Evaluating Legal Implementation Readiness Decision-Making

Aaron K. Massey, *Member, IEEE*, Paul N. Otto, *Member, IEEE*, and Annie I. Antón, *Senior Member, IEEE*

Abstract—Software systems are increasingly regulated. Software engineers therefore must determine which requirements have met or exceeded their legal obligations and which requirements have not. Requirements that have met or exceeded their legal obligations are *legally implementation ready*, whereas requirements that have not met or exceeded their legal obligations need further refinement. In this paper, we examine how software engineers make these determinations using a multi-case study with three cases. Each case involves assessment of requirements for an electronic health record system that must comply with the US Health Insurance Portability and Accountability Act (HIPAA) and is measured against the evaluations of HIPAA compliance subject matter experts. Our first case examines how individual graduate-level software engineering students assess whether the requirements met or exceeded their HIPAA obligations. Our second case replicates the findings from our first case using a different set of participants. Our third case examines how graduate-level software engineering students assess requirements using the Wideband Delphi approach to deriving consensus in groups. Our findings suggest that the average graduate-level software engineering student is ill-prepared to write legally compliant software with any confidence and that domain experts are an absolute necessity.

Index Terms—Legal implementation readiness, regulatory compliance software engineering, legal requirements, requirements engineering

1 Introduction

Software systems are increasingly becoming regulated. The engineers who design, build, and deploy these systems have an ethical obligation to understand the laws and regulations to which they must comply. Unfortunately, understanding the law is not easy. Even the concept of the law is missleading because there is no single, objective understanding of the law. Asking a lawyer to explain the law is like asking a philosopher to explain what truth is: all that can be provided is an argument in favor of or against a position. Laws and regulations must be interpreted, whether by lawyers litigating a case, judges discerning a case in court, or by engineers building software for regulated domains. Engineers may be the first interpreters of laws and regulations that apply to software systems, but little research has been done to examine how software engineers interpret laws and regulations.

In this paper, we examine the processes by which software engineers make decisions about whether a requirement meets or exceeds its legal obligations. A requirement is considered to be legally implementation ready (LIR) if it meets or exceeds its legal obligations as expressed in relevant regulations. We conducted a multi-case study to examine how legal implementation readiness decisions are made. This study employs a set of 31 Electronic Health Records (EHR) system requirements, coupled with a regulatory section of the US Health Insurance Portability and Accountability Act (HIPAA)[1] to which the system must comply. In each case, we ask participants to determine whether or not the set of software requirements provided are LIR with respect to a provided subset of HIPAA. Participant responses are compared against a canonical set of answers developed by three subject matter experts—individuals with both software engineering and HIPAA compliance expertise—who evaluated the same set of requirements for legal implementation readiness.

Each case in this study focuses on the same set of 31 software requirements. These requirements describe a portion of the iTrust Medical Records System,[2] an open-source EHR system designed by faculty and staff at North Carolina State University. iTrust has been in continual development since 2004. The project is designed to provide students with realistic challenges in software engineering: working within non-trivial code bases, managing information security and privacy issues, and using modern integrated development and testing environments and tools. The iTrust system shares many characteristics with other existing software systems that must comply with new laws and regulations. Such systems must be evaluated, updated, and improved in order to achieve compliance with each enactment or amendment of relevant laws and regulations.

Our multi-case study consists of three cases. In our first case, 32 graduate students in software engineering made LIR decisions individually for each requirement. For our second case, 34 different graduate students in software

- *A. K. Massey is a Postdoctoral Fellow at the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: akmassey@gatech.edu.*
- *P.N. Otto is with the Association of Computing Machinery, District of Columbia, 555 13th St. NW, Washington, DC 20004. E-mail: potto@acm.org.*
- *A.I. Antn is Professor and Chair of the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: aianton@cc.gatech.edu.*

Manuscript received 30 Jan. 2014; revised 4 Nov. 2014; accepted 16 Nov. 2014. Date of publication 17 Dec. 2014; date of current version 17 June 2015. Recommended for acceptance by L. Williams.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSE.2014.2383374

1. Pub. L. No. 104-191, 110 Stat. 1936 (1996).
2. http://agile.csc.ncsu.edu/iTrust/wiki/doku.php

0098-5589 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

engineering made LIR decisions individually for the same set of requirements. This second case was conducted to replicate the results of our first case. For our third case, 14 graduate students in software engineering made LIR decisions as a group using Wideband Delphi to derive consensus.

