Capturing Software Security Practices using CBR: Three Case Studies

Ikram Elrhaffari[1], Ounsa Roudies[2]
Ecole Mohammadia d'Ingénieurs, SIWEB Team
University Mohammed V
Rabat, Morocco

Abstract—Generally, software security can be regarded as one of the most important issues in software engineering field since it may affect the software product effectiveness due to the various technological vulnerabilities and menaces. Most traditional software security approaches provide security activities through the software development lifecycle (SDLC) from requirements to design, implementation, testing and deployment. This paper focuses on embedding security concerns in the software development lifecycle (SDLC) using a bottom-up approach that is based on case based reasoning (CBR) paradigm. Thus, we study three high security-focusing cases for software projects, namely "e-shop", "Mobiling" and "intranet" using a structured case study method. Then, we populate these three cases in the proposed framework that is an excerpt of the case project base. Furthermore, this paper identifies the specificity of each case, discusses completeness of the proposed framework and proposes suggestions for improvement. Finally, usages scenarios are defined sustaining the use of the proposed framework.

Keywords—CBR; project features; case base; e-shop; mobiling; intranet; mutualize; security practices; security requirements

I. Introduction

Software engineering security has been discussed in many works with different perspectives. There are numbers of top-down security engineering approaches that cover the entire secure development life-cycle [1][2][3][4]. However, to the authors' knowledge, non bottom-up approaches have been performed on secure software development processes. In that sense, we put more focus on empirical approaches to consider security concerns in software development field.

The software engineering team has to learn more about software security in order to adopt, express, conduct, apply, review, and judge it properly [5]. According to [6] each technical member of a project (developer, tester, etc) should have a basic software security knowledge including concepts like security design, threat modeling, secure coding, security testing, etc.

The original intention was to act in the real world and change the software engineering team attitudes toward security aspects. So, the overall objectives were as follows:

- Allows project managers to maintain security practices repository for software engineering projects;
- Assist software engineering team in order to select and apply security practices in building software products.

The proposed bottom-up approach was inspired from the case based reasoning paradigm (CBR) [7][8][9]. The idea is to learn through experience and use it to make a new one successful. In this work, we focus on the first and second steps in the CBR cycle that correspond to retrieve and reuse security requirements and practices suggested by similar previous project cases.

In this way, we propose a framework that is an excerpt of the case project base. It helps the software engineering team to populate a project case base with the pertinent elements. It serves as a support to describe all project case characteristics that correspond to case descriptors. The proposed framework is structured as a class diagram, which models the project case features, in particular, security requirements and practices.

According to [10], structured method is adopted to select and design the case studies and to collect the related data. We emphasize on case quality that is indeed critical. Therefor, qualitative aspect is more important for us than quantitative. We experiment the proposed framework using three real case studies: "e-shop", "mobling" and "intranet". We notice that these cases have been selected due to their high level of reusability. In addition e-shop, mobling and intranet cases are security focused. Next, the framework is reviewed in order to ensure completeness. At the end, some usage scenarios are provided to ensure the use of the framework (see Fig. 1).

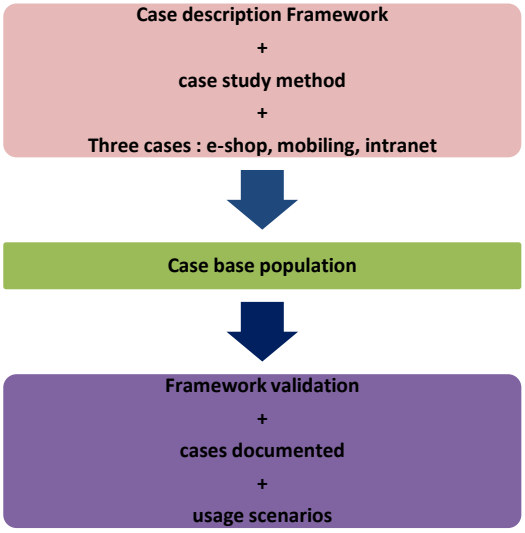

Fig. 1. The Proposed Approach.

