ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

Learning Software Security in Context

An Evaluation in Open Source Software Development Environment

Shao-Fang Wen and Basel Katt
Norwegian University of Science and Technology
Gjøvik, Norway
{shao-fang.wen, basel.katt @ntnu.no

Abstract. Learning software security has become a complex and difficult task today than it was even a decade ago. With the increased complexity of computer systems and a variety of applications, it is hard for software developers to master the expertise required to deal with the variety of security concepts, methods, and technologies that are required in software projects. Although a large number of security learning materials are widely available in books, open literature or on the Internet, they are difficult for learners to understand the rationale of security topics and correlate the concepts with real software scenarios. We argue that the traditional approach, which usually organizes knowledge content topically, with security-centric, is not suitable to motivate learners and stimulate learners' interest. To tackle this learning issue, our research is focused on forging a contextualized learning environment for software security where learners can explore security knowledge and relate it to the context that they are familiar with. This learning system is developed base on our proposed context-based learning approach and based on ontological technologies. In this paper, we present our evaluation study in the open source software (OSS) development environment. Our results demonstrate that contextualized learning can help OSS developers identify their necessary security information, improve learning efficiency and make security knowledge more meaningful for their software development tasks

1. INTRODUCTION

Security has become an important part of today's software development projects. Improving software security requires that software engineers acquire relevant knowledge and skills to secure software development such that they can resist attacks and handle security errors appropriately [1]. However, learning software security has become a complex and difficult task today than it was even a decade ago [2, 3]. Nowadays, with the increased complexity of computer systems and a variety of applications, the intricacy of software development projects have been grown consistently. Each software product and process is different in terms of goals and contexts. It is hard for software developers to master the know-how required to cope with the variety of security concepts, methods and technologies that are required in software projects. Developers are often exposed to this diversity, which makes the software discipline inherently experimental [4, 5].

On the other hand, security knowledge can be both dynamic and situation specific [6], and the complexity of knowledge usually exceeds the capacity of individuals to solve problems by themselves. Learners must not only deal with a variety of security attacks and countermeasures but also have to demonstrate the applicability of the knowledge through experience in order to understand their practical use. Although much security information is widely available in the form of checklists, standards, and best practices in books, open literature or on the Internet [7-9], it remains difficult for software engineers to correlate relevant pieces of security knowledge to apply to their application-specific situations. The traditional learning materials, which

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

usually organize knowledge content topically, with security-centric, are difficult for developers to understand the rationale of security topics and correlate the concepts with real software scenarios. Developers or security learners often feel that the security knowledge is such extensive and software security is so difficult to achieve, that they simply cast it aside.

Keeping in view of aforementioned facts, our position is that security knowledge should be contextualized and placed in a meaningful situation that makes sense to the learners to enhance their understanding and make the concepts more relatable. As Gary McGraw points out, the domain of software security is rather context-specific, and the real project situation is necessary to apply the security concepts within the specific system [6]. Researchers have indicated that studying from a context and then abstracting the knowledge gained to be able to use it in a new context is a common way of learning programming that has been observed extensively in both new and experienced programmers [10, 11]. In computer science education, there is also a broad agreement that teaching units should start from a "real-world" context or phenomenon, aiming to create connections to prior knowledge, to increase the relevance of the material to students or to show application situations of the intended knowledge, thereby increasing motivation [12-15].

To this end, our research is focused on forging a software security learning environment where learners can explore security knowledge and relate it to the context that they are familiar with. We have proposed a learning system for software security with a context-based learning approach, which adaptively places security knowledge in the appropriate context of the software development. We have previously carried out two evaluations for the proposed learning approach and the learning tool in a university learning environment. The experiments showed that both the context-based learning approach and the tool not only yielded significant knowledge gain compared to the conventional approach but also gains better learning satisfaction of students. As part of an investigation into contextualized learning in the domain of software security, we are also interested to discover and examine the impact of the learning approach in real software-project environments. In this paper, we present our evaluation study in the open source software (OSS) development environment. Our results demonstrate that the contextualized learning can help OSS developers

The paper is organized as follows. After the introduction, in section 2, we introduce the theoretical background of this study. Section 3 describes the proposed contextualized learning system. In section 4, we describe the method of the evaluation study. Section 5 presents the result of the evaluation. In section 6, we discuss the results. Lastly, the conclusion is presented in section 7.

2. CONTEXTUALIZED LEARNING

Contextualized teaching and learning builds upon a similar concept of putting learning activities into perspective to achieve the best teaching and learning outcomes. Researchers Berns and Erickson define contextualized learning as a practice that endeavors to link theoretical constructs that are taught during learning, to practical, real-world context [16]. The underlying theme behind contextual learning activities is simple. It recognizes that by embedding instructions in contexts that adult learners are familiar with, learners more readily understand and assimilate those instructions. Naidu [17] also points out that learning is most

effective when learners work on realistic problems with guidance. Contextualized experience helped them develop deeper understanding that positioned them to better comprehend the abstract idea, and see how it manifested in actual contexts [18].

