# Experiences Using an Interactive Presentation Platform in a Functional and Logic Programming Course

Vasileios Triglianos⋆, Martin Labaj†, Robert Moro†, Jakub Simko†,

Michal Hucko†, Jozef Tvarozek†, Cesare Pautasso⋆, Maria Bielikova†

⋆Software Institute, Faculty of Informatics, USI Lugano, Switzerland

name.surname@usi.ch

†Slovak University of Technology in Bratislava,

Faculty of Informatics and Information Technologies, Bratislava, Slovakia

name.surname@stuba.sk

## ABSTRACT

Modern constructivist approaches to education dictate active experimentation with the study material and have been linked with improved learning outcomes in STEM fields. During classroom time we believe it is important for students to experiment with the lecture material since active recall helps them to start the memory encoding process as well as to catch misconceptions early and to prevent them from taking root. In this paper, we report on our experiences using ASQ, a Web-based interactive presentation tool in a functional and logic programming course taught at the Faculty of Informatics and Information Technologies at the Slovak University of Technology in Bratislava. ASQ allowed us to collect immediate feedback from students and retain their attention by asking complex types of questions and aggregating student answers in real time. From our experience we identified several requirements and guidelines for successfully adopting ASQ. One of the most critical concerns was how to estimate the time when to stop collecting the students' answers and proceed to their evaluation and discussion with the class. We also report the students' feedback on the ASQ system that we collected in the form of the standard SUS questionnaire.

## 1 INTRODUCTION

Active learning, which falls within a constructivism philosophical viewpoint of education, has been associated with increased student performance in STEM sciences [3, 7]. In the context of computer programming courses, active learning would require students to perform tasks that include discussing study material concepts, working with modelling tools and writing and executing programs. A recent paper that studies the influence of format and depth of questions during effortful retrieval practice [6] showed that participants who practice using free recall questions types of applied depth (as opposed to factual) outperformed all other combinations of recognition/cued recall questions with factual/applied depth.

We wanted to bring the active learning concepts into our lectures for the Functional and Logic Programming Course at the Slovak University of Technology in Bratislava to move away from the passive paradigm of traditional lectures. One of the reasons for this was the fact that during the course, the students are taught concepts that are entirely new to them (almost all of them encounter the functional and logic programming paradigm for the first time during the course) and they have to learn the basics of three new programming languages. Therefore, it is important for them to confront their understanding of the taught concepts as soon as possible. Moreover, the instant feedback on the students' understanding is beneficial also for instructors who can adapt their lecturing style and pace to the needs of the students. An added incentive to do so was the increasing usage of smart devices by students in classrooms that leads to increased off-task behavior and reduced performance in on-task activities [4, 8]. Our aim was to make students use their devices primarily for on-task activities like following the lecture slides and answering questions.

We wanted to move to a lecture cycle where we teach our students short segments of study material and after each segment we ask them pertinent questions to gauge their level of understanding of the taught concepts. When we set to do so we faced challenges in three main areas: (i) time and cognitive load constraints in aggregating student responses and giving the students feedback, (ii) time and cognitive load constraints when switching between presentation software and audience response systems, and (iii) the need to have different question types to match different needs of recall difficulty (recognition, cued recall or free recall) and answer expressiveness.

In this paper we report on our experiences from using ASQ, an interactive presentation platform that manages some of those challenges, discuss about lessons learned and identify some of the problems that arose and how to potentially address them.

## 2 ASQ

ASQ is a Web platform for delivering interactive lectures in brick-and-mortar classrooms [12]. Students connect with their smart device to the ASQ Web Server and get a copy of the presentation. The active slide of each student is kept synchronized with the slide shown on the beamer, which is controlled by the teacher. When

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*UMAP'17 Adjunct, July 09–12, 2017, Bratislava, Slovakia*
© 2017 ACM. 978-1-4503-5067-9/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3099023.3099082

authoring a presentation, teachers can embed questions of various types to an ASQ slide. Student answers are aggregated in real time and displayed to the teacher, who may choose to share them with the rest of the class. Depending on the course format and goals, the teacher may give generic feedback to the students, share individual answers on the beamer and give specific feedback on them. Also, students may be allowed to correct their answers based on the discussion.

