HADMÉRNÖK
FÓRUM
19. évfolyam (2024) 4. szám 213–232.
DOI: 10.32567/hm.2024.4.15

Zlatko Čović[1]

New Approaches in the Education of Software Engineers in the Field of Cybersecurity

Abstract

This paper presents innovative methodologies for cybersecurity education, with a focus on their application at Subotica Tech – College of Applied Sciences. The study highlights hackathon-based learning (HBL), challenge-based learning (CBL), and the integration of Artificial Intelligence (AI) tools to enhance teaching effectiveness. These approaches are incorporated into professional courses related to the development of web applications and integrated web systems, emphasising practical work and real-world scenarios to better prepare students for industry challenges. The primary objective of this paper is to introduce these novel educational approaches and provide a detailed description of their implementation, underscoring the importance of hands-on experience and student engagement in real-world systems and business environments. The results of the research on the impact of hackathon participation on final exam performance are presented. Additionally, the increasing use of AI tools in software engineering education is explored, with examples of their application. Finally, the paper outlines future research directions based on feedback from the current implementation of these methodologies.

Keywords: cybersecurity, software engineering, practical engineering education, hackathon, challenge-based tasks, AI

Introduction

In the rapidly evolving digital landscape, where technology plays an integral role in nearly every aspect of modern life, the importance of securing online systems cannot be overstated. Every day, huge amounts of information are exchanged through websites, applications, and other information systems. This information exchange is crucial to

[1] Subotica Tech – College of Applied Sciences, e-mail: chole@vts.su.ac.rs

the operation of modern digital systems, making secure data transfer essential. In addition to securing the exchanged data, it is equally important to ensure the safety and reliability of the systems themselves. As cyber threats continue to evolve, it is increasingly important for software and cybersecurity engineers to be well-versed in identifying potential security risks and vulnerabilities, while also possessing the knowledge and skills needed to address these security problems effectively.

Higher education should follow present trends and listen to the needs of industry, continuously adapting and improving curricula accordingly. In addition to the traditional model of engineering education, which includes theoretical lectures and practical laboratory exercises, it is essential to develop and implement new teaching methods that will contribute to the better development of key competencies in students, as well as enhance their competitiveness in the job market. Furthermore, it is crucial to find ways to integrate industry into the educational process, enabling students to directly engage in real-world challenges and market needs, thereby increasing the relevance and applicability of the knowledge acquired.

The importance of hands-on and practical methodologies in cybersecurity education has been extensively emphasised in recent research. For instance, authors analyse the educational value of Capture the Flag (CTF) challenges, demonstrating how they align technical knowledge areas like cryptography and network security with established curricular guidelines.[2] However, the authors highlight the need to incorporate non-technical aspects, such as social engineering, to address advanced cyber threats comprehensively. Similarly, authors in the study identify learning obstacles in CTF-based education, such as a steep learning curve for beginners, and propose varying difficulty levels and adequate support to enhance their effectiveness.[3]

Further, authors explore the role of serious games in enhancing cybersecurity skills. Their findings show that game-based learning can maintain student engagement while balancing educational objectives.[4] Additionally, authors in study advocate for using CTF competitions as an introductory tool for cybersecurity education, emphasising their potential to motivate students and providing practical exposure to key concepts.[5]

Beyond CTFs, authors investigate the integration of hackathons into online cybersecurity courses. Their study reveals that hackathons enhance collaboration and practical understanding, supporting teamwork and maintaining student engagement through real-world problem-solving.[6] These findings align closely with the methodologies implemented at Subotica Tech – College of Applied Sciences, where hackathon-based learning and challenge-based tasks are integrated into professional courses to prepare students for real-world challenges.

In the study by the authors Čović et al., the focus is on evaluating the effectiveness of the hackathon approach in the training of software engineers. This student-centred method employs a constructivist pedagogical framework. Throughout a hackathon, students work closely with programmers and other software development experts

[2] Švábenský et al. 2021.
[3] Chung–Cohen 2014.
[4] Vykopal et al. 2020.
[5] McDaniel et al. 2016.
[6] Obot Affia et al. 2022.

on collaborative projects. This approach enables students to acquire new knowledge, enhance their skills, and develop essential competencies.[7]

Another paper discusses the use of Challenge-Based Learning (CBL) in cybersecurity education, where challenges were based on students' interests in securing systems.[8] Students worked collaboratively to devise solutions, applying their knowledge in two cybersecurity competitions. Formative assessments revealed that while the benefits varied among students, skills in computer security, teaching, and interest in cybersecurity were enhanced. Despite the need for additional resources and flexible meeting schedules, the increased learning outcomes justified the effort.

As authors in Žagar et al. have shown, research in software engineering (SE) indicates that recent graduates are often unprepared for workplace challenges.[9] The authors presented two approaches used in SE courses at the University of Zagreb: a distributed project-based course where Croatian and Swedish students collaborate throughout the software development life cycle, solving technical and cultural problems, and an approach focused on self-constructing knowledge, presenting it to peers, and concluding with student discussions and a group project. Both approaches foster the development of soft skills, which are often underrepresented in SE education.