The remainder of this paper is organized as the following: Section II presents the methodological approach that was adopted for the cases selection. Section III describes the "e-shop", "Mobiling" and "intranet" case studies through the proposed framework. In Section IV we discuss the completeness and validity of the framework. Section V presents some usage scenarios of the framework. Section VI provides a review of related studies. Finally, Section VII summarizes and set plans for future research.

II. Case Study Method

Yin [1] defines case study as "an empirical enquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident". In software engineering field, this can be particularly verified. In [1], the authors provide guidance for case study research in software engineering. this paper, we have followed the main suitable steps of this method.

A. Rationale

Case-based reasoning (CBR) can be seen as "solving a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation" [11]. In order to benefit from CBR in security engineering field, we propose a structured framework for software projects that describes the set of project characteristics, in particular, security requirements and related security practices (see Fig. 2). The idea is to retrieve security knowledge in similar previous situation (project cases) and reuse this knowledge in new project cases.

B. Objectives

The original objective is to investigate how to change the software engineering team attitude toward security and how to assist them to perform security practices through the entire software development life cycle. The objectives of this research are as the following:

- Objective 1: Validate the completeness of the framework and reveal any lack, ambiguity or superfluous elements;
- Objective 2: Populate the case base by the three cases e-shop, Mobiling and intranet;
- Objective 3: Illustrate use of the framework by two scenarios.

C. The Cases and there Context

In this paper, three projects are studied in a Moroccan company in three different application domains, both security focusing and using security practices. The initial definition of the context was only tentative to define all security project characteristics.

The company operates in the postal, banking and e-gov services. It takes careful consideration of information security issue and adopted ISO 27000 standard [12] since 2009. It is involved in global Moroccan information security concerns. The company is involved in the governmental initiatives for enforcing Digital Economy. We notice that first author of this paper is project manager and software practitioner in the company in which these projects took place.

The cases e-shop, Mobiling and intranet have been selected due to their high level of reusability. In addition, these projects are both security focusing and absolutely vital for the selected company. Finally, these three projects relates to heterogeneous domains of the company's management.

D. Theoretical Frame of Reference

Recent literature review reveals several research studies tackling software engineering security issues. There are several works about embedding security in software development lifecycle, but they often propose an ordered series of security activities that start from the SDLC input and are closely linked to SDLC phases [3][5].

Although extra attention is given by the academic literature to security engineering issues, there is a lack of theories devoted to bottom-up approaches. These approaches starts with real engineering projects and aims to capture and mutualyse organisation's know how. So, the idea is to retrieve security practices in order to capitalize them for new similar project cases. We build a case description framework according to Case Based Reasoning (CBR) theory.

E. Research Questions

We focus on the following research questions:

- RQ1: What are security requirements/practices captured from each case?
- RQ2: Is the framework accurate i.e. properly describing these security practices and the main case characteristics?
  - Is the framework appropriate and comprehensive?
  - If not, what are its limitations?
  - Are there any lacks or ambiguous aspects?
  - Are there any superfluous elements?
- RQ3: Are the selected cases pertinent?

F. Data Collection and Case base Populating

Data were collected through interviews with some project stakeholders, in particular, Project managers and software engineers and security engineers. we also received some documentations and deliverables from the project stakeholders.

Overall, the time needed to carry out interviews and to collected all possible data was 5 days hours.

III. Case Base Description and Population

In this section, we will use the proposed framework for mutualizing the selected cases and for illustration of applicability. Fig. 2 presents the proposed framework that is an excerpt from a global base case. This framework includes all classes and attributes needed to describe and document the case projects. The abstract class "feature" represent all case characteristics. We can clearly observe three main classes : "scope", "requirements" and "progress". The class scope is

used to define the case. it provide specific features like “man days”, “title”, “software engineering method”, etc. The class "requirements" is a special class that describes all project requirements. The class "progress" provides an overview of all characteristics defined in the project progress including steps and delivrables. The association "perform" links a security requirement to its security practices. This association allows to access performed security requiremements through the SDLC lifecycle.

We notice that the propoed framework is a template in which generic features can be enriched by additional information, so to provide more specific features. For example, we separated the generic feature “requirements” as “functional requirements” and “non functional requirements” (see Fig. 2).