For our course, we decided to use six question types, five of which could be automatically assessed for correctness. The list of question types is as follows:

(1) **multi-choice**. Multi-choice question (auto-assessed)
(2) **short answer**. Single-line text input (auto-assessed)
(3) **highlight the text** (example in Figure 1). Highlight portions of text using the appropriate color (auto-assessed)
(4) **classify**. Classify the label dropping it into the correct bucket (auto-assessed)
(5) **order**. Place items in the correct order (auto-assessed)
(6) **code**. Code editor with syntax highlighting supporting many programming languages. In this specific course, we used tasks in LISP and Prolog.

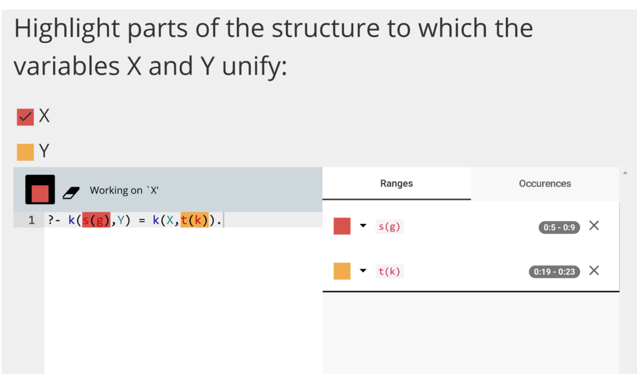


**Figure 1: Example of a highlight question posed in the lectures *Prolog 2a* and *Prolog 2b*.**

The instructors have access to a dedicated web page, called the cockpit which allows them to preview slides, monitor classroom activity or focus on the progress of individual students.

## 3 COURSE AND FORMAT

The Functional and Logic Programming (FLP) course, on which we base the discussion in this paper, is taught at the bachelor level of study. It is organized bi-annually and is usually taken by students of the 2nd and 3rd year. The course is valuated for 6 ECTS credits. The course lasts for 12 weeks. Its format comprises weekly lectures (100 minutes each), laboratory practices (100 minutes each) as well as individual project activities. In this work, we are specifically interested in the improvement of lecturing, therefore, we disregard the practice sessions or individual project work.

The particular instance of the course took place in the spring semester of 2016. It was attended by 46 students.

From the syllabus perspective, the FLP course is the introduction to the functional and logic programming paradigms. It is taught with the assumption that the students have prior knowledge of imperative (procedural) programming, data structures, algorithms, and computing in general. Functional programming is introduced through LISP and Python languages. Logic programming is introduced through Prolog language. The goal of the course is to develop an understanding and practice of non-conventional programming paradigms.

The FLP course is *elective*. Consequently, it was populated with students that showed some degree of interest in functional and logic programming.

### 3.1 Lecture Style of Previous Years

The FLP course had about 10 past editions (over 20 years). During this time, it was lectured by a single professor – one of this paper's authors.

The professor's lecturing style relied heavily on handwriting (drawing) that is simultaneously projected for the students through the beamer (historically, the drawing was done physically on celluloid slides and more recently, virtually, using the OneNote tool). This style of lecturing keeps students more engaged than using prepared slides as it enables to improvise according to the actual context and the students' reactions. The lecture matter contains mostly technical topics and examples (particularly program fragments), which are written/drawn and discussed in depth. When lecturing, the professor relies on detail stressing using virtual pencil and ad-hoc graphical explanations. Prepared materials such as slides are used only sparsely to cover some topics. After the lecture, students receive the drawings as pdf files.

Even though the professor has been using a virtual pencil, the lectures in the past course editions were seldom interactive. The interaction was usually ad-hoc, initiated verbally by the professor (e.g., "how will this code be interpreted?"). Only few students usually answered such calls. This meant that students were easily getting bored by the "one-wayness" of the lecture. Furthermore, if the matter discussed became overwhelming (after all, the subject was often complex and abstract), the professor had no option how to detect this. Also, it was hard to detect, whether the students were not accidentally developing misconceptions.

Our requirements for course lecture reform thus followed mitigating of three issues:

- boredom,
- loss of focus because of overwhelming lecture content, and
- misconception development.

### 3.2 Lecture Style for this Year

The lectures of the latest edition (in 2016) of the course were done with help of ASQ. ASQ was used in the majority of the lectures (8 out of 12). The cases when the system was not used included the introductory lecture and also the guest lectures, which regarded the functional programming in Python. On the other hand, ASQ was heavily used for the LISP and Prolog lectures (see Figure 2 for how the classroom looked like during one of these lectures).