Contextualized instruction in general, starts with presenting a context from which the concepts are developed on a need-to-know basis [19]. This requires teachers to teach in a more constructivist way, i.e. to position the concepts of the learning subject in contexts recognizable to students and to stimulate active learning of the students [20]. The contextualization of the learning on demand can not only be seen from the point of view of an actual problem or learning situation but also in a longer lasting process of learning activities that are integrated [21]. Therefore, a context for a software security topic includes the circumstances in which its technical content exists. To talk about software security in context is to say that knowledge would not only include the basic principles and processes of software security but would consider how security knowledge is used in one or more particular domains or application areas.

The concept of the learning in context has been widely addressed in education and psychology literature over the years, and the effectiveness of contextualized learning has been demonstrated in the setting of interactive school classrooms. However, it is still unclear how this concept can be synthesized and applied in the domain of software security. Our study aims to mitigate this research gap by delivering a tool-based contextualized learning approach to facilitate software security learning in a way that can motivate learners.

3. CONTEXTUALZIED LEARNING SYSTEM FOR SOFTWARE SECURITY

The basic concept of the contextualized learning system is to facilitate the contextual learning process by providing contextualized access to security knowledge through real software application scenarios. To develop this kind of learning system, we first proposed a context-based learning approach to regulate contextualized learning process about software security. Following the proposal of the learning approach, we designed the kernel ontology-based knowledge repository and the system user interfaces. Figure 1 depicts the design consideration of the contextualized learning system.

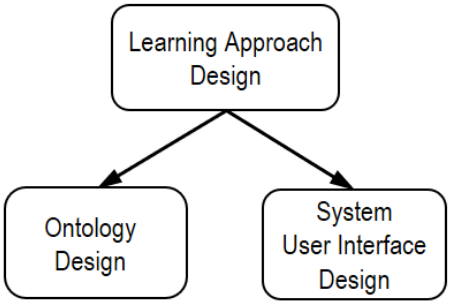

Figure 1. Design consideration of the learning system

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

3.1 The Proposed Context-Based Learning Approach

To facilitate contextualized learning about software security and create engaging learning experiences for learners, we proposed a contextualized approach for software security learning with three strategies.

3.1.1 Starting with a Meaningful Scenario

Contextualized learning often takes the form of real-world examples or problems that are meaningful to the learners personally [22]. Creating the relevance of the learning knowledge before going into the details could provide a stronger foundation for the learning process. Therefore, to begin the process of learning, a meaningful situation for learners must first be established. In our study, the learning situations are created through the use of contextual software scenarios, which refer to different manifestations within an application context. We choose a scenario-based approach because scenarios can be easily adapted to the situation of the represented applications and can be easily integrated with the contextualized security knowledge. In essence, this scenario-based strategy draws on situated knowledge - that is, understandings particular to the software problems or situations in which they are generated. At the same time, scenarios, inherently possess the dramatic potential to optimize learning processes and outcomes.

3.1.2 Stimulating Mental Models for Learning

Contextual learning is a learning approach that ties brain actions in creating patterns that have meaning [23]. In order to help learners make sense of complex security knowledge and create a strong and lasting bond among security concepts while they are engaged through various anchoring events, our strategy is to elicit learners' mental models for the navigation of security knowledge. Such mental models allow learners to gain insight regarding their world by building a work scheme, which makes it easier for them to access the information needed to understand the knowledge domain, make predictions, and decide upon action to take [24]. In order to be useful explanatorily, a mental model has to have a similar relation-structure to the reality it models. Then the constructed mental model can be used to answer questions or solve problems [25].

Generally, our intention was to guide learners in answering three questions while dealing with each software scenario:

- What are the possible attacks?
- Why does it encounter attacks?
- How can these attacks be prevented?

The knowledge structure serves as the basis for both knowledge retention and retrieval, as well as transfer. Once learners answer what–why–how questions, the relationships between the security concepts are revealed in their midst, and thus, their representation of mental models expands.

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

3.1.3 Moving from Concrete to Abstract

To help learners gain a more flexible understanding of the study concept in a range of situations with varying levels of abstraction, we organize security knowledge by blending abstract and concrete perspectives; presenting it with a sequence from concrete to abstract. The used concrete-to-abstract approach in knowledge presentation differs from the traditional, where the concepts are of foremost importance and are usually explained first before concrete examples and applications are discussed. In such concrete-to-abstract knowledge presentation, learners discover meaningful relationships between practical functions and abstract knowledge in the context of real applications. Psychologists and educators have indicated that abstract understanding is most effectively achieved through experience with perceptually rich, concrete representations [26], while concrete materials make concepts real and therefore easily internalized [27]. As long as the concrete knowledge and the underlying abstract explanation are understood by learners, learning transfers from one context to another will be more effective.

3.2 The Underlying Ontology

The role of the ontology in this learning system is to provide a vocabulary for representing knowledge about the software security domain and for providing linkages with specific situations in the application context. Ontologies facilitate capture and construction of domain knowledge and enables representation of skeletal knowledge to facilitate the integration of knowledge bases irrespective of the heterogeneity of knowledge sources [28]. Figure 2 shows the ontology-based knowledge model, which consists three sub-models: application context model, security domain model and security contextualization model. With this model, the learning system can handle contextualized security knowledge with multiple scenarios in different application-specific contexts and integrates security concepts of security domain knowledge.