Authors in a study investigated challenges in teaching software engineering through a survey of 21 faculty and experts. Findings revealed that student engagement was the most significant challenge, with nearly half of respondents also reporting difficulties in designing practical activities.[10] Problem-based learning was the most used approach, followed by newer methods like gamification and role-playing, both shown to enhance engagement. Based on these insights, a conceptual model for improving student involvement is proposed, with future studies suggested to evaluate its impact in practice.

To ensure the security of web applications and information systems, developers must conduct appropriate tests. Raising awareness of potential vulnerabilities and threats is also crucial. The OWASP Top 10, developed by the OWASP Foundation, serves as a standard reference for developers, highlighting the most critical security risks for web applications.[11] This framework should also be incorporated into engineering education.

The primary goal of this paper is to introduce and evaluate innovative educational methodologies designed to enhance the training of software engineers in cybersecurity. Specifically, the study focuses on Subotica Tech – College of Applied Sciences, highlighting the integration of hackathon-based learning, challenge-based tasks, and AI tools. These methods emphasize real-world applicability through industry collaborations and advanced pedagogical techniques.

This paper is organised as follows. The first section provides an overview of engineering education at Subotica Tech – College of Applied Sciences, with a focus on cybersecurity education, including an introduction to the OWASP Top 10 web

7 Čović et al. 2022.
8 Cheung et al. 2011.
9 Žagar et al. 2008.
10 Ouhbi–Pombo 2020.
11 Čović 2024.

application security risks and vulnerabilities and OWASP API Security Top 10. The second offers a detailed description of the new approaches implemented in the education of software engineers, including Hackathon-based learning, challenge-based tasks, and the use of AI tools. The results of the research on the impact of hackathon participation on final exam performance are presented also in this section. Plans are discussed in the next section. In conclusion, the paper summarises the key points discussed throughout.

Engineering education at Subotica Tech

The Subotica College of Applied Sciences, established in 1960, is one of the most respected state higher education institutions in Serbia. For over half a century, the college has educated thousands of engineers in the field of technical and technological sciences, who now form the backbone of the skilled workforce in local and wider industry sectors.

The studies are organised at two levels: undergraduate applied studies (BSc) lasting three years and master applied studies (MSc) lasting two years. The undergraduate programmes offered include Electrical Engineering, Computer Science, Engineering Management, Mechanical Engineering, and Mechatronics. Currently, master's studies are available only in the field of Information Technology, with a focus on security.

At Subotica Tech – College of Applied Sciences, the focus of engineering education is on practical knowledge. In most professional courses, especially within the Informatics programme, teaching is delivered through a Project-Based Learning approach. During these courses, students, either individually or in teams, work on developing a software product (such as a web system, mobile application, or integrated information system), creating projects that include both software and hardware components, as well as managing and maintaining network systems. In addition to practical skills, students acquire theoretical knowledge in these areas, as well as in other foundational and advanced fields. In the final semester of undergraduate and master's studies, students complete a mandatory internship at one of the partner companies and conclude their studies with the preparation and defence of a final thesis.

The collaboration with companies at Subotica Tech involves organising student internships, expert lectures and workshops, joint projects, conference sponsorships, participation in competitions, student scholarships, and mentorship in student projects. All undergraduate and master's students are required to complete internships, often extending beyond the minimum duration.

Subotica Tech makes every effort to prepare future engineers with the necessary theoretical knowledge and practical skills across various engineering disciplines, including informatics. These skills can be further developed through internships and early work experience. The involvement of industry in both curricular and extracurricular activities significantly contributes to the development of key competencies and professional growth in engineering fields.

The College maintains close contact with industry to align study programmes with market needs. The programmes are not strictly tailored to specific job roles but

provide a broad foundation, enabling students to quickly adapt and pursue ongoing professional development. It is crucial that students acquire skills to approach problem-solving logically and apply algorithmic thinking and acquired knowledge to address challenges.

Additionally, new teaching methods are introduced in most professional courses, such as project-based learning, teamwork, hackathons, and challenge-based tasks. The aim is to engage students in real-world scenarios they will encounter in their professional careers.

Education in the field of cybersecurity

The experiences of students and professors gained through participation in various competitions, conferences, and hackathons, as well as strong industry relations, have contributed to the introduction of new methods in software engineering education. Security aspects of information systems have been studied for many years, but with the rise in risks and threats, the amount of coursework focused on cybersecurity has also increased.

Professional courses place a strong emphasis on various aspects of security. Many of these courses incorporate protection methods and techniques in laboratory exercises, employing diverse approaches to test information systems, web applications, mobile apps, and other software and network types.

According to the needs of cybersecurity experts, Subotica Tech participated in the ISSES (Information Security Services Education in Serbia) project under the Erasmus+ programme. The project brought together experts from higher education institutions and companies in the region to improve the higher education capacities in Information Security in the Republic of Serbia. As a result of the project, new courses and cutting-edge laboratories were developed, providing students with practical experience that can be directly applied in the information security industry. Subotica Tech also developed a new laboratory for information security, equipment for network and server room, and an MSC study programme in Information Technology with a focus on information security.

