

# SELF-TEST INTEGRATION IN LECTURE VIDEO ARCHIVES

Martin Malchow, Matthias Bauer, Christoph Meinel

*Hasso Plattner Institute (GERMANY)*

## Abstract

Lecture video archives offer hundreds of lectures. Students have to watch lecture videos in a lecture archive without any feedback. They do not know if they understood everything correctly in comparison to MOOCs (Massive Open Online Course) where a direct feedback with self-tests or assignments is common. In contrast to MOOCs, video lecture archives normally do not offer self-test or assignment sections after every video. Due to this behavior of lecture archives questions have to be made visible on the video page. Furthermore, lecture recording videos are typically longer than videos in MOOCs. So, it is not so reasonable and sometimes even demotivating to ask a lot of questions after a long video when not all information is already memorized by the student. The approach of this paper is to overcome these self-test problems in lecture video archives and to finally solve them in a reasonable way to increase the learning experience and support students to learn more efficient with recorded lecture videos.

Keywords: self-test, e-learning, assessment, user feedback, lecture video archive.

## 1 INTRODUCTION

Students watching lectures of lecture video archives normally have no feedback if they understand everything correctly due to the lack of feedback and tests. If students understand something wrong they will keep the wrong information in mind since there is no feedback possibility like self-tests which are known from MOOC platforms.

To handle these problems this approach deals with integrated self-tests in the video player of lecture archives. This approach is based on previous work in the MOOC and teleteaching area [1]. The idea is to show quizzes during the video at certain time stamps after the related topic of the question was described in the video. So, the users can get instant feedback based on the current lecture content and will not get bored because of the interactivity during the lecture playback. When a question is placed at a certain time the video is paused and the question will overlay the video. This behavior can be controlled by the student. Hence, no self-tests will be shown if the student already saw these self-tests which otherwise could be disturbing for the student. If this is the case the user can deactivate the self-test feature in the player. Possible question types in this approach are multiple choice, multiple answer, and text answer.

In order to validate the acceptance of the feature the students' quiz answers were analyzed and also the students were asked to complete a survey to evaluate this feature of the video lecture archive. The overall feedback of the students is positive and they predominantly like the learning support with interactive questions during the video. Furthermore, it motivates the students to go on watching the lecture when they are able to test their knowledge directly.

As a result the MOOC-like enhancement of the learning experience in video lecture archives is reasonable. This in-player feature enhances the learning experience and motivation of students using video lecture archives as e-learning resource.

## 2 RELATED WORK

Self-tests are a typical way for self-evaluation in MOOC platforms [2][3]. The idea in MOOC platforms is to test the user's knowledge after each video sequence. This allows self-evaluation and students can decide to review the topic of the video depending on their self-test results. In an evaluation most students agree that self-tests help them to test their knowledge [4]. Besides the self-evaluation self-tests are an opportunity to check students' knowledge in a MOOC course with thousands of learners easily and automatically. These advantages should be transferred to lecture video archives. Some adaptations are necessary since there is no course structure like in MOOCs and the videos are longer. So, we have decided to use an in-video approach which is described in the following Section 3 to overcome these challenges.

Before showing the video and integrating self-tests the video has to be recorded with a recording system or software. Since this work is based on a dual-video approach a specialized recording system called tele-TASK is used which offers recording of two videos synchronously. The first dual lecture video recording by tele-TASK was introduced in 2002. Typically, one screen shows the lecturer and the other one shows the lecture slides [5]. An advantage of the tele-TASK system which is used for lecture recording at the Hasso Plattner Institute (HPI) in comparison to other systems is the independence from platform and presentation software and contents. It does not matter if the lecturer uses Windows, Linux, OS X and also no specific presentation tool like PowerPoint [6] is necessary.

Additionally, a player is necessary which allows the synchronous playback of the dual videos. In the beginning the technical background of this player was the Real Media Player with a Real Media Server serving the videos. Over the time the player was replaced with a more feature-rich Flash player application. This also led to the introduction of the Adobe Flash Media Server to serve the videos correctly [6]. In addition to the synchronous web player podcasts are offered which are videos where the lecturer video overlays the slides. As result only one video is created which can be played also on mobile devices [6]. The newest used technique of the video player is the HTML5 technology to play lecture videos synchronously [7]. Due to HTML5 technology the video playback is available natively by the browsers and no additional browser plugins are necessary anymore [8]. For serving the videos the already existing Flash Media Server can be used with enabled HTTP pseudo-streaming. The approach in this paper is to use the HTML5 technologies to embed self-tests in the player. The main reasons are the easy integration with the common web technologies HTML and JavaScript and the native browser support.