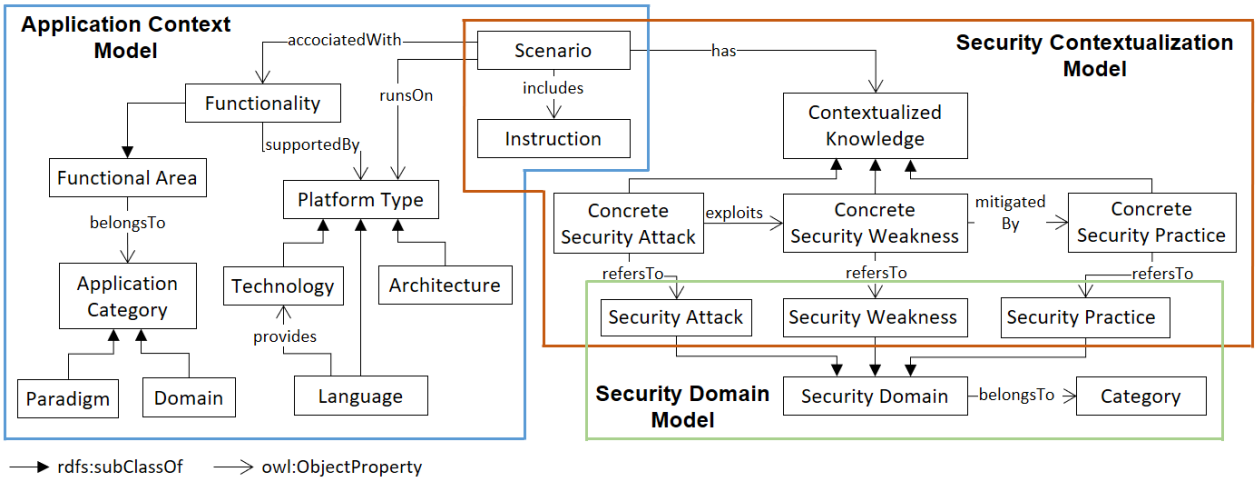

Figure 2. The ontology-based security knowledge model

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

3.2.1 Application Context Model

The context model represents a definition of what context is in a specific domain. In our ontology, the context for software security knowledge is supported by the creation of scenarios in different application contexts. The scenario presents a snapshot of possible features and corresponding code fragments in the specific functionality that is included in the *Instruction* class. It also draws on situated security knowledge, that is, understandings particular to the application context in which they generate. In addition to scenarios, we focus on characteristics that are highly relevant for retrieval within a software application, concerning three perspectives:

- The application category that scenario/functionality belongs to,
- The platforms that the scenario functionality used, and
- The functional area (and the corresponding functionalities) that the application associated with.

3.2.2 Security Domain Model

The security domain model describes the knowledge that is an object of teaching through a set of concepts (topics to be taught). To design a security knowledge structure (schema) that is easier to store in the learners' memory for learning, the schema should be simplified and kept to the point for reducing the content load. Therefore, we identify three security concepts that are most widely used throughout the security domain. Ultimately, three classes were incorporated into the security domain model: *Security Attack*, *Security Weakness*, and *Security Practice*. From a security domain point of view, we only want to indicate which principles or abstract ideas are needed, not their practical implementation. Therefore, we describe security knowledge in this model at a level of abstraction. The instances of these classes specify only the fundamental characteristics of the security concepts, not specific software application aspects. The main advantage of this design is to share a common understanding of the conceptual security knowledge among different security contexts.

3.2.3 Security Contextualization Model

The term contextualization is used here to describe the process of drawing specific connections between security domain knowledge being taught and an application context in which the domain knowledge can be relevantly applied or illustrated. To this extent, the security contextualization modeling manages security knowledge in the context of specific scenarios and brings together the conceptual knowledge that is described in the security domain model. The including security concepts are aligned with those defined in the security domain model, which are *Security Attack*, *Security Weakness*, and *Security Practice*. However, in order to clearly state the purposes and distinguish them from the security domain model, we use different classes, namely *Concrete Security Attack*, *Concrete Security Weakness*, and *Concrete Security Practice*. The abstract class *Contextualized Knowledge* is used from which these three classes inherit common attributes such as tags or external resources. Once the domain knowledge model is defined, each security concept in the contextualized model is able to be connected to the corresponding classes in the security domain model.

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

3.3 System Prototype

3.3.1 System Architecture

A proof-of-concept prototype of the contextualized learning system was implemented as described above. A general architecture of the system is presented in Figure 3.To construct the ontology, we used Protégé and OWL1 Editor because of its simplicity and popularity [29]. When searching the ontology, we use SPARQL protocol to extract information from the RDF graph. The front-end was designed as a web-based user interface with PHP and JavaScript languages and through it, learners can access the knowledge content. The backend was implemented in Java and access to the ontology repository was provided through the Jena API[2], a Java framework for building semantic web applications. Jena provides extensive Java libraries for helping developers develop code that handles RDF, OWL, and SPARQL in line with published W3C recommendations[3].

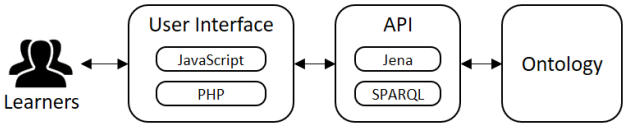

Figure 3. System architecture diagram

3.3.2 Learning Process

The sysem features are shown in Figure 4, in which a scenario of "Accessing database using user input" under Web Application paradigm is demonstrated. Figure 5 illustrates how learners are guided by the learning process of the system. The learning process begins with a selected contextualized scenario in the application context familiar to learners and then gradually leads to an understanding of the abstract part of security knowledge. First of all, the learner defines criteria from the application-context menu to scope the learning session based on his (or her) desired knowledge. The instructional part of the scenario is made up of practical demonstrations of the pre-described application functionality and the code fragments behind it.