One of the major changes in the lecturing style was that the lecture scenarios were now driven by ASQ. The topics and concepts that professor wished to be covered, were sequenced and represented mostly as interactive questions and exercises and sometimes as

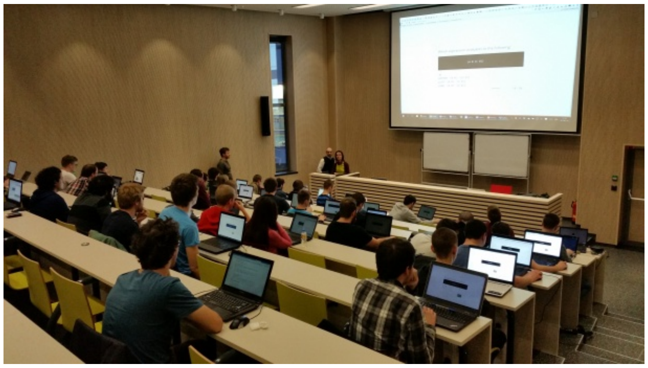

**Figure 2: The students use ASQ on their own devices during one of the lectures. The pace of a lecture is managed by the instructor; the content is synchronized. The question is also beamed for all to see.**

static slides in ASQ. This sequence was then followed in the lectures, which thus contained a substantial portion of interaction, when students were answering questions and solving exercises conveyed to them through ASQ.

At the same time, the lectures retained their key feature – handwriting and drawing. This was done using the same tools, although some drawing was also done in ASQ. The professor resorted to this techniques every time that a concept needed to be explained (technically, she left the ASQ environment, performed the explanation and then returned to ASQ to further follow the lecture structure). Drawings in ASQ served mainly for the discussion on the students' answers and the comparison of various alternatives. The effect was that the lectures were slower-paced; on the other hand, the concepts were covered in more depth and the students' misconceptions were discovered and addressed almost immediately after they had been introduced in the class.

# 4 RESULTS

## 4.1 Activity in ASQ

ASQ was used during 8 lectures of the FLP course in the spring 2016. All student and professor activity in ASQ was recorded, so that it is possible to assess in real time the number of the connected students, the number of the students answering the questions as well as the number of the students that have already submitted their answer.

However, outside of ASQ, we have no idea what the students do on their devices; therefore, it is possible for the lecturer to think that they are off-task, while they can be actively searching for the solution on their smart device. For this reason, we decided to record the last three lectures in our User Experience and Interactions Research Center[1], in which all the working stations are equipped with an eye tracker. In this paper, we focus solely on the analysis of the students' activity and their feedback; the evaluation of the recorded eye tracking data remains a future work. However we report this fact since it slightly changed the course setup: As the room can accommodate at most 20 students at once, the three lectures were divided into two equal sessions; in the first session, the first half of students was lectured, in the second, the second half (the order of the groups switched between the lectures). The two sessions (e.g., *Prolog 1a* and *Prolog 1b*) were essentially the same lecture (with the same content and questions). Additionally, since the students were asked to use the lab computers during the recorded lectures, their interaction patterns might have changed. On the other hand, the activity in ASQ was not anonymous in our case also in the previous lectures, as the students were required to log in (with the perspective of gaining activity points); although, they still might have opened multiple anonymous sessions in other browsers or on other devices.

The summary of the students' activity in ASQ is presented in Table 1 (for Prolog lectures, we report the individual sessions separately, which explains the lower number of connected students). We can see that majority of the students submitted an answer to at least one question; however, the number of students answering all questions was quite low. This can indicate that they did not have enough time to solve the tasks or that the tasks were too complex. On the other hand, we can see an increase in percentages of students submitting all answers between first lectures on LISP or Prolog compared to the later ones; this might indicate that as the students got acquainted with the basic concepts, their performance on tasks improved.