As part of the project, the Serbian Cybersecurity Challenge competition was organised five times to enable student teams to apply their acquired knowledge and gain new awareness in the field of information security. These efforts aim to enhance the competitiveness of students who graduate from participating higher education institutions in Serbia.[12]

It is essential to offer students opportunities to apply their knowledge. Assigning tasks in laboratory sessions and information security-related homework strengthens both their understanding and practical skills.

[12] Serbian Cybersecurity Challenge 2024.

Web application security risks and vulnerabilities

One of the most popular areas for developing various information systems today is the web. It is almost unimaginable for a system or application to lack a component that utilises data and resources from the internet. Therefore, significant attention is devoted to such systems.

Web application security, a critical aspect of information security, focuses on safeguarding web applications, websites, web systems, and web services. It extends the principles of application security, adapting them to address the unique challenges posed by web-based systems and the internet. One major issue in both developing and using web applications is the general lack of knowledge and awareness regarding potential threats. Developers often overlook the implementation of essential security techniques and methods, and they may neglect to perform necessary security assessments on their applications. One of the key aspects in creating secure web applications is the practical application of appropriate methods and technologies, based on theoretical knowledge. Additionally, it is essential to promptly identify security risks and threats and respond to them appropriately.

Categories of web application security risks that are listed in OWASP Top 10 document are:

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery[13]

The release of the OWASP Top 10:2025 is planned to be announced in the first half of 2025.[14] During their studies, students become familiar with most of the risk categories.

With the growing need for integrated systems using APIs, ensuring their security has become crucial. Students learn about API threats and protection methods from the OWASP Top 10 API Security Risks document and web portal. The OWASP API Security Top 10 project highlights key API security risks, helping developers and security professionals recognise and mitigate threats. It provides a Top 10 Risks document and a best practices portal, ensuring up-to-date guidance through collaboration with the security community.[15]

[13] OWASP Top 10 2024.
[14] OWASP Top 10 2024.
[15] OWASP API Security Top 10 2024.

New approaches in education

To allow students to participate in situations like real-life scenarios through the educational process, new teaching methods have been introduced in several specialised courses within the computer science programme, both during lectures and laboratory exercises.

The courses where new methods have been introduced include on the BSc level, Web Programming, Advanced Web Programming, and Integrated Web Systems. At the master's level, the course is Security in e-business systems. One of the innovations was also implemented in the education of students from Óbuda University in Budapest, who spent three months at Subotica Tech as part of the Erasmus exchange programme.

Hackathon-based learning (HBL)

A "hackathon" is a blend of "hack" and "marathon" referring to a 1- or 2-day brainstorming event where participants develop solutions for a specific challenge. Originally popular in the software industry, it typically involves developers working intensively to create software, applications, or hardware aimed at solving particular problems. This approach was first implemented in the introductory course Internet technologies, and after its successful integration, it has been organised annually in the course Web programming. Web programming is the first course where students must create a complex web project as a team, utilising various digital services and tools.

During hackathons, students can develop various key competencies such as collaboration, teamwork, negotiation, project management, time management, communication, and troubleshooting. Learning based on hackathons involves several activities and is carried out in multiple phases. Some of these phases can be categorised as "pre-hackathon" phases, as they aim to better prepare students for participating in a hackathon. At the beginning of the semester, since students are not yet familiar with the concept of HBL, each student is given a detailed description of the project task, which they will work on in teams of two. The document includes a detailed description of the functional requirements, as well as requirements for the implementation of security techniques and methods, and adherence to coding standards. All teams are provided with access parameters for the virtual web servers they will use during the project work.

Figure 1 illustrates the Hackathon-based learning process implemented at Subotica Tech, highlighting its structure and phases.

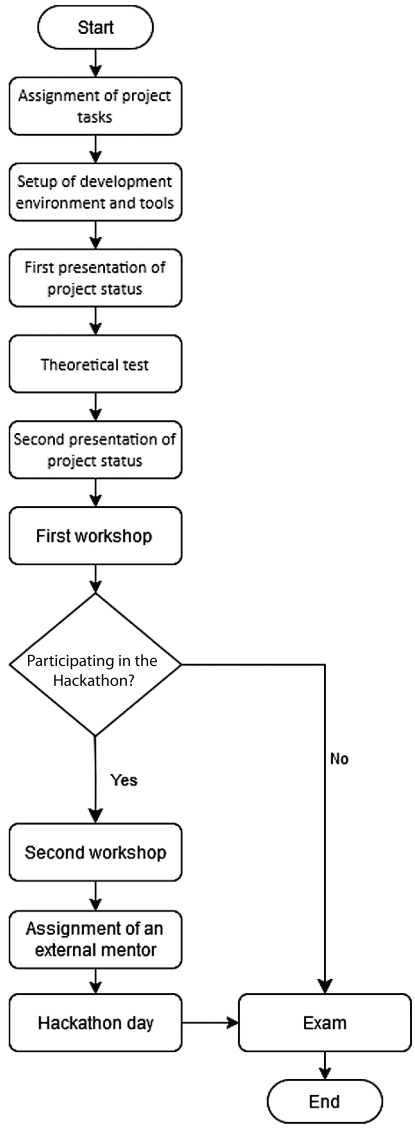

Figure 1: Phases of Hackathon-based Learning (HBL)
Source: compiled by the author

