# Active Learning Impact in Lecture Delivery in The Software Patterns Course

*Dragica Ljubisavljević[*], Milica Bačić[**], Siniša Vlajić [***]*

***Keywords**: Design patterns, Knowledge transfer, Active learning, Online classes, Software engineering*

**Abstract: This paper provides an analysis of the process of lecture delivery in the Software Patterns course at the Faculty of Organizational Sciences, University of Belgrade. The content of the course and the manner of knowledge transfer to students are described in detail. In order to improve the process of delivering lectures at this course, three types of research were conducted among students. At the end of the paper, the results and conclusions of the research are presented.**

## 1. INTRODUCTION

The field of higher education is increasingly facing the challenges of adapting traditional forms of teaching in order to obtain greater engagement of students in this process [1]. Research has shown that greater involvement and engagement of students during the teaching process (related to programming) has led to better development of their programming skills, as well as to greater competencies in the labor market. This is considered a very important aspect of the professional development of students, considering that the next step, after higher education, is the placement of their knowledge and skills in the labor market [2]. The topic of this work is the analysis of the process of delivering lectures in the **Software Patterns** course. The pedagogical model applied in the process of delivery of lectures at the Software Patterns course represents a combination of the traditional manner of teaching and active learning. In the traditional manner of teaching, the engagement of students during the teaching process is minimal. Students are passive and not motivated to actively participate in the teaching process. On the other hand, in active learning, students are directly involved in the teaching process through continuous evaluation and testing of their knowledge. Active learning can be seen as a series of instructions given to students, which implies their active involvement in the teaching process [3]. The concept of active learning leaves the possibility for professors and

[*] Dragica Ljubisavljević is with the Faculty of Faculty of Organizational Sciences, University of Belgrade, Serbia (phone: 381-65-8222320; e-mail: dl20223706@student.fon.bg.ac.rs).

[**] Milica Bačić is with the Faculty of Faculty of Organizational Sciences, University of Belgrade, Serbia (phone: 381-64-5485517; e-mail: mb20223704@student.fon.bg.ac.rs).

[***] Siniša Vlajić is with the Faculty of Faculty of Organizational Sciences, University of Belgrade, Serbia (phone: 381-69-8893133; e-mail: sinisa.vlajic@fon.bg.ac.rs).

associates to adapt it to their own teaching method, as well as to encourage students to be actively involved in the teaching process, through their own predefined set of activities [4]. Additionally, it can be observed that active learning grows into an imperative for studying software technologies over time, because based on a lot of research, it was concluded that the effectiveness of students during the process of acquiring knowledge and the quality of acquired skills rapidly increases [5].

Software Patterns is an elective course of the **Information Systems and Technologies** study program at the Faculty of Organizational Sciences, University of Belgrade. This course is taught in the scope of the seventh semester, at the fourth year of study. The aim of the course is to understand the definitions and different forms of patterns, the possibility of practical use of patterns in different stages of software development, and familiarization with the existing mathematical formalisms for describing patterns. Taking the exam in this course consists of three parts:

*A. The written part of the exam – it is taken in person on a computer and refers to the material from the exercises.*

*B. The oral part of the exam – it is taken in person in the form of a test and refers to the material from the lectures.*

*C. Seminar paper (not mandatory) – it is taken in person, through the explanation of personal examples that are listed in the seminar paper.*