Further related work shows the possibility to integrate additional features like chapters, slide preview, adjustment of the playback speed and the full screen mode [7]. These approaches and related work show that the integration of additional features is possible with HTML5 technologies also for this approach. This highlights the feasibility of this approach with HTML5.

### **3 APPROACH**

The idea of this approach is to enhance the current HTML5 player in a video archive to achieve more user motivation and interaction with the user. The current player is shown in Fig. 1. Besides the playback of one video it also offers a dual-screen playback. Typically one screen shows the lecturer and the other one shows the recorded slides. Both video files are played back synchronously. Furthermore, the player offers a slide preview. Above the timeline there is a "slideline" which matches the slides shown in the corresponding time frame of the timeline. When hovering the cursor over "slideline" blocks a preview is shown as an overlay. Furthermore, when clicking on a block of the "slideline" the video jumps to the point in time this slide is shown first. Another feature of the player is the possibility to change the video playback speed. If a lecturer held a lecture speaking very slowly, it can be useful to speed up the lecture to save time. Or when a lecturer speaks very fast or the students have problems to understand everything due to the high speed the video can be slowed down. Finally, also chapters can be shown and a jump between several chapters is possible by a click on the chapter name.



## Improved Opportunities for Intrusion

- Rapid adoption of computer and network technology
- Internet explosion and E-commerce
- Thousands of exploitable vulnerabilities
- Lack of awareness regarding information security



Fig. 1: Basic HTML5 player to play two videos synchronously.

### 3.1 Architecture

To realize self-tests in the HTML5 player of the video archive some additional changes have to be made to the system like shown in Fig. 2. As base we use the already existing web server. In our case it is an Apache server with Python and a MySQL database. An addition to the infrastructure for the realization is an administration area for creating, handling and evaluation of self-tests. Furthermore, a REST API is going to be implemented as additional service for the web server to provide self-tests for the player and student submission handling which have to be implemented within the player. Finally, the HTML5 player has to be customized to provide the ability to show self-tests which are hosted on the web server as it is visualized in Fig. 2.

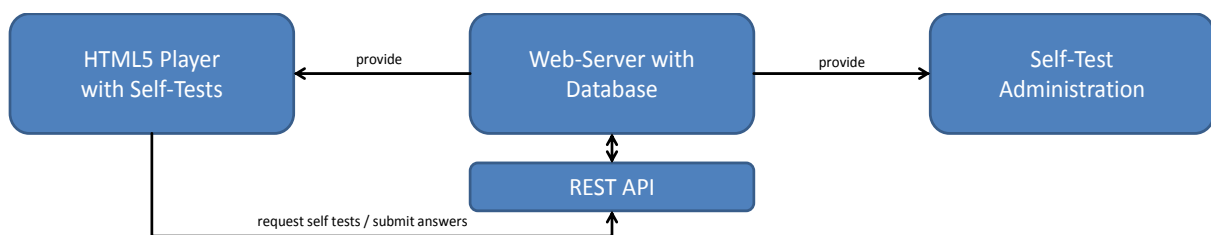


Fig. 2: System architecture to realize the HTML5 self-tests.

#### 3.1.1 Self-Test Administration

In the self-test administration area responsible teachers can create quizzes and corresponding wrong and right answers for 3 different types of self-tests. Possible self-test types are multiple-choice, multiple-answer or text answer. In multiple-choice self-tests there is only one correct answer and several wrong answers. In multiple-answer self-tests there can be several right and wrong answers. When applying text self-tests, there is one right answer which has to be typed in a textbox. There is no display of possible correct or wrong answers users can choose from. After the teacher creates self-tests for a corresponding course he or she also has to set a display time. This time is used to show the self-test at a specific point in time. When the video is played and the set time is reached the video is

going to be paused and the self-test shows up. When the user jumps within a video to a time which is after two earlier self-tests both self-tests will be shown one after the other.

