Empir Software Eng (2017) 22:2127–2178
DOI 10.1007/s10664-016-9481-1

Identifying the implied: Findings from three differentiated replications on the use of security requirements templates

Maria Riaz[1] · **Jason King**[1] · **John Slankas**[1] · **Laurie Williams**[1] · **Fabio Massacci**[2] · **Christian Quesada-López**[3] · **Marcelo Jenkins**[3]

Published online: 16 December 2016

Abstract Identifying security requirements early on can lay the foundation for secure software development. Security requirements are often implied by existing functional requirements but are mostly left unspecified. The Security Discoverer (SD) process automatically identifies security implications of individual requirements sentences and suggests applicable security requirements templates. The objective of this research is to support requirements analysts in identifying security requirements

Communicated by: Andreas Zeller

✉ Maria Riaz
mriaz@ncsu.edu

Jason King
jtking@ncsu.edu

John Slankas
john.slankas@ncsu.edu

Laurie Williams
lawilli3@ncsu.edu

Fabio Massacci
fabio.massacci@unitn.it

Christian Quesada-López
cristian.quesadalopez@ucr.ac.cr

Marcelo Jenkins
marcelo.jenkins@ucr.ac.cr

[1] Department of Computer Science, North Carolina State University, 890 Oval Drive, Box 8206, Engineering Building II, Raleigh, NC 27695-8206, USA

[2] University of Trento, Trento, Italy

[3] University of Costa Rica, San José, Costa Rica

by automating the suggestion of security requirements templates that are implied by existing functional requirements. We conducted a controlled experiment in a graduate-level security class at North Carolina State University (NCSU) to evaluate the SD process in eliciting implied security requirements in 2014. We have subsequently conducted three differentiated replications to evaluate the generalizability and applicability of the initial findings. The replications were conducted across three countries at the University of Trento, NCSU, and the University of Costa Rica. We evaluated the responses of the 205 total participants in terms of quality, coverage, relevance and efficiency. We also develop shared insights regarding the impact of context factors such as time, motivation and support, on the study outcomes and provide lessons learned in conducting the replications. Treatment group, using the SD process, performed significantly better than the control group (at p-value <0.05) in terms of the coverage of the identified security requirements and efficiency of the requirements elicitation process in two of the three replications, supporting the findings of the original study. Participants in the treatment group identified 84 % more security requirements in the oracle as compared to the control group on average. Overall, 80 % of the 111 participants in the treatment group were favorable towards the use of templates in identifying security requirements. Our qualitative findings indicate that participants may be able to differentiate between relevant and extraneous templates suggestions and be more inclined to fill in the templates with additional support. Security requirements templates capture the security knowledge of multiple experts and can support the security requirements elicitation process when automatically suggested, making the implied security requirements more evident. However, individual participants may still miss out on identifying a number of security requirements due to empirical constraints as well as potential limitations on knowledge and security expertise.

Keywords Security requirements · Controlled experiment · Replication · Requirements engineering · Templates · Patterns · Automation

1 Introduction

Security requirements, like requirements in general, come from various sources. Project artifacts, such as requirements documents and privacy policies, contain important information related to security, privacy protection, and compliance. Implied security requirements from these artifacts may be overlooked during the requirements elicitation process given the lack of security expertise and lack of focus on security during early stages of system development. Consider the sentence “The system shall provide a means to view discharge instructions for a particular patient.”[1] This sentence does not explicitly state a security requirement but implies security requirements for confidentiality (*of patient's discharge instructions*) and accountability (*who viewed the discharge instructions*?). Identifying these implied security requirements is important to lay the foundation for secure software development and to strengthen the overall security of the system.

Security community has established a number of knowledge sources, including security catalogues and controls, that capture security expertise and can support elicitation of security requirements. However, a security non-expert may not readily know how and when to leverage the security knowledge sources in the context of a given system. Security requirements

[1] http://www.hl7.org/

templates are a particular example of a broader class of security knowledge that can be sourced from the community by abstracting out common security requirements. Moreover, providing guidance regarding the applicability of various security requirements templates based on a system's functionality can support the analysis of security requirements. Empirical evaluation on the use of security requirements templates can inform how well the templates support the requirements elicitation process as well as the context in which the findings are applicable.

The objective of this research is to support requirements analysts in identifying security requirements by automating the suggestion of security requirements templates that are implied by existing functional requirements.

The Security Discoverer process, developed by Riaz et al. (2014a supports automatic discovery of security requirements implied by existing functional requirements. The process suggests applicable templates by automatically parsing individual requirements sentences in the input artifacts and identifying the security implications of the existing functional requirements of a system. The suggested templates can be manually instantiated in the context of the given system to generate specific security requirements. Riaz et al. (2014b) conducted a controlled experiment, involving 50 graduate students enrolled in a software security course, to evaluate the use of Security Discoverer process in eliciting implied security requirements. Participants were divided into treatment (automatically-suggested security requirements templates) and control groups (no templates provided). Based on the findings, automatically-suggested templates helped participants (security non-experts) gain awareness about security implications for the software system and consider more security requirements than they would have otherwise.

We conducted three differentiated replications of the original experiment to examine whether the findings can be replicated in different settings, incorporating lessons learned from the original experiment. Replication studies are beneficial to evaluate the validity of prior study findings, either by reproducing results or by isolating factors that can influence results and that lead to variations. Based on the objective, we analyzed the following research questions in the original experiment and all subsequent replications:

> *RQ1*: What is the quality of security requirements elicited through the use of automatically-suggested security requirements templates?
> *RQ2*: What is the coverage of security requirements elicited through the use of automatically-suggested security requirements templates?
> *RQ3*: How relevant are the security requirements elicited through the use of automatically-suggested security requirements templates?
> *RQ4*: How efficient is the process of eliciting security requirements through the use of automatically-suggested security requirements templates?

For clarity, we use the following codes when referring to various experiments:

- NCSU13: Original study conducted at North Carolina State University (NCSU) in 2013 and reported in 2014 (Riaz et al. 2014b).
- UT14: First replication conducted at University of Trento (UT) in 2014.
- NCSU14: Second replication conducted at NCSU in 2014.
- UCR15: Third replication conducted at University of Costa Rica (UCR) in 2015.

The replications have several differences in terms of context factors such as participants and experimental setting. Some of these differences are inherent to all replications involving a different sample than the original experiment. Other differences were introduced to incorporate lessons learned from the original experiment related to quality, coverage and relevance of the responses. Based on these differences, we qualitatively examine the following additional research questions:

> *RQ5*: Are participants more inclined to fill in the templates when additional support to fill the templates is provided by explicitly indicating subject, action and resource elements in the input requirements?
> *RQ6*: Can participants differentiate whether a suggested security requirements template is relevant to the given use case scenario?
> *RQ7*: Are there context factors, such as more time on task, which are conducive to producing better outcomes overall?

The contributions of our research include:

- An extensive empirical evaluation of the systematic Security Discoverer process for identifying security requirements implied by natural language requirements artifacts.
- Practical implications related to the use of security requirements templates in eliciting implied security requirements based on the findings across multiple studies.
- Insights related to conducting multiple replications and supporting synthesis across studies.

The rest of the paper is organized as follows: We present the background and related work in Section 2. We provide an overview of the Security Discoverer process in Section 3. In Section 4, we outline the research methodology. In Section 5, we present the results based on the analysis of the replication studies and address RQ1-RQ4. We provide a synthesis of our findings in Section 6 and address RQ5-RQ7. We provide a qualitative summary of the feedback from participants regarding security requirements and templates in Section 7. We discuss the lessons learned conducting the replications in Section 8. We discuss threats to validity in Section 9 and conclude the paper in Section 10.

2 Background and Related Work

We discuss the relevant background concepts and related work in this section.

2.1 Replications in Software Engineering

Replication studies are beneficial to evaluate the validity of prior study findings, either by reproducing results or by isolating factors that can influence results and that lead to variations. According to Lindsay et al. (Lindsay and Ehrenberg 1993), a *close* replication study attempts to recreate the known conditions of original study and is very similar to the original study. Close replications are often used to establish whether the original outcomes are repeatable at all i.e., initial outcomes were not unduly influenced by confounding factors. A *differentiated* replication study has a known or deliberate variation in terms of a major effect of the original study conditions. Differentiated replications allow researchers to explore the impact of variations in treatments on the study outcomes (Lindsay and Ehrenberg 1993). The replications that we have performed are differentiated replications. We outline the differences between our original experiment and the replications in Section 4.

Empirical replications, although an essential part of the experimental paradigm to produce generalizable knowledge, have not been frequently carried out in the field of software engineering (Carver et al. 2014). By conducting replications, and reporting aggregate results, we can have a more in-depth understanding of the software engineering theory or methodology being studied. Moreover, a combined analysis of studies can lead to important insights that may not be otherwise obvious. Often times, even subtle differences in study design and execution can limit comparison of findings. Providing adequate details related to the original experiment design and availability of experimental artifacts can support the replication effort. Moreover, using similar metrics for participants' experience and employing commonly used measures to evaluate performance can support comparison of results across studies (Riaz et al. 2015).

2.2 Security Objectives and Requirements

Security objectives are the security goals or desired security properties of a system (Schumacher et al. 2006). Security requirements are functional and non-functional requirements that operationalize security objectives without specifying how to achieve those objectives. Functional security requirements describe the desired security behavior of a system (2012) and, if incorporated, can achieve the corresponding security objectives. Mellado et al. (2007) argue the effectiveness of reusing related security requirements that act as a group to meet security objectives. In our previous work (Riaz et al. 2014a), we have identified six core categories of security objectives. In addition to the widely known confidentiality, integrity, and availability (CIA) triad, we consider objectives related to identification and authentication, accountability and privacy. Each security objective counters one or more threats in the Microsoft STRIDE[2] threat model as listed in the brackets following the definition of security objectives below:

- *Confidentiality (C)*: The degree to which the data is disclosed only as intended. [*Information Disclosure*]
- *Integrity (I)*: The degree to which a system or component guards against improper modification or destruction of computer programs or data. [*Tampering*; *Elevation of Privileges*]
- *Availability (A)*: The degree to which a system or component is operational and accessible when required for use. [*Denial of Service*]
- *Identification & Authentication (ID)*: The need to establish that a claimed identity is valid for a user, process or device. [*Spoofing; Elevation of Privileges*]
- *Accountability (AY)*: The degree to which actions affecting software assets can be traced to the actor responsible for the action. [*Repudiation*]
- *Privacy (PR)*: The degree to which an actor can understand and control how their information is used. [*Information Disclosure*]

[2] https://msdn.microsoft.com/en-us/magazine/cc163519.aspx

2.3 Security Requirements Elicitation and Evaluation

Methods for eliciting and documenting security requirements include both conceptual frameworks and requirements models. Framework methods such as the SQUARE (Mead et al. 2005) provide a Capability Maturity Model-like reference model for coordinating various technical activities and artifacts related to security requirements engineering (SRE). Another framework-based approach represents security requirements as constraints which are used to develop satisfaction arguments for the security requirements (Haley et al. 2008). Approaches for modeling security requirements include misuse cases (Alexander 2003) (Sindre and Opdahl 2005), misuse activities (Braz et al. 2008), and abuse cases (McDermott and Fox 1999) that document an attacker's perspective and support identification of security requirements that mitigate the attack scenarios. Our approach for identifying security requirements differ from modeling-based approaches in that we use security requirements templates to identify security requirements. Various approaches for identifying security requirements may be used complementary to each other for a comprehensive analysis. Mellado et al. (2010) have conducted a systematic review of SRE approaches to summarize existing methodologies. Fabian et al. also provide a comparison of existing SRE methods (Fabian et al. 2010).

A comprehensive analysis of security requirements, starting from scratch, is time and resource consuming. A recent case study, documenting the use of SQUARE methodology, reported an effort of around 12 person-weeks for applying the methodology, with 3 person-days for identifying security goals (Suleiman and Svetinovic 2013). Consequently, empirical evaluation of security requirements engineering approaches within a controlled setting is challenging and not many approaches have been empirically evaluated. Taubenberger et al. have developed a security requirements-based risk assessment approach, learning from their experience of performing security risk assessment of two systems (Taubenberger et al. 2011). Taubenberger et al. have also evaluated their approach in resolving errors related to the identification of vulnerabilities in comparison to an existing approach. Their findings indicate that explicitly evaluating security requirements during the course of business can help in resolving vulnerability identification errors (Taubenberger et al. 2013). Karpati et al. have recently reported empirical evaluation of misuse case maps in identifying security threats through two controlled experiments (Karpati et al. 2015). Their findings indicate the usefulness of misuse case maps in comparison to an approach based on the combination of misuse cases and system architecture diagrams. Misuse case maps helped the students, involved in the experiment, to suggest better mitigations and were also viewed more favorable as compared to the alternate approach. The authors have emphasized the need for replicating the experiments to evaluate the generalizability of findings beyond their current empirical setup. These recent studies also indicate a trend towards increasing empirical evaluation of SRE approaches.

2.4 Security Requirements Templates and Patterns

Security requirements are potentially reusable across systems that share the same security objectives such as confidentiality or integrity. Firesmith (Firesmith 2004) proposed the use of parameterized templates to model reusable security requirements. Firesmith's parameterized templates do not include information about when a template is applicable and how to instantiate it as part of the template. Toval et al. (2002) present hierarchically structured parameterized and non-parameterized templates for reusable security requirements that adhere to IEEE standards for specifying quality requirements. Some of the quality attributes are:

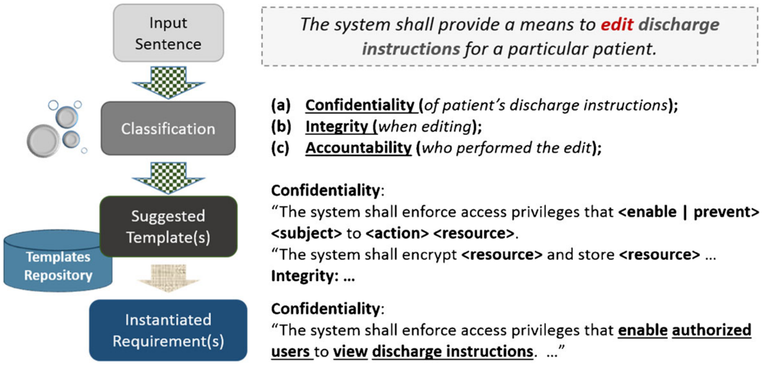
identification (unique), priority, criticality, viability, risk, source, traceability. Toval's requirements elicitation process model is based on spiral model for requirements engineering (SIREN) and explicitly incorporates requirements reuse. A repository for reusable requirements is maintained with annotations about the domain (e.g., accounting or finance) and profile (e.g., information systems security) to indicate when a requirement is applicable for reuse. Toval's reusable requirements are closely related to security requirement patterns. However, they lack concrete examples on how to instantiate the templates or the consequences of reusing requirements from these templates.

Security requirements patterns support an analysis of the security requirements for a system. Security requirements patterns available in the literature cover only a small subset of the security requirements landscape (Ito et al. 2015) Withall's (2007) requirement patterns related to security include access control (registration, authentication, and authorization), audit (chronicle), and some aspects of privacy (archiving, comply-with-standard). Schumacher et al. (2006) have developed a security patterns catalog including security requirements patterns on access control, auditing, intrusion detection, non-repudiation and accounting. Both Withall's and Schumacher et al.'s catalogs use natural language pattern representation. Wen et al. (2011) have proposed security requirements patterns of ownership, authorization, attack and protection based on problem frames and i* for modeling and analysis of assets, threats and attacks. Security requirements patterns available in literature focus mostly on attack patterns (Yoshioka et al. 2008). Security requirements templates are a particular example of a broader class of security knowledge that can be sourced (Gray and Meister 2004) from the community such as software security patterns or security catalogues. Other researchers have studied the usage of security patterns, particularly to design secure architecture from requirements (Yskout et al. 2015) or security catalogues in security risk assessment (De Gramatica et al. 2015). With the notable exception of Zhang and Budgen (2012), all experiments showed that patterns, catalogues or other forms of knowledge sourcing improves the performance of novices.

In our work (Riaz et al. 2014a), we have identified six key security objectives for software systems. We provide an initial set of security requirements templates to meet each of the identified objective, building on the observations from previous work. As a next step, to support ways to realize the specified requirements, we can map security requirements templates to related security patterns available in the literature that offer security solutions for the respective requirements.