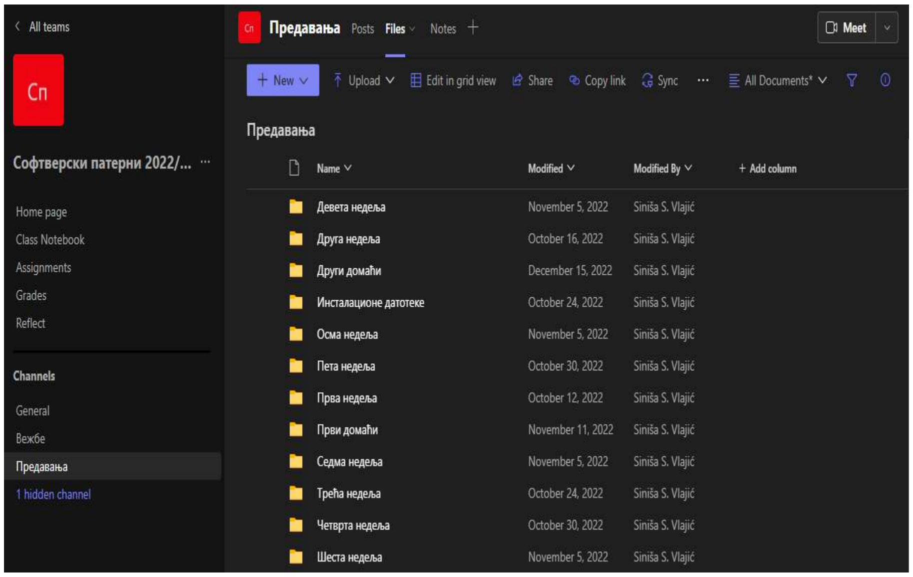
In Chapter 2, a detailed description of the lecture delivery at the Software Patterns course is given. In order to discover new methodologies that can be applied to improve the teaching process, as well as to take into consideration the opinions of students on possible changes in this process, three pieces of research were conducted among students. The afore mentioned research and the results will be presented in Chapter 3. Chapter 4 presents the conclusion of this scientific research work and points to further directions for the improvement of lectures in this course.

# 2. THE PROCESS OF DELIVERING LECTURES IN THE SOFTWARE PATTERNS COURSE

Lectures of the Software Patterns course are held in person, in the computer rooms of the faculty. In addition to in-person classes, a channel has been created on the MS Teams platform including the professors and associates engaged in this course, as well as the students attending it. Within the channel, there is a section "Lectures" in which notices and materials relevant to students are posted. Additionally, within this channel, there is a link placed for students, using which they can download the online version of the book for this course for free, the author of which is Prof Siniša Vlajić, Ph.D., the main and responsible professor for this elective course. In the Lectures/Files/ folder, there are folders named according to the number of the week of the lecture to which they refer. Within the corresponding week, there is a Word document containing the curriculum for the given week organized by classes, as well as folders with the programming code and video materials that are aligned with the curriculum. Materials for the upcoming work week are available in advance so that students could have the opportunity to take a look at the material that will be covered in advance and prepare for the next lecture. Video materials were created during the period of online classes due to the Coronavirus pandemic. After the teaching mode returned to live teaching, these materials have been used as additional help for students to master the curriculum studied in this course.

The lectures are held for 9 work weeks and the thematic units covered within individual weeks are the following:

*1st week:* Fundamentals of patterns
*2nd week:* Program concepts of a design pattern
*3rd week:* Independent work on tasks and creational patterns (Abstract Factory, Builder)
*4th week:* Creational patterns (Factory Method, Singleton, Prototype)
*5th week:* Structural patterns (Adapter, Bridge, Composite)
*6th week:* Structural patterns (Decorator, Facade, Flyweight, Proxy)
*7th week:* Behavioral patterns (Chain of Responsibility, Command, Interpreter, Iterator)
*8th week:* Behavioral patterns (Observer, Strategy, State, Template Method)
*9th week:* Behavioral patterns (Mediator, Memento, Visitor)