### 3.1.2 REST API

The REST API is the connection point for AJAX calls of the HTML5 player to gather self-test information of the currently viewed lecture. This API offers two functions: the “get\_lecture\_selftest” and “submit\_lecture\_selftest” function. “get\_lecture\_selftest” offers all available self-tests for the selected lecture with the corresponding times in JSON format. The function “submit\_lecture\_selftest” receives the selected self-test result of the student and then evaluates it against the teacher’s target. This will result in a JSON response which contains information about the correct and wrong answers. Furthermore, the results are saved to the database so that statistics can be created which then can be shown in the self-test administration area. Currently, only a distinction between right or wrong is shown in the statistics and how often which answer was selected. Nevertheless, the students get a detailed result which of the selected results are right and which are wrong.

### 3.1.3 HTML5 Player with Self-Test Support

The HTML5 player has to be customized for self-tests. First, an additional overlay has to be created which can be shown to display the question above the current video. Furthermore, another additional visual element of the player is the checkbox which is marked red in Fig. 3. Its purpose is to enable and disable the self-test functionality to let students decide independently to use this feature. Finally, the HTML5 player uses JavaScript to access self-tests available on the server in JSON format via the REST API. Another task realized with JavaScript code is to manipulate the playback and to show or hide the self-tests according to the user’s settings and video time.

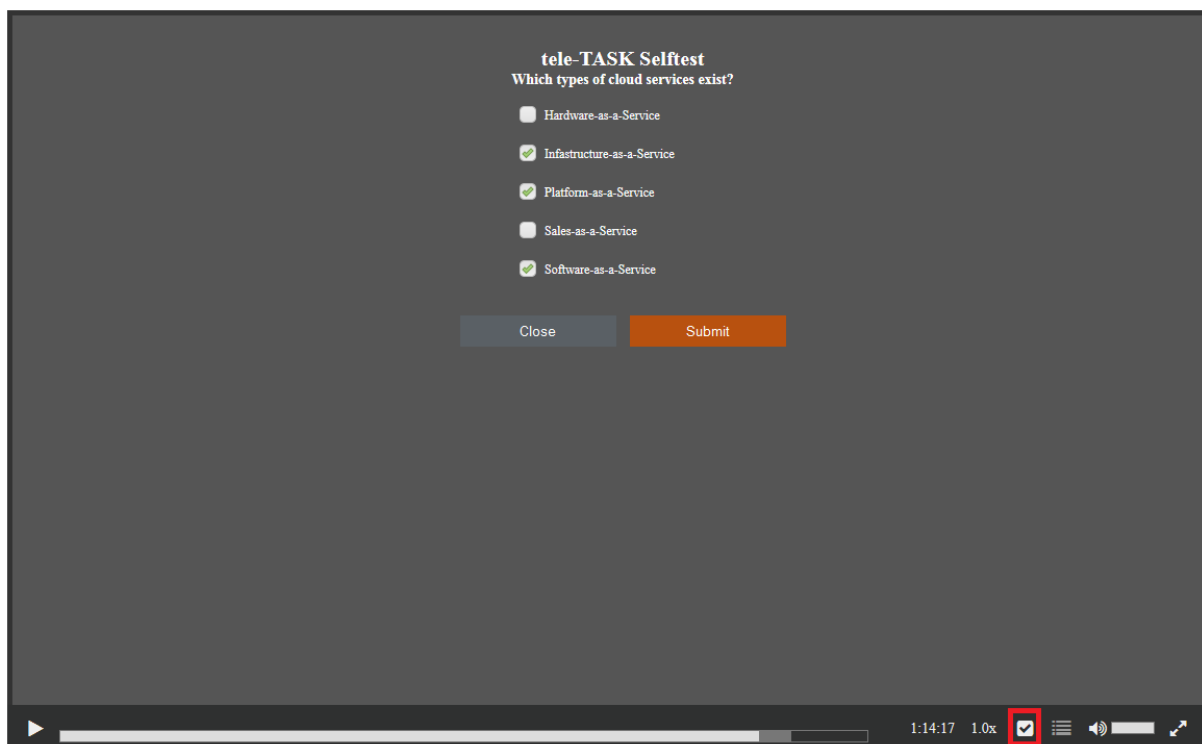


Fig. 3: HTML 5 player showing multiple-answer integrated self-test.

