Using Cybersecurity Body of Knowledge (CyBOK) Case Studies to Enhance Student Learning

Anne Kohnke
Cybersec. & Info Sys Department
University of Detroit Mercy
kohnkean@udmercy.edu

Bastian Tenbergen
Computer Science Department
SUNY Oswego
bastian.tenbergen@oswego.edu

Nancy R. Mead
SEI Fellow,
Carnegie Mellon University
nrmcmu@gmail.com

Abstract

One of the central aspects of specialization in modern software engineering is security engineering. With contemporary systems being networked and entrusted with mission-critical functionality, cybersecurity is an essential quality that must be developed into the system from the first moment. This comprises issues such as privacy, authentication, robustness against vulnerabilities, and hardness against external attacks. To do so, software engineering specialists with appreciation for the detailed intricacies of security engineering as well as broad experience are required. The Cybersecurity Body of Knowledge (CyBOK, [1]) has been developed to serve, among other uses, as an instructional reference for educators to prepare the next generation of security engineers in this respect.

While the CyBOK describes the intricacies of security engineering in plentiful detail, it remains up to the instructor to convey this curriculum in a way that fosters understanding and forms experience as well as competencies in the learner. To aid the instructors who use the CyBOK, we have devised a library of 18 case studies that are specifically designed to target CyBOK knowledge areas. The case studies are sufficiently detailed to allow adoption with minimal overhead on the instructor. In this paper, we describe the case study mapping to the CyBOK, and classroom results of one exemplary case study, demonstrating improved understanding by students.

1. Introduction

As the increase and dependence on digitally enabled technology continues to impact almost every area of life, it has created a demand for innovative software-based solutions. However, developing secure software is a multi-faceted activity that can strain a project's budget, design, and overall functionality [2]. The demand for software often pits delivering value at high speed against high quality. In 2020, poor quality software cost organizations $2.08 trillion in the United States alone [3]. The U.S. government tracks software vulnerabilities in their National Vulnerability Database, which is fed by the Common Vulnerabilities and Exposures list. By 2020, more than 18,000 software code vulnerabilities had already been included [4].

In her 2000 paper, Mary Shaw [5] called, among other things, for software engineering education to start at the earliest feasible point during the students' university career and to seek out ways to improve role-specific software engineering education. Now, more than 20 years later, her call has been answered with many software engineering curricula offering broad experiences as well as avenues for specialization, for example, in requirements engineering [6], [7], testing [8], or supply chain risk management [9], [10]. Yet, in today's rapid development environment, security engineering has become a specialization that will only grow in demand [11]. As modern systems are increasingly interconnected and exchange mission-critical, confidential data with one another, they become attractive targets for attackers. Hence, systems must be sufficiently hardened against any type of vulnerability.

Designing such systems requires a substantial amount of security-relevant knowledge, attention to detail, and a considerable level of experience. To help educate the new generation of security engineers, a recent effort lead by the University of Bristol compiled and produced a substantial resource called the "Cybersecurity Body of Knowledge" (CyBOK, [1]). CyBOK 1.0 is structured in five parts and 19 chapters, each of which suggests knowledge areas related to social, organizational, technical, and procedural issues in cybersecurity. CyBOK is intended to serve as a reference curriculum and resource material for instructors to structure cybersecurity education.

Yet, faculty developing new courses on the topic might additionally require suitable resource artifacts to foster summative learning (as opposed to formative learning, e.g., through rote memorization of required reading [12]). Resource artifacts may comprise case studies, homework assignments examples, and assessment options such as exams. These artifacts, while sometimes publicly available, are often buried in complete sets of course material passed from one instructor to another and are not documented in a consistent or necessarily usable format.

URI: https://hdl.handle.net/10125/79461
978-0-9981331-5-7

To alleviate this issue, we present a library of ready-to-use case studies in this paper, tailored to select CyBOK knowledge areas. Case studies are derived from and describe real-world examples and resources or rich, fictive contexts. They feature assignment descriptions and application guidelines for the instructors as well as example solutions (if applicable) and/or assessment criteria. Herein, we give a brief overview of the case studies included in our library, their mapping to the CyBOK curriculum, and give an example of their initial application, including results.

This paper is structured as follows. Section 2 gives some background on the CyBOK and reviews the related work on case study application in Software Engineering Education. Section 3 overviews our library and associates the case studies with CyBOK learning objectives. In Section 4, we discuss how we applied a selection of the case studies in a real course, and Section 5 concludes this paper with an outlook on future work.

2. Background & Related Work

In this section, we briefly introduce the CyBOK. We also discuss the use of case studies in software engineering education.

2.1. The CyBOK Version 1.0

The Cyber Security Body of Knowledge Version 1.0 (CyBOK) is a freely accessible community resource funded by the National Cyber Security Programme in the United Kingdom and published under the Open Government License in October 2019 [1]. CyBOK is an attempt to consolidate cybersecurity as a discipline, which in the past has been fragmented [13]. By contrast, in fields such as software engineering, computer science, or chemistry, there have been collaborations with leading professional societies that have codified key foundational knowledge on which educational programs have been designed and developed (e.g., the Software Engineering Body of Knowledge, SWEBOK, see [14]). Other efforts have established skills, tasks, competencies, risk, and cyber frameworks that exposed many facets to the discipline [15]. A more recent global undertaking with four leading professional societies and a host of academics and practitioners forming a Joint Task Force, resulted in a comprehensive curricular volume to structure the cybersecurity discipline and provide guidance for cybersecurity education [16]. However, among the diverse community of academics, practitioners, and researchers, there has not been progress in reaching a consensus of what is considered the foundational knowledge in cybersecurity [13], [16].