**Table 1: Overview of the lectures in the case study. Lectures are reported in temporal order of delivery. For Prolog, lectures were divided into two separate sessions (e.g., *Prolog 1a* and *Prolog 1b*), i.e., the same content was lectured to two different groups of students. Legend. #ASQ: number of students connected to ASQ. #Q: number of questions in the lecture slides. #A: number of total answers in the lecture. %SQ: the percentage of students connected to ASQ and answering at least one question (≥1 question submitted). %75Q: percentage of students connected to ASQ and submitting an answer to at least 75% of the questions in the lecture. %AQ: percentage of students connected to ASQ and submitting an answer to all questions. %SN: percentage of students' initiated inputs to the questions that resulted in no submit.**

| Lecture | #ASQ | #Q | #A | %SQ | %75Q | %AQ | %SN |
|---|---|---|---|---|---|---|---|
| Lisp 1 | 46 | 10 | 355 | 86.96 | 4.35 | 4.35 | 23.44 |
| Lisp 2 | 51 | 11 | 501 | 84.31 | 41.18 | 3.92 | 8.13 |
| Lisp 3 | 50 | 8 | 275 | 74.00 | 44.00 | 16.00 | 33.41 |
| Lisp 4 | 45 | 14 | 534 | 86.67 | 51.11 | 17.78 | 33.63 |
| Lisp 5 | 54 | 14 | 625 | 68.52 | 61.11 | 25.93 | 17.32 |
| Prolog 1a | 27 | 20 | 446 | 88.89 | 74.07 | 3.70 | 10.99 |
| Prolog 1b | 22 | 23 | 475 | 95.45 | 77.27 | 9.09 | 10.55 |
| Prolog 2a | 23 | 26 | 469 | 82.61 | 52.17 | 8.70 | 25.20 |
| Prolog 2b | 22 | 26 | 483 | 81.82 | 59.09 | 22.73 | 20.43 |
| Prolog 3a | 19 | 11 | 209 | 94.74 | 73.68 | 31.58 | 15.73 |
| Prolog 3b | 20 | 10 | 220 | 100.00 | 80.00 | 50.00 | 20.86 |

## 4.2 Students' Feedback

At the end of the lecture before the last one (i.e., at the end of one of the recorded lectures which was divided into two sessions,

[1]http://uxi.sk

namely *Prolog 2a* and *Prolog 2b*), we collected the students' feedback on ASQ using the SUS (System Usability Scale) questionnaire that was administered anonymously and the student participation was voluntary. First introduced in [2], it consists of 10 statements (5 worded positively, 5 worded negatively) targeting the aspects of usability and learnability of a system [5]. The students expressed their agreement with the statements on a 5-point Likert scale. In addition, the students were presented with three statements concerning the lecture:

(1) I always managed to submit all the tasks in time (in_time).
(2) I found the lecture interesting the whole time and I payed attention to it (interested).
(3) I understood the content of the lecture (understood).

Lastly, we collected their feedback on two open questions, namely what they liked about the use of ASQ during the lecture and what they would like to improve about their experience.

Overall, 36 students out of 45 that were connected to ASQ during the lecture (in the first or the second session) responded the questionnaire. The average SUS score was 75, which can be interpreted as a good level of usability according to [1]. We also computed the two factors contained in SUS, namely usability and learnability, obtaining the average scores of 71 and 93 respectively, which suggests that the students found the system intuitive to use. Figure 3 shows the distribution of the SUS scores based on the individual students' responses. We can observe that it is skewed to the right and that, in fact, half of the scores were above 77.5, which is the median.

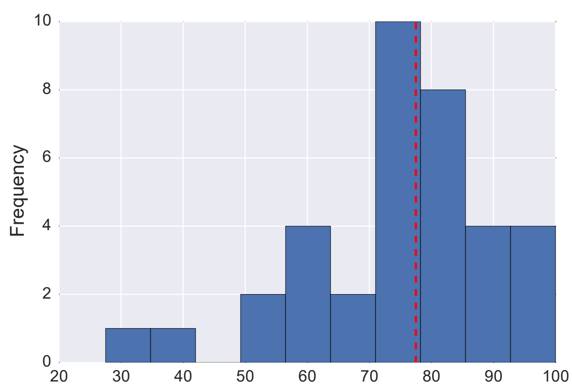


**Figure 3: The histogram of the SUS scores based on the students' responses. The mean is at 75, while the median (shown as the red dashed line) is 77.5.**

As to the lecture-related questions, we can see the distribution of students' answers in Figure 4; the perceived understanding of the topics covered in the lecture was on average quite good, while the interest in the lecture was more widely distributed. The problem turned out to be the ability of the students to submit their answers in time. The instructor moved to the next slide (with submitted and correct answers), when approximately a half of students submitted their answers; this might have been reduced if it took them too long to solve the task (e.g., when it was too complex). The students