Students are required to use the *Trello* tool for project management, *Git* for version control of the code, and an appropriate hosting platform for repositories. Professors involved in the course are granted access to these services to monitor the overall progress of the project. The project work was supported and complemented by a series of lectures and lab exercises that were held throughout the entire semester. These sessions provided students with both theoretical knowledge and practical skills,

which were essential for the successful completion of their projects. In the first and second thirds of the semester, teams were required to publicly present the progress of their projects, allowing them to receive valuable feedback from both their peers and professors. These presentations were structured to not only assess the technical progress of the projects but also to foster a collaborative learning environment where students could share ideas, challenges, and solutions.

In the second third of the semester, an important milestone was reached: all students took a comprehensive theoretical test aimed at assessing the knowledge they had acquired up to that point. The test covered various aspects of the course material, including key concepts, methodologies, and tools relevant to the projects. The test results provided instructors with a clear understanding of each student's grasp of the subject matter, and helped in identifying areas that might require further attention.

In partnership with a local IT company, a workshop was organised for all students, with an additional session specifically for those who chose to participate in the hackathon. The workshops focused on the use of Git and included challenge-based tasks designed to enhance students' understanding of web application security.

After organised workshops, students were given the opportunity to participate in an additional, voluntary activity hackathon. While the hackathon required a significant amount of time and effort from the students, it was designed to further enhance their learning experience by providing an intensive, hands-on environment where they could apply their knowledge in real-world scenarios. This activity was expected to help students develop critical skills such as teamwork, problem-solving, and time management, while deepening their IT expertise.

The primary objective behind incorporating this hackathon activity was to stimulate the development of more polished and professional projects by the end of the semester. It was anticipated that the use of an HBL approach would encourage students to engage more deeply in their projects, enhance the quality of their work, and improve their overall IT competencies. By actively participating in such a dynamic and challenging environment, students were expected to gain valuable insights into the complexities of real-world problem-solving, making them better equipped to tackle similar challenges in their future careers.

Each team was paired with an external mentor-programmer from the local IT company. The mentors were given detailed descriptions of the project tasks and were responsible for identifying three additional functionalities that the teams were expected to implement during the hackathon. Teams and mentors communicated online during the first 24 hours. Mentors were also tasked with providing guidance on the use of specific classes or libraries if the additional functionalities required such resources.

After 24 hours of preparation and collaboration, the hackathon took place at Subotica Tech. The event lasted for a full 10 hours, during which teams had the opportunity to meet their mentors in person. The mentors played an active role in guiding teams through the process of developing and refining the additional functionalities assigned to them.

The following day, each team had the chance to publicly present their completed projects to the other teams, professors, and mentors. The presentations were an

important part of the process, allowing teams to showcase their work and demonstrate how their projects functioned as a whole. After each presentation, the mentors asked questions to gain a deeper understanding of the teams' approaches. The primary focus of the presentations was on the overall functionality, security and performance of the projects, rather than on a detailed analysis or review of the code itself.

When the day for project presentations was completed, both students and mentors received electronic surveys containing questions about the entire hackathon process. This practice was followed by each hackathon. The results from these surveys and feedback were used to assess participants' satisfaction with their involvement in the event, both from the students' and the companies' perspectives. Additionally, after certain hackathons, further evaluations were conducted to assess the impact of this learning method on the development of key competencies. The feedback collected helped to improve the organisation of future events.

The following day, students were required to defend their projects in front of professors, focusing specifically on the programming code level.

During the hackathon, students were required to apply appropriate security mechanisms and identify risks from all the categories listed in the OWASP Top 10 document. However, they focused most on the following risks:

- Server-Side Request Forgery
- Cryptographic Failures
- Injection
- Identification and Authentication Failures
- Broken Access Control

The surveys sent to participants after the last hackathon included 33 questions for students and 11 questions for mentors from the companies. When asked whether they would participate in another hackathon, 100% of the students responded positively. Furthermore, when asked about the importance of hackathons for companies, 100% of the mentors agreed that such events are very important for the companies as well.

The impact of hackathon participation on final exam performance

The most recent hackathon was organised for second-year computer science students as part of the Web Programming course during the summer semester of the 2023/2024 academic year. A total of 88 students were enrolled in the course of whom 64 successfully passed the final exam. For analysis, these students were divided into two groups:

- Experimental group: hackathon participants (14 students, all of whom passed the exam)
- Control group: non-hackathon participants who passed the exam (50 students)

To assess the impact of hackathon participation on academic performance, Welch's t-test was conducted to compare the final exam scores between the experimental and control groups. The grading scale ranges from 6 to 10 for students who passed.