An analysis of the Joint Task Force work along with the ACM Computing Classification System taxonomy, technical certifications, calls for papers, standards, and tables of contents in a variety of textbooks were text-mined using natural language processing and automatic text clustering to group relevant topics and identify the relationships between the topics. Consulting with academics, practitioners, key experts, as well as garnering community feedback, the CyBOK Version 1.0 was developed and ultimately identified 19 Knowledge Areas (KAs) that form the scope of the CyBOK [1]. The 19 KA are grouped into the following five categories and knowledge areas:

I. Human, Organisational & Regulatory Aspects

1. Risk Management and Governance
2. Law & Regulation
3. Human Factors
4. Privacy & Online Rights

II. Attacks & Defences

5. Malware & Attack Technologies
6. Adversarial Behaviour
7. Security Operations & Incident Management
8. Forensics

III. Systems Security

9. Cryptography
10. Operating Systems & Virtualisation
11. Distributed Systems Security
12. Authentication, Authorisation & Accountability

IV. Software Platform Security

13. Software Security
14. Web & Mobile Security
15. Secure Software Lifecycle

V. Infrastructure Security

16. Network Security
17. Hardware Security
18. Cyber-Physical Systems Security
19. Physical Layer & Telecommunications

A detailed description of the categories, knowledge areas is available in the CyBOK Version 1.0 companion text [17].

2.2. Case Studies and Summative Learning in SE Education

It is widely accepted in the education literature [18], that strictly relying on formative learning approaches leads to poor theory retention beyond the end of instruction [19]. "Formative" in this sense encompasses lecturing, rote memorization, and high-stakes assessments (e.g., a single exam to determine grades). Instead, summative approaches have been frequently proposed in a variety of disciplines [20], [21], including software engineering [22], and their use in conjunction with formative learning is advocated [12]. "Summative" refers to stimulating knowledge discovery, e.g., by using real stakeholders [7], industry-realistic projects [19], low-stakes assignments [23], games [24], or collaborative teams [25].

Effective instruction of summative approaches, however, relies on a solid theoretic foundation, which is why a combination of both educational styles is required [18], especially in disciplines, in which application and experiences outweigh theory in terms of value for the learner [12], such as software engineering. To achieve this, illustrative non-trivial examples such as vignettes and case studies are an effective tool. The difference between the two is essentially their complexity. Yet, vignettes are usually too brief to provide a proper context and thereby require a thorough investigation of the problem to arrive at a proper solution [26]. An example of a vignette could be: *"Paul holds open the door to the clearance level 3 office doors for Jamie, who walks right behind Paul, such that Jamie doesn't have to swipe her access card. Describe the security-related problem with Paul's behavior."*

Case studies [27] on the other hand do not have this limitation. Case-based teaching has been part of instructional pedagogy since the late 1880's, primarily starting with law courses and later adopted by both schools of business and medicine from the early 1900's forward [28]. Written case studies can range from a brief outline to illustrate a theoretical point to more elaborate cases, organized and separated into sections with relevant questions and discussion points to integrate theoretical and practical content. What constitutes a case study in education depends largely on the educational goals and pedagogical approach, and of course the instructor. Educational case studies must not be confused with using case studies for empirical evidence in software engineering research (cf. [29]). Case study-based instruction may comprise video materials, curriculum modules, and educational materials for faculty use in software engineering courses [30], ranging from case studies to entire video courses for classroom use. What case studies in this sense have in common is that (a) they provide sufficient context for a problem domain, often involving fictive examples or real-world cases from journalistic or popular scientific sources, and (b) provide task descriptions for students that allows exploring several alternative, equally acceptable solutions rather than one ideal solution.

Application areas of case studies in software engineering include, for example, efforts to increase student motivation for theoretical concepts in requirements engineering [19] or software engineering [31], validation and verification activities [32], and also cyber security engineering [33]. Case studies have a thoroughly positive influence on learning outcomes as concepts are more easily adopted [31], yet occasionally at the expense of an (often unfounded) anxiety over students' final grades [34]. One reason for the popularity of software engineering case studies is that they represented real situations that may be encountered in practice [35]. Such findings have carried over to our experience teaching cybersecurity courses (see, e.g., [36]), where realistic case studies resonate more with students than artificial problems. Yet, the development of a library of case studies specific to a curriculum is a novel approach, in the experience of the authors.

3. Overview of CyBOK Case Studies

Our aim in creating a library of case studies and mapping them to the CyBOK, was to provide educators with relevant and high-quality materials which they can use 'as is' or customize for use in their classrooms. An advisory board consisting of volunteer faculty members and experts in systems, software, and cybersecurity worked to create a collection of robust case studies and citations, thus saving faculty members from having to do the work of researching and structuring the case studies for instructional use or developing their own. Most realistic case studies do not have a single "correct" solution, but where sample solutions exist, these accompany the case studies.