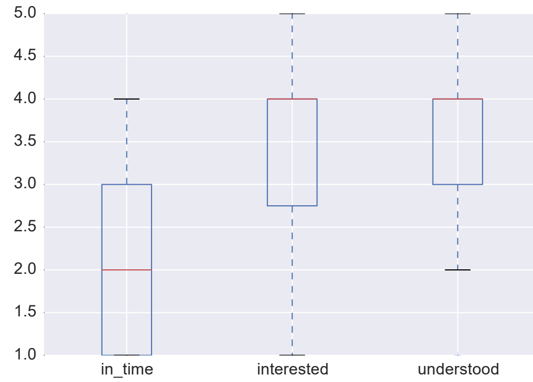


**Figure 4: The distributions of the answers to the three lecture-related questions.**

felt stressed when solving complex tasks and they started to loose motivation if they were always slow to answer compared to other students.

This was also confirmed by the analysis of the answers to the open questions (which we first clustered using text processing methods to simplify their evaluation); the extension of time allocated for answering the questions in ASQ occurred the most in the students' answers on possible improvements. If we look at the percentages of students that started to solve a question, but did not submit it in the end, it was 20.53% on average. The exact values varied between the lectures (see Table 1), but it was always at least about 10% of inputs, which demotivated students to some extent.

The multi-choice questions turned out to be the most problematic (see Table 2; for the *buckets* and *order* types we do not have data on students' inputs, just the submissions); one of three distinct inputs (grouped by a student and a question) did not lead to a submit.

**Table 2: Metrics per question type. #Q: number of questions of this type. #A: number of total answers for this type. %SQ: the percentage of students connected to ASQ and answering at least one question for this question type (≥1 question submitted). %SN: percentage of students that initiated inputs to the questions of the corresponding question type but resulted in no submit.**

| | Question type | #Q | #A | %SQ | %SN |
|---|---|---|---|---|---|
| 1 | Code | 11 | 323 | 70.67 | 9.06 |
| 2 | Highlight | 4 | 131 | 79.12 | 6.82 |
| 3 | Multi-choice | 37 | 1284 | 80.74 | 31.22 |
| 4 | Text input | 62 | 1777 | 79.16 | 13.78 |
| 5 | Buckets | 1 | 39 | 84.78 | N/A |
| 6 | Order | 2 | 36 | 37.11 | N/A |

Therefore, we were also interested in the temporal arrivals of students' answers starting at the time when the question was posed (i.e., when the slide with the question was presented to the students). We can see that, e.g., in the case of one of the lectures (*Prolog 3a*, in

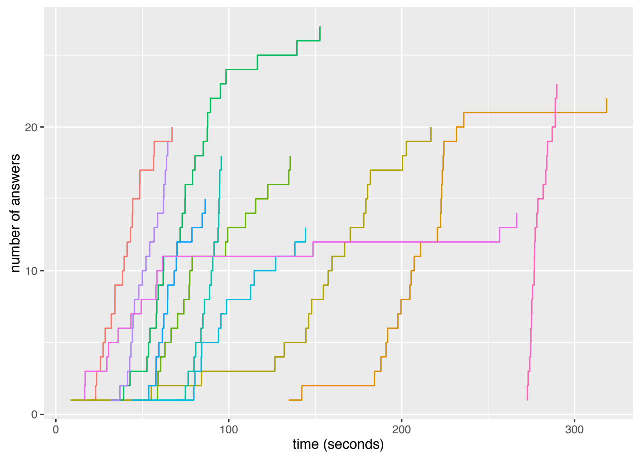


**Figure 5: The arrival of students' answers in time for each question in a lecture (results for *Prolog 3a* are shown here).**


**Figure 6: The correlations between SUS score and students' ability to submit the answers in time, their interest in the lecture, and their understanding.**

which the number of initiated inputs to the questions that resulted in no submit was more than 15%), about 50% of questions reached 20 answers corresponding to the number of connected students (sometimes there were more answers when students changed and resubmitted their answers). The rest of them did not. For some of these, we can observe a steep increase of answers just before the slide change, while for others there was almost no new answers for a longer period of time. This suggests that the instructors need to base their decision for when to advance to the next slide also by considering the number of students still actively submitting. Moreover, the information about the steepness, i.e., the increase of answers in time (which is not provided by ASQ at the moment) might be also helpful.