## 3.2 Quiz Results

The administration area also offers statistics for every lecture and every quiz. The view of a statistic for every lecture is shown in Fig. 4. In this view all questions are listed in a column chart displaying the number of wrong and right answers for the specific question. When hovering the mouse over this chart the detailed numbers of every quiz is visualized in a new box. Next to the "per quiz" results is a pie chart which displays all correct and wrong answers given for the whole lecture.

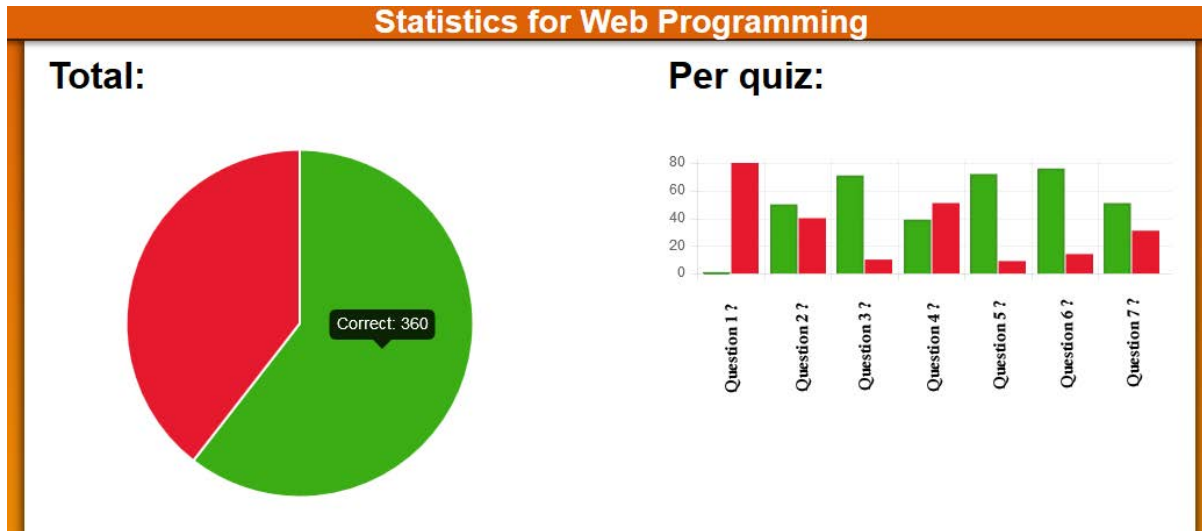


Fig. 4: View of lecture quizzes results.

Furthermore, also a detailed statistic for every quiz is available. An example is shown in Fig. 5. On the left the total number of completely right and wrong answers is shown. Correct answers are answers where all right statements are selected and none of the wrong statements are selected. Even when there is only one wrong statement selected wrongly, the overall answer is wrong. This rule is also effective when the quiz is a multiple-answer quiz. When the mouse is hovering over the pie chart the detailed number is shown in a small box. Next to the total number there is a column chart which indicates how often an answer was chosen by the users. In multiple-answer quizzes, every time an answer is selected, it will be counted up in the user selection.