3 Security Discoverer Process

Riaz et al. (2014a) have developed a tool-assisted process, Security Discoverer (SD), incorporating supervised machine learning techniques to identify a set of security requirements implied by an input set of natural language requirements artifacts (e.g., scenario description). The SD process automatically parses individual sentences in input artifacts and identifies implied security objectives, such as confidentiality or integrity. Based on the implied objectives, the process suggests applicable security requirements templates that can be selected and instantiated by a security requirements engineer into a set of functional security requirements. In Fig. 1, we provide an example of how the SD process works. For the example input sentence in the figure, SD process identifies the security objectives of confidentiality and accountability in the classification step. Security requirements templates, developed by the researchers, are then suggested by the



*The system shall provide a means to **edit** discharge instructions for a particular patient.*

(a) **Confidentiality** (*of patient's discharge instructions*);
(b) **Integrity** (*when editing*);
(c) **Accountability** (*who performed the edit*);

Confidentiality:
"The system shall enforce access privileges that **<enable | prevent> <subject>** to **<action> <resource>**.
"The system shall encrypt **<resource>** and store **<resource>** …
Integrity: …

Confidentiality:
"The system shall enforce access privileges that **enable authorized users** to **view discharge instructions**. …"

Fig. 1 Example of security discoverer process [*Checkered arrow* indicates automated part of the process]

process. Each template is associated with a particular security objective and identifies the conditions under which the template becomes applicable (e.g., based on the subject, resource or action in the input sentence). In the figure, we also show a snippet of the suggested template for 'authorized access' to support the objective of confidentiality. The selected templates are then instantiated by filling in subject, resource and action elements from the input sentence to generate security requirements.

The pool of 19 security requirements templates used by the tool for matching suggestions has been developed by NCSU researchers (Riaz et al. 2014a). The 19 templates capture commonly-used security requirements that support core security objectives and were empirically developed from the analysis of ~11,000 sentences drawn from six different natural language requirements artifacts from the healthcare domain (Riaz et al. 2014a). For developing the templates, the researchers identified commonly occurring security requirements in the requirements artifacts and abstracted the subject, resource and action elements out to form reusable parameterized security requirements. Related set of reusable security requirements were then grouped by security objectives to form the templates. The 19 security requirements templates are named below. Details of these security requirements templates are available at our project website.[3]

- *Confidentiality (C)*:

  C1 – authorized access;
  C2 – confidentiality during storage;
  C3 – confidentiality during transmission;

- *Integrity (I)*:

  I1 – read-type actions;
  I2 – write-type actions;
  I3 – delete actions;
  I4 – unchangeable resources;

[3] http://go.ncsu.edu/secreqtemplatesstudy

- *Availability (A)*:

  A1 – maintaining availability of data;
  A2 – appropriate response time;
  A3 – service availability;
  A4 – backup and recovery capabilities;
  A5 – capacity and performance;

- *Identification & Authentication (IA)*:

  IA1 – select context for roles;
  IA2 – unique accounts;
  IA3 – authentication;

- *Accountability (AY)*:

  AY1 – logging transactions with sensitive data;
  AY2 – logging authentication events;
  AY3 – logging system events;

- *Privacy (PR)*:

  PR1 – usage of personal information;

As an example, we provide two security requirement templates associated with the objectives of accountability and integrity, respectively, in Table 1. Each template groups a set of related security requirements and can be used to instantiate one or more security requirements for the system. For instance, template AY1 can be filled in with details about subject, resource and action elements from input sentence to instantiate two security requirements as shown in the last column of Table 1. The newly-composed security requirements also contain related security objectives themselves. Consider the instantiated security requirements for AY1 in Table 1. These requirements suggest an integrity objective to prevent modification of log files (I4). The template for AY1 captures this relationship between accountability and integrity by suggesting the requirements analyst to consider template I4 for integrity when identifying the security requirements for accountability.

The original experiment (Riaz et al. 2014b) and the subsequent replications reported in this paper evaluate the part of SD process related to the use of automatically-suggested templates.

4 Research Methodology

In the following subsections, we provide details about the methodology for conducting the experiments. We also document the similarities and differences between NCSU13 and the subsequent replications. In all the studies, participants were assigned the task of identifying security requirements based on a given use case scenario. Participants were randomly placed into treatment and control group. The treatment group carried

Table 1 Example context-specific security requirements templates

Input sentence: *The system should provide a means to view discharge instructions for a particular patient.*

Security objective	Automatically-suggested security requirements templates		Instantiated security requirements
Account-ability	AY1	Logging transactions with sensitive data *Given*: <subject>= user or role <resource>= sensitive information <action>= create/read/update/delete *Add Security Requirements*: • The system shall log every time<subject>[performs the]<action><on|for><resource>. [*see templates C1*, *I4*] • At a minimum, the system shall capture the following information for the log entry: <subject>identification, timestamp, <action>, <resource>, and identification of the owner of<resource>.	• The system shall log every time user views discharge instructions for a particular patient. • At a minimum, the system shall capture the following information for the log entry: user identification, timestamp, view discharge instructions, patient identification. • The system shall not allow modification of the log by any user.
Integrity	I4	Maintaining integrity of unchangeable resources *Given*: <resource>= write-once information (e.g., log files) *Add Security Requirements*: • The system shall not allow modification of <resource > by any user.	

out the task with the support of automatically-suggested security requirements templates whereas the control group did not receive such support.

The motivation for conducting the three differentiated replications of the original experiment is to examine whether the findings can be replicated in different settings (in-class vs. take-home), different level of support (in filling templates), difference in motivation and different problem domain (healthcare vs. mobile banking), incorporating lessons learned from the original experiment (see Table 4). An initial set of guidelines for reporting experimental replications have been proposed by Carver (2010). We have included all the recommended details about the original study as well as the replications in this paper. We also detail the results of individual studies as well as analysis across multiple studies in Sections 5 and 6 respectively.

4.1 Goals, Hypotheses and Metrics

In the original experiment, and all subsequent replications, we want to determine whether the use of automatically-suggested security requirements templates leads to efficient and effective elicitation of security requirements when compared to a manual approach based on personal expertise. We test the following null hypotheses to address our research questions RQ1-RQ4:

> H_{01}: The mean quality of elicited security requirements is unrelated to the use of automatically-suggested security requirements templates. *[RQ1]*
>
> H_{02}: The mean coverage of elicited security requirements is unrelated to the use of automatically-suggested security requirements templates. *[RQ2]*

H_{03}: The mean relevance of elicited security requirements is unrelated to the use of automatically-suggested security requirements templates. *[RQ3]*
H_{04}: The mean efficiency of the requirements elicitation process is unrelated to the use of automatically-suggested security requirements templates. *[RQ4]*

We compute the metrics listed in Table 2 for each participant's response and use the results to test the preceding hypotheses. By using the same criteria and comparable metrics, we support comparison of findings across the original experiment and subsequent replications (Kitchenham and Charters 2007).

For computing the metric for quality of identified security requirements, we used a Likert-like scale of 1 to 5 (1: poor; 2: below average; 3: average; 4: above average; 5: good). Two researchers independently assigned quality scores for each participant's response, using following questions as guide:

- Are the requirements too general or too specific?
- Have all necessary elements of the requirements been identified (e.g., subject, resource, action, data to be logged)?
- Are there any logical inconsistencies in the requirements?
- Are different types of security objectives considered?
- For the treatment group, have the selected templates been filled-in with appropriate details?

For coverage and relevance, we respectively used the metrics for recall and precision computed based on an oracle of security requirements developed a priori to the conduct of the study (Section 4.6). Metrics for quality, coverage and relevance evaluate the security requirements identified by the participants. The metric for efficiency evaluates the requirements elicitation process.

4.2 Participants

In this section, we report the demographic summary of the participants for the original experiment as well as all its replications.

Table 2 Metrics used for evaluating participants' responses

Evaluation criteria	Metric type	Metrics used
Quality, *of security requirements*	Qualitative	Likert-like scale (1–5): lower score indicates lower quality.
Coverage, *of security requirements*	Quantitative	• Recall w.r.t security requirements in the oracle. • Recall w.r.t security requirements templates in the oracle. (*for UCR15*)
Relevance, *of security requirements*	Quantitative	• Precision w.r.t security requirements in the oracle. • Precision w.r.t security requirements templates in the oracle. (*for UCR15*)
Efficiency, *of process for eliciting security requirements*	Quantitative	• # of security requirements in the oracle identified per minute. • # of security requirements templates in the oracle identified per minute. (*for UCR15*)

w.r.t with respect to

For the original study, NCSU13, participants were graduate students enrolled in a 16-weeks software security course[4] offered at NCSU in Fall 2013. Researchers conducted the study as an online web-based activity during the last week of the course, after students had learned various software security concepts. The task for this study was mandatory for all the students to complete, similar to other class exercises. Based upon the IRB approval obtained for the study, students could opt-out of participating in the study, which would preclude the inclusion of their work in the study results. Of the 54 students enrolled in the course, 50 gave consent to use their responses for the study. Each student received coursework credit for completing the task as a classroom exercise, irrespective of their decision to participate in the study or of the quality of their responses.

The first replication, UT14, was conducted at the University of Trento (UT), Italy within a course in Security Engineering at the Master level offered in Fall 2014. The total enrollment in the course was 35 students, out of which 32 gave consent to participate in the study. Of the 35 students, 13 students were enrolled in a special curriculum on security and privacy while the other students were enrolled in the general Master of Science in Computer Science. The exercise was given after 8 h of lectures on introduction to security concepts covered over 4 lectures. The concepts include the Confidentiality, Integrity, and Availability (CIA) triad, security controls, security management methodologies (e.g. COBIT[5]) and security risk management (e.g. NIST 800–30,[6] ERM COSO[7]). The exercise was presented in class and the material was given as a take home exercise from Wednesday to the following Tuesday.

The second replication, NCSU14, was conducted by the researchers of the original study at NCSU. The participants were students enrolled in the same course as the original experiment, taught in Fall 2014. Of the 110 enrolled students, 107 agreed to participate in the study. Each student received coursework credit for completing the task as a classroom exercise irrespective of his or her decision to participate in the study. However, in contrast to the original study where all students received a standard participatory grade, the coursework credit given for NCSU14 was based on the quality of each student's response as evaluated by the teaching assistants for the course. The change in incentive was based on findings of the original study to explore whether using differentiated credit based on quality might lead to improved overall quality of the responses (Section 6.2.1).

The third replication, UCR15, was conducted at University of Costa Rica (UCR) in Summer 2015. The participants were first year Master's degree students enrolled in a course on Software Metrics. All the 16 enrolled students participated in the study. The exercise performed as part of the study was graded and represented a significant percentage (25 %) of the final grade. Participants, therefore, had a strong incentive to produce good quality results.

At the end of the task, we asked participants to report their experience in three academic categories (CS: Computer Science, SE: Software Engineering, and Security related education) and three work-related categories (CS, SE, and Security related work experience). Table 3 summarizes the participant's background across the replication studies.

Within each category, participants could select one of the four experience categories as listed in the table (>5 years; 3–5 years; 1–2 years; <1 year). In Table 3, we have highlighted the maximum frequency for each study for treatment and control group. Within each study,

[4] https://sites.google.com/a/ncsu.edu/csc515-software-security/

[5] http://www.isaca.org/cobit/pages/default.aspx

[6] http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf

[7] http://www.coso.org/ERM-IntegratedFramework.htm

Table 3 Frequency of participants' academic and work experience across studies

Study	Group	Number of participants	Experience (yrs)	Academic (A)			Work (W)		
				CS	SE	Sec	CS	SE	Sec
NCSU13	Treatment	30	>5 years	16	4	2	0	0	0
			3–5 years	9	11	2	11	6	1
			1–2 years	1	7	7	4	5	2
			<1 year	0	4	15	11	15	23
			Not Responded	4	4	4	4	4	4
	Control	20	>5 years	11	3	0	0	0	0
			3–5 years	8	10	2	9	7	0
			1–2 years	0	4	5	3	6	5
			<1 year	1	3	13	8	7	15
			Not Responded	0	0	0	0	0	0
UT14	Treatment	17	>5 years	0	0	0	0	0	0
			3–5 years	10	0	0	3	1	0
			1–2 years	3	5	4	2	1	1
			<1 year	4	12	13	12	15	16
			Not Responded	0	0	0	0	0	0
	Control	15	>5 years	1	0	0	0	0	0
			3–5 years	10	2	0	0	0	0
			1–2 years	1	5	6	5	1	1
			<1 year	3	8	9	10	14	14
			Not Responded	0	0	0	0	0	0
NCSU14	Treatment	55	>5 years	15	1	0	0	0	0
			3–5 years	36	18	0	11	5	0
			1–2 years	2	27	9	31	31	3
			<1 year	2	9	46	13	19	52
			Not Responded	0	0	0	0	0	0
	Control	52	>5 years	23	2	0	1	1	0
			3–5 years	20	18	1	15	8	0
			1–2 years	9	24	10	24	21	9
			<1 year	0	8	41	12	22	43
			Not Responded	0	0	0	0	0	0
UCR15	Treatment	9	>5 years	5	4	2	2	3	0
			3–5 years	3	2	0	2	1	0
			1–2 years	0	2	1	3	2	2
			<1 year	1	1	6	2	3	7
			Not Responded	0	0	0	0	0	0
	Control	7	>5 years	2	1	0	1	1	0
			3–5 years	5	4	0	5	4	0
			1–2 years	0	2	2	0	1	1
			<1 year	0	0	5	1	1	6
			Not Responded	0	0	0	0	0	0

CS Computer Science, *SE* Software Engineering, *Sec* Security participants in the treatment and control groups have comparable experience based on the frequency of participants across various experience categories, minimizing potential biases. Since the groups are heterogeneous (to provide a greater power of generalization), we do not make comparison across different replications in terms of participants' experience. Experience is self-reported by participants and the semantics and interpretation of the word "experience" might vary for each participant.

4.3 Study Environment

In this section, we present the details related to the study environment that were shared among all the experiments. We also provide details in terms of setting that were different across the experiments.

4.3.1 Shared Aspects of Study Context

The original experiment and all subsequent replications were conducted as an online activity. All students received a URL to access the online site for the study. On accessing the site, the students first viewed the consent form. Students then read the consent form and could either allow or deny use of their data in the study results. Next, the system assigned each student an auto-generated random access code. The student was randomly assigned to treatment or control group and shown a screen with instructions for completing the task based on the group the student was assigned to. We provide more details about the grouping in Section 4.5 when we discuss the experiment design. Having read the instructions, the student could continue to the task of identifying security requirements.

Students could save the task at any point during the experiment and return to the task by entering their access code at the provided URL. For each participant, we recorded total time spent completing the task and whether the participant submitted the task.

After completing the task, we asked participants to:

- Briefly explain the process used for identifying applicable security requirements (e.g., what information you looked at in the use case or reference material).

We solicited feedback on the security requirements templates from the participants in the treatment group (explained in Section 4.5) by asking the following additional open-ended questions:

- What is your opinion regarding the use of requirements templates?
- What is your opinion regarding the use of generated requirements?

4.3.2 Differences in Study Context

The original experiment and replications differed in terms of certain aspects related to the setting of the study. In Table 4, we summarize the context factors for each experiment. We also indicate the metrics that we expect to be affected by the differences in the context factors.

NCSU13 was conducted as an in-class activity at NCSU. Students were encouraged to work on the task during the 60-min lecture period in a classroom setting. Participants received

Table 4 Summary of context factors across different experiments

Study → / Context factors ↓	Original study [NCSU13]	Replication-1 [UT14]	Replication-2 [NCSU14]	Replication-3 [UCR15]
Participants	50 graduate students	32 graduate students	107 graduate students	16 graduate students
Setting	In-class Activity, NCSU [60 min]	Take-home Activity, UT [1 week] [**Coverage & Efficiency*]	In-class Activity, NCSU [60 min]	In-class Activity, UCR [180 min]
Security training provided	• Software Security course • 4 page reference material	• Security Engineering course • 4 page reference material	• Software Security course • 4 page reference material • 10 min video on security objectives and requirements	• 4 page reference material • 10 min video on security objectives and requirements • 15 min explanatory presentation in Spanish
Problem domain	Healthcare	Healthcare	Healthcare	Mobile banking
Other changes	NA	No other changes	• Suggestion to fill in templates • Feedback on quality of responses [**Quality*] • Extraneous Templates [**Relevance*]	• Suggestion to fill in templates • Feedback on quality of responses [**Quality*] • Extraneous Templates [**Relevance*]

*metrics expected to be affected by differences in study context

related reference material 2 days prior to the conduct of the study. Researchers provided a 5-min overview of the task at the beginning of the lecture period. In addition to the remaining 55 min, students had a total of 2 days to complete the task and were required to submit the task before the start of the next lecture period. Of the 50 participants, 40 completed the task during the lecture period.

The UT14 study was initially planned as an in-class activity. However, almost 30 % of the students did not have a laptop so a last minute change was made to assign the task as a take-home activity. In NCSU13, we found a positive correlation between time on task and number of requirements identified (Riaz et al. 2014b). Conducting UT14 as a take-home activity provides opportunity to assess if more time on task leads to improvement in the coverage of identified security requirements. Participants received related reference material 2 days prior to the conduct of the study, as was done in NCSU13. The measurement of time for UT14 might not be as reliable as NCSU13 or NCSU14. The tool cannot differentiate whether the task window is merely opened or if the participant is actually working on the task. Though the University of Trento is in Italy, the participants in UT14 attended a Master degree where the language of instruction was English and the audience was mostly international. The participants provided responses in English.