According to the framework, we populate the Tables I, II, and III from metadata sources in order to characterize these different types of projects with the typical values for each feature. Table I illustrates an excerpt from case base that descripbe the “project scope” feature and related sub-features. We observe that, for each case, wide scope of functionalities are gathred. The “man day estimation” feature gives an indication about the size of the project.

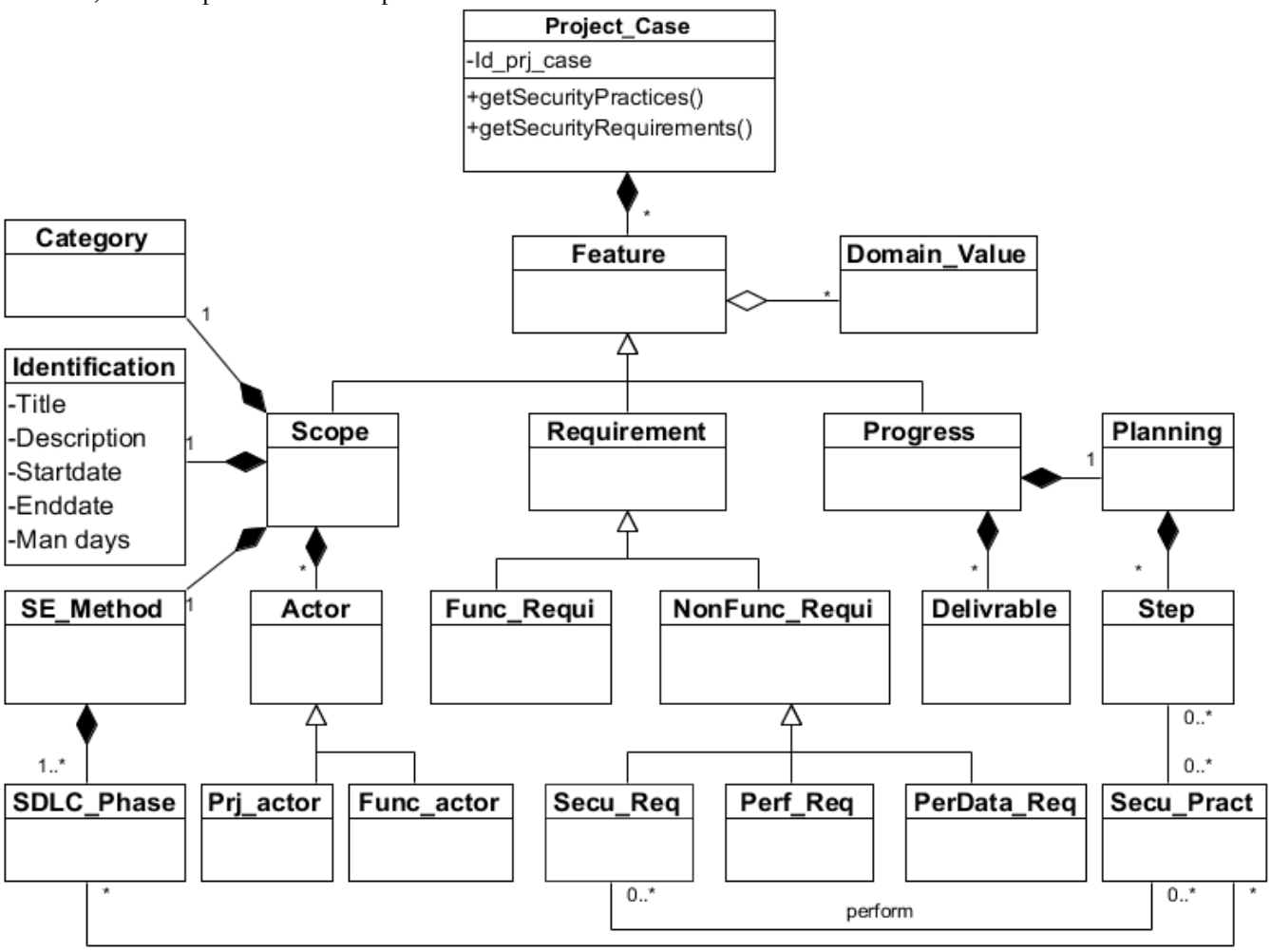

Fig. 2. Case Description Framework.