Table 1 on the next page provides an overview of the case studies with a brief description of the content and context[1].

3.1. Common Case Study Structure

Most of the case studies share a common structure to foster quick and easy adoption by the instructor. The subsections that comprise the format of the cases are as follows:

Background. This section provides a brief overview of the real-world and/or fictional example at hand and provides sufficient context to frame the problem space. This section makes references to externally available resources, if applicable, or suggests further reading.

Case Study Overview. This section takes a step back from the subject matter provided in the "Background" section and describes the learning activities to be carried out on the basis of the information given in "Background" to meet CyBOK learning outcomes.

[1] A detailed overview over all case studies is available here: https://www.cybok.org/media/downloads/Overview_of_CyBOK_Case_Studies.pdf

The complete case study library is available here:

https://www.cybok.org/media/downloads/CyBOK_Case_Study_Library_upload.zip

Table 1 Overview of the CyBOK Case Studies

Case Study Name	Topic Overview
ACME Water	Provide a secure operating environment for SCADA, Telemetry and Control Systems associated with assets owned and operated by ACME.
Aircraft Service Application	Develop the requirements for a secure aircraft service management application to replace a legacy system with hand-held device support.
Archetypal Users: Personae non Gratae	Support malicious user identification and assessment by developing personas of unwanted, possibly nefarious users and derive security requirements pertaining thereto.
Driver Assistance System Safety & Security	Use a real-world owner's manual for a car to "reverse engineer" the requirements specification with special focus on safety and security requirements.
Drone Swarm	Conduct threat modeling with secure cards for deliveries with search & rescue drones.
FAA ERAM Outage	Model the strategic importance of Federal Aviation Administration's EnRoute Automation Modernization project and find flaws in its software testing and cybersecurity plan. Conduct a risk assessment and threat analysis.
GPS Spoofing of UAV	Review real-world incident reports to investigate necessary design changes to path a security vulnerability that allowed attackers to hijack a military Unmanned Aerial Vehicle.
Heartland Payment System Breach	Investigate and re-create the anatomy of an SQL injection attack and develop possible countermeasures to avoid risks.
Mt. Gox Bitcoin Theft	Review popular science articles on the famous Bitcoin theft to discover procedural, organizational, and technological flaws in the Mt. Gox cryptocurrency trading system and derive recommendations on how to avoid it in the future.
National Grid SAP Adoption	Review popular science articles on a secure acquisition project discover procedural, organizational, and technological flaws that lead to project failure and avenues to avoid them.
Organizational Risk Management: The Widget Company	Investigate the organizational structure of a fictive company against organizational risks. Develop a mitigation plan and a protection strategy.
Secure Acquisition (Case Studies 1-4)	Four case studies centered around adopting off-the-shelf components for a development project in a secure way.
SQUARE	Elicit and document security requirements for a software development project that expands existing infra-structure of a mission-critical system in a subsidiary of a fictitious company.
Tokeneer ID Station Project	Conduct a compliance and cost-effectiveness analysis of a development project for a top-secret level governmental development project.
Using Malware Analysis to Improve Security Requirements	Suggest a process model to conduct malware analysis and derive misuse cases to identify vulnerabilities in a software development lifecycle.

Student Instructions. As the name suggests, this section contains concrete work assignments for students with sufficient detail to understand what is expected but with enough leeway to allow the learner to explore the problem space. This section may be subdivided into multiple tasks or provide partial solutions to get started.

Instructor Notes. This section discusses pedagogical strategies on how to apply the case study. For example, this may entail ways to tailor one case example for group vs. individual project assignments or exam questions, or solution templates.

Example Solution. If one is available, this subsection contains example solution(s), key grading criteria, success factors, or caveats depending on the case study at hand.

References. This section contains references to external resources and/or further reading.

All case studies are freely available and non-commercial usage is permitted, provided respective copyright and attributions are honored. An example case study is provided and discussed in Section 4.

3.2. Development Process and Quality Criteria

The case studies were created based on the respective author's experience and knowledge of the subject matter. Each (team of) authors presented the advisory team with a choice of case study topics, along with a brief overview of the content. Once approved, case studies were independently worked on by each author during the Spring 2021 semester. At regular intervals, status updates were reported to the advisory team, who would then ensure that the following quality criteria were met. Specifically, each case study was designed and formatted to:

1. Involve real-world examples from journalistic or popular scientific sources or fictional examples sufficient to provide a rich problem space;
2. Provide sufficient context for a problem domain by referencing said sources or providing sufficient explanation;
3. Provide detailed task descriptions that allow exploration of alternative solutions;
4. Provide instructor guidance on how to apply the case study in a given educational setting; and
5. Contain example solution descriptions, common pitfalls to avoid, and/or critical success factors to attain (if applicable, given the case study topic).

Members of the advisory board validated the submissions against the above criteria, where possible enforced a common structure, and maintained a reporting structure pertaining to the mapping of each case study to CyBOK knowledge areas.