In NCSU14, the students performed the task as an in-class activity during the 60-min lecture period, like the original experiment. Participants received related reference material 2 days prior to the conduct of the study. However, instead of the introductory overview at the start of the lecture period, participants watched a 10-min video introducing the general concept of security objectives that are implied by natural language requirements artifacts before the lecture. The video did not include details related to the security requirements templates, and only treatment group had the knowledge of the templates during the conduct of the study, as with all other studies.

In UCR15, the students performed the task as an in-class activity during a 180-min lecture period. Participants received related reference material, including the 10-min video presentation given to participants in NCSU14, 1 week prior to the conduct of the study. Researchers expected the participants to have at least read the provided material (see Section 4.4). Participants also received a review of the reference material in the form of a 15 min presentation in Spanish before the start of the study. Participants in UCR15 had access to the research paper in which the findings of the original experiment were reported (Riaz et al. 2014a, b) however we do not know if they read the paper before the experiment. Participants in UCR15 were mostly native Spanish speakers. Almost half of the participants in the control group provided responses in Spanish. Researchers at UCR translated the responses to English. The templates suggested to the treatment group in NCSU14 and UCR15 included some extraneous suggestions as well, however participants were not aware that some of the suggestions may be extraneous.

4.4 Experiment Artifacts

All participants received four pages of reference material containing a description of software security objectives as well as textual clues that can indicate an implied security objective. Reference material also contained a total of 40 example security requirements grouped by security objectives. We provided this standard reference material to participants prior to the start of the experiment. During the experiment, the control group had access to the same reference material online. However, for the treatment group, we presented example security requirements in two forms: i) reusable security requirements templates grouped by security objectives; and ii) concrete example security requirements (also available to control group) that were generated from the templates. The reference material, use cases and other study documents are available on our project website.[8]

As part of the task, participants identified security requirements based on a given use case scenario. We used the following criteria to select the use cases:

- The use case must focus on a single unit of functionality, such that participants could easily understand the scope of the requirements.
- Understanding the use case shall require no understanding of domain-specific taxonomies.
- The use case shall imply at least four different types of security objectives.
- The use case specifications shall be openly accessible.

For the NCSU13, we selected two use cases for participants to identify security requirements for, both from the electronic healthcare domain that met our specified criteria. First use case (UC1 - Document office visit) is from the iTrust[9] electronic health record (EHR) system (Meneely et al. 2012), an open-source system developed by students at NCSU. The second use case (UC2 - Retrieve exam results by patient ID) is based on a user story[10] from Virtual Lifetime Electronic

[8] http://go.ncsu.edu/secreqtemplatesstudy

[9] http://agile.csc.ncsu.edu/iTrust/wiki/doku.php?id=requirements

[10] http://www.va.gov/vler/vlerdocs_userstories.asp

Record (VLER), a business and technology initiative that allows secure and standardized electronic exchange of health and benefits information for United States Veterans and Service members. Participants in UT14 and NCSU14 identified security requirements related to the same use cases as NCSU13.

For UCR15, to assess generalizability of findings across domains, we selected two use cases from the Cyclos[11] mobile payment software. Cyclos offers a complete mobile banking platform including SMS banking and Mobile application. The first use case (UC1- Make payment) is related to the functionality of making online payment to another member via SMS. The second use case (UC2- Retrieve account information) is related to the functionality for querying account information, such as current account balance, via SMS. Both use cases have a similar number of sentences and readability scores[12] as the use cases in the original study.

We did not introduce any differences in terms of the experiment material given to participants between NCSU13 and UT14. However, since participants in NCSU13 and UT14 were enrolled in different courses, they may differ in terms of knowledge and experience related to security requirements. For the second and third replications, NCSU14 and UCR15, we provided a 10-min video to participants introducing the concept of security objectives and requirements. UCR15 also got separate introductions in Spanish. Participants in UCR15 were also instructed to find as many security requirements as they could.

We also introduced two differences in terms of the support provided to the treatment group as compared to NCSU13. Firstly, participants in the treatment group for NCSU14 and UCR15 received additional details for filling in templates such as the subject, action and resource elements in the use case sentences. Participants in NCSU14 and UCR15 also received course work grade based on the quality of responses. With the additional support and strong incentive on quality, participants were expected to provide better quality responses as compared to participants in NCSU13 where one-third of the participants did not fill in the templates, impacting the quality score. Secondly, we intentionally suggested some extraneous templates to participants in NCSU14 and UCR15 that were not relevant to the given use case sentence. By suggesting extraneous templates, we wanted to assess whether the participants randomly select any suggested template or if they can differentiate between applicable and extraneous templates. The second change can have an impact on the relevance of the requirements identified by the participants. We have summarized these differences in Table 4.

5 Experiment Design

We used a 2 × 2 between-subjects design (Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/2/research_design/designs.html). Project Leader: David M. Lane, Rice University) for the original study and all subsequent replications. We automatically assigned study participants (students who agreed to participate in the study) to one of four groups in a round-robin fashion based on the process used for identifying requirements and the use case assigned (UC1 or UC2). All groups were given the same task of identifying security requirements. We provided specific sets of instructions, reference material, and task screens depending on the process (treatment vs. control) and the use case (UC1/UC2). To minimize potential bias, participants did not know about the existence of different groups or use cases. They were just informed that they will be given a use case scenario and will have to identify security requirements in accordance with the

[11] http://www.cyclos.org/mobilebanking/

[12] https://readability-score.com/

instructions. Participants could be assigned to one of the following processes for identifying security requirements:

- Treatment (T): automatically-suggested security requirements templates for identifying security requirements.
- Control (C): no templates, manual identification of security requirements.

In Table 5, we document the number of participants in each group for the original and replicated experiments. We recorded no personally identifiable information about the participants (e.g., name, student identifier).

In Fig. 2, we provide the task screen for treatment group for UCR15 (only showing two sentences from the use case for brevity). We indicate the subject, resource and action elements at the end of each sentence in the use case to help with filling in the security requirements templates.

The task screen for control group is similar to treatment group but there are no suggestions of applicable security objectives or templates for individual sentences in the use case. Participants have to manually identify applicable security requirements and enter into the text area at the bottom half of the page. For traceability, participants entered the security requirements in the text area followed by sentence number(s) from use case scenario to which the security requirement relates.

5.1 Evaluation Methodology

In this section, we present the methodology for evaluating the participants' responses to compute the metrics.

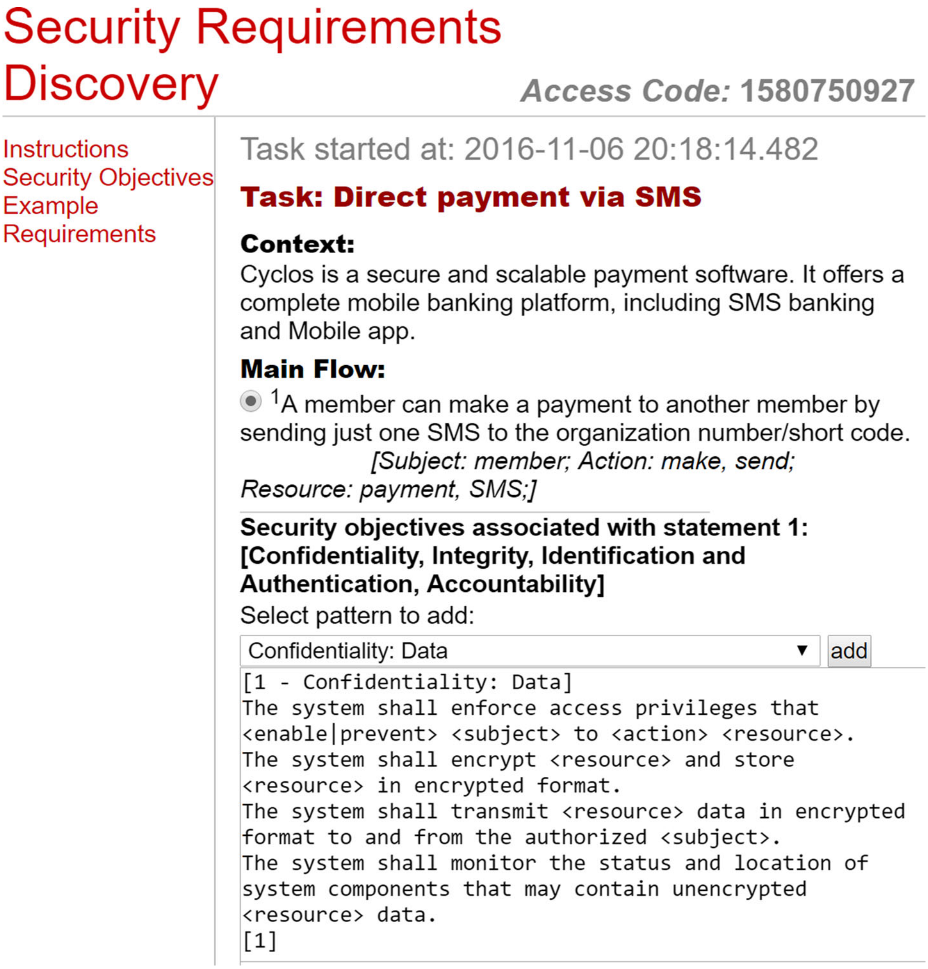
5.1.1 Oracle of Security Requirements

Five software security researchers, including the first three authors, created an oracle of the security requirements for each use case to evaluate the coverage and relevance of security requirements identified by the participants. The manual steps for creating the oracle are similar to the steps of SD process, as listed below:

- For each sentence in the use case, identify the security objectives associated with the sentence.
- For each identified objective, select the security requirements templates that are applicable.

Table 5 Number of participants in each group

Experiment	Treatment		Control		Total
	UC1	UC2	UC1	UC2	
NCSU13	16	14	10	10	50
UT14	9	8	9	6	32
NCSU14	29	26	25	27	107
UCR15	5	4	4	3	16
Overall	59	52	48	46	205
	111		94		



Security Requirements Discovery

Access Code: 1580750927

Task started at: 2016-11-06 20:18:14.482

Task: Direct payment via SMS

Context:
Cyclos is a secure and scalable payment software. It offers a complete mobile banking platform, including SMS banking and Mobile app.

Main Flow:
● [1]A member can make a payment to another member by sending just one SMS to the organization number/short code. *[Subject: member; Action: make, send; Resource: payment, SMS;]*

Security objectives associated with statement 1: [Confidentiality, Integrity, Identification and Authentication, Accountability]

Select pattern to add:

Confidentiality: Data ▼

```
[1 - Confidentiality: Data]
The system shall enforce access privileges that
<enable|prevent> <subject> to <action> <resource>.
The system shall encrypt <resource> and store
<resource> in encrypted format.
The system shall transmit <resource> data in encrypted
format to and from the authorized <subject>.
The system shall monitor the status and location of
system components that may contain unencrypted
<resource> data.
[1]
```

Fig. 2 Task screen for treatment group in UCR15

We used the same process for creating the solution oracle for all the use cases. Three of the five researchers who participated in the creation of study oracle for the original study, NCSU13, were involved in the creation of the oracle for UCR15. For consistency, we used the same classification guide for UCR15 as for NCSU13 when deciding on applicable objectives and templates. The classification guide lists textual clues that can indicate an implied security objective and the guide was provided to participants as part of the reference material as well.

Additionally, for the oracle related to the use cases from healthcare domain for the original study, we performed the following steps:

- For each applicable security requirements template, instantiate the templates by filling-in contextual details from the original sentence to generate concrete security requirements.
- Remove duplicate or redundant requirements.

In Table 6, we provide a summary of the templates associated with each of the use case sentences in the oracle for the use cases selected from the domains of healthcare (used in NCSU13, UT14,

Table 6 Security requirements templates associated with use case sentences in the oracle

Use case	Security requirements template	Sentences implying the template	Requirements in oracle (#)
UC1-Health [*NCSU13*, *UT14*, *NCSU14*]	C1: Confidentiality – Data	3, 4, 5, 7, 8, 9, 10	22
	I1: Integrity – Read-type actions	9, 10	2
	I2: Integrity – Write-type actions	3, 4, 5, 6, 7, 8	41
	I3: Integrity – Delete actions	9	1
	A1: Availability – Maintaining availability of data	9	1
	IA2: Identification and authentication – Unique accounts	1, 2	1
	IA3: Identification and authentication – User authentication	2	1
	AY1: Accountability – Logging transactions of sensitive data	3, 4, 5, 6, 7, 8, 9, 10	18
	AY2: Accountability – Logging authentication events	2	2
	PR1: Privacy – Usage of personal information	3, 4, 5, 7, 8, 9	21
UC2-Health [*NCSU13*, *UT14*, *NCSU14*]	C1: Confidentiality – Data	1, 3, 4, 5, 6, 8, 9	15
	A1: Availability – Maintaining availability of data	9	1
	A2: Availability – Maintaining response time	9	1
	AY1: Accountability – Logging transactions of sensitive data	2, 3, 4, 5, 6, 8, 9	12
	PR1: Privacy – Usage of personal information	1, 3, 4, 5, 8, 9	6
UC1-Mobile [*UCR15*]	C1: Confidentiality – Data	1, 2, 3, 4, 5, 6, 7, 8, 9	N/A
	I1: Integrity – Read-type actions	6, 7	
	I2: Integrity – Write-type actions	1, 3, 4, 5	
	IA2: Identification and authentication – Unique accounts	1, 2, 3, 4, 5	
	AY1: Accountability – Logging transactions of sensitive data	1, 2, 3, 4, 5, 8, 9	
	AY3: Accountability – Logging system events	6, 7	
UC2-Mobile [*UCR15*]	C1: Confidentiality – Data	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	N/A
	I1: Integrity – Read-type actions	5, 6, 7	
	I2: Integrity – Write-type actions	1, 2, 3, 4, 10	
	IA2: Identification and authentication – Unique accounts	1, 2, 3, 4, 8, 9, 10	
	AY1: Accountability – Logging transactions of sensitive data	1, 4, 8, 9, 10	
	AY3: Accountability – Logging system events	5, 6, 7	

NCSU14) and mobile banking (used in UCR15). For UCR15, we defined the oracle in terms of the templates (i.e., did not instantiate the templates to generate individual security requirements in the oracle), indicated by N/A in the last column.

5.1.2 Mapping Responses to the Oracle

The security requirements in the participants' responses were mapped to the requirements in the oracle to compute the metrics for coverage and relevance. For each participant's response, if a requirement in a response could be mapped to a requirement in the oracle, it was considered as a true positive (TP). If a requirement in a response could not be mapped to a requirement in the oracle, we considered two cases: a) the requirement is not related to the given use case and is thus a false positive (FP); and b) the requirement is related to the given use case scenario and should be marked as true positive (TP). In the latter case, we would also update the oracle to include the newly identified requirement. However, none of the participants identified any security requirement that was relevant to the scenario but not in the oracle already. Lastly, the requirements in the oracle not identified by a participant would be considered false negatives (FN).

For the treatment group, the requirements in the responses could be directly mapped to the requirements in the oracle as the requirements were generated using the same templates in both cases (TP). If a requirement was generated using an extraneous template, the requirement was marked as FP. Control group did not have the templates however they had example security requirements (one example requirement generated using each template) available with them. A number of participants in the control group used those examples as guide when specifying security requirements. Participants in the control group could also use their own words to phrase security requirements in which case we would not have a direct mapping to requirements in the oracle. In such cases, we mapped the requirements in the participants' response to one or more of the closest matching requirements in the oracle (TP). For instance, if a participant in the control group specified a general confidentiality requirement (e.g., all data should be encrypted during transmission), we mapped it to all the requirements for confidentiality during transmission in the oracle (e.g., encrypt passwords during transmission, encrypt health records during transmission) to minimize the potential advantage that treatment group had through the availability of templates. A participant in the treatment group would have to select the corresponding template (Confidentiality during transmission) for sentences related to each individual resource (e.g., passwords, health records) to get a similar mapping. In some cases, a requirement in the response for control group would partially map to a requirement in the oracle, in which case we still counted the requirement to be identified (TP). If a requirement in the response did not map to any of the requirements in the oracle, and was not relevant to the given scenario, we marked the requirement as FP.

For URC15, we defined the oracle in terms of the templates rather than individual requirements as explained in the previous section. Control group in UCR15 explicitly mentioned which security objective the requirement was related to and we mapped the requirement to the corresponding template (e.g., if the participant mentioned a requirement related to integrity, we mapped it to the template related to the objective of integrity). Essentially, we are comparing which security objectives each participant considered for a given sentence in the use case for treatment and control groups in UCR15. We may not get as granular mapping as other studies, but we still found significant differences in coverage between treatment and control groups in UCR15 indicating that participants in the treatment group considered significantly more security objectives and corresponding templates.