To guide learners navigating through the contextualized knowledge efficiently, we outline the knowledge contents in a graphical Concept Map, which shows in the left-corner of the screen. Concept Map is a visual representation of different concepts and their relationships. The contextualized concept map demonstrates how security knowledge can be made more relevant with linkage of real-world items by demonstrating their relationships. With the use of concept mapping, the learning arena becomes transparent and can be virtualized in a learner's mind [30]. This transformation is essential for learners in order to integrate the semantical impact of the knowledge structure into the mental models for efficient learning.

While a node is clicked on the concept map, the knowledge content correspondent to this concept is displayed in the right half of the screen, where the upper part is the contextualized

[1] Web Ontology Language (OWL), a markup language based on Resource Description Framework/Extensible Markup Language (RDF/XML).
[2] https://jena.apache.org/
[3] https://www.w3.org/2001/sw/

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

knowledge and the lower part is the abstract explanation, following the concrete-to-abstract presentation strategy. By concrete representations, we include perceptually detailed and rich materials, such as demonstrating security attacks with different exploits, identifying mistakes in the source code, and showing the secure coding practices to fix the mistakes. With the scenario instruction displaying aside, learners can easily recall the demonstrations of the software functions without interrupting the learning process. After experiencing the facts, learners then move on to the section of abstract knowledge, where the corresponding conceptual knowledge is presented. In such an environment, learners discover meaningful relationships between the abstract explanation and the practical demonstration in the context of real software applications; security concepts are internalized through the process of discovering, reinforcing, and relating.