Welch's t-test was selected due to its robustness in handling groups with unequal variances.

In this study, the null hypothesis (H_0) states that there is no significant difference in final exam performance between students who participated in the hackathon and those who did not:

$$H0: \mu_{hackathon} = \mu_{non-hackathon}$$

where $\mu_{\text{hackathon}}$ represents the mean exam score of hackathon participants, and $\mu_{\text{non-hackathon}}$ represents the mean exam score of non-participants.

The alternative hypothesis (H_1) posits that hackathon participation leads to significantly higher final exam performance:

$$H1: \mu_{hackathon} > \mu_{non-hackathon}$$

Table 1: Group Statistics

Group	Sample Size (n)	Mean Scaled Mark	Standard Deviation (σ)
Experimental Group (Hackathon)	14	9.86	0.53
Control Group (Non-Hackathon)	50	8.42	1.49

Source: compiled by the author

Table 2: T-Test Results

Statistical Test	Value
T-Statistic	5.66
P-Value	4.99×10^{-7}

Source: compiled by the author

The analysis of exam performance reveals a significant impact of hackathon participation on student outcomes. The average final exam score for hackathon participants was 9.86, notably higher than the 8.42 recorded for non-participants. Furthermore, score variability was lower among hackathon participants, with a standard deviation of 0.53 compared to 1.49 in the control group, indicating more consistent performance. Inferential statistics further support these findings, as the computed t-value of 5.66 suggests a substantial difference between the two groups. The extremely low p-value of 4.99×10^{-7} confirms that this difference is highly unlikely to have occurred by chance. Consequently, the null hypothesis is rejected, and it is concluded that hackathon participation had a statistically significant positive effect on final exam performance.

These findings suggest that students who engaged in hackathons not only achieve higher scores but also demonstrated greater performance stability. This highlights

the potential of hackathon-based learning as an effective educational strategy for improving student outcomes in web programming courses. Future research should investigate the broader applicability of this approach across different academic disciplines and assess its long-term impact on knowledge retention.

Challenge-based tasks

In cybersecurity education, challenge-based tasks are hands-on activities that engage students in identifying, analysing, and justifying security threats, simulating real-world cyber challenges to build practical skills and resilience against potential vulnerabilities. These types of tasks were first implemented in the course Security in E-Business Systems. Succeeding the successful initial implementation and student feedback, the tasks were slightly adjusted and integrated into the courses Web Programming, Advanced Web Programming and Integrated Web Systems.

Assigned to students toward the end of the semester, these tasks were introduced once they had acquired sufficient theoretical and practical knowledge. Covering all recognised categories of web application security risks, the assignments primarily focused on:

- Security logging and monitoring failures
- Cryptographic failures
- Injection
- Identification and authentication failures
- Security misconfiguration

The difficulty levels varied, and some challenges spanned multiple categories of web application security.

The tasks were designed as challenge-based assignments focused on web application security. Each task was assigned a specific timeframe for completion, after which students submitted their solutions. Their submissions included program code as well as a text file that documented their experiences, challenges faced, and, if necessary, a detailed explanation of their solution. For certain tasks, students were also required to identify potential security issues and vulnerabilities, discuss how these could be mitigated, and suggest the most suitable techniques or methods for prevention. Following the submissions, a discussion was held to explore different approaches to problem-solving and to address issues encountered. These tasks were highly practical, with created code or code snippets required to be submitted by students, giving them the characteristics of code-entry challenges. In the task description, the use of AI tools was not specified, but some students chose to use them while working on tasks.

The completion of each task involved several steps, each essential for successfully finishing the assignment. Figure 2 shows the steps of challenge-based task.

Following the initial implementation of these tasks, all participants completed a survey. Based on their feedback, some adjustments were made for the next round, with additional changes planned for future activities. The implemented updates include

anonymous logging of all attempts and testing, which provides useful material for discussion, and assigning some tasks to pairs of students for collaborative work.

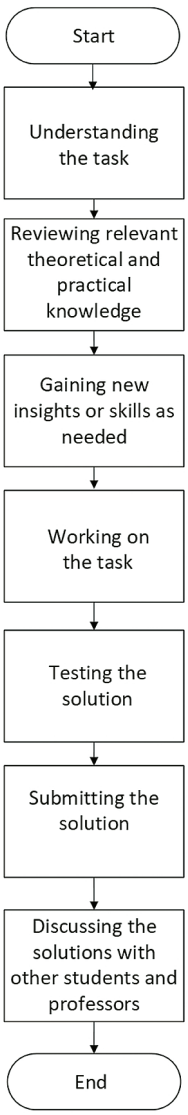

Figure 2: Steps of challenge-based task
Source: compiled by the author

During the discussion, the first focus was on analysing anonymous logs from the testing phase. The logs were reviewed to identify records that provided information indicating the correct approach to solving the problem, as well as those that demonstrated an incorrect approach or a misunderstanding of the task instructions. Following this,

a discussion of the submitted solutions took place. Interestingly, in some cases, parts of the final solutions did not differ from the information recorded during testing. Specifically, each attempt during testing received feedback from the task webpage on whether the solution was correct or not. If the response was negative, students were expected to continue working on solving the problem. The discussion and review of solutions provide a phase in which students gain new knowledge and insights into their weaknesses in a particular area. If none of the students could solve the task, the professor uploaded the solution to an internal platform. This was followed by a solution analysis, testing, and then a discussion.