The software engineering literature is almost silent with respect to case studies of professional software engineers performing regulatory compliance tasks. To our knowledge, Maxwell conducted the first and largest study of this type [1]. His findings indicate that practitioners with a median industry experience of over 12 years were only able to accurately classify 5.5 legal cross-references out of 10, even when provided with a taxonomy designed to aide them in this task [1]. Furthermore, participants in Maxwell's pilot study, who had a median industry experience of fewer than two years, out-performed the practitioners and accurately classified seven legal cross references out of 10 [1]. This finding suggests that education and training specifically focused on legal requirements activities may prove more valuable than industry experience for legal requirements analysis activities. In this study, we examine how graduate students trained in legal requirements analysis make legal implementation readiness decisions. Although we did not study practitioners, it is not clear that practitioners would be better suited to this task. In fact, Maxwell's research suggests education and training in regulatory compliance for software engineering is more valuable than industry experience when performing regulatory compliance tasks [1].

Results from our first and second cases indicate that graduate-level software engineering students exhibit little consensus about LIR decisions. To our knowledge, our research is the first work to examine how people make legal implementation readiness decisions. Examining graduate student performance supports our goal of establishing a baseline for future work. Individual participants in our study tended to err on the side of determining a requirement to be LIR when it was not. Errors of this nature may result in legal compliance concerns in the implemented software. These results validate the need for subject matter expert involvement or other additional guidance for making legal implementation readiness decisions. Results from our third case indicate that when participants are required to come to a consensus LIR decision for each requirement they are only slightly more accurate than when making these decisions alone. Participants continued to err on the side of determining a requirement to be LIR when it was not. These results indicate that graduate-level software engineering students are ill-prepared to accurately make these decisions. When combined with Maxwell's work, these studies demonstrate the need for new tools and techniques for establishing regulatory compliance in software engineering.

The remainder of this paper is organized as follows. Section 2 describes related work in regulatory compliance for software engineering. Section 3 provides an overview of the methodology used in our multi-case study. Sections 4, 5, and 6 detail the application of our methodology to each case and provide results. Section 4 describes our initial examination of LIR decision making. In Section 5, we replicate our results from the first case. Section 6 evaluates LIR decision making as a group rather than as individuals. In Section 7, we describe the threats to validity and limitations of this work. Finally, we summarize our results in Section 8.

2 Related Work

Bringing software into legal compliance is an important and challenging software engineering concern [2]. Ambiguities, cross-references, the need to comply with multiple regulations, the need to deploy software systems in multiple jurisdictions, and a rich set of domain knowledge make legal compliance particularly challenging for requirements engineers without legal training [2]. In addition, amendments from administrative agencies or interpretations from judicial proceedings may change laws and regulations years after their initial passage [2]. The US Department of Health and Human Services (HHS) enforces HIPAA. Legal norms suggest that HIPAA regulations could be revised as often as every year [3].

Yin et al. developed Eros, which is an approach to examine business process outcomes, rather than software requirements or other software artifacts, for compliance with relevant laws and regulations [4]. This approach is similar to checklists that focus on evaluation of a final product against a law or regulation; such checklists are established tools for establishing legal compliance of software systems [3]. Although focusing on outcomes is a useful approach for measuring compliance of an existing software system, the research presented herein focuses on decisions made well before outcomes can be assessed.