The metrics for coverage and relevance provide an assessment of the participant's performance in terms of how many requirements in the oracle a participant identified as well as how much of the effort was spent in identifying relevant requirements (TP) verses irrelevant ones (FP). We can have cases where for high relevance score, the participant has low coverage (e.g., participant only identified a few requirements and those requirements were in the oracle) and vice versa (e.g., participant identified a large number of requirements in the oracle, but also identified a large number of irrelevant requirements). We may also have cases where for the same relevance score, participants have different coverage scores (e.g., participant A identifies only 5 % requirements in the oracle and no irrelevant requirement while participant B identifies 80 % requirements in the oracle and no irrelevant ones – both have 100 % relevance score) and vice versa. In such cases, we may see no relation between the two metrics (Menzies et al. 2007). Moreover, some participants in the treatment group may select the suggested templates without deliberate consideration that may inflate the mean coverage scores. Researchers should identify such cases during evaluation. We have discussed mitigation of this threat in threats to validity.

6 Results Based on Individual Experiments

We present the results from each replication below and discuss whether the results support the hypotheses give in Section 4.1. We have used 2x2 ANOVA for unbalanced groups, adjusting for multiple comparisons (Tukey), to test the four null hypotheses, as in the original study. We used SAS version 9.4 for the statistical analysis. The two factors for grouping are: i) requirements process (tgroup) which can be either treatment or control; and ii) use case (ucid) which can be either 1 or 2. Using the analysis, we determine whether the factors or the interaction between the factors leads to significantly different group means for the four metrics. Results with $p < 0.05$ are considered significant for our analysis. Our data meets the ANOVA assumptions of independence, normality and homogeneity of variance (Levene's test) for the treatment and control groups across the studies in general. However, for UT14 and NCSU14, the homogeneity of variance assumption doesn't hold for the metric of relevance due to the large variations in the relevance scores for the control group as compared to the treatment group. For efficiency, the distribution of treatment group is slightly skewed due to a couple of outliers with high efficiency scores however the outliers do not affect the significance of results. Quality is a likert-like scale and the quality scores are normally distributed. Considering quality as an interval scale (assuming difference between scores 1 and 2 is similar to difference between scores 2 and 3), ANOVA can be applicable given that the parametric assumptions are satisfied (McCrum-Gardner 2008).

Two raters individually evaluated the responses and consolidated the evaluation through discussion. The two raters also assigned the quality scores based on a pre-specified criterion for assessing quality. For the metric of quality, we applied weighted kappa (Viera and Garrett 2005) using linear weights to assess how far apart the two raters were in assigning the quality scores. The weighted kappa scores[13] are 0.689 (good agreement), 0.948 (very good agreement) and 0.848 (very good agreement) for UT14, NCSU14 and UCR15 respectively.

We provide overall mean scores for all metrics across studies in Table 7 for a high-level overview of the findings across studies. We discuss the results for each study in the following

[13] http://graphpad.com/quickcalcs/kappa1/?K=5

Table 7 Overall mean scores for all metrics across studies

Study	Time available for completing the task	Mean time on task	Mean quality (1–5)	Mean coverage (0–1)	Mean relevance (0–1)	Mean efficiency (req./min)
NCSU13	60 minute in-class	~20 minutes	2.88	0.31	0.88	1.14
UT14	One week to complete at home	~47 minutes	2.70	0.36	0.77	1.07
NCSU14	60 minutes in-class	~25 minutes	2.66	0.37	0.73	1.01
UCR15	180 minutes in-class	~102 minutes	3.69	0.51	0.66	0.64*

*scaled for comparison

subsections. The results discussed in this section provide insights into the answers for research questions RQ1 to RQ4.

6.1 UT14: Replication at UT

Based on the results of UT14 as listed in Table 8, we found the requirements process (treatment vs. control, p-value = <0.0001) to be a significant factor in determining the relevance of identified security requirements. Thus, we reject the null hypothesis H_{03} that ratio of relevant requirements to total requirements identified is unrelated to the use of security requirements templates. Almost 90 % of the security requirements identified by the participants using the templates were relevant (88 % for UC1, 92 % for UC2). Only 61 % of the security requirements identified by participants in the control group were relevant (66 % for UC1, 53 % for UC2) on average. We did not find any other significant differences based on the requirements process or use cases. Thus, we fail to reject the null hypotheses H_{01}, H_{02} and H_{04} for UT14. The difference in coverage of security requirements between treatment and control group (41 % vs. ~30 %) is not significant at $p < 0.05$ but it is significant at $p < 0.1$. The coverage scores are also close to the corresponding values for NCSU14. The interaction between requirements process and use case is not significant for any of the metrics.

We provide box-plots capturing group means and variance for each requirements process (tgroup) and use case (ucid) in Fig. 3. On average, participants in the treatment group

Table 8 Results of 2 × 2 ANOVA for UT14

Factor ↓ / Metric →		Quality (1–5)		Coverage (0–1)		Relevance (0–1)		Efficiency (req/min)	
		Means	p-value	Means	p-value	Means	p-value	Means	p-value
Requirements Process (tgroup)	Treatment	2.91	0.1230	0.41	0.0782	0.90	<0.0001	1.37	0.2731
	Control	2.47		0.30		0.61		0.74	
Use case (ucid)	1	2.75	0.6220	0.30	0.0836	0.77	0.4674	1.33	0.2988
	2	2.64		0.43		0.76		0.74	

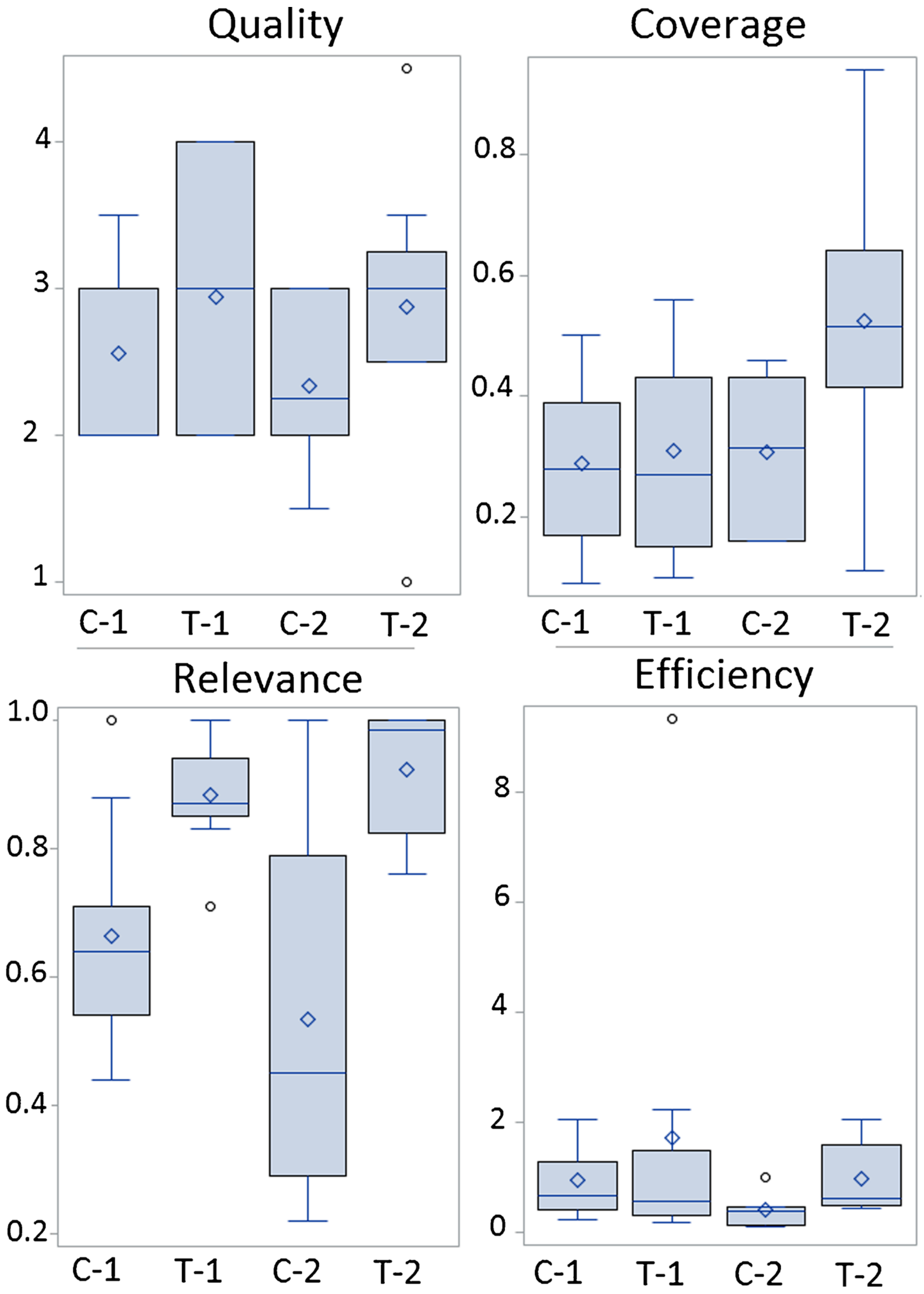


Fig. 3 Box-plots with results from UT14 across each factor (tgroup: T/C; ucid: 1/2) and metric

performed better than participants in the control group for each use case across all the four metrics.

Table 9 Results of 2 × 2 ANOVA for NCSU14

Factor ↓ / Metric →		Qualitys (1–5)		Coverage (0–1)		Relevance (0–1)		Efficiency (req/min)	
		Means	p-value	Means	p-value	Means	p-value	Means	p-value
Requirements Process (tgroup)	Treatment	3.33	<0.0001	0.46	<0.0001	0.91	<0.0001	1.18	0.0221
	Control	1.95		0.26		0.54		0.84	
Use case (ucid)	1	2.71	0.7706	0.27	<0.0001	0.75	0.3932	1.28	0.0003
	2	2.60		0.46		0.70		0.74	

6.2 NCSU14: Replication at NCSU

Based on the results of NCSU14 as listed in Table 9, we found that participants in the treatment group performed significantly better than the control group across all four metrics. Thus, we reject the null hypotheses H_{01}, H_{02}, H_{03} and H_{04} that the performance of participants based on metrics for quality, coverage, relevance and efficiency is unrelated to the use of security requirements templates. Participants in the treatment group produced significantly better quality requirements (3.33 versus 1.95 for control group). Participants in the treatment group also identified significantly more requirements (~46 % vs. ~26 % for control group) and the identified requirements were more often relevant (~91 % vs. ~54 % for control group). The participants in the treatment group were also 40 % more efficient as compared to the control group overall. This relative efficiency translates to the identification of an additional requirement every 3 min for the treatment group and may not provide any practical significance in terms of the efficiency of the process.

We also found significant differences between the coverage (p-value <0.0001) of security requirements identified for each use case. The difference in efficiency (p-value = 0.0003) is also significant. We have a total of 110 security requirements for UC1 versus 35 security requirements for UC2 in the oracle. Although the participants working on UC1 were more efficient, as they identified more total security requirements per unit of time, the percentage of identified requirements for UC1 was less compared to the percentage of requirements identified for UC2.

The interaction between requirements process and use case was not significant for any of the metrics at $p\text{-value} < 0.05$. We provide box-plots capturing group means and variance for each requirements process (tgroup) and use case (ucid) in Fig. 4. Participants in the treatment group performed better than participants in the control group for each use case across all the four metrics.

6.3 UCR15: Replication at UCR

Based on the results of UCR15 as listed in Table 10, we found the requirements process (treatment vs. control) to be a significant factor in determining the metrics for coverage (p-value = 0.0162) and efficiency (p-value = 0.013). Thus, we reject the null hypotheses H_{02} and H_{04} that the performance of participants based on metrics for coverage and efficiency is unrelated to the use of security requirements templates. Participants in the treatment group identified almost 63 % of all the requirements in the oracle (65 % for UC1, 60.5 % for UC2).

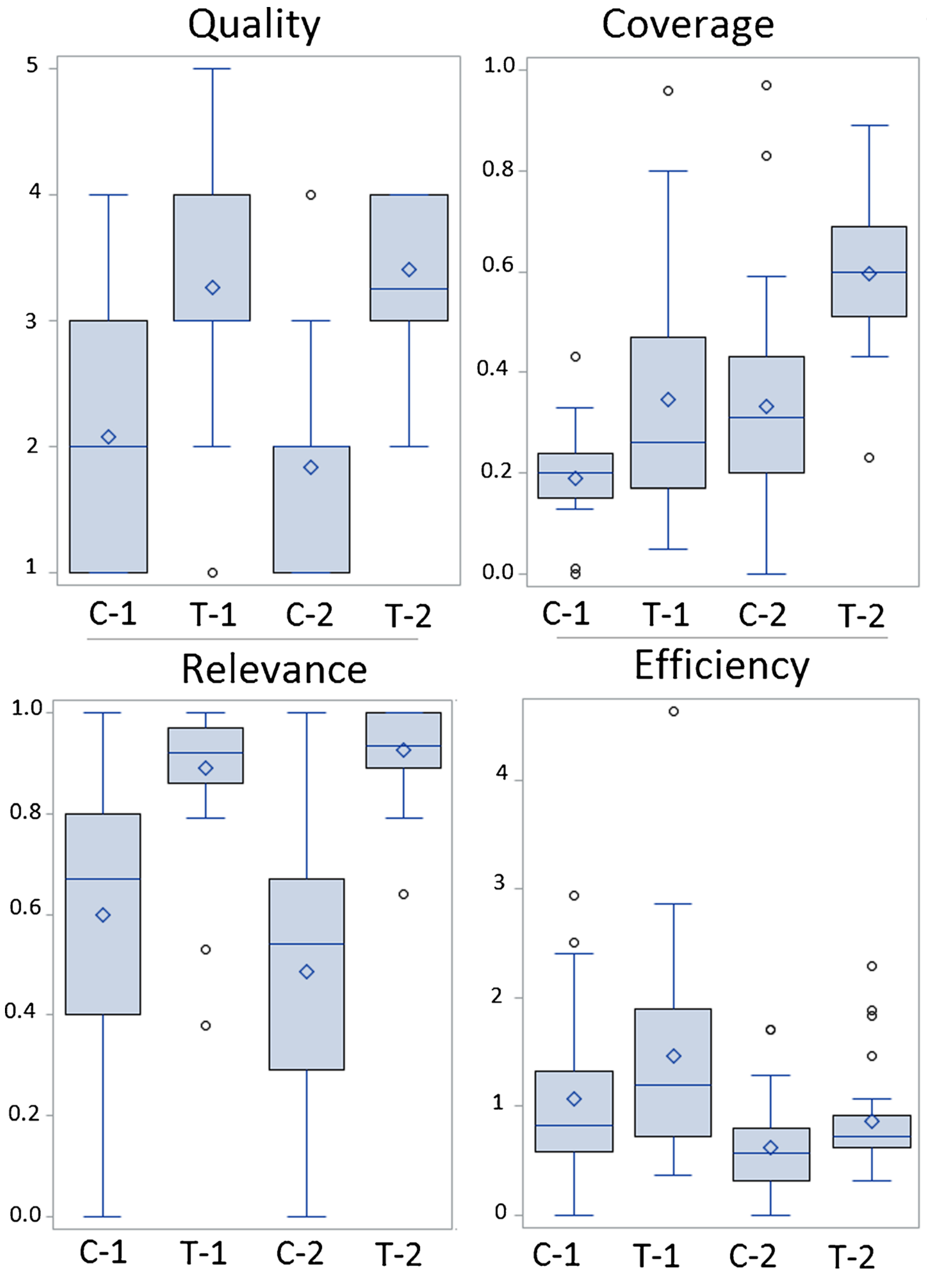


Fig. 4 Box-plots with results from NCSU14 across each factor (tgroup: T/C; ucid: 1/2) and metric

In comparison, participants in the control group identified only 36 % of all the requirements in the oracle (40.5 % for UC1, 30 % for UC2) on average.

Participants in the treatment group were also 77 % more efficient as compared to control group, although the efficiency scores for both the groups is much lower as compared to other

Table 10 Results of 2 × 2 ANOVA for UCR15

Factor ↓ / Metric →		Quality (1–5)		Coverage (0–1)		Relevance (0–1)		Efficiency (templates/min)	
		Means	p-value	Means	p-value	Means	p-value	Means	p-value
Requirements process (tgroup)	Treatment	3.94	0.1565	0.63	0.0162	0.71	0.0631	0.20	0.0130
	Control	3.36		0.36		0.59		0.11	
Use case (ucid)	1	3.83	0.3998	0.54	0.4650	0.66	0.9695	0.14	0.2679
	2	3.50		0.47		0.66		0.18	

experiments. The reason for this difference is primarily that, for UCR15, we computed the metrics based on the number of security requirements templates identified per minutes instead of the number of individual security requirements identified per minute. Each template generates multiple security requirements as discussed earlier.