3.3. Mapping to CyBOK Knowledge Areas

One goal of this project was to collect many useful and completed case studies, specific to the software security engineering discipline, and map them to the knowledge areas of the CyBOK [17]. We hoped that a large quantity of case studies would roughly cover the entire CyBOK, ideally with multiple case studies, thus providing different examples and assignments for many knowledge areas, thereby allowing for variety.

The result was 18 different case studies covering all but three knowledge areas, thus yielding case studies related to 84% of the CyBOK knowledge areas. Seven knowledge areas (36%) are addressed by a single case study. Nine knowledge areas (47%) are addressed by at least two case studies. In particular, the knowledge areas "risk management & governance" and "secure lifecycle management", which arguably are at the core of secure software engineering projects, are addressed by six and eight case studies, respectively. Table 2 on the right shows the collection of case studies mapped to the knowledge areas and their respective categories. Note, that the Secure Acquisition case study consists of four individual cases, each of which builds upon the previous one to create an overall comprehensive project.

The collection of case studies provides a robust degree of coverage of the CyBOK knowledge areas and learning outcomes, especially in the fundamental topics of "risk management & governance" and "secure lifecycle management". There are three knowledge areas presently uncovered by our case study library and include "law & regulation", "network security", and "physical layer & telecommunications". Future work will therefore be concerned with recruiting additional cases focused on these areas, along with the other areas. We welcome and invite readers to contribute their case studies to provide a well-rounded library to help other software engineering instructors with teaching the CyBOK in their curriculum.

Table 2 Mapping of Case Studies to CyBOK Knowledge Areas

Cat.	Knowledge Area	Case Study Mapping
Human, Organizational & Regulatory Aspects	Risk Management & Governance	ACME Water Arch. Users Personae non Gratae FAA ERAM UAV GPS Spoofing Nat. Grid SAP Adoption Widget Company
	Law & Regulation	
	Human Factors	ACME Water FAA ERAM
	Privacy & Online Rights	Driver Asst. Sys.
Attacks & Defences	Malware & Attack Technologies	Mt. Gox Theft Malware Analysis for Sec. Reqs
	Adversarial Behaviour	Heartland Breach Mt. Gox Theft
	Security Operations & Incident Mgmt	Heartland Breach Mt. Gox Theft
	Forensics	Mt. Gox Theft
Systems Security	Cryptography	Mt. Gox Theft
	Operating Systems & Virtualisation	Heartland Breach Mt. Gox Theft
	Distributed Sys. Sec.	Driver Asst. Sys.
	Authentication, Authorisation & Accountability	ACME Water Heartland Breach
Software Platform Security	Software Security	Driver Asst. Sys. FAA ERAM
	Web & Mobile Security	Driver Asst. Sys.
	Secure Software Lifecycle	ACME Water Aircraft Serv. App. Drone Swarm Nat. Grid SAP Secure Acquisition SQUARE Tokeneer ID Station Malware Analysis for Sec. Reqs
Infrastruct. Security	Network Security	
	Hardware Security	Driver Asst. Sys.
	Cyber-Physical Systems Security	Driver Asst. Sys.
	Physical Layer & Telecommunications	

4. Preliminary Experiences from Applying the Case Studies

The complexities of software engineering and the competencies expected of software application developers are continually increasing. Central to building competencies is knowledge that must be organized, systematically communicated, and applied to real-world situations. Learning the requisite knowledge is critical for the security of an organization, however, educators have often struggled in understanding how learning occurs. To aid in this understanding, a variety of learning models have been developed along with measuring specific outcomes, setting threshold standards, and the development of learning frameworks [36]. The many learning models that have been developed provide the basis to help understand learning behaviors and ultimately to inform the design of instruction in the classroom. Real-world case studies have been instrumental and are often utilized to assist software engineers in obtaining requisite knowledge as well as to develop problem solving skills for projects they will encounter after graduation.

Since 1984, the Software Engineering Institute (SEI) has been committed to improving the practice of software engineering [30]. In an early effort to influence software engineering curriculum development throughout the education community, the SEI recruited a software educator to lead the effort. Workshops were conducted that included leading software engineering educators and practicing engineers to develop 'curriculum modules', which led to the curriculum guidelines that became the model curriculum at many universities [30]. An outgrowth of the curriculum project was the development of freely available educational materials, which included case studies, to support faculty members teaching software engineering courses, presented in various workshops.

In addition to these workshops, curricula and educational materials were presented and discussed at the Conference on Software Engineering Education

(CSEE). Feedback from faculty at the CSEE conference indicated that the case studies and examples were among the most useful materials, along with curriculum modules exploring a single software engineering topic, as they could be used directly in courses developed by faculty at their own universities. The full courses were seldom used directly by faculty as they naturally preferred developing their own courses once they became familiar with the material. Additionally, faculty members and industry trainers like the fact that the educational materials are structured so they could easily be tailored for international variations and incorporated into courses developed by faculty world-wide [30].

It is in this tradition that we developed the case studies presented in this library, specifically for the CyBOK curriculum. Many of the case studies presented herein have been applied for years in many SEI courses (e.g., SQUARE and Software Acquisition). Others have been designed specifically for this CyBOK library (e.g., Mt. Gox and Heartland Breach). Again, others have been derived from previous experience, such as the Driver Assistance System case study, which we describe exemplarily in the following.