Many researchers have developed approaches to modeling laws and regulations for software systems. Barth et al. employ a first-order temporal logic approach to model HIPAA, the Childrens Online Privacy Protection Act (COPPA),[3] and the GrammLeachBliley Act (GLBA)[4]—they use this model to demonstrate legal compliance [5]. May et al. employ an access control approach to model laws and regulations, also using HIPAA as an example [6]. Massacci et al. use goal-oriented modeling to analyze Italian data protection laws [7]. Jureta et al. proposed an approach to benchmarking legal requirements models with the goal of establishing more accurate comparisons of legal requirements modeling techniques [8]. Although promising, benchmarks for legal requirements modeling remains nascent. The utility of these modeling techniques depends on the relative ease with which software engineers are able to make legal compliance decisions for software systems. In this paper, we examine one such decision: legal implementation readiness.

Kharbili et al. developed a more recent approach towards modeling requirements called CoReL [9], [10]. CoReL is a domain-specific modeling language for representing compliance requirements [9], [10]. The goal of CoReL is to provide a more user-friendly approach to compliance checking and legal requirements traceability [9], particularly with respect to business process management [10]. Interestingly, CoReL allows users to document both possible violations and intentional violations of policies [10]. As this work demonstrates, legal implementation readiness decisions are non-trivial. The ability to document decisions is, therefore, valuable.

3. Pub. L. No. 105-277, 112 Stat. 2681 (1998).
4. Pub. L. No. 106-102, 113 Stat. 1338 (1999).

Hassan and Logrippo developed a logic-based approach to compliance checking for requirements based on Alloy [11], [12], [13]. Their most recent work includes a model for extracting legal requirements from legal texts, a set of rules to translate those requirements into a formal logic, and tool support to analyze the resulting logical model for compliance [13]. Although this research provides a reasonably comprehensive methodology for establishing compliance in legal requirements, it does not examine how people make decisions when asked to consider regulatory compliance for software requirements. We believe studies of how people make requirements compliance decisions are both important and too rarely conducted.

Maxwell and Antón employ production rules, an artificial intelligence technique, to model regulations for requirements engineers and check requirements for compliance [14]. In their work, requirements engineers query a production rules model and receive specific answers to questions regarding a modeled regulation [14]. Maxwell and Antón also use the HIPAA regulations to illustrate their approach and trace requirements directly to elements of the legal text [14]. Tracing requirements directly to specific elements of the legal text is a required prerequisite to our approach in this work.

Ghanavati et al. developed and examined methods for integrating requirements compliance with goal-oriented requirements engineering techniques [15], [16]. Early work in this area provided an approach to model software artifacts that contributed to compliance, focusing on scenarios where a single artifact alone could not establish compliance [15]. Recently, Ghanavati et al. have begun examining requirements engineering modeling and analysis for systems that must comply with multiple regulations [17], [18], [19]. In this paper, we examine legal implementation readiness, which may benefit from an understanding of compliance contributions. A set of requirements that meet or exceed their legal obligations, are not necessarily meeting all legal obligations. The identification, elicitation, specification, and analysis of all legal requirements, including those from more than one legal text, are all activities outside the scope of this work, but a complete requirements compliance process must take these considerations into account.

Software systems are often regulated in multiple jurisdictions, and researchers have begun investigating the impact of this reality on requirements engineering. Gordon and Breaux developed an approach in which they adopt a single standard for software systems that meets the minimum or maximum requirements of a collection of laws and regulations in multiple jurisdictions [20], [21]. Their framework also highlights potential conflicts and important trade-offs resulting from multiple laws and regulations. Ingolfo and Silva Souza developed an adaptive approach to legal requirements engineering intended to provide flexibility to requirements for systems that must comply with multiple legal texts in multiple jurisdictions [22]. Rifaut and Ghanavati support measurement-based methodologies for comparing multiple regulations using the Goal-Oriented Requirements Engineering Language (GRL) [19]. We consider multi-jurisdictional concerns to be out of scope for the work presented herein. Our goal is to produce a baseline understanding of legal implementation readiness decision-making, and consideration of multiple laws or multiple jurisdictions adds unnecessary complexity for establishing a baseline. However, compliance in multiple jurisdictions is certainly a concern for practicing software engineers, and determining how implementation readiness decisions are made in these instances is an area for potential future work.