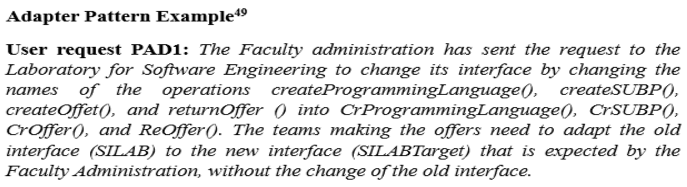


| Name | Modified | Modified By | + Add column |
|---|---|---|---|
| Девета недеља | November 5, 2022 | Siniša S. Vlajić | |
| Друга недеља | October 16, 2022 | Siniša S. Vlajić | |
| Други домаћи | December 15, 2022 | Siniša S. Vlajić | |
| Инсталационе датотеке | October 24, 2022 | Siniša S. Vlajić | |
| Осма недеља | November 5, 2022 | Siniša S. Vlajić | |
| Пета недеља | October 30, 2022 | Siniša S. Vlajić | |
| Прва недеља | October 12, 2022 | Siniša S. Vlajić | |
| Први домаћи | November 11, 2022 | Siniša S. Vlajić | |
| Седма недеља | November 5, 2022 | Siniša S. Vlajić | |
| Трећа недеља | October 24, 2022 | Siniša S. Vlajić | |
| Четврта недеља | October 30, 2022 | Siniša S. Vlajić | |
| Шеста недеља | November 5, 2022 | Siniša S. Vlajić | |

Fig. 1. Material organization on MS Teams platform

Within the independent work on tasks that takes place in the fifth lesson, students are assigned ten programming tasks for testing their knowledge related to the general concepts of patterns and object-oriented programming. The results achieved when solving these tasks are recorded in points for additional student activities. *In this way, the active participation of students during lectures is encouraged.* The curriculum that is processed related to a specific pattern includes an explanation of the definition of the given pattern and a presentation and clarification of the structure of the pattern with a corresponding class diagram. The definition and structure are accompanied by two examples and one task that students create independently during lectures, after listening to the examples. *In this manner, a combination of traditional (explanation of examples) and active learning* *(creating a solution to the assigned task) is performed.* Both examples shown to students keep the same domain problem through each lecture, and their structure and specific user request change depending on the software pattern.

The first example refers to the domain model of communication between the Faculty Administration, the Head of the Software Engineering Laboratory, and its employees, and this example is less complex (Fig. 1).


**Adapter Pattern Example**[49]

**User request PAD1:** *The Faculty administration has sent the request to the Laboratory for Software Engineering to change its interface by changing the names of the operations createProgrammingLanguage(), createSUBP(), createOffet(), and returnOffer () into CrProgrammingLanguage(), CrSUBP(), CrOffer(), and ReOffer(). The teams making the offers need to adapt the old interface (SILAB) to the new interface (SILABTarget) that is expected by the Faculty Administration, without the change of the old interface.*

Fig. 2. User request of the study example

The second example is more complex and is a model according to which students should create their own examples as part of their homework (students are given guidelines on what needs to be changed when creating their own example). The domain of the second example refers to processing applications for the exam using a desktop application consisting of a display form, a controller, and a database broker. The structure and organization of the code change depending on the pattern being processed.

The solution of the user request example (Fig. 1), as well as a more complex example, which refers to a software system[6] for processing exam applications, are available to students for download on the Teams platform. Students are expected to download these examples and run them on their computers during lectures. Within these materials, there is also a task for independent work. What is characteristic of each more complex example is the organization of the programming code. In order for students to intuitively understand which part of the code refers to a specific element of the structure of the software pattern, the code is divided into folders of the same name. Furthermore, the classes within the code are named the same as the given classes in the example class diagram.

As already mentioned, the second example within the lecture for a specific pattern is used as a model for doing the individual example. Given that patterns are divided into three groups: creational patterns, structural patterns, and behavioral patterns, the students create their examples in three phases. The first two groups refer to the preparation of the first and second homework assignments, while the preparation of the third group of individual examples (behavioral patterns) requires unification with the first two homework assignments, and the final result is a seminar paper. The seminar paper must follow the appropriate structure and instructions, and guidelines for its creation are available for students to download. In addition, the instructions for making an individual example can be found within the Documents folder of the given project. The homework assignments and the seminar paper are not mandatory, but they carry 20% of the grade, which can only be obtained in this way.

The oral part of the exam, which carries half of the grade, is passed through two theoretical tests. The first test is organized in the middle of the semester in which this course is attended, while the second test is done at the end of the semester.