We did not find any significant differences for the metrics of quality and relevance based on the requirements process or use cases. Thus, we fail to reject the null hypotheses H_{01} and H_{03} for UCR15. The difference in relevance of security requirements between treatment and control group (71 % vs. ~59 %) is not significant at $p < 0.05$ but is significant at $p < 0.1$. Although participants in the treatment group received suggestion for extraneous templates, the requirements identified are still more relevant as compared to the control group. The overall quality of responses in UCR15 was better than all other replications at 3.7 compared to a range of 2.7 to 2.9 for other studies. The difference might be due to the change in use cases for the study or the fact that participants spent the most time on task, two to five times more than other studies. Interaction between the requirements process and use case is not significant for any of the metrics.

We provide box-plots capturing group means and variance for each requirements process (tgroup) and use case (ucid) in Fig. 5. Participants in the treatment group performed better than participants in the control group for each use case across all the four metrics.

6.4 Summary of Findings from Individual Studies

We compare the performance of participants in the treatment and control groups across the studies in terms of the four metrics for quality, coverage, relevance and efficiency to address research questions RQ1-RQ4. In Table 11, we provide a study-wise summary of differences in mean scores between treatment and control groups for each of the four metrics (treatment mean – control mean). We also indicate whether the difference is significant or not. We plot the mean scores for the four metrics across studies for the treatment and control groups in Fig. 6. The actual mean scores for each study are provided in previous sections.

Participants in the treatment group performed better than participants in the control group across all the four metrics in all the studies for both use cases. The differences were most evident for the metrics of coverage and efficiency as shown in Fig. 6. We address RQ1-RQ4 based on the findings below.

> *RQ1*: What is the quality of security requirements elicited through the use of automatically-suggested security requirements templates?

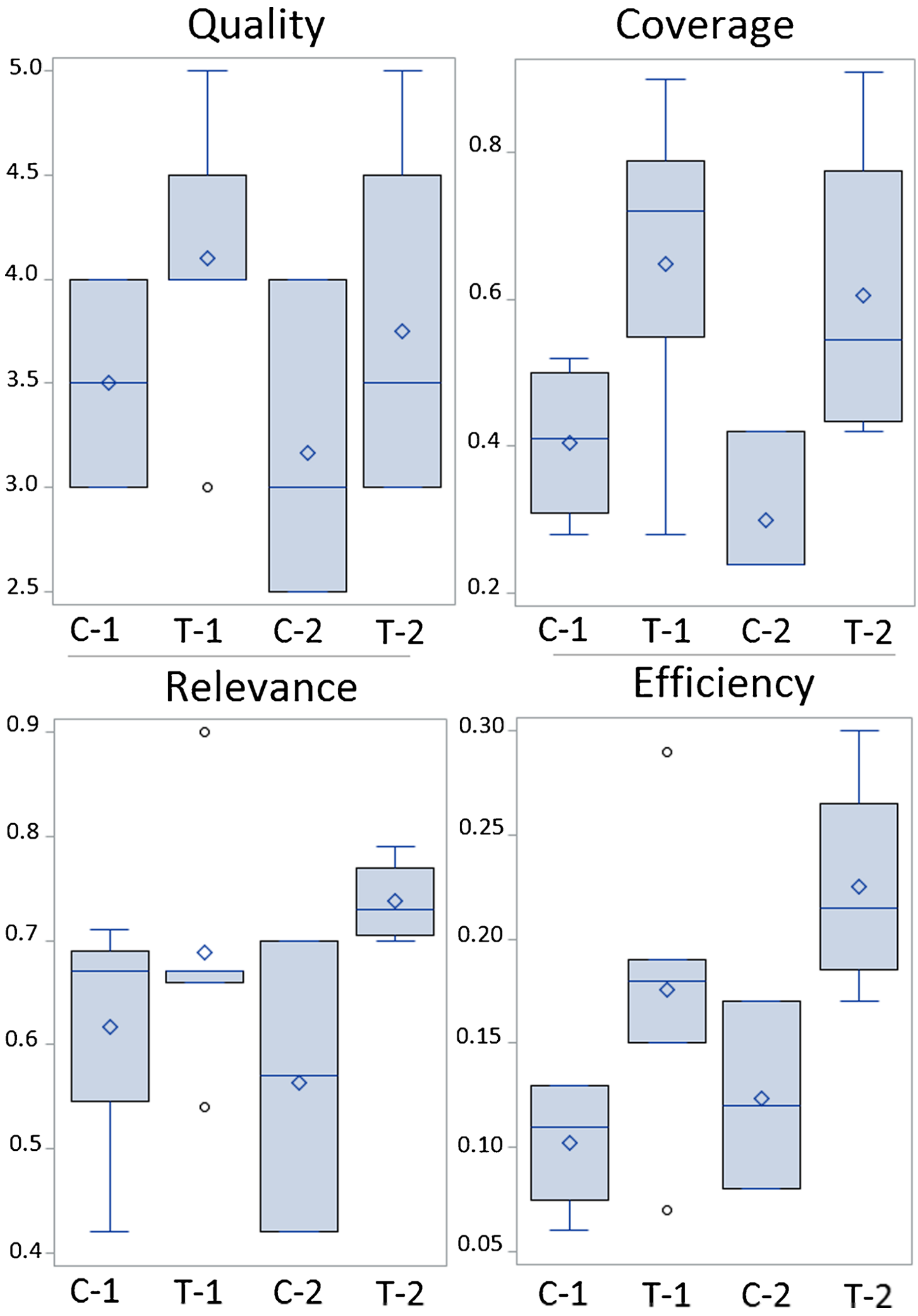


Fig. 5 Box-plots with results from UCR15 across each factor (tgroup: T/C; ucid: 1/2) and metric

We found the least differences in the treatment and control groups in terms of the quality of the identified security requirements. The overall quality of the treatment group was 3.2 compared to 2.3 for the control group. Participants in the control group

Table 11 Difference between treatment and control group means across studies

Study	Δ Quality (−4 to 4)	Δ Coverage (−1 to 1)	Δ Relevance (−1 to 1)	Δ Efficiency (−2 to 2)
NCSU13	−0.03	0.27 **	0.05	0.58 **
UT14	0.44	0.11 *	0.29 **	0.63
NCSU14	1.38 **	0.20 **	0.37 **	0.34 **
UCR15	0.58	0.27 **	0.12 *	0.36 **

*Significant at p-value < 0.1; **Significant at p-value < 0.05

performed slightly better in terms of the quality of the identified security requirements for one of the use cases (UC1) in the original study NCSU13 (2.95 vs. 2.78), providing the only result where control group performed better. However, the difference is not statistically significant. In NCSU13, almost half of the participants in the treatment group for UC1 did not fill in the templates which negatively impacted the overall quality score. For the replication studies, participants in the treatment group produced better quality requirements as compared to the control group. The difference is significant for NCSU14 with the largest number of participants (107). Participants in the treatment

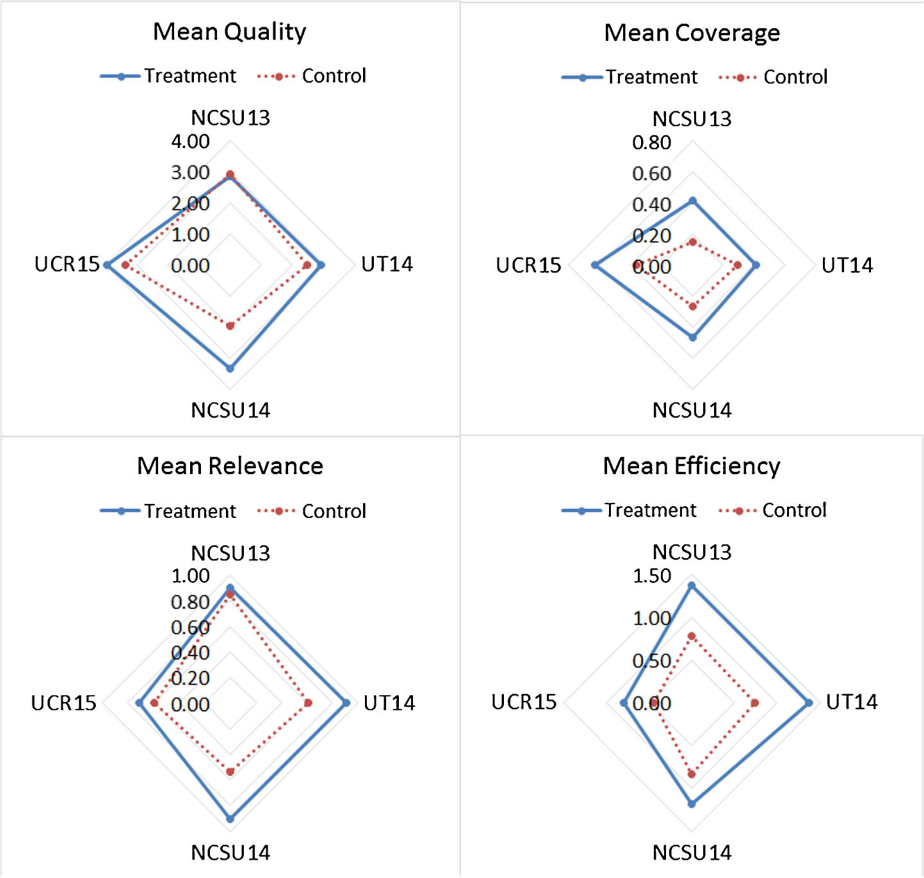


Fig. 6 Mean scores for treatment and control groups across studies

group for NCSU14 had additional support in filling the templates which could have positively impacted the quality scores.

> *RQ2*: What is the coverage of security requirements elicited through the use of automatically-suggested security requirements templates?

In all the studies, requirements coverage of the treatment group is better than the control group. The treatment and control groups differ significantly (at p-value < 0.05) in terms of the coverage of the identified requirements in three of the four studies. If we consider a significance level of 0.1, treatment and control group differ significantly in all the studies for the metric of coverage. Participants in the control group identified 25 % of the security requirements in the oracle overall whereas participants in the treatment group identified almost 46 % of the security requirements in the oracle.

Missing requirements is a common problem in requirements engineering (Walia and Carver 2009) and we identified that participants in our studies also had missing security requirements, indicated by the low coverage scores. Overall, between 49 and 69 % of the relevant security requirements in the oracle were not identified by the participants across studies. To some extent, this lack of security requirements coverage may be due to limited security expertise of the participants, limited resources and time constraints, and to the fact that no one individual may identify all applicable security requirements. One of the most commonly cited reasons for missing requirements is the lack of knowledge or expertise. Given that participants in our studies were not security experts and not involved in the on-going development of the systems they analyzed, they may miss more requirements as compared to an individual with additional security expertise and domain knowledge.

> *RQ3*: How relevant are the security requirements elicited through the use of automatically-suggested security requirements templates?

The requirements identified by treatment group were more relevant as compared to the control group in all the studies. The difference was significant in two of the four studies at p-value < 0.05. If we consider significance level of 0.1, three of the four studies differed significantly in terms of the relevance of identified security requirements. In three studies, the relevance of treatment group is over 90 % whereas in UCR15, with the smallest sample size, the relevance is the lowest at 71 %. Overall relevance of treatment group is 89 % compared to 62 % of the control group.

> *RQ4*: How efficient is the process of eliciting security requirements through the use of automatically-suggested security requirements templates?

In all the studies, participants in the treatment group, using the automatically-suggested templates, were more efficient as compared to the control group. Treatment and control group differ significantly in terms of the efficiency of the requirements elicitation process in three of the four studies. Participants using the templates (treatment group) to elicit security requirements were also 57 % more efficient as compared to the control group overall (1.23 vs 0.78).

For study UCR15, we selected use cases from the domain of online banking instead of healthcare. The results for UCR15 are similar to the findings of the NCSU13 where the treatment group is significantly better than the control group for the metrics of coverage and efficiency. Security requirements templates support elicitation of applicable security requirements in the selected use cases for both healthcare and mobile banking domains. The automatically-suggested templates capture the security knowledge of multiple experts and can support the security requirements elicitation process as indicated by the results.

7 Results Based on Analysis Across Studies

We present the synthesis of results from the original experiment and subsequent replications below. We provide a comparative analysis of studies in terms of the metrics for quality, coverage, relevance and efficiency. We have raw data available from all the studies. Despite some differences across studies, we use the same or comparable metrics for computing the results. We perform a three-way ANOVA for unbalanced groups, adjusting for multiple comparisons (Tukey), to test the four null hypotheses for the combined data from all four studies. The third factor, in addition to the group and use case, is the study itself. Results with $p < 0.05$ are considered significant for our analysis. We also qualitatively analyze the studies to address research questions RQ5-RQ7.

7.1 Combined Analysis of Variance

We combined the raw data from all the studies to perform the combined analysis of variance across the three factors: requirements process (treatment, control), use case (1-health, 2-health, 1-mobile, 2-mobile) and study (NCSU13, UT14, NCSU14, UCR15). The combined analysis is equivalent to performing an aggregate analysis of the individual studies, assigning weights based on the number of participants in each study. The combined results are most affected by the performance of participants in NCSU14 since NCSU14 has the largest number of participants, almost half of the total participants. We scaled the metric for efficiency for UCR15 by a factor of four (average number of requirements per template), converting from templates per minute to requirements per minute, to

Table 12 Results of 3-way ANOVA for combined data from all studies

Factor ↓ / Metric →		# of Obs.	Quality (1–5)		Coverage (0–1)		Relevance (0–1)		Efficiency (req./min)	
			Mean	p-value	Mean	p-value	Mean	p-value	Mean	p-value
Requirements process (tgroup)	Treatment	111	3.2	0.0014	0.46	<0.0001	0.89	<0.0001	1.23	0.0168
	Control	94	2.3		0.25		0.62		0.78	
Use case (ucid)	1-health	98	2.76	0.7214	0.26	<0.0001	0.79	0.6723	1.34	0.0005
	2-health	91	2.69		0.45		0.76		0.74	
	1-mobile	09	3.83		0.54		0.66		0.57	
	2-mobile	07	3.5		0.47		0.66		0.73	
Study (study)	NCSU13	50	2.88	0.3075	0.31	0.0808	0.88	0.0002	1.14	0.9372
	UT14	32	2.70		0.36		0.77		1.07	
	NCSU14	107	2.66		0.37		0.73		1.01	
	UCR15	16	3.69		0.51		0.66		0.64*	
Significant interactions ($p < 0.05$)			tgroup*study (<0.0001)		tgroup*study (0.0001)		tgroup*study (0.0001)		NONE	

*scaled for comparison

make it comparable with other studies. Based on the results of combined analysis of data using a three-way ANOVA, as listed in Table 12, we found that participants in the treatment group performed significantly better than the control group across all the four metrics for quality, coverage, relevance and efficiency of the identified security requirements. Thus, we reject the null hypotheses H_{01}, H_{02}, H_{03} and H_{04} that the performance of participants in eliciting security requirements in terms of quality, coverage, relevance and efficiency metrics is unrelated to the use of security requirements templates. Participants in the treatment group produced significantly better quality requirements (3.2 vs 2.3 for control group). Participants in the treatment group also identified significantly more requirements (~46 % vs. ~25 % for control group) and the identified requirements were more often relevant (~89 % vs. ~62 % for control group). Participants in the treatment group were also more efficient in identifying the security requirements (1.23 vs 0.78 for control group).

We found significant differences between the two use cases from the healthcare domain for the metrics of coverage and efficiency. Due to the large number of total requirements in the oracle for UC1 in the healthcare domain, the coverage percentage is lower, but efficiency is higher as there are more requirements to be identified overall for UC1. Results also indicate significant differences between the original study NCSU13 and NCSU14 (the largest study) for the metrics of relevance. These differences are due to the lower relevance scores of the control group in NCSU14 as compared to NCSU13. No other differences between studies are significant.

Considering interactions between the factors, we found significant interactions between requirements process and study for the metrics of quality and relevance as shown in Table 12. Specifically, we observed large variations in the quality of responses by both control and treatment groups across the studies. The relevance of responses also varied across the studies for both control and treatment groups. In terms of coverage, we found significant interaction between the requirements process and use case. No other interactions were significant at p-value <0.05. We provide group means and variance for each metric across all the groups in the studies in Table 13.

Participants in the treatment group produced significantly better quality requirements as compared to the control group overall. Participants in the treatment group also identified significantly more requirements, almost twice as many as control, and the identified requirements were more often relevant. Participants in the treatment group were also more efficient in identifying the security requirements with the support of automatically suggested templates.