The next use of this type of task is planned to be implemented in multiple iterations throughout a single semester. Each iteration will focus on a specific area covered during the coursework, with the final iteration encompassing content from the entire curriculum. Additionally, there is a plan to integrate challenge-based tasks with other new methods and to include them as part of a future research study.

The following text presents an example of a task that was given to the students.

"Visit the provided URL and analyse the HTML code of the form and the entire web page. Attempt to upload files and gather relevant information. Your objective is to locate and extract the secret word contained in the file named 'secret'. This file has no extension, and it is one level below the root directory. It is inaccessible via the browser and can only be reached through PHP code. Write PHP code to exploit the vulnerability in the web form and retrieve the contents of the 'secret' file. Once you find the secret string, decode the message to reveal the plaintext. For decoding, create a file named decode.php."

The output of this task is that students, by analysing the code of a web page, tested their knowledge of attributes essential for creating secure forms and gathered information obtained after successfully uploading files to the server.

Specifically, after a file was successfully uploaded to the server, the page displayed a message confirming that the file had been successfully uploaded. The web page with the form also displayed an image, and by analysing its URL, students could determine the directory name and its location relative to the root directory.

Next, students created a PHP script containing code that accessed the secret file, located one level below the root folder. They uploaded this file to the server via the web form. By analysing the location of the displayed image, they could attempt to directly open uploaded files in the browser. If the image was successfully displayed, this indicated that files uploaded via the form were accessible at that location. In this way, they managed to manually execute the PHP script, which reads the contents of the secret file.

The file's content was encoded, so students entered the obtained data into a separate script, *decode.php* and performed its decoding. They needed to recognise that Base64 encoding was used and apply the appropriate decoding function.

This task allowed students to identify security risks associated with such web forms and recognise vulnerabilities in the server-side code. Some key issues and security measures include:

- Restricting uploads to specific file types
- Disabling directory listing on the server
- Automatically renaming uploaded files using functions that generate unique names with additional parameters

After these initial tests of this type of task, a study is planned with an experimental and a control group to determine whether and to what extent these tasks contribute to better knowledge acquisition in web application security.

Artificial intelligence (AI) tools in education of software engineers

Incorporating Artificial Intelligence (AI) into the education of software engineers is transforming how students acquire knowledge and build practical skills. Today's learning environment extends beyond traditional resources, encompassing digital tools, online platforms, and a rapidly growing suite of AI technologies that enhance educational experience. By using AI, students can streamline and enhance their learning processes, accelerating development while working on assignments, homework, and projects.

AI tools provide students with real-time assistance, helping them debug code, optimise algorithms, and explore complex problem-solving approaches. Additionally, these tools allow students to test and analyse their projects more effectively by simulating various scenarios and identifying potential improvements. Many AI platforms also offer APIs, enabling students to integrate sophisticated AI functionalities into their applications, whether for machine learning models, natural language processing, or predictive analytics. This hands-on use of AI APIs deepens their understanding of both AI and software engineering principles, fostering skills directly applicable to the tech industry.

Moreover, by working with AI-driven tools, students are exposed to emerging trends and gain insights into ethical considerations, responsible AI use, and the impact of AI on society. In this way, AI not only supports their technical education but also prepares them to address broader issues in the field of software engineering. Integrating AI into their projects allows students to tackle real-world challenges, experiment with innovation, and develop a more versatile skill set that is critical for their future roles as software engineers.

Student tasks and AI

During the laboratory exercises in the courses Advanced Web Programming and Integrated Web Systems, students are given tasks that they need to solve during the class. The use of AI tools in solving some of the tasks is evident in the solutions that students submit to the internal service. Some students are noticeable for immediately forwarding the entire task text to AI and downloading the complete solution generated by the tool. However, most students approach the problem logically. The first thing they do is carefully review the task text to understand all the necessary

components, and then they start working on the program code. Some of them use AI tools when they need quick help with a minor issue or if they want to optimise a part of the code.

To encourage students who strictly use AI for generating solutions to read, analyse, and understand the problem in detail, they were given tasks containing illogical requirements that could affect the result and lead to a solution that functionally does not match the given problem. When these students received such solutions, they began to analyse the task text and try to solve the problem correctly.

As previously mentioned, it is important for students to be exposed to situations that are like real business environments. A new type of task was assigned to students in which they are placed in the role of project managers within an IT company primarily focused on developing software solutions.

Using AI tools and guidelines provided, students were tasked with creating a project description that would be randomly assigned to another student. The guidelines were of a technical nature: specifying which PHP classes to use, the database operations that needed to be implemented, security risks and threats, as well as a limited number of functional requirements.