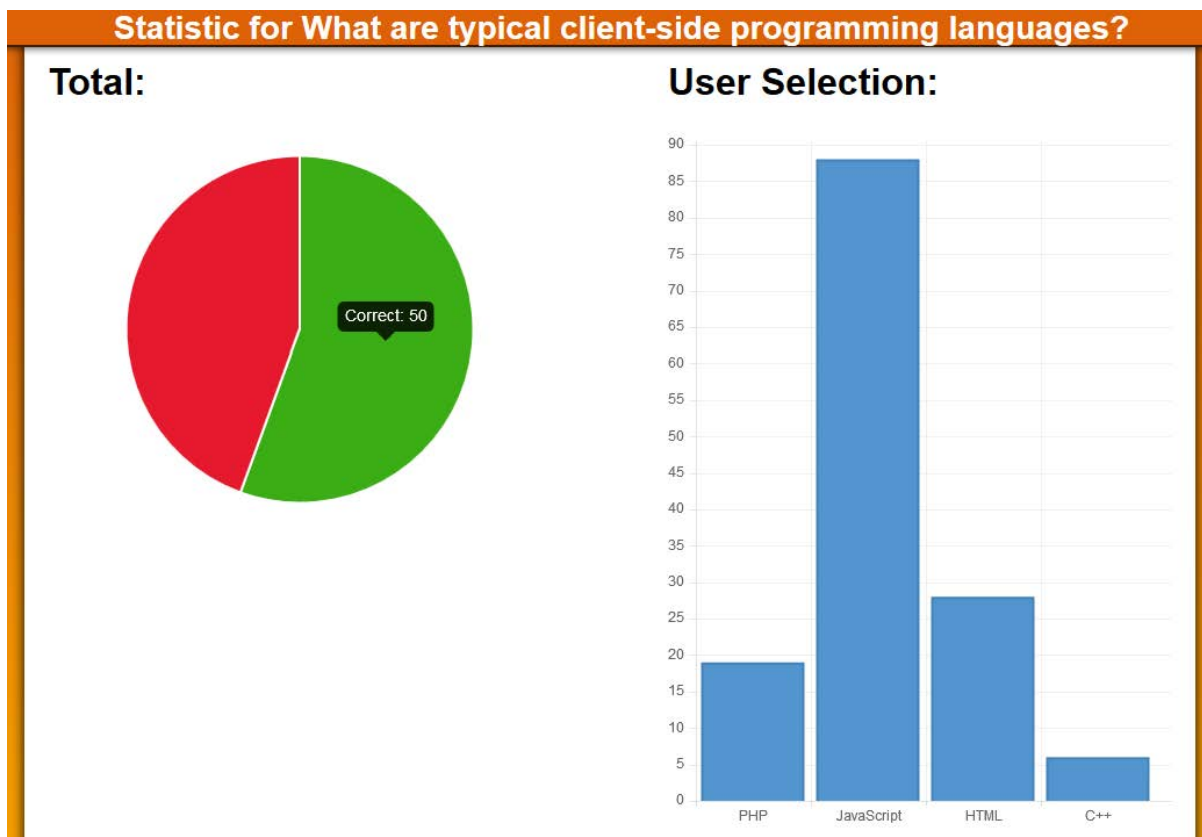


Fig. 5: View of quiz results.

## 4 EVALUATION

This approach was tested within a homework assignment at a university course. Around 90 students answered several questions as part of the homework. The questions were in the area of internet technologies. In our case three to five self-tests per lecture were added to the lectures “HTML”, “Web Programming” and “Cloud Computing” which then had to be solved by the students.

As a result, two thirds of the given answers were correct. The detailed numbers of correct and wrong self-test answers are shown in Fig. 6. The results fluctuate extremely. Nevertheless, this is a good result. This means that some of the topics were not well understood by students, yet. So, the students have the possibility to get direct feedback so that they know which topics they will have to learn more intensely for the exam, if they did a mistake. Also the teacher can evaluate which topics should be repeated in the next on-site tutorial.