TABLE. I. EXTRACT FROM CASE BASE : PROJECT SCOPE

Project scope				
Section	***e-Shop project***	***Mobiling project***	***Intranet project***	***Values domain***
1. Project title	e-shop	Mobiling of postal distribution activities	Intranet	
2. Description	e-commerce website for third	Provide a solution that allows postmans to carry out their operations in the field using a mobile terminal	Provide a collaborative space within an organization, Various stakeholders are involved.	
3. Man days estimation	270	360	750	
4. Functionalities	e-marketing (promotion, best sales); e-Shop customer registration; basket management; Online payment; Order tracking; Merchants registration in Interbank center; Newsletters management (customer and others)	Mobiling of postal distribution activities; Monitoring postman activity; Frontoffice and backoffice solutions	Suggestion box, forum for the discussion,Newsletter, human resources data, exchange of experience through shared platform, intranet document repository	
5. Customer (target)	Enterprises (cooperatives…)	Postman, supervisors	Employees	Employees, enterprises
6. Actors				
6.1. Functional actors	merchant, customer, functional administrator, technical administrator, commercial	Postman, supervisor, manager, functional administrator	Employees, functional approver, functional administrator, technical administrator	Functional administrator, technical administrator, manager
6.2. Project actors	Project manager(provider side), developers, system administrator, Project manager (owner side), tester	Project manager (provider side), developers, system administrator, Project manager (owner side), security advisor, tester,	Project manager (provider side), developers, system administrator, Project manager (owner side), tester	Developer, project manager, system administrator, tester
7. Project type				
7.1. Category 1	Outsourcing	Outsourcing, backsourcing, internal development	Outsourcing , internal development,	Internal development, outsourcing, backsourcing
7.2. Category 2	Web-based	Domestic, mobile	Domestic	Web-based, domestic, mobile, app for smartphone, app for tablets
7.3. Category 3	Third	Business	Collaborative	Third, e-gov, collaborative, business
8. Software engineering method	Agile	Agile	UP	Agile, UP
9. Software lifecycle coverage	Total	Total	Total	Total, partial
10. Tools/technologies	Websphere e-commerce, DB2 (IBM), Jee, web services	Visual studio .NET, windows mobile, SqlServer, GPS, 3G, 4G, Wifi	SharePoint, SqlServer	

Table II contains a section for each requirement category. Across interviews, we identified three non functional requirements categories : security requirements, performance requirements and personal data requirements. We can conclude that several security requirements are commun to both the three cases, other security requirements are expressed for a specific use. For example the security requirement “the connection to the payment website should be encrypted by a certificate delivered by a certificate authority” is dedicated for “online payment” funcltionality. We can clearly distinguish between organisationla and operational security requirements. An example of organizational security requirement is “demilitarized zone (DMZ) Front Office must be isolated form demilitarized zone (DMZ) Back office”. This requirement must be performed when we have two platforms : one exposed to internet and the other reserved to backoffice users and services.

Performance requirements are crucial when we cannot predict simultaneous user connections. The typical example is: e-shop case project.

In Table III, we describe some elements dealing with project progress. For instance, one of these features is “deliverables” which provide a guidance for each step in the project planning. In addition, deliverables serve as a gateway between all project stakeholders.