Lastly, we examined the correlations between the SUS score and the students' ability to submit the answers in time, their interest in the lecture, and their understanding (see Figure 6). There was a statistically significant correlation between

- students' interest in the lecture and their understanding ($p < 0.005$) as well as between
- the SUS score and their interest ($p < 0.01$),
- the SUS score and their ability to submit in time ($p < 0.05$), and
- the SUS score and their understanding of the lecture ($p < 0.05$).

Overall, based on the answers to the open question concerning what the students liked about the lectures with ASQ, the students mostly appreciated the new style of lectures in the form of the interaction with the lecturer and the opportunity to immediately test their understanding. They also appreciated the possibility to see all correct answers for different types of questions and compare their solution to them. Other proposed improvements included the ability to auto-assess the *code* question type. In the current setup, only an exact match with the example solution was evaluated as correct and students often inserted code with syntax errors. We observed an even higher variation between different answers in the case of the programming tasks in Prolog.

## 4.3 Instructors' Feedback

In general, the overall experience from using ASQ left a positive impression, however there are also areas that need improvements.

The students spent more time on on-task activities compared to the previous years which we assume benefited their learning. The addition of cued recall (highlight the text, classify, order) and free recall questions (short answer, code) on top of recognition questions (multi-choice) opened the possibility for answers that we did not expect. We saw both original answers with correct solutions that we did not think of beforehand as well as misconceptions for the presented material that we had not met in the past. ASQ allowed us to catch said misconceptions early as well as present and discuss in class different approaches to solve the same problem. This way, the lectures adapted to the students' needs (more attention was given to the concepts they had problems to understand). We observed this especially during the three recorded lectures which consisted of two sessions targeting different groups of students; even though the same content was prepared for both sessions, they had different pace and sometimes different explanations were needed.

In terms of classroom orchestration, the integration of slides and questions was deemed beneficial, since it removed context switching overhead between specialized software and the effort needed to keep the collective attention focused on the same activity. We encountered several challenges with time management. We observed that the amount of curriculum material presented in each lecture decreased. This is somewhat expected, since the amount of time spent in quizzes and feedback was increased but, at the same time, it calls for better classroom time scheduling on the instructor's side as well as picking out material for home studying. Moreover, sometimes there was unnecessary time spent during the first minutes of the lectures waiting for all the students to log in and connect to the lecture.

We also encountered challenges associated with question time management. Sometimes it was hard for the instructors to decide what was the best time to move on with the rest of the lecture instead of waiting for more submissions from the students. We would like to have a mechanism in place to warn the students that quiz will be over in a specified amount of time which would be dynamically determined taking into account factors as overall answer progress, question difficulty, remaining time and more.

An interesting phenomenon we observed when instructors gave feedback for a question while still accepting submissions was that the students would modify their answers to align with the perceived opinion of the instructor. We consider this to be beneficial to clearing misconceptions and making the instructors aware of whether the students understood their explanation.

Preparing slides for an ASQ lecture required HTML5 knowledge. Once the material for a lectured was finalized, it took on average two hours to prepare it. The current version of ASQ does not allow to dynamically add new questions during a lecture. It is therefore important to carefully plan the lecture material during authoring time. However, during classroom time unexpected events happen which may require to adapt the quiz material. We hope that in the future there will be a graphical user interface (GUI) editor for ASQ presentations as well as the ability to inject questions on a live ASQ presentation.

## 5 CONCLUSIONS

The ASQ interactive slideshow platform was successfully deployed at the Faculty of Informatics and Information Technologies at the Slovak University of Technology in Bratislava. It was positively received by the students as well as by the instructors. For this reason, we continue using ASQ also on other courses taught at the University with a larger number of students in the class (more than 100). Nevertheless, we identified several areas for improvement (support for time management during question time, the effort of preparing interactive material and the need for questions to be dynamically injected during a presentation) that we summarized in the previous section.