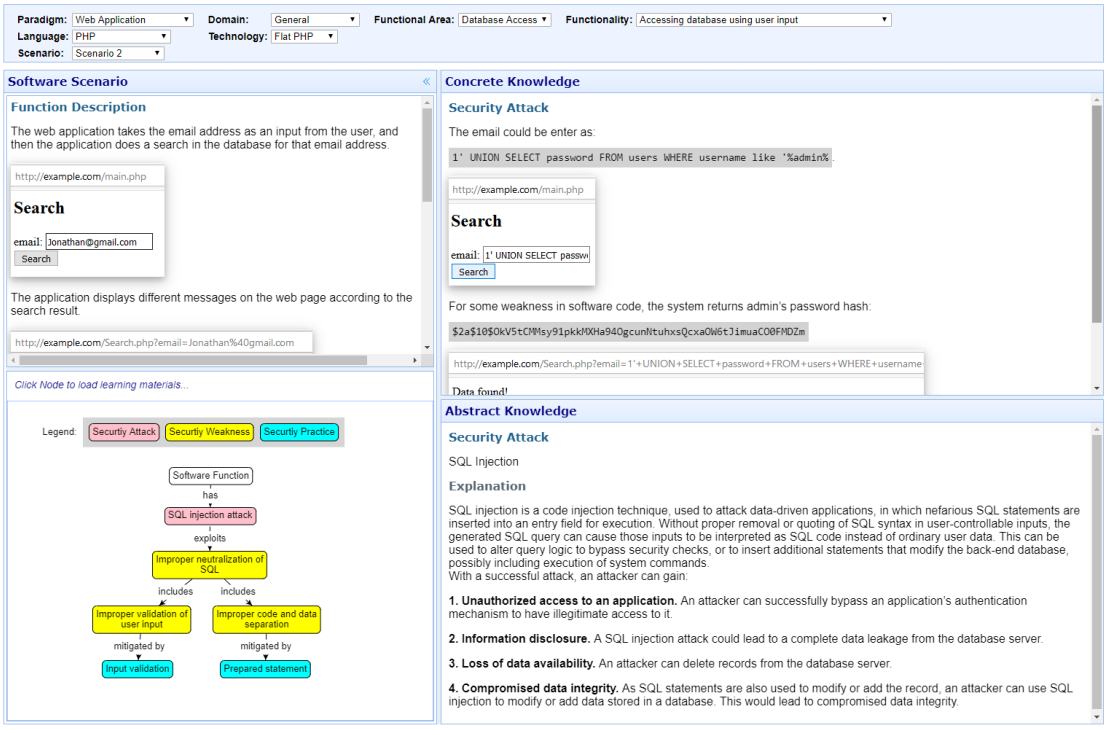

Figure 4. Snapshot of the contextualized learning system

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

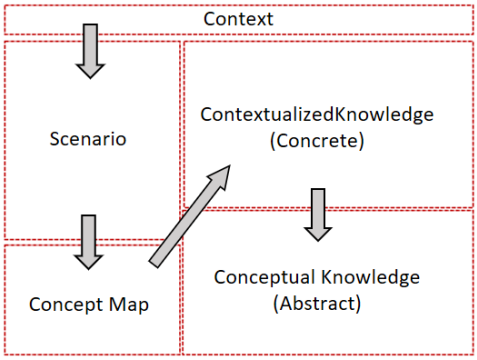

Figure 5. Learning process in the system

4. STUDY METHOD

4.1 Study Setup

In preparation for the study, we identified two common software vulnerabilities in web applications: SQL Injection (SQLi) and Cross-Site Scripting (XSS) as the learning subjects. SQLi and XSS were among the OWASP's Top 10 [31] most critical web application vulnerabilities in the past decade. For preparing ontology of the system, we first set up the learning environment in a web application paradigm, an e-Store. For this specified context, the author developed two sets of functionalities to operate a web-based e-Store application using two different programming languages: PHP and Java, including a login module, data input/output features, data processing, database access, and payment functions. Three scenarios were manipulated under critical functionalities to demonstrate the two vulnerabilities within the scope of the e-Store system, including the corresponding vulnerable code fragments, exploits and mitigations. With the readiness of the real software scenarios, we then constructed all learning materials and filled the ontology via Protégé application.

4.2 Data Collection

This study was designed to examine the potential of adopting the idea of context-based learning system for software security for OSS developers. For the purpose, the use of a survey is deemed appropriate in this study, as the survey enables clear, direct, and objective answers to the questions presented to the respondents [32]. In this study, a self-administered web-based questionnaire was used to collect individual-level perception data from participants in OSS projects. The purpose of the questionnaire was to validate the learning system by eliciting respondents' perception and opinions of the learning approach and system features that support software-security learning in OSS projects. The survey instruments, which consisted of four sections, was created and hosted using Google Forms. Section 1 addressed demographics information of participants. In section 2, respondents were asked to rate the system features (Table 1), ranging from "very impractical" to "very practical", administered in accordance with the 5-point Likert scale. Section 3 dealt with the learning approaches embedded in the system. Respondents were required to choose the answer that reflects their own views and stance on

the statements which were ranged from “strongly disagree” to “Strongly agree”, with 5-point Likert scale (Table 2). In the last section, participants were allowed to share their thoughts or suggestions on all aspects of the learning system.

Table 1. Survey items for learning satisfaction about system features

Evaluation Item	Question
Software Scenario	• The system introduces security subjects using common software functions.
Concept Map	• The system uses a graphical concept map to outline the knowledge content.
Security Concepts	• The system forms the main theme of security learning using three concepts: Security Attack, Security Weakness and Security Practice.
Contextualized Knowledge	• The system demonstrates practical security knowledge in connection with the scenario.
Concrete-to-Abstract	• The system guides learners studying concrete/practical security knowledge first, then the abstraction/theory.

Table 2. Survey items for learning satisfaction about learning approaches

Evaluation Item	Question
Effectiveness	• This system can effectively assist learners in obtaining software security knowledge.
	• The learning approach reduces the difficulty of learning software security.
Experience correlation	• The approach helps me relate security knowledge to what I knew or experienced before.
Interest Promotion	• The approach promotes my interest in learning software security.
Learning Preference	• The system guides learners studying concrete/practical security knowledge first, then the abstraction/theory.

All data collected through the survey was non-identifiable. Each participant received the research invitation and survey link via email. Implied consent was obtained by the informational letter sent through the email. Participants were sent two invitation e-mails over a period of 4 weeks in April and May 2016.

4.3 Participants

For setup of this study, we recruited OSS developers on GitHub by sending out research invitation between January 2019 and February 2019. The email invitation included an introduction of the research, and links to the learning system and to the survey site. The only participation requirement of participants was the experience of web application development. Total 21 voluntary participants accepted the invitation, and completed the questionnaire after trying out the system.

5. RESULT

5.1 Respondent Demographics

Table 3 describes the general demographic information of the 21 respondents, in terms of gender, age and seniority in OSS development. 90% of respondents were male, while there

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

were only 2 female respondents. A large body of participants, that is 85%, was between 20 and 40 years old, and over 70% of respondents had over 3 years of experience in OSS development. As shown in Figure 6, Java, Python and PHP are the top 3 programming languages that most respondents are familiar with in this study.

Table 3. Demographic analysis of the respondents (*n*= 21)

Item	Category	Frequency	Percentage
Gender	Male	19	90.48%
	Female	2	9.52%
Age	<20	1	4.8%
	20–30	12	57.1%
	31–40	6	28.6%
	41–50	2	9.5%
Seniority in OSS development	6 months to 1 year	1	4.8%
	1 to 3 years	5	23.8%
	3 to 5 years	9	42.9%
	More than 5 years	6	28.6%

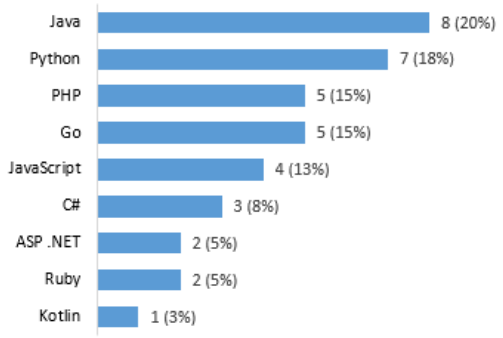

Figure 6. The programming languages that respondents are familiar with

5.2 Satisfaction Analysis for System Features

The mean scores of the system features are plotted as a radar chart with five axes (Figure 7) according to each evaluation item. As can be seen from the chart, the mean scores of the system features ranged from 4.00 (for Contextualized knowledge) to 4.67 (for Concept map). The highest rating category made by the respondents was “Concept map”. Most of the respondents expressed that the design of Concept map was attractive and though it was useful to guide the learning process. They commented:

“I like the color-design concept. Neat and simple. Easy to follow.”