TABLE. II. EXTRACT FROM CASE BASE : REQUIREMENTS DESCRIPTION

Requirements				
Section	***e-Shop project***	***Mobiling project***	***Intranet project***	***Values domain***
11. Functional requirements	- Display category and related products - Customer registration - User registration for newsletter - e-marketing (promotion, best sales) - Reporting (orders, customers…)	- Automatic Transmission of the postman rounds on their mobile devices - delivery operations using mobile terminal and transmission of real-time information - Monitoring postman activity	- All stakeholder contributions must be validated by the functional approver -	
12. Non functional requirements				
12.1. Security requirements	- Financial transactions between the customer's bank and the merchant bank on a closed network and not open on the Internet - Merchant does not have access to banking information of its customers either in consultation neither in treatment - The connection to the payment website should be encrypted by a certificate delivered by a certificate authority - All transactions must be traced (log files, logging) - Licensing and code ownership	- Automated connectivity with online and offline modes - User access must be restricted - All data exchange must be encrypted - demilitarized zone (DMZ) zone Front Office must be isolated form DMZ Back office - The password rules must be in line with the organism password policy - The reset or change of password must be available for users - All security rules must follow the organism security policy - contractual requirements for quality and code security level	- Single sign on (SSO) - User authentication system includes Lightweight Directory Access Protocol (LDAP) directory - User access must be completely restricted - Data must be transmitted to the user over a secure connection - External access must be denied - Code Test must be performed to detect potential malware or Trojan horse code	
12.2. Privacy [13]	- The compliance process must be initiated	- The compliance process must be initiated		
12.3. performance requirements	- 1000 simultaneous user connections	- 500 mobile users, 100 BackOffice users - Fast response time		

TABLE. III. EXTRACT FROM CASE BASE : PROJECT PROGRESS

Project progress				
Section	***e-Shop project***	***Mobiling project***	***Intranet project***	***Values domain***
13. Planning (macro steps)	Project scope, requirements analysis, design, development, test, production	Same	Same	Project scope, requirements analysis, design, development, test, production
14. Deliverables	Installation procedure, backup procedure, user guide, acceptance document	User guide, operation procedures, acceptance document, functional specification document, technical specification document	Acceptance document, functional specification document, technical specification document	User guide, acceptance document,
15. Tests				
15.1. Functional /Non-functional tests	See Table I			
16. Pre-production				
16.1. incidents/bugs	some product images are not displayed, adding individual product is not operational	Mobile configuration are not adapted, some data are not updated on the mobile	intranet document repository link is not available,	
17. Post-production				
17.1. incidents/bugs	Low performance of web service calculating the charge of distribution; hard disk space exceed the limit	unavailability or data transmission, unavailability of the network	Unauthorized user can access to restricted data,	

IV. Discussion

After studying, documenting the cases and clarifying the need of a project case base, the proposed framework must be verified. The validation will be done by answering the research questions defined in Section II.

- **RQ1:** What are security requirements/practices captured from each case?

The interviews which we conducted give information on security requirements and practices related to the selected cases. Security requirements are also described in the "functional specification document", but no document was found for security practices or how they perform the security requirements (see Fig. 3, "perform" association). We present in Table IV, the main security practices through interviews and related SDLC phases.

- **RQ2:** Is the framework accurate i.e. properly describing these security practices and the main case characteristics?

On the positive side, the framework has been defined to guive some guidance on how to document the cases. Thus, fundamental knowledge has been gathered during the case studies and the execution of framework experiment (see Tables I, II and III). This knowledge includes generic features (scope, requirements and progress) and specific features (software engineering method, functional requirements, security requirements, project actors, functional actors, etc.).

On the negative side, it is interesting to observe that, the framework lacks several elements that could play a leading role in documenting and selecting cases. Thus, the framework shows how the security requirements can be integrated through the SDLC phases (see "perform" association). However, the framework does not explicit who is performing these requirements (involved roles).

Another two important features of a case base of security projects are security bugs and incidents. So, to ensure a high security level of the software product, software team has to learn about security bugs and incidents of previous cases, in particular critical ones. The framework does not seem to cover this. In Table V, we illustrate some examples of security bugs and incidents that are collected through the case studies.

The class "category" in the framework describes the case category. This feature is not fully clear since many interpretations were provided by the interviewers. Fig. 3 shows example values for this feature. Thus, we suggest a three-level categorization for this feature.

There is a correspondence between security and technologies used in almost all software project and. For example, when we use web services, several security activities must be performed to deal with this technology. Therefor we can capitalize them for future similar situations.

In general, additional classes must deal with all security aspects involved in a project case including actors, technologies, project categories, bugs and incidents, etc. these features will be used as case descriptors (characteristics) for eventual new cases in order to perform similarity rules and to select the suitable similar cases.

Depending on the kind of the organization and its main strategies, there is a set of security aspects related to software security. For example, standards, policies and guides, requirements, principles, practices and activities.

The framework has very limited support in this way. The only aspect that the framework covers is that the security practices should perform the secure requirements. In the authors' opinion, the framework must deal with all software security issues.