4.1. Driver Assistance System Case Study

The Driver Assistance System case study is based on an industry-realistic case example provided by our industry collaborators during a publicly funded research project. The case study follows the approach in [19], [34] regarding its application and achievements. Results and experiences pertaining to student motivation and retention in a safety requirements engineering (RE) course are described in [37]. Yet, to meet ABET accreditation requirements, the RE course presented in [37] needed to be adapted after Spring 2019 to provide additional cybersecurity learning outcomes. This was done based on CyBOK, for which the Driver Assistance System Case Study was created. While the case study is freely available in the link provided in Section 3, we give a very brief overview here to frame the results from application, presented in Section 4.2.

Background. The case study describes the purpose of modern driver assistance systems (e.g., adaptive cruise controls or lane keeping support) and alleges that a nefarious hacker may be able to gain access to the car's safety-critical features through OEM-specific cloud-based connectivity systems (e.g., OnStar, BlueLink, meConnect, ConnectedCar, etc.).

Case Study Overview. The purpose of the case study is to familiarize students with the similarities and differences of safety and security requirements while building a complete, consistent, safe, and secure requirements specification consisting of natural language and model-based requirements as well as a safety argument and security assessment report.

Student Instructions. Students are asked to build the requirements specification based on the feature descriptions of car systems found in a real-world glovebox manual for a modern car. In three milestones (which in turn are subdivided into tasks), students will:

1. "Reverse engineer" the user requirements for one of the car's driver assistance systems and document goals and scenarios as well as natural language requirements in an IEEE 830-compliant requirements specification.
2. Conduct a Safety Hazard Analysis and Security Risk Assessment using a provided template and tutorial slides, derive safety mitigations as well as countermeasures, and document them as requirements in the specification from milestone 1.
3. Develop UML class, activity, and state machine diagrams to refine the requirements from milestones 1 and 2 to "a degree that would enable implementation" and develop a safety argument.

Instructor Notes. Instructors are advised that the case study at hand is intended as a semester-long team project for 2-5 students and recommends frequent team presentations of partial and preliminary solutions so other teams in the course can get exposed to solution alternatives.

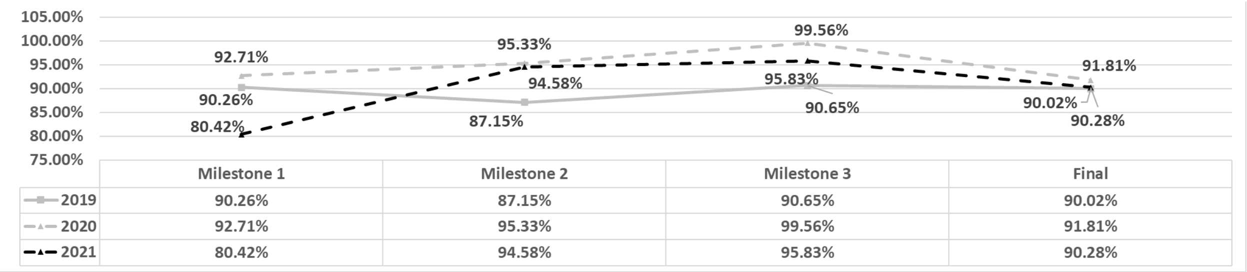
Example Solution. There is no example solution for this case study, however, notes are presented that frame the degree to which aspects such as requirements completeness would allow for, e.g., hypothetical implementation.

References. This section contains references to the glovebox manual in question, hazard analysis template and tutorial, as well as some further reading.

4.2. Results and Experiences from Applying the Driver Assistance System Case Study

As mentioned in Section 4.1, the RE course has employed industry-realistic case studies before [37] and was modified after 2019 with cybersecurity learning outcomes. Specifically, from Spring 2019 to Spring 2020, lecture units were added to convey risk and vulnerability analysis and modeling, and an exam question was added for formative assessment. However, in Spring 2020, the case study was not yet modified to foster CyBOK learning outcomes. Specifically, the case examples used in that semester largely mimicked the case study outlined in Section 4.1 yet lacked cybersecurity-related milestones and task.

In Spring 2021, the Driver Assistance Case Study described in Section 4.1 was used in addition to the lecture and exam assessment on cybersecurity that were added in Spring 2020. Assessment in the case study took the form of grading student solutions out of 15 points based on criteria such as correctness of used notations, consistency of information throughout the specification document, specificity (i.e., lack of vagueness and



	Milestone 1	Milestone 2	Milestone 3	Final
2019	90.26%	87.15%	90.65%	90.02%
2020	92.71%	95.33%	99.56%	91.81%
2021	80.42%	94.58%	95.83%	90.28%

Fig. 1 Project Performance in all three Case Study Milestones and Final Project Grades across Semesters

ambiguity), and completeness. Specificity and completeness in this sense mean that requirements should be specific and complete enough to allow hypothetical implementation, or alternatively highlight missing information to be determined in future hypothetical work (i.e., during system architecture design by another team, which was beyond the scope of the RE course).