“Have a node graph that helps me a lot to see stuff not in paragraph form, but to capture the cause and effect.”

“The sense of connecting security problems and solutions, is really good.”

Respondents also recognized the use real software scenarios in introducing security knowledge. One respondent stated:

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

"When I learn [software] security, I have a very fuzzy view to begin with, and then I kind of work at it, read about it, and wait for lightbulb to go on. I think [to start with] cases help me turn those lightbulbs on immediately."

In addition, most also appreciated the arrangement of contextualized and abstract security knowledge in the system. Some of the comments were indicated below:

"[…] clear and concise. Straight to the point, easy to understand"

"That way the sample code and the description are put together helps me learn the [security] concepts."

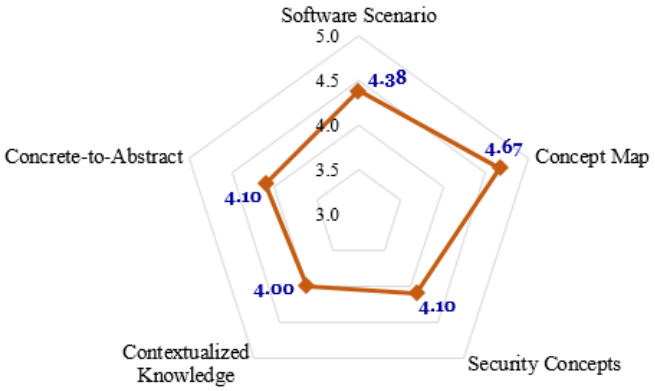

Figure 7. The mean score of system features

5.3 Satisfaction Analysis for the Learning Approach

We carried out reliability tests using IBM SPSS software by calculating Cronbach's alpha to examine the internal consistency of the five evaluation items within the category of "Learning approach", and determine the scale in questions is unidimensional (Figure 8). The resulting alpha value derived 0.834, which were above the acceptable threshold (0.70) suggested by Nunnally [33]. Thus, the survey items on the instrument are deemed highly reliable and appropriate for such research.

Case Processing Summary

		N	%
Cases	Valid	21	100.0
	Excluded[a]	0	.0
	Total	21	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.834	5

Figure 8. Reliability test

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

To understand respondents' perception regarding the learning approaches embedded in the system, we carried out descriptive statistical analysis for the five survey items. Table 1 shows the analysis result, including the frequency of the valid values, means and the standard deviation. The result shows that mean scores for the five survey items all reached 4, indicating a high overall satisfaction for the learning approach expressed by the respondents. To obtain a closer view of the respondents' perception with our proposal, we depicted the proportion of responses of each survey item in Figure 9. From the perspective of simplicity learning, the vast majority of respondents (91%) expressed their agreement that the learning approach can reduce difficulty of learning software security. In line with this, 85% of respondent agreed that the leaning approach create conditions for effective leering about software security. In addition, over 80% of respondents thought that the learning approach fits their learning preference and promote their interest in learning software security. They expressed their though about the advantages of the proposed learning approach. For example:

"I highly recommend your method. Teaching practice first. Developers can derive understanding for the theory easier from the practice instead of doing it the other way round."

"Software security needs to be practical; it needs to be related to something, to be given contrast to something. So it becomes really interesting when I reach your ideas. But where there is so much theory it's also a bit hard to understand."

Last, 71% of respondents agreed that the learning approach helped them relate security knowledge to their prior experience. One respondents supporting the statement commented:

"When I relate the cases to the practical things that I do in my project, the security concepts become more applicable and easier to understand."

However, we found that the survey item, Experience correlation, got the least satisfaction (Mean = 4.05) in the category. Seven respondents, that is one-third, did not hold a positive agreement with the statement, and the neutral responses were relatively high (six respondents). Probing into this issue, we identified respondents' comments related to this survey item. They reported that their specialties were not within the knowledge scope that the system currently provided. For example, a respondent who was familiar with Python stated:

"I've used Python for many years. I expect this [programming language] will be included in your code examples."

Table 4. Descriptive analysis of satisfaction for the learning approach

	Frequency						
Item	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Mean	Std. Deviation
Difficulty reduction	0	0	3	11	7	4.19	0.700
Effectiveness	0	0	2	14	5	4.14	0.573
Learning preference	0	0	4	10	7	4.14	0.973
Interest promotion	0	0	4	8	9	4.24	0.949
Experience correlation	0	1	5	7	8	4.05	0.928