TABLE. IV. The main Security Practices Gathered

Project case	*security practices*	*SDLC phases*
Mobiling	Use secure coding guidelines during implementation	Implementation
	Reduce priviliges	Design
	The reset or change of password must be available for users	Design
E-shop	Specify operational environment	Design
	Reduce priviliges	Design
	Perform manuel code inspection	Implementation
	Perform penetration testing	Test
	Perform database log files	Implementation, Maintenance
Intranet	Reduce priviliges	Design
	Perform database log files	Implementation, Maintenance
	User authentication system includes LDAP directory	Design, implementation

TABLE. V. Examples of Security Bugs and Incidents Gatherded through the Case Studies

Project case	**Security bug/incident (pre and post-production)**
e-shop	• Low performance of web service calculating the charge of distribution • Hard disk space exceed the limit
Mobiling	• Unavailability or data transmission • Unavailability of the network
Intranet	• Unauthorized user can access to restricted data

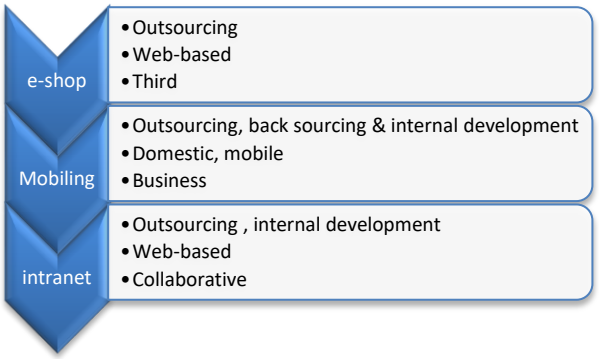

Fig. 3. Values Gathered for the "Category" Feature.

- **RQ3:** Are the selected cases pertinent?

The case pertinence can be analysed from different points of view. On the one hand, the selected cases are pertinent as they are leader in their respective fields. For instance, if you start any new project using mobile technology, you can find guidance about deadlines, project actors, requirements, security practices, technologies, etc. To assess the optimal reuse of both the "Mobiling" and "e-shop" cases, we introduce in Section V, a usage scenario that provide baselines about security requirements to be performed.

TABLE. VI. QUALITY EVALUATION OF CASE STUDIES

Quality criteria	*e-shop*	*Mobiling*	*Intranet*
A theoretical basis including research questions is described	3	3	3
Triangulation is ensured by using multiple sources of evidence (data collection and interpretation)	1	2	1
A chain of evidence is designed with traceable reasons and arguments	2	2	2
The case study research is fully documented	3	3	3
The case study report is compiled through an iterative review and rewriting process	0	0	0

On the other hand, software engineering team is not oriented to achieve security activities. We observed this in the context of the case studies. Unfortunately, the selected cases are not appropriate to provide guidance on software security. Thus, one of the problems that we have encountered is the weakness of a strucuctured approach for embedding security in software engineering field.

Author in [1] summarizes Yin's quality criteria for a good case study. In Table VI, we evaluate the case studies regarding these criteria. We consider 4 levels to evaluate the case studies as follow:

- Level 0: Lack; Level 1: low; Level 2: medium; Level 3: strong.

V. EXPERIMENT : REUSE OF CASE KNOWLEDGE

We present here briefly some usage scenarios of the cases knowledge.

A. *Reuse scenario 1 : Identify Security Requirements for Mobiling e-Shop Project*

In this scenario, we consider a new e-shop project on mobile. A typical question that can be asked by the project manager is: what security issues should we focus on? So, for a project team, it is of crucial importance to have access to security requirements and best practices identified in both e-shop and Mobiling experiences.

In Fig. 4, we illustrate this scenario and we represent some useful security requirements for the Mobiling e-shop project.