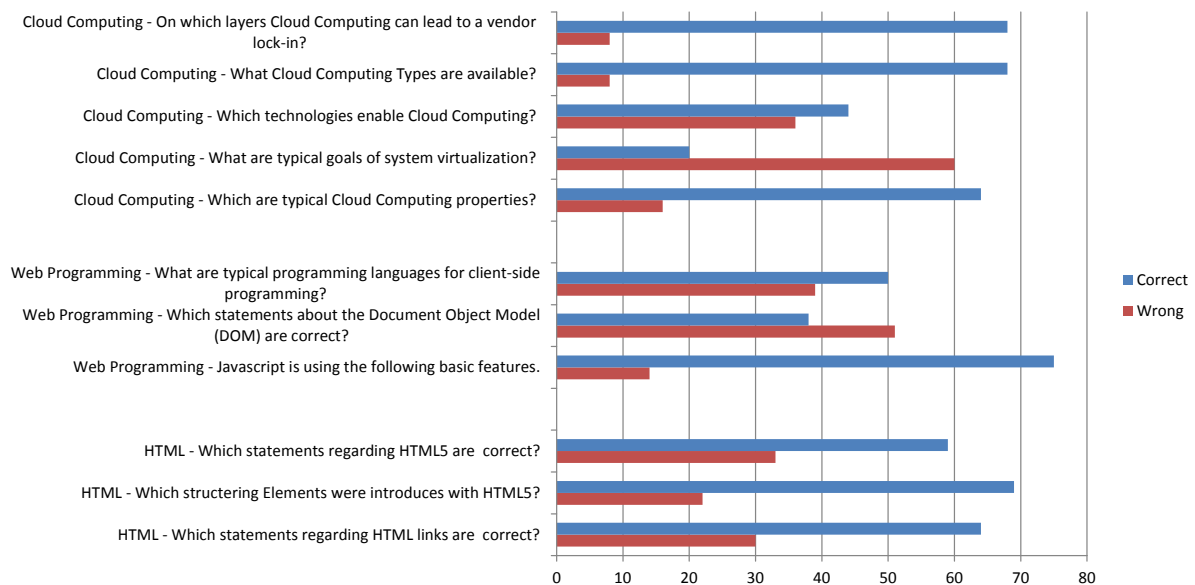


Fig. 6: Self-tests used for evaluation (questions are translated).

Besides answering self-tests 80 students evaluated their user experience with self-tests in the HTML5 video player. They answered four questions about their experience and the player's features. The questions and the given answers are shown in Fig. 7. As an overall result, the students like the self-test feature integrated in the player. 71 students think the approach is helpful to check their knowledge during video playback. Only eight students have the feeling that there is no advantage and one person feels distracted by self-tests. Furthermore, 70 students would like to have self-tests available in other lectures, too. Only 10 students do not want to have this feature enabled in other lectures. When it comes to the questions if integrated self-tests enable students to follow this lecture more easily 50 students think it is easier to follow the lecture due to the reason they actively participate in the recorded lecture. 26 students are not able to decide if they can learn better with the integrated self-tests or not. Only 4 people do not like the interruption of the lecture and think it is more difficult to learn. For those students the player also offers the option to disable self-tests. Students also evaluated this functionality. This was a multiple-answer evaluation. So students could decide for more than one possible answer. In conclusion, the option that students can decide by themselves if they want to see self-tests or not received 94 votes. Another reason for disabling the self-test feature is that self-tests will not be shown if the user has already solved them earlier.