Each student was required to record all input parameters provided to the AI tool, along with the responses. All conversations with the AI tool were to be documented. Based on the input parameters and the generated task description, students were also expected to request an estimation of the time required to complete the task. They had to take on the role of a project manager, using all available information about the task and the prior knowledge of their peers to estimate how long it would take to complete the task.

Once the proposed tasks and input parameters were uploaded to the internal service, the professor reviewed them and randomly assigned them to students. After completing the tasks, a group discussion was held to review all aspects of the task, with a focus on the estimated and actual time taken to complete it.

These activities have shown that incorporating AI tools into student tasks can enhance their problem-solving approach when structured thoughtfully. By designing assignments that require critical analysis, documentation, and estimation, students are encouraged to engage more deeply with the problem rather than relying solely on AI-generated solutions. This method helps bridge the gap between academic exercises and real-world software development challenges.

Integration of AI APIs in the development of tools for education

Multiple models developed by OpenAI are available, each designed to address specific purposes and use cases. Different approaches can be employed to interact with the API, with each tailored to distinct use cases.

The implementation of the AI API resulted in the creation of a web service to assist students in testing their knowledge in specific topics.

The model employed in this application is "GPT-4o Mini", which stands as the most advanced model within the category of smaller models. It is characterised by

its cost-effectiveness, speed, and efficiency. For this application, the "Chat Completion" method is utilised, as it best aligns with the specific requirements of the task. Essentially, this approach entails submitting a request to AI, which then generates and returns the corresponding output.

The application was developed using the Laravel PHP framework, along with Filament and Livewire. For data storage, the MySQL relational database management system is used.

The developed application is a web-based quiz platform where students can select the language in which they wish to receive questions, the difficulty level of the questions, and enter key terms related to the subject area for which the AI will generate the quiz. Each quiz consists of 10 questions of various types: true/false, multiple choice, and one correct choice. Data about the generated quizzes, entered key terms, and answers are stored anonymously in the database for future statistical analysis.

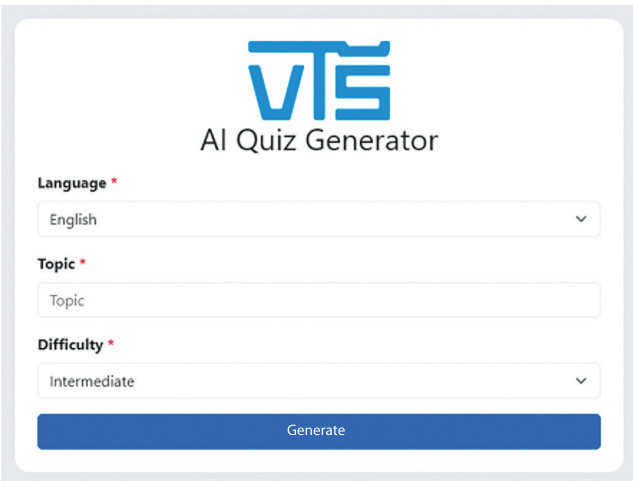

Figure 3: Index page of AQ Quiz Generator
Source: screenshot of web-based quiz platform

Future work

The plan for future research is to determine the extent to which the use of AI tools and challenge-based tasks contributes to better knowledge acquisition among students. The research would begin with an initial testing phase after which experimental and control groups would be formed. Members of the experimental group would have the opportunity to participate in challenge-based tasks and use the AI tool to test and improve their knowledge. The AI tool would offer predefined categories for generating quizzes, most likely based on material from the OWASP Top 10 and OWASP API Security Top 10 documents. After the period of using the tool and participating in challenge-based tasks, retesting would be conducted for both groups.

Another plan is to further involve companies in the future and deepen the collaboration in the field of implementing new methods in the education of software engineers. Companies would bring in their industry experience, thereby contributing to the increased awareness and knowledge of cyber threats and vulnerabilities. This approach would result in the development of experts who can quickly adapt to the demands of the labour market.

Future research will employ a structured framework involving experimental and control groups to evaluate the impact of these methodologies on student learning outcomes.

Additional studies will focus on the incorporation of more comprehensive empirical data, such as pre- and post-test results, to validate the effectiveness of AI tools and challenge-based learning tasks.

The plan for future research is to determine the extent to which AI tools and challenge-based tasks contribute to enhanced knowledge acquisition among students. The study will begin with an initial testing phase, followed by the formation of experimental and control groups. The experimental group will engage in challenge-based tasks while utilising AI tools to test and improve their knowledge. These tools will provide predefined quiz categories, likely based on materials from the OWASP Top 10 and OWASP API Security Top 10 documents. After a designated period of AI-assisted learning and participation in challenge-based tasks, both groups will undergo a retesting phase to assess knowledge retention and improvement.

Additionally, future research aims to strengthen collaboration with industry partners to integrate real-world cybersecurity challenges into educational practices. By involving companies, students will gain insights into industry-relevant security risks and best practices, ultimately developing expertise that aligns with the evolving demands of the job market.