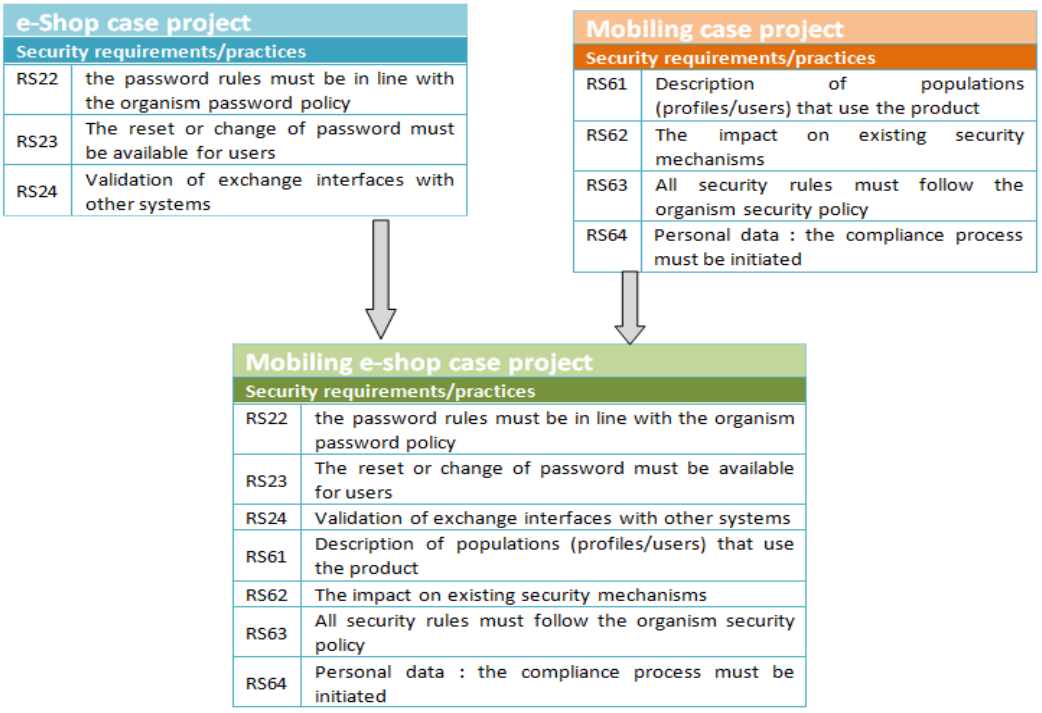

Fig. 4. Capturing Security Requirements for Mobiling e-Shop Project.

B. Reuse scenario 2 : Security Requirements for Outsourcing Projects

According to Table I, the “e-shop” case is only outsourcing, while the "Mobiling" and "Intranet" cases are both outsourcing and internal projects. Thus, by analyzing the Tables I and II, we can conclude that several security requirements and derived security best practices are extremely important when we decide to develop an outsourcing project, for example:

- An outsourcing project is managed by several contract documentations such as “contractual requirements for quality, code security level”, “licensing”, “code ownership” and “intellectual property rights”.
- All security rules must follow the organism security policy. In particular, The password rules must be in line with the organism password policy.
- Code test to detect potential malware or Trojan horse code.

VI. Related Works

Our research builds on multiple streams of related research, including software security, reuse, case based reasoning, and case study research.

In the context of software security, a good number of approaches and tools have been proposed to support security into the software development lifecycle. These approaches tend to be centered on building security activities and best practices gathered from standards, security processes and methodologies [5][6][13][14]. In this way, each security best practice is mapped with the corresponding phase of development. However, although security in the SDLC can be fine-built and managed by experienced teams, we have found that the integration of security best practices still difficult, even for skilled software engineering teams [6][15][16]. Reusing previews experienced best practices can reduce the complexity of security adoption in the SDLC. In that respect we propose to reuse security requirements and practices suggested by similar previous project cases.

One of the aspects that we focused on is the adoption of the CBR paradigm. In this way, there are currently several approaches focusing on the CBR approach to ensure similarities and to improve quality of search in the software engineering field [7][8][17].

As mentioned in section II, the case study methods provide guidelines that are helpfull to researchers when deciding how to select and evaluate case studies. Author in [10] outlined a case studiy method in software engineering and provided usefull examples.

VII. Conclusion

Security aspects are critical to secure software products and to meet end-user needs. The global aim is implementing a detailed case base for security focused software project that helps both software engineering and security teams to maintain a knowledge repository.