Researchers are also exploring automated processes for generating traceability links for regulatory compliance [23], [24], [25]. Cleland-Huang et al. use a machine learning approach to automatically generate traceability links for regulatory compliance [24]. However, their links trace from a requirement to an information technology responsibility identified in the law rather than to a specific section of the legal text itself. Berenbach et al. describe techniques for just-in-time regulatory traceability [23]. They intentionally trace requirements to higher level segments of the legal text rather than to individual sections [23]. To complete the process, a requirements engineer must be involved manually [23]. Breaux developed a frame-based approach to eliciting legal requirements from legal texts [25]. As a part of this elicitation, requirements can be traced to specific elements of the text from which they were elicited. Breaux and Gordon developed an approach to preserve meaning and ensure traceability during extraction of legal requirements from source documents [26]. These approaches may reduce the effort required to trace requirements to specific sections of a legal text, which is required to conduct the multi-case study described in this paper.

3 Case Study Methodology

The experiment design for our multi-case study employs the Goal/Question/Metric (GQM) approach [27], [28], which is based on the idea that measurement is useful for making intelligent decisions and improving those decisions over time, but that measurement must be focused, and based on goals and models. The GQM approach proposes that every measurement activity has one or more goals. Each goal must then have at least one question that helps achieve the goal. Finally, each question is answered using one or more measures or metrics. Our research goal as formulated using the GQM template is as follows:

> *Goal*. Analyze empirical observations for the purpose of characterizing legal implementation readiness with respect to software requirements from the viewpoint of software engineers in the context of an EHR system that must comply with HIPAA regulations.

Given this research goal, we formulate the following research questions:

- *Q1*. Is there a consensus among subject matter experts on which requirements are LIR?
- *Q2*. Is there a consensus among software engineering graduate students on which requirements are LIR?
- *Q3*. Can graduate students accurately[5] assess which requirements are LIR?

5. Objectively accurate assessment of LIR requirements is impossible due to the inherently subjective nature of interpreting laws and regulations. Thus, we compare against a canonical set of requirements created by experts as described in Section 3.2.

In Section 3.1, we describe the materials for our research experiment. In Section 3.2, we describe how we created our canonical set of correct LIR responses by achieving consensus among experts. In Section 3.3, we discuss our analysis methodology and the statistical tools we used in our multi-case study. We introduce our participant populations and measures for answering our research questions on a case-by-case basis as described in Sections 4, 5, and 6.

3.1 Materials

Each participant for each case in our multi-case study received three inputs:

Legal text. HIPAA 45 C.F.R. Section 164.312
Requirements. 31 iTrust requirements and a glossary of terms.
Traceability mapping. A traceability mapping of the individual requirements to the legal text.

These same three inputs were provided to our subject matter experts (described in Section 3.2) used to formulate the canonical responses against which each case was measured.

The legal text chosen for this experiment is HIPAA 45 C.F.R. Section 164.312, which governs technical safeguards. We selected this section for three reasons: First, it is a part of HIPAA, with which we have extensive experience [14], [29], [30], [31]. Second, it is focused on technical measures for protecting healthcare information, with which software engineers are more likely to be familiar than a less technical or purely legal section of HIPAA. Thus, if there is a gap between the results from the experts' canonical classification and that of the graduate students, we can reasonably infer that this gap would widen if we were to perform the experiment again on more ambiguous or less technically oriented HIPAA sections.