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

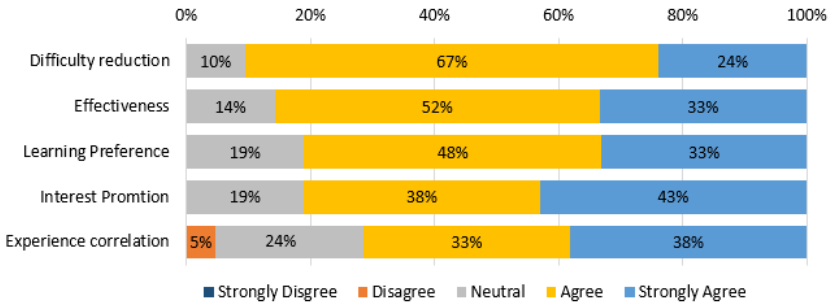

Figure 9. The proportion of responses for survey items of learning approaches

6. DISCUSSION

The results of this study indicate that our proposed learning system has the potential to be an effective learning tool that can motivate OSS developers to learn about software security. First, the respondents overall evaluated the practicality of system features with a positive degree. They highly recommended the use of software scenarios with the graphical and contextualized security knowledge presentation. With a clear and visualized layout, they could sort out the desired knowledge quickly. Second, the results also indicated the learning approach kept developers interested and engaged. They overwhelmingly expressed their satisfaction with the learning sessions. Such benefits of the contextualized approach can be explained by the effective mechanism of intrinsic motivation, where a learner is drawn to engage in a task because it is perceived as interesting, enjoyable, and/or useful [34-36].

Based on the findings presented in the study, we deem contextualized learning a suitable approach to support developers' security training and education in software projects. In OSS, development and maintenance of qualified and secured software products relies mainly on the ability of participants to acquire, refine and use new aspects of secure programming knowledge in their projects [37]. With proper contextual guidance, developers can identify their necessary security information, improve learning efficiency and make security knowledge more meaningful for their software development tasks. The contextualized approach helped the developers to see how the various security concepts were inter-related in their works and gave them the personalized perspective that they valued. Therefore, their learning experience can be related to a similar programming topic that they want to learn about or a problem to be solved in their projects. In addition, when developers encounter the security problems within the context they are already familiar with, the consequences of exploiting the code's vulnerabilities will be understood with a strong and personal effect, which become more real and less theoretical.

From this study, we also draw some lessons for further improvements to this learning system. First, we need to create more contextual scenarios and equip corresponding security knowledge in the system to expand the knowledge scope. The learning sessions can then be cast in the contexts which are more closed to learners' working environments. Additionally, the respondents also indicated that they cannot grasp abstract explanation of security concepts because of the heavy embedded textual descriptions. The abstraction knowledge we built was

extracted from the resources on the internet (e.g. OWASP and CWE[4]). It is suggested that we decompose the vast information into smaller knowledge objects to further ease leaners' loading. With the defined relationships in the ontology, these new instances can also be illustrated in the concept map to support knowledge navigation. For example, the security practice of "Input validation" can be broken down into flat text validation, rich text validation and file upload validation, etc. We are proactively working on these improvement in preparing for longer-term studies.

7. CONCLUSION

In this study, a web-based learning system was conceptualized and developed to support contextualized learning about software security. We have presented the design rationale, including the embedded learning strategies and underlying ontological knowledge repository. Our approach attempts to place security learning in the context of software projects that can draw developers' attention to similar software events and conditions. We aims to help learners organize security knowledge by connecting concepts to real software scenarios, to motivate learners and stimulate their interest. The contextualization of security knowledge make it possible to support developers reflect on their learning to bridge ideas from a familiar concrete context so they can recognize their own personal relationship to these concepts.

The proposed learning system was evaluated through an online survey with 21 developers in OSS projects. Overall, the analysis of the survey data yielded positive and promising results, in which OSS participants overwhelmingly expressed their satisfaction with our proposal, in perspectives of system features and the embedded learning approach. They enjoyed the experience, found the subject matter interesting and found the presentation helpful. This finding demonstrate that our approach is not only possible but also practical to be adopted by software development projects. We are encouraged by the results of the context-based approach and believe it provides a formula for increasing the attitude and understanding of security subjects for developers without sacrificing rigor or quality of learning. We believe this implies a direct effect of the contextualized learning approach on higher overall learning satisfaction, which motivates developers to learn.

Several limitations of this study should be noted. First, this evaluation was based on self-reported data from voluntary participants about their experience and perceptions for the proposed learning system. It is not certain their actual behavior on the system, the span of time they practice the system, and for how long the knowledge will be retained. Moreover, the number of respondents obtained from the survey was relatively small compared with the enormous number of OSS projects and field workers today. We intend to invite more OSS participants from various domains joining future sessions, meanwhile, to conduct in-depth interviews to collect more detailed information about their thoughts and learning behaviors.

[4] Common Weakness Enumeration (CWE) is a universal online dictionary of weaknesses that have been found in computer software. https://cwe.mitre.org/

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

REFERENCE

[1] Bishop, M. (2010), "A Clinic for" Secure" Programming". IEEE Security & Privacy, volume 8, issue 2, pages.

[2] Viega, J. and G.R. McGraw (2001), "Building secure software: how to avoid security problems the right way". volume: Pearson Education.

[3] Barnum, S. and G. McGraw (2005), "Knowledge for software security". IEEE Security & Privacy, volume 3, issue 2, pages 74-78.

[4] Basili, V.R. and H.D. Rombach (1991), "Support for comprehensive reuse". Software engineering journal, volume 6, issue 5, pages 303-316.

[5] Lindvall, M. and I. Rus (2000), "Process diversity in software development". IEEE software, volume 17, issue 4, pages 14-18.

[6] McGraw, G. (2006), "Software security: building security in". volume 1. MA,USA: Addison-Wesley Professional.

[7] Shuaibu, B.M., et al. (2015), "Systematic review of web application security development model". volume 43, issue 2, pages 259-276.