7.2 Qualitative Analysis Based on Differences Introduced Among Studies

Studies differed in terms of participants, empirical setting, time and support available to participants in performing the task. We have summarized these differences in the context factors in Table 4. We can qualitatively assess if these factors impacted the results to address the additional research questions.

7.2.1 Filling the Details in Security Requirements Templates

RQ5: *Are participants more inclined to fill in the templates when additional support to fill the templates is provided by explicitly indicating subject, action and resource elements in the input requirements*?

Table 13 Group means and variances for the combined data from all the studies

			Control			Treatment		
Study	Use case	Metric	# of Obs.	Mean	Std. dev	# of Obs.	Mean	Std. dev
NCSU13	1	Quality	10	2.95	1.09	16	2.78	0.97
		Coverage	10	0.12	0.06	16	0.24	0.13
		Relevance	10	0.86	0.31	16	0.90	0.06
		Efficiency	10	1.15	0.68	16	1.70	1.52
	2	Quality	10	2.85	0.82	14	2.96	1.22
		Coverage	10	0.19	0.07	14	0.62	0.31
		Relevance	10	0.84	0.24	14	0.91	0.16
		Efficiency	10	0.40	0.13	14	1.00	0.39
UT14	1	Quality	9	2.56	0.68	9	2.94	0.88
		Coverage	9	0.29	0.14	9	0.31	0.18
		Relevance	9	0.66	0.18	9	0.88	0.09
		Efficiency	9	0.95	0.70	9	1.71	2.94
	2	Quality	6	2.33	0.61	8	2.88	0.99
		Coverage	6	0.31	0.14	8	0.52	0.24
		Relevance	6	0.53	0.30	8	0.92	0.10
		Efficiency	6	0.41	0.33	8	0.99	0.69
NCSU14	1	Quality	25	2.08	1.00	29	3.26	0.99
		Coverage	25	0.19	0.11	29	0.35	0.25
		Relevance	25	0.60	0.32	29	0.89	0.14
		Efficiency	25	1.07	0.81	29	1.46	0.95
	2	Quality	27	1.83	0.91	26	3.40	0.63
		Coverage	27	0.33	0.23	26	0.60	0.17
		Relevance	27	0.49	0.29	26	0.93	0.09
		Efficiency	27	0.62	0.46	26	0.87	0.47
UCR15	1	Quality	4	3.50	0.58	5	4.10	0.74
		Coverage	4	0.41	0.11	5	0.65	0.24
		Relevance	4	0.62	0.13	5	0.69	0.13
		Efficiency	4	0.41	0.14	5	0.70	0.32
	2	Quality	3	3.17	0.76	4	3.75	0.96
		Coverage	3	0.30	0.10	4	0.61	0.23
		Relevance	3	0.56	0.14	4	0.74	0.04
		Efficiency	3	0.49	0.18	4	0.90	0.22

We hypothesize that providing additional support in filling templates by explicitly indicating subject, action and resource elements in the input requirements will lead to a higher percentage of participants filling in the templates.

Participants in the treatment group for NCSU14 and UCR15 received additional support in filling templates as we explicitly indicated the subject, action and resource for each of the input sentences. Participants in both studies also received course work credit based on the quality of their responses as opposed to just a participatory grade in the original study, NCSU13. For

NCSU14, the course work credit was minimal whereas for UCR15, it was a significant percentage of the final grade. Participants in UCR15 also had the most dedicated time for the task.

We looked at the percentage of participants who filled the templates in the treatment group for different studies. In the original study, NCSU13, only 62 % of the participants filled in the templates. In UT14, only 59 % of the participants filled in the templates. In both cases, participants did not have support in filling the templates. In NCSU14, 78 % of the participants filled in the templates which is an increase of ~26 % as compared to NCSU13. In UCR15, 100 % of the participants filled in the templates. These results suggest that participants might be more inclined to fill in the templates with the additional support. However, participants had stronger motivation for NCSU14 and UCR15 as compared to NCSU13 and UT14.

To control for motivation, we ran a confirmatory experiment at UT in October 2015, UT15, as an in class exercise with compulsory participation but no impact on grading. The goal of UT15 was to see whether participants in the treatment group fill-in the templates if we provide support in filling the templates, but not a lot of time or incentive. The experiment was setup similar to NCSU14 and UCR15 and was only intended to investigate RQ5 further. Participants in UT15 had support in filling templates, similar to NCSU14 and UCR15. However, the participants had limited time and did not have strong motivation. Using the same metrics that we have used for the other studies, we found that 15 of the 18 participants (83 %) in the treatment group filled the templates despite lack of strong motivation and time on task.

The mean quality score of the treatment group is impacted positively if more participants fill in the templates. The mean quality of participants in the treatment group in NCSU14 was significantly better than the control group (3.33 vs. 1.95 of control) and also better than the treatment group in NCSU13 (2.86). The overall quality of responses in UCR15 was better than all other studies as shown in Table 7. The improved quality can be attributed to a number of factors including the additional support and considerable motivation to perform well, different problem domain, or the fact that participants spent the most time on task, two to five times more than other studies.

Strong motivation and more time on task are only partial drivers to increase the rate of filling in the templates. Providing additional support in filling templates by explicitly indicating subject action and resource elements in the input requirements leads to a higher percentage of participants filling in the templates.

7.2.2 Differentiating Between Relevant and Extraneous Templates

> *RQ6*: *Can participants differentiate whether a suggested security requirements template is relevant to the given use case scenario*?

We hypothesize that participants will consider whether a suggested template is relevant to the given use case scenario when selecting applicable templates.

In addition to the relevant security requirements templates, participants in the treatment group for NCSU14 and UCR15 were intentionally suggested extraneous templates. We can qualitatively examine if the participants selected relevant templates or extraneous ones as well. Based on the relevance scores given in Table 7, participants in NCSU14 and UCR15 identified less relevant requirements overall. Looking at the relevance scores for only the treatment groups, 90 % of the security requirements identified in the NCSU13 were relevant as compared to 91 % for NCSU14 and 71 % for UCR15. Participants in NCSU14 and NCSU13 identified almost the same percentage of relevant requirements indicating that participants did not randomly select the suggested templates in NCSU14 and that participants may be able to differentiate between good and bad suggestions. The relevance of requirements is lower for UCR15 compared to NCSU13. In addition to the lack of knowledge about presence of extraneous templates, this difference can be attributed to the different problem domain, more time on task and the instructions to identify as many security requirements as possible.

We also looked at the counts of extraneous templates selected by the participants in both studies. The four use cases analyzed in the studies have between 22 and 35 relevant templates, based on aggregating the third column in Table 6. In NCSU14, 26 of the 55 participants (47 %) in the treatment group selected at least one extraneous template. The 26 participants selected 4 extraneous templates on average. If we average over all 55 participants in the treatment group, we have ~2 extraneous templates per participant. In UCR15, all 9 participants in the treatment group selected at least one extraneous template with an average of 8 extraneous templates selected by each participant. Participants in UCR15 selected four-times the number of extraneous templates on average as compared to NCSU14 and spent thrice the time on task. Consequently, the relevance score for UCR15 was the least (71 %) whereas relevance score in all other studies is over 90 % for the treatment group.

Based on the feedback from the participants, as discussed in Section 7 participants considered the templates to be a good starting point for identifying security requirements. Although participants were asked to select the templates they thought were applicable, some participants may consider extraneous templates as valid suggestions if they found most of the other suggested templates relevant.

Participants may be able to differentiate whether a suggested security requirement template is relevant or extraneous to the given use case scenario. However extraneous suggested templates may confuse the participants if they consider them a good starting point and always potentially correct suggestions. Spending more time on task may also lead to identifying some extraneous security requirements.

7.2.3 Impact of Task Time on Study Outcomes

> *RQ7*: *Are there context factors*, *such as more time on task*, *which are conducive to producing better outcomes overall*?

We hypothesize that differences in context factors, such as more time on task, account for some of the differences in findings across the studies.

UT14 was conducted as a take-home assignment with 1 week to complete. NCSU14 was conducted as an in-class activity of 60 min as was NCSU13, but participants had slightly stronger motivation to produce better outcomes as compared to NCSU13. UCR15 was conducted as an in-class activity, however participants had 180 min to work on the task as compared to 60 min in the original study and had strong motivation to produce good quality outcomes.

In UT14 and NCSU14, participants identified almost 36 and 38 % of all the requirements in the oracle respectively, slightly more than the 31 % identified for NCSU13 as shown in Table 7.

Participants in UCR15 identified 51 % of all the requirements on average, which is an increase of almost 64 % as compared to the NCSU13. Overall, participants in UCR15 spent the most time on task, 102 min on average, which is almost five times that of NCSU13. The significant increase in the coverage of security requirements identified in UCR15 could be due to other factors as well, such as different problem domain or strong motivation to perform well.

Participants may use the available time as cue for the expected time they ought to spend on the task. Another factor that may influence the time spent on task is the expected credit or reward. Participants may look at both the available time and expected credit to decide how much of the available time to spend on task, especially if the time available is not dedicated time for task. With comparable credit for the task in NCSU13 and UT14, participants in UT14 spent more time on task on average (20 vs 47 min) even if the time was not a dedicated slot for the task. However, time spent in case of UT14 was still less than time spent in UCR15 where availability of more dedicated time coincided with higher credit for the task as well.

Considering the overall mean relevance and coverage scores for the studies, with more time spent on task, the mean coverage scores increased but the mean relevance scores went down as shown in Table 7. For studies UT14 and NCSU14, the decrease in relevance can be attributed to the performance of the control group as treatment group had high relevance scores (over 90 %). However, for UCR15, the decrease in relevance is also due to the performance of the treatment group where relevance is 71 %. This may be attributed to the availability of extraneous templates and availability of more time on task. We provided extraneous templates to participants in NCSU14 as well, but the relevance scores didn't go down. Under time constraints, as in NCSU14, participants may select fewer templates that seem the most relevant and thus irrelevant templates are mostly left out as well as some relevant templates. Whereas when ample time is available, as in UCR15, once the participants have identified most relevant security requirements, they may try to improve the response by selecting additional templates, consequently increasing the chance of selecting irrelevant templates as well. This can explain the simultaneous increase in coverage (63 %, whereas other studies have between 41 and 46 %) and decrease in relevance (71 %, whereas other studies have over 90 %) observed in the treatment group in UCR15. However, we don't have a reliable way of knowing the order in which requirements were identified by each participant.

In terms of efficiency, participants in NCSU13 were more efficient as compared to participants in the subsequent replications as shown in Table 7. The results indicate that although participants in UT14, NCSU14 and UCR15 identified relatively more requirements than NCSU13, they were not more efficient and identified some extraneous requirements.

We plot the relation between tasktime (in minutes) and the metrics for quality, coverage and relevance in Fig. 7 for the combined data of 205 participants across the four studies. By definition, tasktime and efficiency are negatively correlated. Tasktime was much larger for UCR15 as compared to other studies as visible in Fig. 7. For NCSU14, the study with the largest number of participants, most of the responses are clustered towards higher relevance and lower coverage. If we plotted the figure grouped by treatment and control groups instead, we would see that treatment group participants are clustered towards higher relevance and control group participants are clustered towards lower coverage.

We identified a significant positive correlation between tasktime and quality of identified requirements overall (Pearson correlation coefficient of 0.39 at p-value < 0.0001). We also found a significant positive correlation between tasktime and coverage of identified requirements overall (Pearson correlation coefficient of 0.29 at p-value < 0.0001), similar to our findings in NCSU13. We did not identify any significant correlation between tasktime and

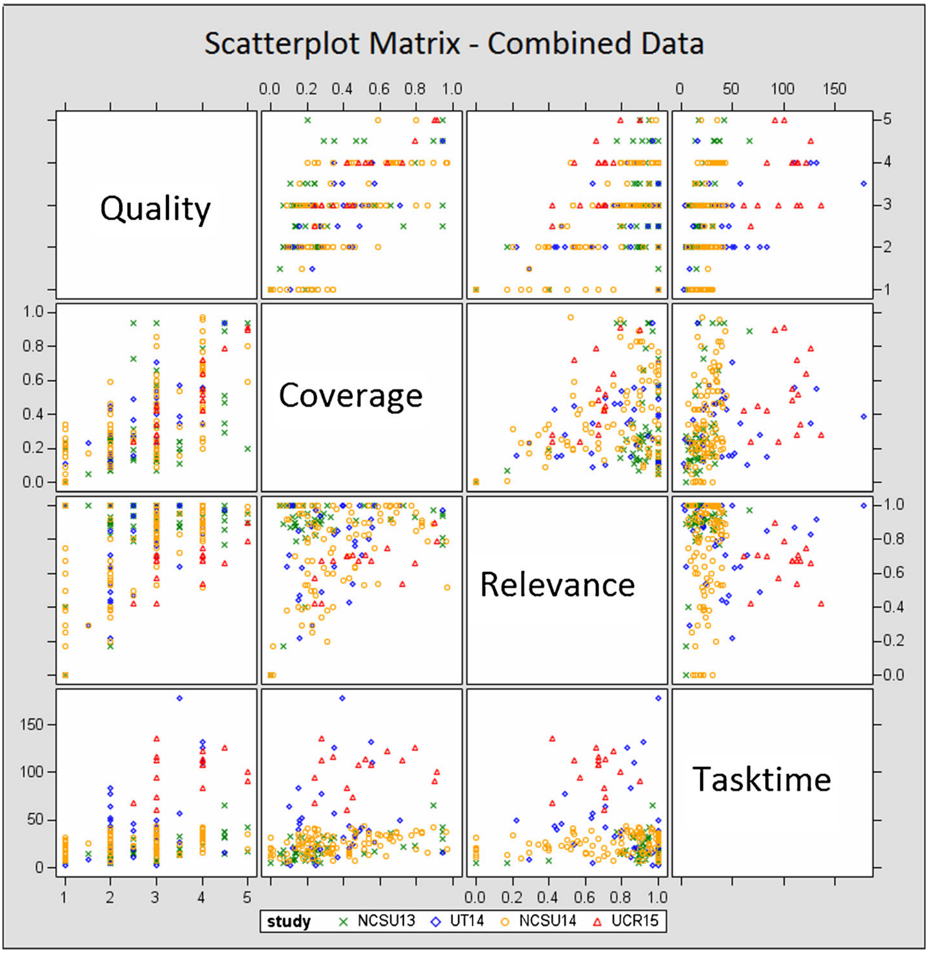


Fig. 7 Relation among tasktime and metrics of quality, coverage and relevance for all participants

relevance (Pearson correlation coefficient of −0.04, p-value = 0.5756). In terms of relation among the metrics, quality is positively correlated with coverage (Pearson correlation coefficient of 0.64 at p-value < 0.0001) and relevance (Pearson correlation coefficient of 0.48 at p-value < 0.0001). Coverage and relevance metrics are also positively correlated (Pearson correlation coefficient of 0.32 at p-value < 0.0001), indicating that the participants who performed well, performed well across multiple metrics. There is no apparent correlation between other data points.

Participants who performed well performed well across multiple metrics. Participants may look at both the available time and expected credit to decide how to approach the task. Under time constraints participants may select fewer templates that seem the most relevant. Spending more time on task can lead to identifying more relevant security requirements. However some participants may spend the additional time in improving the quality of the identified requirements (filling in templates using reference material) or in identifying some irrelevant requirements.

7.3 Breakdown of Identified Security Requirements

We group security requirements by the objectives that will be supported if a system satisfies the security requirement. For example, security requirements related to ensuring access control over sensitive resources support the objective of confidentiality. Similarly, security requirements related to logging user activities will support the objective of accountability. The first three studies, NCSU13, UT14 and NCSU14, involved identifying implied security requirements based on use cases selected from the domain of healthcare. The fourth study, UCR15, involved identifying implied security requirements based on use cases selected from the domain of mobile banking. For all the four use cases, we computed the number of requirements in the oracle (Table 6) for each security objective. We also computed how many of the requirements in the oracle were identified by the participants on average in the treatment and control groups for each objective and plot the results in Fig. 8.

