1) Faculty of Mathematics and Computer Science,
Babeş-Bolyai University of Cluj-Napoca
1, Mihail Kogălniceanu Street, RO-400084, ROMANIA
E-mail: cretu[at]cs.ubbcluj.ro

Abstract

Hard and soft skills are mainly acquired by students while being engaged in team activities, with an emphasis on communication and collaboration. Software tool integration in teaching software testing helps to build a learning community. This promotes collaboration principles in teams, participants being allowed to accomplish specific tasks along with valuable competencies gain. The paper details teaching activities specifically designed for students that work in teams. Benefits, challenges, and teaching experience are presented, as activities were held face to face and then online, because of COVID-19 pandemic circumstances.

Keywords: communication and collaboration tools, gamification, soft skills

1 Introduction

Designing tasks focused of hard and soft skills acquisition for face to face or online activities is a demanding task for educators. Software tool adoption and harmonization in teaching can be successfully carried out considering competencies to be grasped. This needs to be complemented with actions that support a collaborative learning environment.

Software Systems Verification and Validation course (VVSS) (Chisalita-Cretu, 2020) at Babeş-Bolyai University of Cluj-Napoca is currently a compulsory course held during the last year in computer science (CS) bachelor degree.

The paper addresses the seminar learning activities that needs to be rigorously developed in order to support communication and collaboration between participants. Tasks that engage students are usually execises that challenge them to provide adequate solutions. Teachers can promote healthy competition among teams that brings immediat goal accomplishment and knowledge acquisition, through collaboration and team spirit. Such course activities ensure the competency advancement for the entire learning group.

The paper is structured as follows: *Section 2* details the VVSS course, its activities and the participants' details. *Section 3* shortly presents the tools integrated in software testing teaching that helped building a collaborative learning environment. The designed tasks for the seminar activities focused on collaboration and goals achievement are presented together with the students' feedback in *Section 4*. Lessons learned on designing seminar tasks are noted in *Section 5*. The paper ends with conclusions and future work.

2. VVSS Course Design

This section details the educational objectives and the assessment rules stated for VVSS course, together with the activities attended by students.

2.1 Educational Objectives

VVSS course takes 12 weeks and provides 5 credit points (ECTS) that requires a total of 125 hours (=5*25 hours) of study. The course follows the guidelines of ACM Curricula

Recommendations (ACM, 2013) for CS. It covers a wide range of topics, from concepts extensively used in practical software testing to the more abstract ones, like formal methods used for program specification and verification. The course is designed as a two hours lecture each week, supported by two hours seminar and lab activities every two weeks. Students can attend a non-mandatory project activity that allows them to deepen their knowledge in software verification topics at their choice.

Assessment. The final grade is obtained by using the following formula: *FinalGrade* = 0.50 * *LabGrade* + 0.10 * *SeminarGrade* + 0.40 * *WrittenExam* + *ProjectGrade*. The *ProjectGrade* helps the students to increase their final grade up to 2 points. The following subsection details the course activities.

2.2 Course Learning Activities

VVSS course consists of activities organized as seminars and laboratories.

Seminar activities. The course covers various topics in the six allocated seminar activities. The main goal is to detail, to experiment and to reflect on the concepts approached during lectures. Students are allowed to further investigate testing related concepts during lab activities, while other notions, concerning correctness and program refinement are approached during seminars only. The topics covered by the seminars are:

- **Sem01:** different types of inspection, from formal ones (Fagan inspection) to less formal and informal reviews, on various types of documents (requirements, architectures, diagrams, source code);
- **Sem02:** investigate black-box test design techniques as equivalence class partitioning (ECP), boundary value analysis (BVA), best representative (BR) and domain testing (DT) on specific problems, at unit testing level; use of testing frameworks (`JUnit`) and test management tools (`Testlink`);
- **Sem03:** investigate white-box coverage criteria by building the control flow graph (CFG), computing the cyclomatic complexity (CC) and studying code coverage criteria (statement/ decision/ condition/ decision and condition/ multiple conditions/ loop coverage);
- **Sem04:** study the integration strategies and apply them according to some specific application development context; use of mocking frameworks (`mockito`), continuous integration/ continuous delivery (CI/CD) tools (`Jenkins`) and report the test results in test management tools (`Testlink`); improving bug reports by using RIMGEA strategy (Kaner and Bach, 2008);
- **Sem05:** apply Floyd’s method to assess the partial correctness, termination and total correctness of small already developed programs;
- **Sem06:** apply refinement rules to develop correct programs based on given specifications; investigate formal static and dynamic analysis tools based on Java Modeling Language (`JML`), as `ESC2Java` and `jmlc` and `jmlrac` tools.