## 3. EVALUATION OF THE RESEARCH CONDUCTED ON STUDENTS

In order to improve the teaching of this course, three pieces of research were conducted on students. Their description and results are presented below.

*Research 1*

On December 2, 2022, within the term of lectures of the Software Patterns course, a survey was conducted among students in order to examine the *satisfaction of students with the delivery of lectures in this course.*

The questions were designed to examine students' satisfaction with the lectures, their perception of the knowledge they acquire in this elective course, and the time they spent mastering the material covered in the lectures.

Additionally, the reasons why students decide to take this course were examined, given that it is an elective course within the seventh semester of basic academic studies.

In order to maintain the continuous improvement of teaching in this course and take into account the opinions of students, a section was added to the survey that allowed current students to express their opinions and advice so that the teaching of this course could be improved.

Given that the lectures are not only a scientific but also a social process, the survey highlighted questions regarding the evaluation of the relationship between the teachers in this course and the students attending the course.

The goal was for students to express their honest observations without any reservation, so the survey was conducted anonymously, using the Google Forms platform.

The survey contains 10 open-ended and closed-ended questions [7].

The survey was filled out by 18 students out of 33, which is the total number of students taking this course.

The survey results showed the following:

- 94.4% of the surveyed students answered "Yes" to the question of whether they attended classes regularly
- On average, students spent 4.2 hours a week preparing for the lectures from this course
- 94.4% of surveyed students always or occasionally used pre-loaded video materials to prepare for the upcoming week's lectures
- 88.9% of the surveyed students had no prior knowledge of software patterns before taking this elective course
- The majority of surveyed students (sixteen students) expressed the practical usefulness of the material covered in the course as a reason for choosing it, while few provided additional reasons such as the recommendation of older students, the presentation of the course, improvement of the knowledge they already have in the field, and teachers and associates who are engaged in this course
- As the most useful elements of the lectures, the students pointed out the tasks done at the end of the lectures and the tests during the lectures

- Areas that students feel they can master without difficulty are the following: recognizing spaghetti code, adapting existing code structure to an appropriate pattern structure, and applying software patterns in the development of new projects
- All students believe that the lectures and video materials are aligned with the material on the tests related to the oral exam
- In general, the majority of students are satisfied with all the mentioned aspects of the relationship between students and teachers
- As the main proposal for improving the lectures in this course, the students state the need to add more diverse examples of the application of software patterns, which will be accompanied by the programming code

*Research 2*

In addition to the survey, in the sixth week of classes, students were given a test in the form of an assignment within the MS Teams platform. The test is conceived to contain eight questions for recognizing the correct answers and two open-ended questions. *The goal of this testing of students is to check how durable the knowledge they acquire during lectures is*, given that the test took place a week after the first theoretical preliminary exam. The test was taken by 24 students out of 33 in total. The maximum number of points that could be achieved is 5 (each question carried 0.5 points). The results show the following:

- 17 students achieved a maximum of 5 points
- 2 students achieved 4.5 points
- 4 students achieved 4 points
- 1 student achieved 3.5 points

From these results, it can be seen that the majority of students completed the test with more than 90% success, which shows that the knowledge of the basics of software patterns that they acquired is permanent.

By conducting the test one week after the colloquium week, an attempt was made to determine whether students had memorized the material intended for the colloquium. This research can be further improved by repeating the same test at specific intervals such as the end of the semester and the end of the school year to examine the long-term retention of knowledge gained in the course.

*Research 3*

In addition to the research that was conducted among the students, research was also conducted on the success of passing the oral part of the exam by year. The school years for which the data are comparatively presented are 2018/19, 2020/21, 2021/22, and 2022/23. With this analysis, it is possible to observe the regularity of the success of passing the exam before, during, and after the online classes that were organized due to the pandemic of COVID-19 virus.

The data on the success of passing the elements of the exam are shown in Table I.