In this work, we have tested the effeteness of the proposed framework that is an excerpt of the case base through three cases e-shop, mobiling and intranet. These cases were selected using a structured case study method. We recall that we conducted this empirical study within a Moroccan company.

As a validation of the framework, we discuss a number of fundamental elements including cases pertinence, framework lacks and completeness. We also suggest improvements, which the framework could benefit from. At the end, we provide two usage scenarios to illustrate of the effectiveness of the framework.

This study indicates that several security requirements and practices can be adapted for specific projects. However, It is clear that both software engineering and security teams have to work synchronized with an incremental approach in order to provide a comprehensive project cas base.

This work can serve as a good reference for many organisations to implement their own case base for security focused projects. In this way, the authors plan to experiment this work based on validation by panels. Furthermore, the development of a software tool to support the managenemnt of the case base will be initiated.

References

[1] Mohammed, N. M., Niazi, M., Alshayeb, M., & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. Computer Standards & Interfaces, 50, 107-115.

[2] Baca, D., Boldt, M., Carlsson B., Jacobsson, A.: A novel security-enhanced agile softwaredevelopment process applied in an industrial setting. In: ARES 2015, pp. 11–19 (2015).

[3] Moradian, E., 2012. Integrating Security in Software Engineering Process: The CSEP Methodology. Ph.D. dissertation, KTH Royal Institute of Technology, Sweden.

[4] D. Noopur, Secure Software Development Life Cycle Processes, Published: July 05, 2006 | Last revised: July 31, 2013, Official website of the Department of Homeland Security, Available on https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes.

[5] Mohammed, N. M., Niazi, M., Alshayeb, M., & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. Computer Standards & Interfaces, 50, 107-115.

[6] Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead (2008). Software Security Engineering: A Key Discipline for Project Managers. part of the IEEE Reliability Society 2008 Annual Technology Report.

[7] Bitar, Ibrahim EL. CBR4WSD: Une approche de découverte de services Web par Raisonnement à Partir de Cas. Diss. Ecole Mohammadia d'Ingénieurs-Université Mohammed V de Rabat-Maroc, 2014.

[8] Martin Andreas, Sandro Emmenegger, and Gwendolin Wilke. "Integrating an enterprise architecture ontology in a case-based reasoning approach for project knowledge." Enterprise Systems Conference (ES), 2013. IEEE, 2013.

[9] de Mantaras, Ramon Lopez. "Case-based reasoning." Machine Learning and Its Applications. Springer Berlin Heidelberg, 2001. 127-145.

[10] RUNESON, Per, HOST, Martin, RAINER, Austen, et al. Case study research in software engineering: Guidelines and examples. John Wiley & Sons, 2012.

[11] A. Aamodt and E. Plaza, “Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches,” Artificial Intelligence Communications, vol. 7, no. 1, pp. 39–59, 1994.

[12] Alexandre Fernandez Toro, Management de la sécurité de l'information, implémentation ISO 27001 : Editions Eyrolles, 2018.

[13] Michael Howard and Steve Lipner. The Security Development Lifecycle, Microsoft Press, 2006.

[14] Uzunov, Anton V., Katrina Falkner, and Eduardo B. Fernandez. "A comprehensive pattern-oriented approach to engineering security methodologies." Information and Software Technology 57 (2015): 217-247.

[15] Busch, Marianne, Nora Koch, and Martin Wirsing. "Evaluation of engineering approaches in the secure software development life cycle." Engineering Secure Future Internet Services and Systems. Springer, Cham, 2014. 234-265.

[16] Mazni Mohamed Jakeri, Mohd Fadzil Hassan, "A Review of Factors Influencing the Implementation of Secure Framework for in-House Web Application Development in Malaysian Public Sector", Application Information and Network Security (AINS) 2018 IEEE Conference on, pp. 99-104, 2018.

[17] Benjamin Maraza-Quispe, Olga Melina Alejandro-Oviedo, Walter Choquehuanca-Quispe, Alejandra Hurtado-Mazeyra and Walter Fernandez-Gambarini, “e-Learning Proposal Supported by Reasoning based on Instances of Learning Objects” International Journal of Advanced Computer Science and Applications(IJACSA), 10(10), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101035.