To ensure a robust evaluation, the research will employ a structured framework with experimental and control groups, supported by empirical data such as pre- and post-test results. This approach will provide a clearer understanding of the effectiveness of AI-driven learning and challenge-based methodologies in improving student outcomes.

Conclusions

The innovative approaches outlined in this paper address critical gaps in cybersecurity education by integrating practical, real-world applications with theoretical instruction. By aligning academic curricula with industry demands, this study highlights the necessity of evolving educational practices to equip students with the skills needed to navigate the rapidly changing landscape of software engineering and cybersecurity.

In an era where secure data exchange is vital for modern systems, the demand for well-prepared software and cybersecurity engineers is greater than ever. Given the evolving nature of cyber threats, it is crucial that education in this field provides students not only with theoretical foundations but also with hands-on experience to identify and mitigate security risks effectively. Higher education must remain

dynamic, continuously adapting curricula to meet industry expectations and ensure graduates are prepared for real-world challenges.

This study explored the integration of innovative teaching methods in software engineering education, particularly in the context of cybersecurity. By embedding these methods into courses on web application and integrated web system development, the findings underscore the value of practical learning experiences in bridging the gap between academia and industry.

The results indicate that hackathon-based learning can be a highly effective strategy for improving student engagement and outcomes in web programming courses. Additionally, the integration of AI tools has proven to enhance the learning process, offering students new ways to tackle complex tasks and refine their problem-solving skills. The structured implementation of these approaches, combined with student feedback, reinforces the need for continuous evolution in educational methodologies to remain aligned with industry advancements.

Future research will further investigate the long-term impact of these strategies and their potential to shape the next generation of skilled software engineers. By refining and expanding these methods, we aim to develop an educational framework that not only enhances technical proficiency but also fosters adaptability and critical thinking – essential qualities in the ever-changing field of cybersecurity and software development.

This article is an excerpt from a presentation given at the II. Alverad-Bánki International Cybersecurity Conference on October 15, 2024.

References

Cheung, Ronald S. – Cohen, Joseph P. – Lo, Henry Z. – Elia, Fabio (2011): *Challenge Based Learning in Cybersecurity Education.* Proceedings of the International Conference on Security and Management (SAM) The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Athens, 2011, 1–6.

Chung, Kevin – Cohen, Julian (2014): *Learning Obstacles in the Capture the Flag Model.* Proceedings of the 1st USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE).

Čović, Zlatko (2024): Threats and Vulnerabilities in Web Applications and How to Avoid Them. In Kovács, Tünde A. – Nyikes, Zoltán – Berek, Tamás – Daruka, Norbert – Tóth, László (eds.): *Critical Infrastructure Protection in the Light of the Armed Conflicts.* Cham: Springer Nature Switzerland, 93–103. Online: https://doi.org/10.1007/978-3-031-47990-8_9

Čović, Zlatko – Papp, Zoltán – Manojlović, Helena – Simon, János (2022): *Hackathon-based Teaching Method in the Training of Software Engineers*. Proceedings of the 12th International Conference on Applied Internet and Information Technologies AIIT, Zrenjanin, 2022, 108–116.

McDaniel, Lukas – Talvi, Erik – Hay, Brian (2016): *Capture the Flag as Cyber Security Introduction*. 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, USA, 2016, 5479–5486. Online: https://doi.org/10.1109/HICSS.2016.677

Obot Affia, Abasi-amefon – Nolte, Alexander – Matulevičius, Raimundas (2022): *Integrating Hackathons into an Online Cybersecurity Course*. arXiv preprint arXiv:2202.06018. Online: https://doi.org/10.1145/3510456.3514151

Ouhbi, Sofia – Pombo, Nuno (2020): *Software Engineering Education: Challenges and Perspectives*. IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 2020, 202–209. Online: https://doi.org/10.1109/EDUCON45650.2020.9125353

OWASP API Security Top 10 (2024). Online: https://owasp.org/API-Security/

OWASP Top 10 (2024). Online: https://owasp.org/www-project-top-ten/

Serbian Cybersecurity Challenge (2024). Online: https://sajberheroj.rs/en/scc/

Švábenský, Valdemar – Čeleda, Pavel – Vykopal, Jan – Brišáková, Silvia (2021): Cybersecurity Knowledge and Skills Taught in Capture the Flag Challenges. *Computers & Security,* 102. Online: https://doi.org/10.1016/j.cose.2020.102154

Švábenský, Valdemar – Vykopal, Jan – Cermak, Milan – Laštovička, Martin (2018): *Enhancing Cybersecurity Skills by Creating Serious Games.* Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, Larnaca, 2018, 194–199. Online: https://doi.org/10.1145/3197091.3197123

Žagar, Mario – Bosnić, Ivana – Orlić, Marin (2008): *Enhancing Software Engineering Education: A Creative Approach*. Proceedings of the 2008 International Workshop on Software Engineering in East and South Europe (SEESE '08), New York, 51–58. Online: https://doi.org/10.1145/1370868.1370878