Third, we sought to identify a concise, self-contained section of HIPAA. This would allow us to test it in its entirety rather than excerpting and testing a longer HIPAA section. Cross-references, which are common in legal texts, make identifying useful, self-contained sections of HIPAA challenging. Although Section 164.312 does contain two cross-references, no other technically-oriented, reasonably short section of HIPAA contains as few cross-references as it does. The first cross-reference appears in the preamble of the text and is not traced to any requirements. The second clarifies a process by which persons or software programs are granted access to protected health information. For some requirements, the details of this process may be relevant. For others, they may not be. We allowed participants to use the existence of the cross-referenced material to influence their determination. This mirrors real-world compliance scenarios for software engineers, where legal cross-references are a fundamental problem that may make a complete examination of all cross-references impractical [30].

For purposes of our experiment, we modified the iTrust requirements specification. Instead of including all iTrust system requirements, we started with the 15 requirements with legal obligations traced within HIPAA Section 164.312. Then we iteratively applied the methodology for evaluating requirements for legal compliance outlined in our prior work [31] to the other iTrust system requirements. After three iterations, we identified an additional 17 requirements for inclusion in our experiment. We selected one of our 32 requirements to be used as a part of the tutorial for providing basic training to participants about how software engineers may reason about legal compliance. As a result, we had 31 remaining requirements to use in our experiment.

> **R-8:** When patient account information is altered, iTrust shall email the personal representatives of the affected patients with a description of the alterations made. [Traces to § 164.312(b)]

Fig. 1. iTrust requirement 8.

To ensure that some requirements were explicitly not LIR, we chose not to cover all elements of HIPAA Section 164.312 and included some requirements that only partially covered the elements of the legal text. Additionally, we modified our selected requirements to remove terminology and references to other aspects of iTrust not relevant to HIPAA Section 164.312. Any remaining terminology was provided to the participants in the form of a glossary. The materials also included a traceability mapping of the 31 selected requirements to the specific section(s) of HIPAA Section 164.312 to which they applied. We constructed the traceability links iteratively using the techniques outlined in our prior work [31], stopping when we had enough requirements for the study. Had we continued applying our iterative approach to evaluate and refine these requirements, we would have removed any requirements that were not ready for implementation.

Our goal in creating these input materials was not to create a set of requirements that were already perfectly aligned with the law; instead, we sought to examine a variety of legal compliance situations ranging from relatively easy to relatively challenging decisions while maintaining as much realism as possible. For example, consider iTrust Requirement eight shown in Fig. 1. This requirement is in almost direct conflict with the traced section of HIPAA, which calls for audit controls for recording and examining changes of this nature. Even a basic understanding of email would allow an engineer to determine that this requirement is not ready for implementation. However, the requirement is not completely unrealistic. It describes functionality similar to many web-based applications. If you update your billing address or other account information on many websites, you will receive an email notification detailing the changes.

The traceability mapping used in this study was constructed using the complete procedure describe in our prior work [31]. This procedure begins with a directed terminology mapping to normalize the terms used in both the legal text and the requirements document to ensure that only one term is used to refer to a particular concept. This is accomplished through the creation of separate hierarchies for both actors and data objects. Once the terminology mapping is completed, the requirements are elaborated and clarified to use the new, normalized terms. Finally, the clarified requirements are mapped to the specific sections of the legal text. It is worth noting that the traceability mapping strongly influences LIR determinations. All participants in this study, including our experts, were told to accept the mapping provided as correct for the purposes of this study.

3.2 Subject Matter Experts

Canonical sets of LIR requirements can be used to check other requirements sets for legal compliance [32]. A canonical set of requirements is a set of requirements evaluated by experts and found to be acceptable for some purpose [32]. The expert assessment is accepted as correct and it is a useful for comparative purposes when no absolute or objective standard is possible. In this research, our canonical set is a set of requirements for which experts have determined whether each requirement is legally implementation ready. To generate a canonical set of LIR requirements, we recruited three subject matter experts: the three authors of this paper. All three experts have varying years of academic experience with HIPAA legal compliance in software engineering; additionally, one expert (Paul Otto) has a law degree.