Lab activities. There are five lab activities designed for this course. Each of them focuses on different verification tasks employed during software development, having different weights in the *LabGrade*. The tasks of each assignment are applied to an already existing software product. Specific task requirements, tutorials and guidelines prepared by teachers were available for students for each assignment. The aim of elaborating such tutorials was to help participants achieve the tasks on their own pace, without the pressure of taking notes and not paying attention during the live demo sessions.

2.3 Participant Details

The data and the analysis refer to students enrolled in VVSS course during 2019-2020 academic year. All of them were young adults studying for their first university diploma.

While attending activities they were organized in groups of 25 to 30 participants for seminar activities and teams of 3 students for the lab assignment.

3. Technology Integration in VVSS Classes

Throughput VVSS classes a plethora of tools and frameworks required to perform software testing is used. In addition to these tools, communication and collaboration software are largely employed for seminar and laboratory activities.

Slack platform (Slack, 2020) is a communication platform that is used by a wide range of teams that collaborate in various types of projects. Over 3,000 higher education institutions use Slack platform to keep their classes and administrative activities online.

Basically, Slack allows building an environment focused on communication and collaboration, by relying on the *workspace* concept that represents the team (group) that works together. Workspace members can send messages on *channels* or as direct messages to designated collaborators. Further discussion on a specific topic can be achieved by creating *threads* for already sent messages.

Skype platform (Skype, 2020) was mainly used to achieve synchronous communication during lectures, seminars and laboratories. This tool allowed organizing smoothly the live teaching and learning sessions by creating id-based *meets* for different groups of study. Still, students were instructed to use Slack platform as the main communication tool, even if during live activities it was recommended to use Skype's chat features. The students were able to keep the chat as a content archive of the previous online activities.

In order to prepare challenging quizzes during activities, **Kahoot!** game-based learning platform (Kahoot, 2020) was employed. Students were invited during live sessions to join quizzes by using given entry codes. Competition among participants was stimulated by the gamification characteristics that Kahoot! presents to its users. Reports and analytics features were actively used by the teachers to rank the players, i.e., students.

Menti application (Mentimeter, 2020) was used to get feedback on spot from the students during face to face and online activities. Teachers designed various types of questions offered by the free version of the platform. Students were able to answer the quizzes after accessing presentations from their mobile phones and entering a unique identification code for the specific interactive presentation.

4. Collaborative Learning Activities for VVSS Course

This section presents several seminar activities developed for face to face and online activities. The student feedback based on Menti quizzes together with the teacher's perception on the deployed activity are reported and analysed. The first seminar was held for all groups as face to face activity, while for others seminars the switch to online activity was made. This situation was caused by the COVID-19 pandemic circumstances that started at Cluj-Napoca on 10 March 2020. From this moment onward all activities were carried out online.

4.1 Activity Details

The topics addressed during seminar activities were carefully selected such that students acquire knowledge required for laboratories tasks. For the online activities the asynchronous communication used Slack and for synchronous communication during online activities Skype was employed. Students were able to use private Slack channels while providing solutions on tasks or other tools at their choice.

Sem01. This seminar introduced several verification activities that referred to: formal inspection (Fagan inspection, walkthroughs, technical review) and informal review (peer review, buddy review, e-mail pass around, pairing).

For a deeper understanding of similarities and differences between these verification tasks, the teacher designed a game that required students to walk in the classroom and assign characteristics to designated verification activities.

After discussing the obtained results, actual inspection activities were proposed to the students. Therefore, context-based inspection solutions were developed by participants for different types of documents elaborated during software lifecycle. Sem01 was held as face to face activity for all students enrolled in VVSS course.

Sem02. The black-box testing seminar focused on working with specific test design techniques (ECP, BVA, BR, DT) for particular problems starting from specifications (pre-condition and post-condition). A team task was designed to allow students to further exercise the test design on several given contexts and reporting the results between teams.

Sem03. This activity was aimed to understand how CFG is built CC is computed together with various code coverage criteria in white-box testing, e.g., statements, decisions, conditions, multiple conditions, loops, Various practical problems related to white-box testing were discussed with the online participants aiming to understand the difference between *decision* and *condition* concepts while addressing code coverage criteria. Besides a proper learning of the particularities of each criterion, actual test case design is essential. The teacher designed a quiz Kahoot! that allowed students to learn over a consistent number of problem contexts issues frequently encountered by developers.