The treatment group not only identified more security requirements overall, but also more security requirements for each security objective as compared to the control group in general. The security requirements related to the objectives of confidentiality and accountability were the most frequently identified by both treatment and control groups and were also the most common requirements in the oracle for all the use cases as shown in Fig. 8. Looking at the different types of confidentiality requirements identified, the majority of the participants

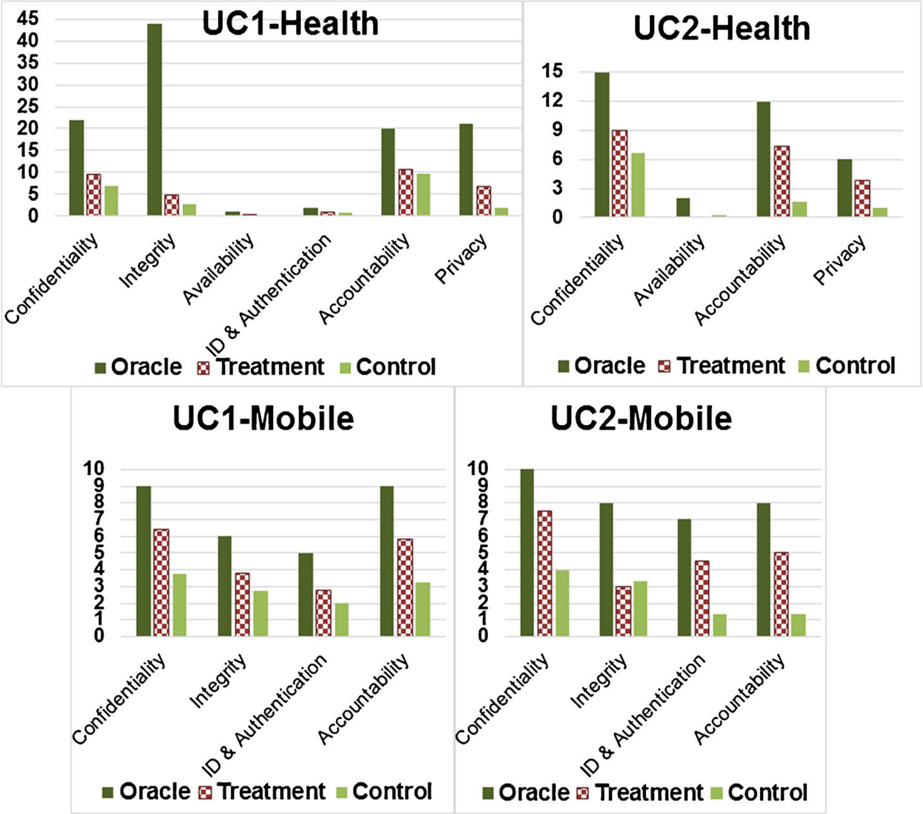


Fig. 8 Requirements in the oracle identified per security objective

identified requirements related to ‘enforcing access privileges’ in both groups. The majority of the participants in the treatment group also considered requirements for confidentiality during storage and transmission as well as monitoring activity of locations where sensitive information is stored. A very small number of participants in the control group considered these additional confidentiality requirements. We observed similar trend for the accountability requirements where almost all of the participants in both groups identified requirements to log one or more of the user actions (e.g., documenting office visit, updating office visit). Moreover, almost everyone in the treatment group considered requirements for integrity of log files whereas almost everyone in the control group ignored these requirements. These findings indicate that participants in the control group may consider only the most obvious security requirements (e.g., access control, logging of activities) for a given security objective whereas participants in the treatment group are able to consider additional requirements as well.

For the healthcare domain, the requirements related to privacy were identified by the treatment group, but not as often by the control group. The oracle for UC1 had a fairly large number of integrity requirements but only a small fraction of these requirements were identified by the participants in both groups. We had the least number of requirements related to identification and authentication and availability in the oracle for UC1-Health and UC2-Health. Almost everyone identified some requirements related to identification and authentication. Whereas almost no one identified requirements related to availability.

We also provide the breakdown of identified requirements for the mobile banking domain in Fig. 8. The requirements identified by the participants in treatment and control groups mirror the breakdown of requirements in the oracle. Treatment group identified more security requirements for each security objective except for UC2-Mobile where control group identified slightly more requirements for integrity as compared to the treatment group.

The treatment group not only identified more security requirements overall but also more security requirements for each security objective as compared to the control group in general. Participants in the control group may consider only the most obvious security requirements (e.g. access control logging of activities) for a given security objective whereas participants in the treatment group are able to consider additional requirements as well. We observed a focus on identifying security requirements related to the objectives of confidentiality and accountability in both healthcare and mobile banking domains. For the healthcare domain requirements related to integrity were among the least commonly identified whereas for the mobile banking domain we did not observe such trend.

8 Feedback from Participants

We solicited feedback from participants in the treatment group on the use of the security requirements templates and the generated security requirements in a post-task survey. The feedback was voluntary for all the studies.

Overall, almost 80 % of the 111 participants in the treatment group provided a favorable opinion related to the use of security requirements templates. We did not get feedback regarding templates from 8 participants (7 % of 111). Of these, six participants did not provide answers for the optional feedback whereas one participant in UT14 and NCSU14 each did not use templates. The participant

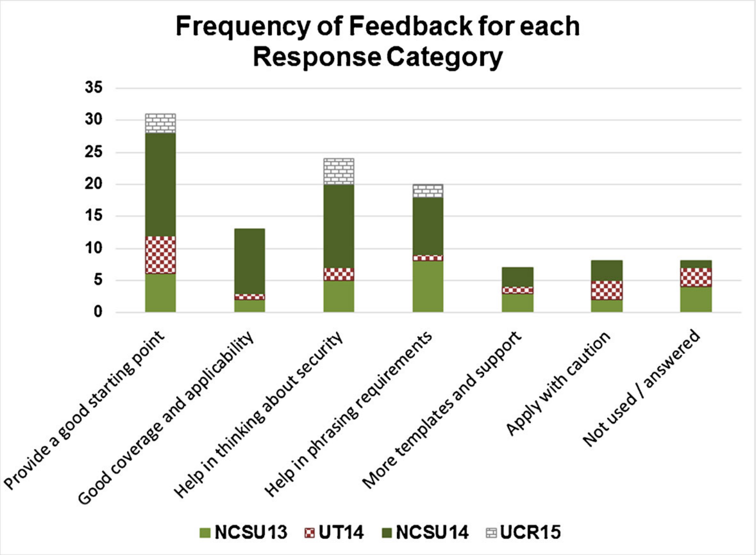


Fig. 9 Feedback related to the use of security requirements templates

in UT14 not using templates still copied the example security requirements generated from templates provided as part of the reference material. The participant in NCSU14 who chose not to use the templates identified security requirements based on keywords in the sentence according to survey response. The participant was the least efficient in the treatment group but not influential on the results. The participant did not provide a reason for not using templates.

Based on the feedback from participants related to the security requirements templates, we identified the following response categories. We provide the frequency of responses for each category in Fig. 9.

- *Provide a good starting point* (*28 %*): provide a direction for developing secure systems; can use the templates and add additional requirements if needed; easy to start defining security requirements with the templates;
- *Good coverage and applicability* (*12 %*): classification based on security objectives is comprehensive and covers all main areas of concern; holistic and ensure that system is analyzed from all perspectives related to security; easy to apply to the given system;
- *Help in thinking about security* (*22 %*): allows users with minimal knowledge to use and understand the templates; helped in thinking about context of the sentences in the use case; helped in considering more security requirements than would have otherwise;
- *Help in phrasing requirements* (*18 %*): saves time by making it easy to type and edit requirements; provided reusable requirements; help in thinking how to go about writing the requirements; helped put forward ideas in formal manner; better if can be auto-filled (e.g., input field for resource, subject, action as these fields are same within a template);
- *More templates and support* (*6 %*): more template choices would be even helpful; good but not exhaustive; auto-filling the templates to generate security requirements will be good;
- *Apply with caution* (*7 %*): not all requirements in the template may be relevant; should not be considered as an exhaustive list; might lead to choosing templates arbitrarily, without carefully thinking about security requirements;

- *Not used / answered* (*7 %*): participant did not provide feedback related to templates.

The templates helped participants think about security and provided a starting point to identify relevant security requirements. Participants considered the six security objectives, which are used to classify the templates, to be holistic in supporting a comprehensive analysis of security concerns. However, the templates should not be viewed as an exhaustive list, as pointed out by some of the participants. Some participants also indicated the need to apply the templates with caution as all templates may not be relevant and availability of templates may lead an individual to choose the templates arbitrarily.

9 Lessons Learned Conducting the Replications

We present some of the lessons learned as part of conducting the replications.

9.1 Communicating with the Original Experimenters

The replications were conducted in close collaboration with the original experimenters. All the experiment material and online tool setup was provided by the original researchers. Moreover, the first two authors evaluated the results for UT14 and NCSU14 replications, as was done for NCSU13. Conversely, one of the original evaluators and researchers from UCR evaluated the responses for UCR15. The first author maintained communication with the researchers conducting the replications over email and voice calls throughout the planning, conducting, evaluating and reporting stages of the experiment. Two of the original researches, including the principal investigator of the original experiment, also met in person with the researchers conducting the replications prior to the conduct of the replications to discuss the research effort behind the original experiment.

Based on our experience, maintaining close communication links is important for successful conduct of replication studies. The researchers need to communicate the context factors associated with the original experiment in detail as well as the tacit knowledge gained by conducting the original experiment. Details about experiment design, context factors and evaluation process should also be reported in research publications. Experiment material and artifacts should be made available for any other researchers who are interested in performing subsequent replications. In addition to communicating the details of the original experiment, feedback and observations from the researchers conducting the replications is valuable in understanding the findings.

9.2 Minimizing Technical Setup

Minimizing the effort needed to setup the experiment supports the replication effort. As observed by Riaz et al. (2015), a replication package, such as the online tool and experiment material provided by the original researchers, is conducive to the conduct of subsequent replications. In our case, the researchers conducting the replications required no technical setup at their end. Each participant needed a computer or laptop to access the online site for participating in the study. On the other hand, if a tool is used to conduct a study, the reliability of the tool is important for the successful conduct of the study. Technical support should be at hand in case problems arise. Moreover, contingency plans, such as availability of offline options to complete the task, should be put in place in case the tool is not available.

9.3 Managing and Reporting Emerging Contexts

Replicating a study, even with all the experimental material and tools available, requires the acquisition of a considerable amount of knowledge. In our case, before conducting the replications, the researchers studied the original experiment, read the additional reference material and tested the tool to familiarize with the environment prior to the conduct of the replications. Despite all the preparation, unexpected events and situations may still arise during the conduct of the study which should be managed and reported.

The replication at UT was initially planned as an in-class activity. However, almost half of the students did not have access to a computer as observed by the instructor at the start of the class. The researchers at UT decided to conduct the study as a take home activity instead. The findings can be interpreted in the context of a take home activity where participants may have more time overall to work on the task but not dedicated time for the task.

For the replication at UCR, almost half of the participants in the control group provided responses in Spanish. Researchers at UCR translated the responses and also participated in evaluating the responses. Researchers conducting replications should be prepared to handle such emerging situations and report the potential impact of changes in the experimental context on the findings of the study.

9.4 Developing Shared Insights

Availability of data from multiple studies with small differences in context factors provides an opportunity to develop insights through qualitative analysis of the combined data, looking at each study's findings in the particular context. Participants in the studies were enrolled in three different graduate courses related to software security and metrics. All the studies were well-integrated with the coursework requirements (Carver et al. 2010) however participants had varying degrees of motivation to perform well. In addition to different participants and courses, we modified the context of the studies in terms of the support in filling the templates as well as the presence of extraneous templates. We were also able to explore the effect of time on task on various metrics. Each group of researchers had unique insights into the study they conducted. By sharing these insights, we addressed additional research questions (RQ5-RQ7) to explore potential reasons for similarities and differences observed across studies.

An open issue is whether results from students' experiments would also be broadly applicable to practitioners. A key observation is that security patterns or similar forms of knowledge sourcing from the community (such as security catalogues or guidelines) are mostly useful to people who are not experts. For example, Gray and Meister's survey of practitioners shows that knowledge sourcing is most sought and most effective when needed in conditions of high intellectual demand w.r.t. users' actual expertise (Gray and Meister 2004). De Gramatica et al. experiments with practitioners show that security catalogues can indeed equalize security experts, without a catalogue, to non-security experts, with a security catalogue (De Gramatica et al. 2015). Therefore, our experiments capture the case of most interest: knowledge sourcing does improve performance of non-experts such as students or junior practitioners in the field. The question remains open whether the performance of experts could be significantly improved by using the security requirements templates. The qualitative evidence from De Gramatica et al. seems to imply that experts and non-expert use knowledge sourcing in essentially different ways. Non-experts may use additional knowledge for finding information while experts may use it as a checklist for noting what may otherwise be forgotten. More experiments would be needed in this respect to develop further insights.

9.5 Working with Diverse Groups of Participants

Participants in the studies were graduate students from three different countries, enrolled in difference courses, and speaking different languages. Different sets of contextual issues, mainly background, previous knowledge and cultural issues, may exist which we did not factor into our results. For instance, due to language differences, researchers at UCR prepared equivalent instructions in Spanish to explain the purpose of the study in the native language to the participants. Moreover, some participants in the control group responded in Spanish and also provided feedback in Spanish which was translated by the researchers at UCR.

As observed by researchers at UCR, students may feel a sense of competition if they are aware that other students in different countries have performed the task of identifying security requirements. Some of the students at UCR also showed interest in knowing about the results across universities at the end of the study. Moreover, participants in larger classes may behave differently than participants in smaller classes. For instance, in NCSU14 with the largest class size, the researchers observed that many participants were confused about the user interface despite the availability of written instructions addressing the questions asked by the participants. Whereas we did not observe such trend in NCSU13.

10 Threats to Validity

We report various threats to validity (Wohlin et al. 2000) that we considered or mitigated during the design and execution of the four studies.

10.1 Internal Validity

Selection The effect of natural variation in human performance can influence the study outcomes. In all the four studies, we used the round-robin assignment approach to randomly assign participants to treatment and control groups, as well as to one of the use cases. Unbalanced groups in terms of participant expertise in the given task could result. However, based on the background information of participants, treatment and control groups were evenly balanced in terms of expertise across all studies.

Instrumentation The effect of experiment artifacts can influence the study outcomes. We observed significant differences in the mean coverage scores for the use cases from healthcare domain (UC1, UC2) used in studies NCSU13, UT14 and NCSU14. UC1 has 110 unique security requirements in the oracle compared to 35 unique security requirements for UC2. Participants identifying security requirements for UC1 would have found a smaller percentage of the total security requirements in the oracle even if they found the same absolute number of security requirements as UC2. However, participants in the treatment group identified significantly more security requirements as compared to the control group, independent of the use cases, in three of the studies as well as in combined analysis, indicating that our findings related to coverage of security requirements (RQ2) still hold.

Diffusion or Imitation of Treatment Three of the fours studies were conducted as an in-class activity where participants did not have the opportunity to share information about various treatments so this threat is minimal overall. Moreover, participants were not aware that they will be assigned to different groups (treatment and control) to further minimize the risk of treatment diffusion and biased responses. However, study UT14 was conducted as a take-home activity and control group participants could have gotten the templates from fellow treatment group participants. We examined the responses by control group in UT14 to see if they resembled closely with treatment responses. However, we did not find any evidence of treatment diffusion across the groups. Some participants in the control group specified security requirements based on the examples available to them while others used their own words, similar to our observation in other studies.

Testing and Training As part of the reference material, we provided example security requirements to the control group generated from the same requirements templates available to the treatment group. Many participants in the control group used the example requirements as a basis for specifying their own security requirements and the examples might have introduced a bias by priming the participants towards certain security requirements. We provided the same examples to the treatment group as well (in addition to the templates), so any potential bias is similar for both groups. Moreover, the process for creating the oracle is similar to the process for identifying security requirements used by the treatment group which may introduce a bias. To minimize the potential bias, we provided control group with examples of security requirements generated from the same templates used to create the oracle.

Interactions with Selection Participants in the treatment group are provided suggestions for applicable security requirements templates. A participant in the treatment group may just blindly accept every suggestion without filling in the templates, or fill in the templates only with the provided suggestions when those are available. In such a case, the participant would be behaving like an automated procedure and may get high coverage and relevance scores without any deliberate effort. This might inherently raise the mean performance of the treatment group in terms of coverage and efficiency metrics. We looked at the participants' responses to see if they behaved in such a manner in our experiments. We found one participant in NCSU14 who had selected almost all the templates. However, the participant had put in effort to fill the templates and spent 10 min above the average time spent by the treatment group indicating that the participant was involved in the activity. Moreover, if we remove this participant's responses from the analysis, the treatment group still performed significantly better than the control group in terms of the coverage of the identified security requirements and efficiency of the elicitation process. Furthermore, in experiments where we suggested extraneous templates, we noticed that most of the participants ignored the extra suggestions, indicating that participants were choosing from the suggested templates rather than including all the suggestions. This threat is thus minimized in our experiments. Experimenters should take this threat into consideration when evaluating participants' responses in any subsequent experiments as well.