The version of ASQ deployed for our course allowed only instructors to ask the students questions, which had to be prepared before the lecture. Allowing students to ask the instructors questions can also be useful as it provides the students with the ability to ask what they are interested in or what they do not understand. The instructors can attend to these questions during the lecture or at the end of it, when it is convenient for them and does not disrupt the flow of the lecture. This is supported by systems such as *sli.do*[2] which also supports voting questions and discussion moderation. Recently, this functionality was added to ASQ as well and we plan to use it in our future lectures.

Both these types of asking questions work in a synchronous setting, i.e., both the ones that ask and the ones that answer are present at the same time; there is also possibility to ask questions asynchronously, e.g., in a CQA (Community Question Answering) system [10]. In the setting of our university, we use educational CQA system *Askalot* [9] that has been since modified to work also in other settings, such as *edX*[3] [11]. It would be interesting to connect ASQ and *Askalot* in the future, so that the selected questions (posed either by the instructors or by the students in ASQ) would be persisted and the discussion would continue in *Askalot*, where it could be archived for future retrieval.

For future work, in order to provide the instructor with a condensed overview of the submitted answers, we are currently working on an approach that would automatically cluster them in the real time. It has the potential to help quickly identify common students' mistakes and misconceptions so that the instructor could comment on them even during answering time, thus affording the students to understand concepts better and be more successful.

In addition, as we mentioned in Sec. 4.1, we recorded several lectures in our User Experience and Interactions Research Center (including all students' browser and operating system activity as well as their gaze, face, and screen) in order to get the whole picture of the students' behavior during the lecture and their interaction with ASQ. Therefore, as future work, we aim to analyze the recorded data with the goal of identifying patterns of the on-task and off-task student time during the lectures.

[2]https://www.sli.do
[3]https://www.edx.org

## ACKNOWLEDGMENTS

This work is a partial result of the collaboration within the SCOPES JRP/IP, No. 160480/2015 and it was partially supported by the projects KEGA 028STU-4/2017 and "Development of research infrastructure of STU", project no. 003STU-2-3/2016 by the Ministry of Education, Science, Research and Sport of the Slovak Republic.

## REFERENCES

[1] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* 4, 3 (2009), 114–123.
[2] John Brooke and others. 1996. SUS – A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
[3] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415.
[4] Robin H Kay and Sharon Lauricella. 2011. Exploring the benefits and challenges of using laptop computers in higher education classrooms: A formative analysis. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie* 37, 1 (2011).
[5] James R. Lewis and Jeff Sauro. 2009. The Factor Structure of the System Usability Scale. In *Proceedings of the 1st International Conference on Human Centered Design – HCD 2009*, Vol. 5619 LNCS. Springer, Berlin, Heidelberg, 94–103.
[6] Jaclyn K Maass and Philip I Pavlik Jr. 2016. Modeling the Influence of Format and Depth during Effortful Retrieval Practice. In *Proceedings of the 9th International Conference on Educational Data Mining*. 143–150.
[7] Michael Prince. 2004. Does active learning work? A review of the research. *Journal of engineering education* 93, 3 (2004), 223–231.
[8] Susan M. Ravizza, Mitchell G. Uitvlugt, and Kimberly M. Fenn. 2017. Logged In and Zoned Out. *Psychological Science* 28, 2 (2017), 171–180.
[9] Ivan Srba and Maria Bielikova. 2015. Askalot: Community Question Answering As a Means for Knowledge Sharing in an Educational Organization. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing (CSCW'15 Companion)*. ACM, New York, NY, USA, 179–182.
[10] Ivan Srba and Maria Bielikova. 2016. A Comprehensive Survey and Classification of Approaches for Community Question Answering. *ACM Trans. Web* 10, 3, Article 18 (Aug. 2016), 63 pages.
[11] Ivan Srba and Maria Bielikova. 2016. Design of CQA Systems for Flexible and Scalable Deployment and Evaluation. In *Web Engineering: Proceedings of the 16th International Conference on Web Engineering – ICWE 2016*, Alessandro Bozzon, Philippe Cudre-Maroux, and Cesare Pautasso (Eds.). Springer International Publishing, 439–447.
[12] Vasileios Triglianos and Cesare Pautasso. 2014. Interactive Scalable Lectures with ASQ. In *Web Engineering: Proceedings of the 14th International Conference on Web Engineering – ICWE 2014*, Sven Casteleyn, Gustavo Rossi, and Marco Winckler (Eds.). Springer International Publishing, 515–518.