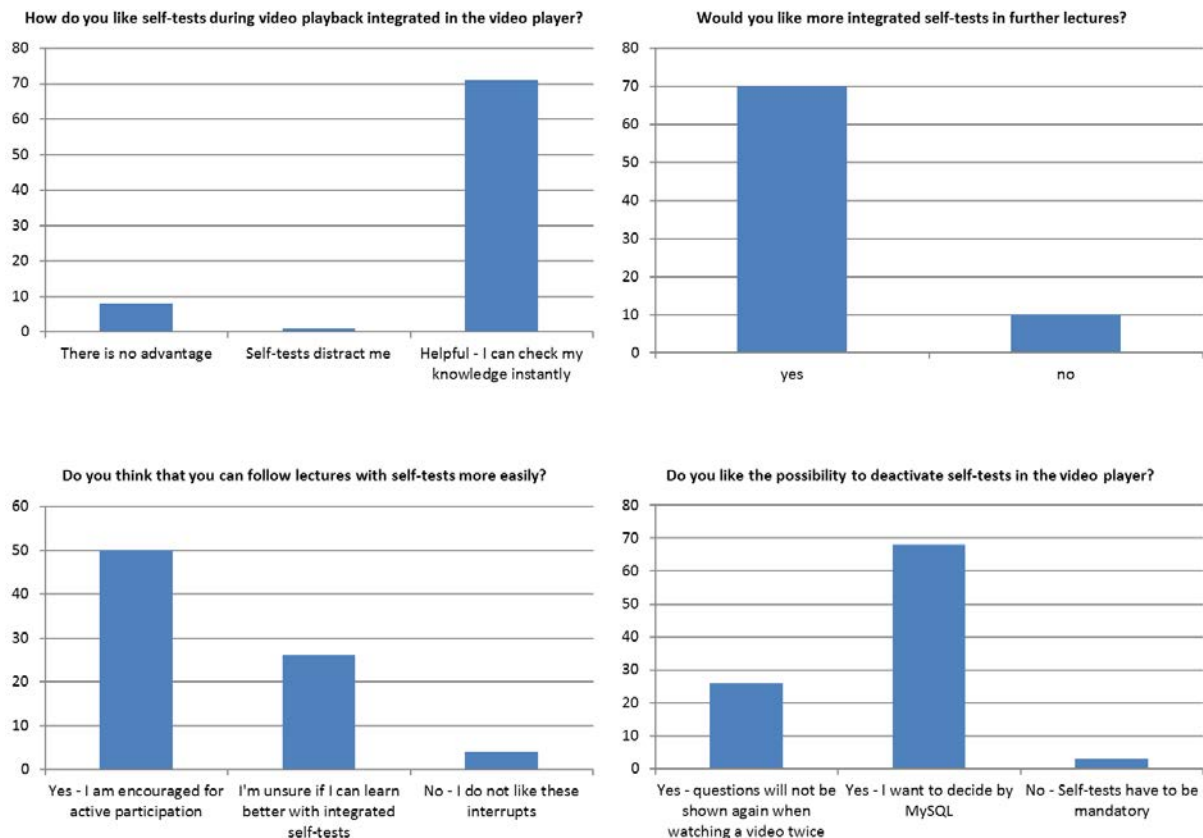


Fig. 7: Student evaluation of the HTML5 player with self-test functionality

## 5 RESULTS AND FUTURE WORK

The result of this work is that integrated self-tests are very useful for students to increase learning experience. Students enjoy the direct feedback. Also for teachers there is a direct feedback of the knowledge and problems students have in certain lecture topics. Another reason to implement this feature in lecture archives is that students like it and they think their motivation is higher. Furthermore, they enjoy the direct feedback of their knowledge and they can jump back if they missed something important. Nevertheless, the experiment showing self-tests every time the video is requested is not reasonable for students. This has to be considered when implementing this feature. Otherwise students could become stressed out by self-tests and would avoid to watch videos again on this platform. In conclusion, self-tests can enhance the user experience and support students during learning in lecture video archives.

Self-tests can be extended with an interactive live-mode feature that displays questions already while the lecture is being held. Students can answer these questions with a mobile device or during the live stream directly in the player.

Finally, also an integration of existing MOOC self-tests can be implemented with a common data exchange format e.g. XML. This will reduce the effort of self-test creation for the teaching team of a course. The same approach is reasonable when a course of a lecture video archive takes place more than once, e.g. recurs every year because the questions can be reused easily. The necessary adaption is slight time adjusting only.

## REFERENCES

- [1] Malchow, M., Bauer, M., & Meinel, C. (2014, December). The Future of Teleteaching in MOOC Times. In Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on (pp. 438-443). IEEE.
- [2] Staubitz, T., Renz, J., Willems, C., & Meinel, C. (2014). Supporting Social Interaction and Collaboration on an xMOOC Platform. Proc. EDULEARN14, 6667-6677.

- [3] Kay, J., Reimann, P., Diebold, E., & Kummerfeld, B. (2013). MOOCs: So Many Learners, So Much Potential... *IEEE Intelligent Systems*, (3), 70-77.
- [4] Kulkarni, C., Wei, K. P., Le, H., Chia, D., Papadopoulos, K., Cheng, J., ... & Klemmer, S. R. (2015). Peer and self assessment in massive online classes. In *Design Thinking Research* (pp. 131-168). Springer International Publishing.
- [5] Schillings, V., & Meinel, C. (2002, November). tele-TASK: teleteaching anywhere solution kit. In *Proceedings of the 30th annual ACM SIGUCCS conference on User services* (pp. 130-133). ACM.
- [6] Grünwald, F., Yang, H., Mazandarani, E., Bauer, M., & Meinel, C. (2013). Next generation tele-teaching: Latest recording technology, user engagement and automatic metadata retrieval. In *Human Factors in Computing and Informatics* (pp. 391-408). Springer Berlin Heidelberg.
- [7] Bauer, M., Malchow, M., & Meinel, C. (2015). Enhance Teleteaching Videos with Semantic Technologies. In *Smart Education and Smart e-Learning* (pp. 105-115). Springer International Publishing.
- [8] Anthes, G. (2012). HTML5 leads a web revolution. *Communications of the ACM*, 55(7), 16-17.