Table I
Data on the success of passing the test

| **School year** | **Number of students** | **Students who did not take the test (%)** | **Between 25 and 35 points (%)** | **Between 35 and 45 points (%)** | **More than 45 points (%)** |
|---|---|---|---|---|---|
| 2018/19 | 42 | 35.73% | 11.90% | 11.90% | 40.47% |
| 2020/21 | 34 | 17.66% | 5.88% | 38.23% | 38.23% |
| **2021/22 (online)** | **33** | **24.25%** | **24.24%** | **30.30%** | **21.21%** |
| 2022/23 | 33 | 45.46% | 6.06% | 15.15% | 33.33% |

Based on the obtained results, the following conclusions can be drawn:

1. *During online classes, the success rate of passing tests decreased significantly, compared to the period before and after the Coronavirus.*
2. *The success of passing tests before and after the Coronavirus period is at a similar level.*
3. *The percentage of students who did not take the test and who did not attend classes regularly increased after the period of online classes.*

We can conclude that after the period of online classes, the number of students who attended lectures and took tests decreased. However, students who were active in the current school year (regular classes) achieved better results on tests compared to the previous year (online classes). One of the reasons for the decrease in class attendance and test-taking is the availability of materials for lectures in the online form. Too much availability of materials leads to the demotivation of students to actively participate during the teaching process. This leaves room for the improvement of the teaching process in order to determine the balance of materials that will be available to students in advance or outside of the lecture dates.

## 4. CONCLUSION

Different programming methods are intended to promote good code design, encourage developers to adhere to good structures, and allow for proper standardization of this process. Software patterns have made it possible to reuse abstractions of appropriate solutions to programming problems in appropriate situations [8].

The aim of the lectures in this course is precisely to show students that it is a good practice to introduce existing solutions of software patterns as standardization for writing their future code.

From the conducted analysis of the lecture delivery process, it can be observed that the process itself can be viewed as a pattern that has been successfully applied for years, with appropriate variations. Research has shown that the newer generations of students attach increasing importance to the concept of active learning and that they tend to adopt new concepts through various practical examples. This leaves an opportunity to improve the teaching process through active student learning during lectures.

One of the ways of providing more active participation of students in the process of delivering lectures, based on the proposal of students from *Research 1*, is a catalog of practical examples of the application of software patterns that can be created as additional material for students. Each example from the catalog would contain an explanation of the example and programming code whose structure is adapted to the suitable pattern.

Further directions of research refer to the creation of metrics for evaluating pedagogical models applied in teaching. The aim of the research is to create unique attributes for the evaluation of the pedagogical model, as well as mathematical formalisms that would allow quantitative evaluations of these parameters. In this way, appropriate uniformity would be enabled in the process of analyzing the success of teaching in any course.

## REFERENCES

[1] S. Hartikainen, H. Rintala, L. Pylväs, & P. Nokelainen, "The Concept of Active Learning and the Measurement of Learning Outcomes: A Review of Research in Engineering Higher Education." in *Education Sciences*, vol. 9, 2019, pp. 1-3.
[2] ED Justo, & A. Delgado, "Change to Competence-Based Education in Structural Engineering." in *Journal of Professional Issues in Engineering Education and Practice*, vol. 143, 2015, pp. 1-8.
[3] M. Prince, "Does Active Learning Work? A Review of the Research." in *The Research Journal for Engineering Education*, vol. 93, 2004, pp. 223-231.
[4] M. Prince, & R. Felder, "Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases." in *The Research Journal for Engineering Education*, vol. 95, 2006, pp. 123-138.
[5] M. R. Lima, P. Hammar Andersson, & E. Saalman, "Active Learning in Engineering Education: A (re)introduction." in *European Journal of Engineering Education*, vol. 42, 2017, pp. 1-4.
[6] Ljubisavljević, D. (2022). Adapter[Source code]. https://github.com/dragicalj/adapter
[7] Ljubisavljević, D. (2022). Anketa. https://github.com/dragicalj/anketa
[8] E.Gamma, R. Helm, R. Johnson, & J. Vlissides, "Design Patterns: Abstraction and Reuse of Object-Oriented Design", Addison-Wesley, 1994. pp. 1-2.