We employed the Wideband Delphi method [33] to determine the consensus expert assessment regarding which requirements were LIR and which needed further refinement. First, we gave each expert the materials described in Section 3.1. Next, we asked the experts to individually identify both the LIR requirements and those needing further refinement, recording their rationale for each decision. This individual analysis was used to address our first research question. Third, we held a coordination meeting in which the experts discussed areas of disagreement and worked to arrive at a consensus for each requirement. Their final, consensus results serve as the canonical LIR requirements set against which we compare responses from the graduate-level computer science participants in both the first and second case studies and the legal requirements triage algorithm.

3.3 Analysis Methodology

We take two basic approaches in our analysis: (1) statistics calculated individually for each participant as compared to the consensus expert results and (2) statistics calculated based on a calculated consensus participant assessment compared to the consensus expert assessment. We are particularly interested in assessing how effective participants are at correctly identifying requirements as not LIR. When software engineers incorrectly believe a requirement is LIR and begin implementation, they risk building a non-compliant system. We also examine how effective participants are at correctly identifying requirements that are LIR. When software engineers incorrectly believe a requirement needs further refinement, they risk over-engineering, which may result in unnecessarily increased system complexity or unnecessary implementation, deployment, and maintenance expenses. We used the R Project,[6] a statistical modeling package, to calculate statistics for the data in our study.

The performance of a participant to classify things as being one of two possible options is often evaluated using two measurements: sensitivity and specificity [34]. Sensitivity measures the ability to predict positives. In our case, this is a measurement of how many LIR requirements participants classify as not LIR. Sensitivity is defined in Equation (1), where tp is the number of true positives and fn is the number of false negatives,

$$Sensitivity = \frac{tp}{tp + fn}. \tag{1}$$

Specificity measures the ability of participants to predict negatives. In our case, this is a measurement of how many non-LIR requirements participants classify as LIR. Specificity is defined in Equation (2), where tn is the number of true negatives and fp is the number of false positives,

$$Specificity = \frac{tn}{tn + fp}. \tag{2}$$

Our choice of sensitivity and specificity is based on the fact that a false negative (identifying a requirement as needing refinement when it is actually LIR) has a low penalty whereas a false positive (identifying a requirement as LIR when it actually needs further refinement to meet its legal obligations) has a high penalty. Precision and recall [34] are similar statistical tools that may be better estimators in some situations, but these statistics treat false positives and false negatives as equally problematic.

We use two statistics to measure agreement of participants with the consensus expert assessment: percent agreement and Cohen's Kappa. Percent agreement is the percentage of requirements where the participant assessment matches the consensus expert assessment. Cohen's Kappa is a measure of inter-rater agreement for two raters that attempts to account for agreement occurring by chance [35]. Cohen's Kappa is considered to be a more robust measure of agreement than percent agreement [35]. Values of Cohen's Kappa range from -1 to 1 where the value -1 represents perfect disagreement and the value 1 represents perfect agreement. Every value between 0 and 1 reflects some level of agreement, with larger values indicating more agreement.

The Fleiss Kappa statistic [36], κ, measures level of agreement between raters on a given set of subjects. Fleiss Kappa is an extension of Cohen's Kappa for groups of raters larger than two. We employ the Fleiss Kappa statistic to calculate the level of agreement among subjects' determinations about whether a requirement is LIR or not. As with Cohen's Kappa, the Fleiss Kappa statistic ranges from -1 to 1. The value 1 reflects perfect agreement between all raters, the value -1 indicates perfect disagreement, and the value 0 indicates the amount of agreement that would be expected by random chance.

We formed a consensus using Wideband Delphi [33] for the three subject matter experts. In Wideband Delphi, consensus is formed in rounds with discussion at the end of each round. First, each participant provides answers independently and records their rationale for the answers they have chosen. Next, participants share their answers and reconsider their positions. Then, participants meet to achieve a final consensus. Although we completed the Wideband Delphi technique to produce our canonical set of LIR requirements, we also chose to analyze the results of our independently recorded first analyses to determine the level of agreement that existed among the subject matter

6. http://www.r-project.org/

experts at the beginning of the process. We also employ Wideband Delphi in our third case study as described in Section 6.