Figure 1 shows the relative performance in all three case study milestones as well as the cumulative final grade for the 2021, 2020, and 2019 semesters, respectively. Note that to increase legibility, the vertical axis is scaled to the interval [0.75..1.05]. As can be seen, project grades remained comparably high across all semesters and all milestones. Since project performance using case studies is typically at a very high level (cf. [34]), this seems to indicate that the specific instructions and structure of the Driver Assistance System Case Study led to a comparable performance. It is notable, however, that milestone 1 in 2021 was roughly ten percentage points below the 2019 reference semester and 12 percentage points below the previous year. We attribute this to students struggling with reading, comprehending, and deriving requirements from the car's real-world glovebox manual, instead of inventing requirements by themselves as in previous years.

Nevertheless, we conclude from this that the Driver Assistance System Case Study was able to repeat the success from previous years' application of industry-realistic case studies in the safety requirements engineering course in question.

To assess whether this case study fostered CyBOK learning outcomes, we compare student performance in an exam question. As outlined above, in partial response to ABET accreditation requirements, lecture material as well as one exam question were added to the course for this purpose. The lecture material consisted of one intensive lecture on threats, attack vectors, and risk assessment which also contrasts safety engineering and security engineering principles with one another. Furthermore, security-related lecture material was interspersed with already-existing lectures, e.g., misuse case modeling as part of scenario-based RE and threat modeling during safety argument construction. The exam question required students to synthesize knowledge on safety analysis and cybersecurity risk assessment by presenting them with a Functional Safety Hazard Analysis template (which they were already familiar with from working on the case study) and evaluating what needed to be changed to accommodate concepts such as "threat", "risk", "vulnerability", and "countermeasure."

Figure 2 shows the difference in the cybersecurity assessment score in the final exam (i.e., the average score achieved by all students in the cybersecurity-related question on the exam) across all three semesters. Please note that while in 2019, only minimal security-related instruction took place in the course, a similar question requiring concept synthesis existed in the final exam. Albeit this is not comparable to the assessment in 2020 or 2021, we present the score in Fig. 2 for reference. In 2020 and 2021, both courses were identical, except that in 2021 the CyBOK case study was applied.

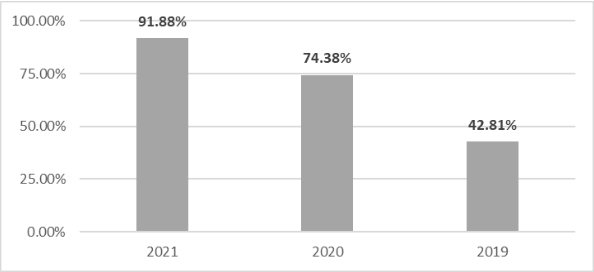


Fig. 2 Cybersecurity Assessment Score in Final Exam

Unsurprisingly, students performed much better in security-related assessment in 2020 onward. Since the CyBOK-related curriculum between 2020 and 2021 differed only in application of the Driver Assistance System Case Study, the 17.5 percentage point increase from the 2020 and 2021 semester must be attributed to it. To test whether this increase is significant, we conducted a T-Test to verify differences in means (after rejecting the assumption of variance equality by means of an F-Test). Results are shown in Table 3.

T-Test results reveal that the difference between the 2021 and 2020 semester is significant ($p < 0.05$). Since the mean for the 2021 offering is higher (91.88% vs. 74.38%), we reject the null hypothesis and accept these results as evidence that the Driver Assistance System Case Study significantly increased students' CyBOK

Table 3 Analysis of Cybersecurity Assessment Score between 2020 and 2021 (2019 shown for Reference)

	2021	2020	2019
Mean	91.88%	74.38%	42.15%
Variance	20.38%	28.50%	20.86%
Sample Size	16	16	31
dF	27		
F	0.5113 (unequal variances)		
Student's T	**0.0318**		
Cohen's d	0.706 (medium-large effect size)		

learning outcome (as it pertained to the case study). A post-hoc power analysis using Cohen's d [38] revealed a medium-large effect size (d = 0.706), indicating a low likelihood of statistical error to cause significance (despite the low sample size, α = 0.037%).

Finally, we would like to share some qualitative experience regarding the Driver Assistance System Case Study. Our experience throughout the semester mimicked experiences reported in related work (see, e.g., [19], [34], [37]). In particular, we noticed a steep learning curve regarding safety-related concepts. However, even though the core concepts of safety and cybersecurity are comparatively relatable, students seemingly struggled less in finding, e.g., threats as opposed to hazards. Cybersecurity concepts seemed to be almost intuitively understandable, while safety concepts were not.

One of the motivating factors of assigning the Driver Assistance System Case Study as a project was that students would pick different vehicle systems and, during the frequent in-class presentations, would actively exchange ideas and recognize logical interfaces between driver assistance systems (for example, the team working on the Lane Keeping Assist system would realize that the forward-facing camera can also be used in the adaptive cruise control system, hence leading to collaboration between the respective teams during safety analysis and threat modeling). Despite the instructor's best efforts to point out similarities across projects and encouraging cooperation beyond class discussions, student reactions rarely exceeded acknowledgement that certain interfaces are similar ("Oh, yeah, we have a camera, too"). Instead, students focused on producing their own isolated solution. A confounding factor may have been the mode of interaction, as due to the COVID-19 pandemic, the RE course had to resort to synchronous online instruction using video conferencing throughout 2021 and partially in 2020.