Sem04. A collaborative assignment was designed to acquire knowledge on testing levels. It was focused mainly on integration testing applied to a particular architecture of software modules. Students were required to work in teams and apply a studied integration strategy. The choice needed to consider project development requirements details provided by the teacher. This information was represented by the integration context that guided the integration strategy to be used. The students selected from the followings: big-bang, incremental top-down depth first, incremental top-down breadth first, incremental bottom-up, and sandwich strategies.

Sem05. The collaborative tasks designed required students to build a cross-word game based on concepts previously discussed during lectures. The task aimed to challenge the students to revisit course content and to help other colleagues to check their knowledge on software verification. Therefore, the cross-word games created by teams of three students needed to be solved by other teams (or by individual students) while the initial team was requested to grade the provided solution.

Sem06. One of the most challenging tasks in testing is the delivery of good bug reports that encourage the developer to allocate resources to investigate the reported issue. Students organized in teams were asked to build a bug report that emphasized the RIMGEA strategy (Kaner and Fiedler, 2008). At the same time, colleagues were able to vote for the best bug report in their opinion, as potential users that exposed the addressed issue.

4.2 Student Feedback Analysis

For each particular seminar activity a specific feedback quiz was created by the teacher, aiming to immediately assess discussed content. Due to the space limitations we present in the following some of the obtained results to indicate the analysis outcomes.

Sem01. The Menti quiz created consisted of three questions that addressed both the topics addressed during the seminar and the learning experience students perceived. *Figure 7* shows the word cloud provided by one seminar group that performed tasks of Sem01. All seven groups provided similar word clouds, formed of relevant terms associated to the concepts in human-based verification activities, applied to different types of documents throughout the development process.

Figure 7. Word cloud for concepts discussed in Sem01

Students reported their opinions on the performed activities when they were asked: *What did you like? What did you not like?*. From the 192 students that provided answers, 69% of them liked the activities, indicating high level of interaction, team based-tasks, not being required to keep their place all times, or being able to learn theoretical aspects in a pleasant manner in group, not by individuals. Other 31% reported they did not liked the activity, claiming several issues as: not being able to read details from the table or cards placed far from them, too many concepts being discussed at once, not enough time to solve the tasks.

Sem02. One of the questions of the Menti quiz referred to the applicability of ECP and BVA techniques. From the 59 students that answered the question, 80% of them were correct, saying they apply to variables' domain of values. Still, many of them choose to not respond even if they were present in the classroom or had joined the web conference on Skype. Further, we have studied the answer on the questions *What was easy to understand? What was hard to understand?* We got a better understanding of the small number of students that offered a response to the first question. Aspects discussed referred to theoretical concepts, while some students did not grasp the differences between ECP and BVA techniques.

Sem03. Students were excited about answering to the quiz on the white-box topics. Still, for some of them it was hard to finish as the quiz was time limited. The CC importance and the difference between a *decision* and a *condition* concepts were the main aspects addressed in the Menti quiz. When asked about them 30% of the students answered that a condition is part of a decision, while 70% responded the other way around. This issue was confirmed by several Kahoot! quiz questions that required to identify the correct number of test cases to achieve condition or decision coverage.

Sem04. Team communication and collaboration based on public/private Slack channels and *meets* on Skype allowed students to prepare the integration solutions following a specific application and architecture context. Students reported to be very useful for them these platforms, as they were able to ask questions while elaborating their solution. All the teams could see the addressed questions and teacher's answers, creating a learning environment that encouraged them to provide an acceptable solution that met the context constraints. In the end, students were able to share their solutions. They stated that it was relevant for their learning experience to see other teams' work and debate upon the integration strategies and the bug types easy to spot. Students noticed that the application context information was essential to uniquely identify the integration strategy.

Sem05. Students enjoyed the task, 30 student teams being enrolled for this group assignment. They felt empowered by creating challenging and diverse questions for their colleagues and were stimulated to provide the correct solution for other teams' quizzes.

Sem06. For the bug report race seven teams enrolled to compete. Students were challenged to find a bug and provide a proper description such that a possible developer is motivated to fix it. Teams reported various bug found in web browsers, operating systems, MS Word application, or team project already developed. Students voted the preferred bug story on the Slack platform by using emoticons and providing feedback on their colleagues' work.

4.3 Teacher Perception Analysis

For each seminar activity the teacher made notes on how the designed seminar activities were carried out.