[8] Mohammed, N.M., et al. (2017), "Exploring software security approaches in software development lifecycle: A systematic mapping study". volume 50, issue, pages 107-115.

[9] Wen, S.-F. (2017), "Software Security in Open Source Development: A Systematic Literature Review". in Proceedings of the 21st Conference of Open Innovations Association FRUCT. Helsinki, Finland.

[10] Ko, A.J. and B.A. Myers (2008), "Debugging reinvented: asking and answering why and why not questions about program behavior". in Proceedings of the 30th international conference on Software engineering. ACM.

[11] Apvrille, A. and M. Pourzandi (2005), "Secure software development by example". IEEE Security & Privacy, volume 3, issue 4, pages 10-17.

[12] Cooper, S. and S. Cunningham (2010), "Teaching computer science in context". Acm Inroads, volume 1, issue 1, pages 5-8.

[13] Guzdial, M. (2010), "Does contextualized computing education help?". ACM Inroads, volume 1, issue 4, pages 4-6.

[14] Diethelm, I., P. Hubwieser, and R. Klaus (2012), "Students, teachers and phenomena: educational reconstruction for computer science education". in Proceedings of the 12th Koli Calling International Conference on Computing Education Research. ACM.

[15] Guzdial, M. (2006), "Teaching computing for everyone". Journal of Computing Sciences in Colleges, volume 21, issue 4, pages 6-6.

[16] Berns, R.G. and P.M. Erickson (2001), "Contextual Teaching and Learning: Preparing Students for the New Economy. The Highlight Zone: Research@ Work No. 5". volume, issue, pages.

[17] Naidu, S. (2008), "Situated learning designs for professional development: Fundamental principles and case studies". in Fifth Pan-Commonwealth Forum on Open Learning.

[18] Giamellaro, M.J.I.J.o.S.E. (2014), "Primary contextualization of science learning through immersion in content-rich settings". volume 36, issue 17, pages 2848-2871.

[19] Bennett, J., F. Lubben, and S.J.S.e. Hogarth (2007), "Bringing science to life: A synthesis of the research evidence on the effects of context-based and STS approaches to science teaching". volume 91, issue 3, pages 347-370.

[20] Parchmann, I., et al. (2006), ""“Chemie im Kontext”: A symbiotic implementation of a context-based teaching and learning approach". volume 28, issue 9, pages 1041-1062.

[21] Specht, M. (2008), "Designing contextualized learning", in Handbook on information technologies for education and training, Springer. pages 101-111.

[22] Rivet, A.E. and J. Krajcik (2008), "Contextualizing instruction: Leveraging students' prior knowledge and experiences to foster understanding of middle school science". Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, volume 45, issue 1, pages 79-100.

[23] Davtyan, R. (2014), "Contextual learning". in ASEE 2014 Zo. 1 Conf.

[24] Rouse, W.B. and N.M. Morris (1986), "On looking into the black box: Prospects and limits in the search for mental models". Psychological bulletin, volume 100, issue 3, pages 349.

[25] Kieras, D.E. and S.J.C.s. Bovair (1984), "The role of a mental model in learning to operate a device". volume 8, issue 3, pages 255-273.

[26] Goldstone, R.L. and J.Y. Son (2005), "The transfer of scientific principles using concrete and idealized simulations". The Journal of the Learning Sciences, volume 14, issue 1, pages 69-110.

[27] Kamina, P. and N.N. Iyer (2009), "From concrete to abstract: Teaching for transfer of learning when using manipulatives". in Proceedings of the Northeastern Educational Research Association (NERA) 2009.6. https://opencommons.uconn.edu/nera_2009/6.

[28] Gruber, T.R. (1995), "Toward principles for the design of ontologies used for knowledge sharing?". International journal of human-computer studies, volume 43, issue 5, pages 907-928.

[29] Tudorache, T., et al. (2013), "WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web". Semantic web, volume 4, issue 1, pages 89-99.

[30] Shambaugh, N. (1995), "The cognitive potentials of visual constructions". Journal of Visual Literacy, volume 15, issue 1, pages 7-24.

[31] OWASP, "OWASP Top 10 Application Security Risks - 2017"; Available from: https://www.owasp.org/index.php/Top_10-2017_Top_10. (Accessed on)

[32] Berry, L.M. and J.P. Houston (1993), "Psychology at work: An introduction to industrial and organizational psychology". volume: Brown & Benchmark/Wm. C. Brown Publ.

[33] Numally, J.C. (1978), "Psychometric theory". NY: McGraw-Hill, volume, issue, pages.

[34] Kozeracki, C.A. (2005), "Preparing faculty to meet the needs of developmental students". New directions for community colleges, volume 129: Responding to thechallenges of developmental education, issue, pages 39-49.

[35] Dean, R.J. and L. Dagostino (2007), "Motivational factors affecting advanced literacy learning of community college students". Community College Journal of Research Practice, volume 31, issue 2, pages 149-161.

ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and SecurityAugust 2019, Article No.: 58, Pages 1–10
https://doi.org/10.1145/3339252.3340336

[36] Cordova, D.I. and M.R. Lepper (1996), "Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice". Journal of educational psychology, volume 88, issue 4, pages 715.

[37] Wen, S.-F. (2018), "Learning secure programming in open source software communities: a socio-technical view". in Proceedings of the 6th International Conference on Information and Education Technology. ACM.