5. Conclusion and Future Work

In this paper, we presented a library of 18 case studies for the Cybersecurity Body of Knowledge (CyBOK), version 1.0 (October 31, 2019). The work was supported by The UK National Cyber Security Centre [1], [17]. Case studies were developed by a team of subject-area expert authors. We present initial favorable results from the application of one case study in a Safety Requirements Engineering course that heavily emphasizes cybersecurity. Results show that the use of the Driver Assistance Case Study had a significantly positive impact on learning outcomes as assessed by a final exam.

The purpose of the creation of the case study library was to provide sample educational materials for instructors to educate the next generation of cybersecurity software engineering professionals in CyBOK's five topic categories and 19 knowledge areas. The library consists of 18 case studies, which share a common structure for expedient and easy adoption, including context, student instructions, instructor notes, and sample solutions. Many CyBOK knowledge areas are covered with multiple case studies, allowing for variety in instruction such as application as group projects or as exam questions.

Three knowledge areas in CyBOK 1.0 (i.e., "law & regulation", "network security", and "physical layer & telecommunications") are currently uncovered by the CyBOK case study library, for which we welcome and invite contributions.

Since the completion of this work and submission of this manuscript to peer review, a new version of CyBOK was released (version 1.1, July 27th, 2021). The new version has been expanded in size by about 20% and includes new knowledge areas such as "Formal Methods for Cybersecurity" as well as "Applied Cryptography." Although the two versions overlap to a considerable degree, the current version of the case study library does not address the revised version and the additional knowledge areas. Therefore, future work ought to address the degree of coverage between the new aspects in CyBOK v1.1 and develop additional case studies for aspects thus far uncovered (i.e., "Law and Regulation," "Network Security", and "Physical Layer & Telecommunications"). Moreover, in the future we plan to continue to collect quantitative data to support improved learning outcomes and we encourage others to join us and contribute to this important data collection effort.

Acknowledgements

The authors appreciate the kind funding from The UK National Cyber Security Centre to facilitate development of the case studies presented herein. We thank Stephan Krusche for detailed feedback on this manuscript.

References

[1] The National Cyber Security Centre, The Cyber Security Body of Knowledge (CyBOK), Version 1.0. ©

Crown Copyright, 2019, UK Open Government License. Accessed 5/24/2021, available at: https://www.cybok.org/

[2] Chowdhury, N., Adam, M., Skinner, G., The Impact of Time and Pressure on Cybersecurity Behavior: A Systematic Literature Review. Behavior & Information Technology 38(12), 2019, pp. 1290-1308.

[3] Krasner, H. The Cost of Poor Software Quality in the US: A 2020 Report. Consortium for Information & Software Quality. 2021. https://www.it-cisq.org/pdf/CPSQ-2020-report.pdf

[4] O'Driscoll, A. 25+ cyber security vulnerabilities statistics and facts of 2021. Comparitech, 2021, https://www.comparitech.com/blog/information-security/cybersecurity-vulnerability-statistics/

[5] Shaw, M., Software Engineering Education: A Roadmap. In Proc. Future of Software Engineering, 2000, pp. 371-380.

[6] Sedelmaier, Y., Landes, D., Systematic evolution of a learning setting for requirements engineering education based on competence-oriented didactics. In Proceedings of the IEEE Global Engineering Education Conference, 2018, pp. 1062–1070.

[7] Gabrysiak, G., M. Guentert, R. Hebig, and H. Giese, Teaching requirements engineering with authentic stakeholders: Towards a scalable course setting. In Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences 2012, pp. 1–4.

[8] Mishra, D., Ostrovska, S., Tuna, H., Exploring and expanding students' success in software testing, Information Technology and People 30(4), pp. 927-945, 2017. DOI:10.1108/ITP-06-2016-0129.

[9] Sonatype, 2020 State of the Software Supply Chain: The 6th annual report on global open source software development, Fulton, MD. https://www.sonatype.com/hubfs/Corporate/Software%20Supply%20Chain/2020/SON_SSSC-Report-2020_final_aug11.pdf

[10] Shoemaker, D., Mead, N., Kohnke, A., Teaching Secure Acquisition in Higher Education. IEEE Security & Privacy 18(4), pp. 60-66, 2020.

[11] Bosch, J., Speed, Data, and Ecosystems: The Future of Software Engineering. IEEE Software 33(1), 2015, pp. 82-88.

[12] Harlen, W., James, M., Assessment and Learning: Differences and Relationships between Formative and Summative Assessment. Assessment in Education: Principles, Policy & Practice 4(3), 1997, pp. 356-379.

[13] Ramirez, R., Choucri, N., Improving Interdisciplinary Communication with Standardized Cyber Security Terminology: A Literature Review. IEEE Access 4, 2016, pp. 2216-2243.

[14] Bourque, P., Fairley, R., Guide to the Software Engineering Body of Knowledge (SWEBOK ®), Version 3.0. IEEE Computer Society Press, 2014.