10.2 External Validity

Representativeness of Sample Population The sample population should be representative of the population for which we want to draw conclusions based on the study outcomes. Participants in the study were enrolled in four different graduate courses across three different universities in two different continents. In three studies, participants had been exposed to concepts related to security principles, practices and tools. In the fourth study, participants were enrolled in a non-security related course. Participants are fairly representative of the graduate students in computer science. Based on the feedback, about 72 % of the participants had less than 1 year of academic experience related to security and about 85 % had less than 1 year of work experience related to security. Rest of the participants had between 1 and 2 years of academic and work experience with only a handful of candidates having more than 3 years security experience. Participants can be considered representative of entry-level, non-expert software and security practitioners accordingly.

Task Representativeness The task should be representative of how security requirements are identified in practice. Each participant identified security requirements based on a single use case scenario. Additional context for the system and problem domain may help in considering additional security requirements.

Templates Representativeness The templates should be representative of how security requirements are specified for different systems. Through the replications, we have demonstrated the applicability of the templates in identifying security requirements for four different use case scenarios selected from two different domains. Our findings indicate that we may use the templates to identify security requirements for other scenarios and potentially other application domains that have similar security objectives as covered by the templates.

Experimental Constraints that Limit Realism Participants used a limited amount of time to complete the task, which may affect the quality and coverage of identified security requirements. Moreover, participants work individually and may not be able to identify all the applicable requirements in a limited amount of time.

10.3 Construct Validity

Hypothesis Guessing Participants may try to guess the purpose of the experiment which could bias their performance. In the first three studies, participants were not aware of the existence of treatment versus control groups or whether they belonged to different groups. Participants were told only that they were supposed to perform the task of identifying security requirements based on a given use case. Thus single blinding was used to minimize biases towards viewing one requirements elicitation process favorably as compared to the other. However, participants in UCR15 had access to the research paper documenting the original experiment and they may have read the paper prior to the experiment. The problem domain for UCR15 was different than the original experiment so the participants could not have known the answers (i.e., which templates are in the oracle) however they may have known about the hypothesis. This threat is limited to UCR15.

10.4 Conclusion Validity

Reliability of Measures Reliability of measures is an important consideration to draw valid conclusions about the outcomes. When measuring time spent on the task, we automatically recorded every time a participant saved or resumed the task. The recorded timestamps helped us assess the actual time spent on the task at a more granular level, compared to self-reporting by participants. However, the measurement of time for UT14 might not be as accurate as other studies. In UT14, participants performed the task as a take-home activity. We record time based on when a participant opened the task screen and when the participant saved or submitted the task. If a participant opened the task and then switched to something else, we may get a lower efficiency score due to higher time on task recorded. Consequently, if a participant performed the task offline and then copied the responses back to the web page, we may get a higher efficiency score due to lower time on task recorded. We noticed one response in UT14 where efficiency was unusually high. However, removing that response from the analysis does not affect the significance of results for UT14 or the combined analysis. For take-home activities, a better approach would be to use a combination of recording time with the tool and comparing with the self-reported time by the participants. This threat is limited to the assessment of efficiency (RQ4) for UT14.

Fishing and the Error Rate Fishing pertains to the threat of experimenters looking for a specific outcome. For instance, we might actively look for results that support the use of security requirements templates. Ideally, the evaluator should be blind to whether they are evaluating the responses for a participant in the treatment group or control group. However, as is often the case with experiments in software engineering, we could not employ double-blinding when reviewing the participants' responses. Determining whether a participant belonged to the treatment or control group was obvious, since participants in the treatment group used security requirements templates with standardized wording. In contrast, participants in the control group specified requirements in their own words. Care was taken to minimize biases during the evaluation of the responses by devising quantitative measures whenever possible, having multiple independent evaluators and using a standard oracle created beforehand. Some participants in the control group for UCR15 provided response in Spanish that were translated by the researchers to English for the purpose of evaluation. Assessment of quality of responses might be impacted due to the translation (RQ1). However, native Spanish speakers took part in evaluating the responses using the same criteria as the other studies. This threat is limited to the control group in UCR15 and is minimal given the high inter-rater agreement between evaluators using Spanish response and corresponding English translation. In terms of the error rate, we adjusted the error rate for multiple comparison to maintain the desired significance level of results ($p\text{-value} < 0.05$).

Violated Assumptions of Statistical Tests For UT14 and NCSU14, one of the ANOVA assumption related to homogeneity of variance doesn't hold for the metric of relevance (RQ3) due to the large variations in the relevance scores for the control group as compared to the treatment group. In both studies, we found the relevance scores of treatment group to be significantly better than the control group. However, the results for the relevance metric in these two studies may not be considered as reliable as other findings where ANOVA assumptions are met.

Number of Participants For UCR15, we have 16 participants divided in four groups and we may not be able to draw reliable conclusions due to limited number of participants. For instance, we have three participants in the control group for UC2 from the mobile banking domain. If one participant makes a mistake, that amounts to ~33 % of the participants making the mistake. However, the threat is limited to UCR15 and the findings of UCR15 are similar to other studies discussed in this paper having a larger number of participants.

11 Conclusions and Future Directions

We have conducted three differentiated replications of a controlled experiment to evaluate the use of automatically-suggested templates in identifying implicit security requirements as compared to a manual approach without the guidance of templates. We presented information about the security objectives implied by sentences in the given use case and suggested security requirements templates to consider when identifying applicable security requirements. Participants in the treatment group performed significantly better than participants in the control group in terms of the coverage of the identified requirements and efficiency of requirements elicitation process in three of the fours studies. Participants in the treatment group also performed significantly better for the metric of relevance in two studies and metric of quality in one study. In the combined analysis of all the studies, participants in the treatment group performed significantly better than the control group across all the four metrics. We did not find a case where participants in the control group performed significantly better. Overall, participants using templates identified 84 % more requirements and were 57 % more efficient as compared to the control group. Almost 80 % of the participants in the treatment group provided a favorable opinion related to the use of security requirements templates. Providing more dedicated time on task and strong motivation, as in UCR15, led to improvement in the overall quality of elicited requirements. The findings hold for the four different scenarios selected from the domains of healthcare and mobile banking, indicating that the results may be generalizable across other scenarios, potentially from different domains, where security is an important consideration.

The automatically-suggested templates capture the security knowledge of multiple experts and can support the security requirements elicitation process as indicated by the results. However, between 49 and 69 % of the relevant security requirements in the oracle were not identified by the participants across studies. To some extent, this lack of security requirements coverage may be due to limited security expertise of the participants, time and resource constraints, and to the fact that no one individual may identify all applicable security requirements. A few participants in the treatment group also voiced the concern that the templates should not be considered an exhaustive list, as reported in Section 7. Participants may tend to over-rely on the technique and overlook security requirements that are not part of the templates. Despite the relatively low coverage scores, requirements coverage of the treatment group is better than the control group across all the studies. The coverage is significantly better in three out of the four studies as well as in the combined analysis. Putting more effort into identifying a comprehensive set of security requirements templates is also warranted (Riaz et al. 2016) based on our findings.

We only provided a use case scenario as input to the participants as a starting point for identifying the security requirements. However, additional resources such as security policies can also be provided as input to the participants and may guide the identification of applicable security requirements. We are currently designing an industrial case study to evaluate the coverage of security requirements identified by our process for a software system, in comparison to a proprietary approach. The results will provide evidence on how the process generalizes when applied with the help of security analysts without the time and other experimental constraints. Moreover, using our process as complementary to other existing approaches is another direction for future work.

Based on our experience, maintaining close communication links between the original researchers and the researchers conducting the replications is important for successful conduct of the replication studies. Researchers need to communicate the context factors associated with the original experiment in detail as well as the tacit knowledge gained by conducting the original experiment. Moreover, researchers conducting the replications should be prepared to handle emerging situations and discuss the potential impact of changes in the experimental context on the findings of the study.

Our findings underscore the prevailing sentiment that the security expertise is limited and a significant proportion of security requirements are left unidentified due to errors of omission. An underlying objective of these experiments is to assess whether automatically suggesting applicable security requirements for a system based on supervised machine learning is useful in generating a baseline set of security requirements for the system. Based on the findings of our studies, such an automated process, in conjunction with expert analysis, is a viable approach for identifying security requirements for the system.

Acknowledgments This work is partially supported by NSA Science of Security lablet. Fabio Massacci is partially supported by the SESAR Joint Undertaking WP-E EMFASE Project. Christian Quesada-López and Marcelo Jenkins are supported by University of Costa Rica Project No. 834-B5-A18, and Ministry of Science, Technology and Telecommunications (MICITT). Special thanks to Patrick Francis and Patrick Morrison with their help in developing the study oracle. We are thankful to the Realsearch group for their collaboration and helpful comments.

References

Alexander I (2003) Misuse Cases: Use Cases with Hostile Intent. IEEE Softw 20(1):58–66

Braz F, Fernandez EB, VanHilst M (2008) Eliciting security requirements through misuse activities. 4th International Conference on Trust, Privacy & Security in Digital Busines (TrustBus'08), Turin, Italy, September 1–5, 2008, pp 328–333

Carver J (2010) Towards reporting guidelines for experimental replications: a proposal. 1st International Workshop on Replication in Empirical Software Engineering Research (RESER) [Held during ICSE 2010], Cape Town, South Africa

Carver J, Jaccheri L, Morasca S (2010) A checklist for integrating student empirical studies with research and teaching goals. Empir Softw Eng 15:35–59

Carver J, Juristo N, Baldassarre M, Vegas S (2014) Replications of software engineering experiments. Empir Softw Eng 19(2):267–276

De Gramatica M, Labunets K, Massacci F, Paci F, Tedeschi A (2015) The role of catalogues of threats and security controls in security risk assessment: an empirical study with ATM professionals. 21st International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ2015), Springer Verlag, pp 98–114

Fabian B, Gürses S, Heisel M, Santen T, Schmidt H (2010) A comparison of security requirements engineering methods. Requirements Engineering - Special Issue on RE'09: Security Requirements Engineering 15: 7–40

Firesmith DG (2004) Specifying Reusable Security Requirements. J Object Technol 3(1):15

Gray PH, Meister DB (2004) Knowledge sourcing effectiveness. Manag Sci 50(6):821–834

Haley CB, Laney R, Moffett JD, Nuseibeh B (2008) Security requirements engineering: a framework for representation and analysis. IEEE Trans Softw Eng 34(1):133–153

Ito Y, Washizaki H, Yoshizawa M, Fukazawa Y, Okubo T, Kaiya H, Hazeyama A, Yoshioka N, Fernandez E (2015) Systematic mapping of security patterns research. Plop 2015

Karpati P, Opdahl AL, Sindre G (2015) Investigating security threats in architectural context: experimental evaluations of misuse case maps. J Syst Softw 104:90–111. doi:10.1016/j.jss.2015.02.040. Elsevier Ltd

Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01 School of Computer Science and Mathematics, Keele University

Lindsay RM, Ehrenberg ASC (1993) The design of replicated studies. Am Stat 47(3):217–228

McCrum-Gardner E (2008) Which is the correct statistical test to use? Br J Oral Maxillofac Surg 46(1):38–41. doi:10.1016/j.bjoms.2007.09.002

McDermott J, Fox C (1999) Using abuse case models for security requirements analysis. In: Computer Security Applications Conference, pp 55–64

Mead NR, Houg ED, Stehney TR (2005) Security quality requirements engineering (SQUARE) methodology. Technical Report CMU/SEI-2005-TR-009 Software Engineering Institute, Carnegie Mellon University

Mellado D, Fernández-Medina E, Piattini M (2007) A common criteria based security requirements engineering process for the development of secure information systems. Comput Stand Interfaces 29(2):244–253

Mellado D, Blanco C, Sánchez LE, Fernández-Medina E (2010) A systematic review of security requirements engineering. Comput Stand Interfaces 32:153–165

Meneely A, Smith B, Williams L (2012) Appendix B: iTrust electronic health care system case study. Software and Systems Traceability, Springer Verlag, pp 425–438

Menzies T, Dekhtyar A, Distefano J, Greenwald J (2007) Problems with precision: a response to “Comments on ‘data mining static code attributes to learn defect predictors”. IEEE Trans Softw Eng 33(9):637–640

Riaz M, King J, Slankas J, Williams L (2014) Hidden in plain sight: automatically identifying security requirements from natural language artifacts. Requirements Engineering (RE 2014). Karlskrona, Sweden, pp 183–192

Riaz M, Slankas J, King J, Williams L (2014) Using templates to elicit implied security requirements from functional requirements − a controlled experiment. International Symposium on Empirical Software Engineering and Measurement (ESEM), Torino, Italy

Riaz M, Breaux T, Williams L (2015) How have we evaluated software pattern application? a systematic mapping study of research design practices. Inf Softw Technol 65:14–38

Riaz M, Elder S, Williams L (2016) Systematically developing prevention, detection, and response patterns for security requirements. 3rd International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE), Beijing, China

Schumacher M, Fernandez-Buglioni E, Hybertson D, Buschmann F, Sommerlad P (2006) Security Patterns: Integrating Security and Systems Engineering. John Wiley & Sons, Ltd., West Sussex

Sindre G, Opdahl AL (2005) Eliciting security requirements with misuse cases. Requir Eng 10(1):34–44. doi:10.1007/s00766-004-0194-4

Suleiman H, Svetinovic D (2013) Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure. Requirements Engineering 18(3):251–279

Taubenberger S, Jürjens J, Yu Y, Nuseibeh B (2011) Problem analysis of it-security risk assessment methods – an experience report from the insurance and auditing domain. Future Challenges in Security and Privacy for Academia and Industry, pp 259–270

Taubenberger S, Jürjens J, Yu Y, Nuseibeh B (2013) Resolving vulnerability identification errors using security requirements on business process models. Inf Manag Comput Secur 21(3):202–223

Toval A, Nicolás J, Moros B, García F (2002) Requirements reuse for improving information systems security: a practitioner’s approach. Requir Eng 6(4):205–219

Viera AJ, Garrett JM (2005) Understanding interobserver agreement: the kappa statistic. Fam Med 37(5):360–363

Walia GS, Carver JC (2009) A systematic literature review to identify and classify software requirement errors. Inf Softw Technol 51(7):1087–1109

Wen Y, Zhao H, Liu L (2011) Analysing security requirements patterns based on problems decomposition and composition. First International Workshop on Requirements Patterns (RePa), pp 11–20

Withall S (2007) Software requirement patterns. Microsoft Press

Wohlin C, Runeson P, Höst M, Ohlsson M, Regnell B, Wesslén A (2000) Planning. In: Basili VR (ed) Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell, MA

Yoshioka N, Washizaki H, Maruyama K (2008) A survey on security patterns. Progress in Informatics, Special Issue: The future of software engineering for security and privacy (5): 35–47

Yskout K, Scandariato R, Joosen W (2015) Do security patterns really help designers? Proc. of ICSE 2015. IEEE, pp 292–302

Zhang C, Budgen D (2012) What do we know about the effectiveness of software design patterns? IEEE Trans Softw Eng 38(5):1213–1231

Maria Riaz has recently completed her Ph.D. in Computer Science from North Carolina State University, Raleigh, NC, USA. Her research interests include software engineering, software patterns, security and privacy requirements engineering, empirical methods in software engineering and distributed computing. She is currently working as a Software Engineer at Google, Inc.

Jason King is a Teaching Assistant Professor in the Department of Computer Science at North Carolina State University. Jason’s research focuses on software security and user activity logging mechanisms for increasing accountability and forensicability following security or privacy breaches. He received his PhD in Computer Science from North Carolina State University.

John Slankas earned his PhD from North Carolina State University in 2015. His dissertation focused on techniques to extract access control rules from natural language texts. He continues to work at NCSU as a researcher studying a variety of machine learning approaches for information extraction.

Laurie Williams is the Associate Department Head of Computer Science and a Professor in the Computer Science Department at North Carolina State University (NCSU). Laurie is a co-director of the NCSU Science of Security Lablet. Laurie's research focuses on software security; agile software development practices and processes; and software reliability, software testing and analysis.

Fabio Massacci has a PhD from the University of Roma La Sapienza. He has been at Cambridge University, University of Siena and IRIT Toulouse and he is now full professor in Security at the University of Trento. He is a member of ACM, IEEE, ISACA, SRA and a chartered engineer. His research interests are at the crossroads between empirical computer security, malware and vulnerability prediction, security economics, and risk analysis. He was the coordinator of the multi-disciplinary SECONOMICS projetc on security and economics and the EMFASE project on empirical methods for security risk assessment.

Christian Quesada-López is a doctoral candidate in the Department of Computer Science at the University of Costa Rica, where he also holds a teaching and research position. He has been continuously involved in empirical software engineering research since his graduate master's thesis, in the Center for ICT Research (CITIC) at University of Costa Rica. Prior to his doctoral work he had a 10 year career in software product development, software project management, and software quality assurance. His main research interests are empirical software engineering, measurement, and software quality assurance. Contact him at: cristian.quesadalopez@ucr.ac.cr http://www.citic.ucr.ac.cr/perfil/cristian-quesada-lópez.