During **Sem01** teacher noticed that students expected a classical seminar, where they are invited to involve in blackboard-based problem solving. Some of them did not expected from their teacher to ask them to go around the classroom and discuss with their colleagues to solve the assigned tasks. Teacher easily spotted students having leader and task management skills, organizing the actions to be performed by the group. The main intent of the first seminar task was to engage all students, regardless of the learning styles (visual, auditory, kinesthetic). Therefore, during the first task all types of learners were able to get involved.

For **Sem02** the teacher expected to have a consistent engagement of students during stated tasks. This expectation was not fulfilled as many students were struggling to remind formal concepts discussed in a previous year. More, while student team aimed to collaborate in teams to test design test cases bad on ECP and BVA, their answers were different from one team to another. Students considered the problems hard to understand, by once understood, the solution was easy to provide. The teacher noticed that more emphasis on mathematical concepts is required before being able to solve easy problems.

Sem03 tasks were the only ones that required to work as individuals not in teams. During the actual seminars, teacher noticed that some students very active compared to others. More, participants seemed to be overwhelmed by the number of coverage criteria to be discussed during a single seminar.

Improvements in team management while achieving assigned tasks were noticed for **Sem04**. Students quickly formed teams, worked on the solution and sent to the teacher the strategy-based integration results following the recommended file format. Students carefully analysed the application contexts and observed several small variations that suggested a certain integration strategy. The cross-word game quickly engaged students during **Sem05**. Many teams tried and succeeded to solve more than one puzzle, even if only solution was finally considered to be graded. Students were stimulated to conceive and deliver puzzles formed of compounded words and thus, having increased complexity level. The collaborative task assigned for **Sem06** granted students the freedom of being creative while building an appealing bug description that helped to fix it. While elaborating the bug story, participants reflected on the relevance of choosing the right details that may motivate someone else, e.g., developer, to investigate the issue. While reading someone else's bug description, students had the opportunity to understand the bug impact from some one's perspective that embodies a certain user profile.

5. Lessons Learned

During the 2019-2020 session of VVSS course that required both face to face and online interaction during activities, teachers were challenged to design tasks that easily engage students. Game-based activities, were successfully accepted by students, helping them to overcome the everlasting question *why do we learn this?* on many undergraduate CS-based classes. The knowledge was perceived like a gift nicely packed, students being curious to find out more about it. Still, from the teachers' perspective, the effort to prepare the working materials cannot be neglected. The educators need to have clear goals about the concepts students should learn during some specific activity. Then, the focus is turned to choose the right software that helps gaining competencies, while creating a relevant learning experience. The instructor's endeavour in creating meaningful task was fully repaid by the feedback that students provided at the end of the activity. For example, Menti features that allowed students to provide answers to a specific question while dynamically building the word cloud (see *Figure 7*) helped them to have a group reflection moment during the class.

Students were stimulated to actively engage during classes even in some theoretical and abstract parts of the assigned tasks. Each time students were told about what follows and what is expected from them to achieve. Clear directions made easy each task they were involved in.

6. Conclusions

The paper addresses the hard and soft skill acquirement during seminar activities for the VVSS course by engaging students in collaborative tasks. Software integration in software testing education like Slack, Skype, Kahoot and Menti allows creating a learning environment that promotes communication, creativity, competition and goal achievement.

Specifically designed tasks were analysed from students' perspective considering the formative provided feedback. The results show an increased commitment for activities that allow idea sharing and team work. Educator's perception while guiding activities indicates the students require clear directions, interactivity and sometimes support to accomplish the assignments.

Future work in this domain includes communication and collaboration software tools analysis, with focus on the integration opportunity in Slack. A more narrow research will be focused on the design of critical thinking-based tasks for testing education using these tools.

References

Association for Computing Machinery (ACM) Society, 2013. Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science, New York, NY, USA, ACM.

Chisăliţă-Creţu, C. (2020), VVSS Course Syllabus, https://www.cs.ubbcluj.ro/files/curricula/2020/syllabus/IR_sem6_MLR5014_ro_cretu_2020_5005.pdf, accessed on June 2020.

Kahoot, 2020, Kahoot! Platform. Retrieved June 30, 2020 from https://kahoot.com/schools-u/.

Kaner, C. and Bach, J., 2008. Black Box Software Testing. Bug Advocacy.

Menti, 2020. Menti platform, https://www.menti.co/.

Skype, 2020, Skype Platform. Retrieved June 30, 2020 from https://www.skype.com/en/.

Slack, 2020, Slack Platform. Retrieved June 30, 2020 from https://slack.com/intl/en-ro/.