[15] Švábenský, V., Vykopal, J., Čeleda, P., What are Cybersecurity Education Papers About? A Systematic Literature Review of SIGSCE and ITiCSE Conferences. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education, pp. 2-8, 2020.

[16] Joint Task Force on Cybersecurity Education: Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity Education. accessed 5/27/21, available at: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/csec2017.pdf

[17] Rashid, A., Chivers, H., Danezis, G., Lupu, E., Martin, A. (Eds.), The Cyber Security Body of Knowledge. © Crown Copyright, The National Cyber Security Centre, 2019. Accessed 5/27/21, available at: https://www.cybok.org/media/downloads/CyBOK-version-1.0.pdf

[18] Man Sze Lau, A., Formative good, summative bad? – A Review of the Dichotomy in Assessment Literature. Journal of Further and higher Education 40(4), 2016, pp. 509-525.

[19] Daun, M, Salmon, A., Tenbergen, B., Weyer, T., Pohl, K., Industrial case studies in graduate requirements engineering courses: The impact on student motivation. In Proceedings of the IEEE 27th Conference on Software Engineering Education and Training (CSEE&T), 2014, pp. 3–12.

[20] Sivan, A., Wong Leung, R., Gow, L., Kember, D., Towards more active learning in hospitality studies, Intl. Journal of Hospitality Management 10(4), 1991.

[21] Bonwell, C., Eison, J., Active Learning: Creating Excitement in the Classroom, ASHE-ERIC Higher Education Report No. 1, The George Washington University, 1991.

[22] Richardson, I., Delaney, Y., Problem Based Learning in the Software Engineering Classroom, In Proceedings of the 22nd IEEE Conference on Software Engineering Education and Training (CSEE&T), 2009, pp. 174-181.

[23] Berry, D., Kaplan, C., Planned programming problem gotchas as lessons in requirements engineering. In Proceedings of 5th International Workshop on Requirements Engineering Education and Training, pp. 20–25, 2010.

[24] Rusu, A., Russell, R., Cocco, R., Simulating the software engineering interview process using a decision-based serious computer game. In Proceedings of the 16th International Conference on Computer Games (CGAMES), pp. 235–239, 2011.

[25] Marutschke, D. M., Kryssanov, V., Brockmann, P., Teaching distributed requirements engineering: Simulation of an offshoring project with geographically separated teams. In Proceedings of the IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), pp. 1–5, 2020.

[26] Grubb, A., Four Opportunities for SE Ethics Education. In Proceedings of the IEEE/ACM 2nd International Workshop on Ethics in Software Engineering Research and Practice (SEthics), 2021.

[27] Garg, K., and Varma, V., "A Study of the Effectiveness of Case Study Approach in Software Engineering Education", 20th Conference on Software Engineering Education Training (CSEET'07), (2007), 309–316.

[28] Kimball, B.A., The Emergence of Case Method Teaching, 1879s-1990s: Search for Legitimate Pedagogy, Bloomington, IN: Poynter Center.

[29] Runeson, P., Höst, M., Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Empirical Software Engineering 14, 131, 2009. https://doi.org/10.1007/s10664-008-9102-8

[30] Druffel, L., A Technical History of the SEI. Special Report CMU/SEI-2016-SR-027, Software Engineering Institute, January 2017. Accessed 5/27/21, available at: https://resources.sei.cmu.edu/asset_files/SpecialReport/2017_003_001_485151.pdf

[31] Saurabh Tiwari. Impact of cbl on student's learning and performance: An experience report. In Proc. of 13th Innovations in Software Engineering Conf. (ISEC 2020), ISEC 2020, New York, NY, USA, 2020. Association for Computing Machinery.

[32] P. Manohar, S. Acharya, P.Y. Wu, A.A. Ansari, and W.W. Schilling, Jr. Case study based educational tools for teaching software V&V course at undergraduate level. In122nd ASEE Annual Conf. and Exposition: Making Value for Society, 2015.

[33] N.R. Mead and E.D. Hough. Security requirements engineering for software systems: Case studies in support of software engineering education. In Proc. of 19th Conf. on Software Engineering Education and Training, volume 2006, pages 149–156, 2006.

[34] M. Daun, A. Salmon, T. Weyer, K. Pohl, and B. Tenbergen. Project-based learning with examples from in-dustry in university courses: An experience report froman undergraduate requirements engineering course. InProc. of IEEE 29th Int. Conf. on Software Engineering Education and Training (CSEE&T), pages 184–193, 2016.

[35] B. Penzenstadler, M. Mahaux, P. Heymans, "University meets industry: Calling in real stakeholders", Proc. 26th IEEE Conf. Soft. Eng. Education & Training, 2013, pp. 1-10.

[36] Voorhees, R.A., Competency Based Learning Models: A Necessary Future, New Directions for Institutional Research, 2001, No. 110, Summer, John Wiley & Sons, Inc.

[37] B. Tenbergen, M. Daun, "Industry projects in requirements engineering education: Application in a University Course in the US and Comparison with Germany," In Proceedings of the Hawai'i International Conference on System Sciences, 2019.

[38] J. Cohen, Statistical Power Analysis for the Behavioral Sciences (2nd ed.), Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1988.