4 First Case

Sections 4, 5, and 6 describe the exact methodology and results of each case in our study. These sections also provide some discussion of the results from each case. In this section, we also describe the process used to develop our canonical evaluation of the requirements using Wideband Delphi to derive consensus for our subject matter experts. The canonical, expert assessment developed in this section is used to evaluate the correctness of responses for all three cases. Our threats to validity are described separately in Section 7.

4.1 Software Engineering Participants

The participants in our experiment were computer science graduate students who had taken or were taking the graduate-level software engineering course at North Carolina State University. Through their coursework, we knew that they were familiar with the basic practices and responsibilities of a software engineer working as a part of a larger team. For this case study, conducted during the Spring semester of 2011, we had 32 participants.

Before participating in the experiment, all participants received lessons in requirements engineering. These lessons consisted of two class sessions specifically devoted to legal compliance concerns in software requirements for systems that must comply with privacy regulations, including HIPAA. Immediately prior to beginning the study, participants also received a 15 minute tutorial on legal compliance in software systems. This tutorial introduced some basic concepts in legal compliance for software requirements. It consisted of an explanation of the importance of legal compliance, as well as an explanation of the traceability mapping of requirements to the legal text and how that mapping might be useful in evaluating legal compliance. The tutorial included the example legal compliance scenario outlined in Fig. 2.

After explaining this example to the participants, we verbally described another potentially conflicting requirement that is a variation on Requirement B. We asked the participants to consider whether Requirement B would become LIR if we added the phrase "so long as the user ID remains unique." After allowing the participants to consider this for a few moments, we showed that it would be possible for a user to adopt a previously discarded user ID as their own. In this situation, a user could change their ID to something totally unique, but their old ID would then become available for another user to use as their ID. This could result in access logs that have a single user ID that represents two separate users, which would be a clear violation of the legal obligations stated.

4.2 Results of First Case

First, we describe reactions and questions from our participants. Next, we discuss some of the points of disagreement found during our consensus meeting for the subject matter experts. Then, we analyze the data according to our analysis methodology.

Consider Requirement A:

Requirement A: iTrust shall generate a unique user ID and default password upon account creation by a system administrator. [Traces to § 164.312(a)(1) and § 164.312(a)(2)(i)]

Here are the two subsections of the legal text to which Requirement A is traced:

(a) (1) Standard: Access control. Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in § 164.308(a)(4).

(2) Implementation specifications: (i) Unique user identification (Required). Assign a unique name and/or number for identifying and tracking user identity.

Requirement A meets or exceeds its legal obligations outlined in § 164.312 because no element of the regulation related to the requirement describes different or additional obligations than those described in the requirement.

In contrast, consider Requirement B:

Requirement B: iTrust shall allow an authenticated user to change their user ID and password. [Traces to § 164.312(a)(1) and § 164.312(a)(2)(i)]

Note that Requirement B traces to the same two subsections of legal text as Requirement A. Requirement B does not meet or exceed its legal obligations outlined in § 164.312 because it describes a situation where it is possible for users to end up with the same identifying name or number, which violates § 164.312(a)(2)(i).

Fig. 2. Example scenario from case study tutorial.

4.2.1 Participant Experiences

We conducted our experiment with 32 graduate students. The majority (21) of these participants completed the experiment in the same room at the same time. The remaining participants completed the experiment individually or in small groups over the course of the next week. Participants were trained using the same training materials, which took about five minutes to introduce and explain at the beginning of the experiment. Each participant was then given 45 minutes to analyze 31 requirements and determine whether each requirement was LIR or not. Training was conducted in a group setting when possible, but all participants worked on their own during the 45 minute analysis portion of the experiment.

We answered clarifying questions during each session, and for each question asked we repeated the question and the answer for every participant. Because the first session was the largest group to participate, we assembled all of the questions that could not be answered by referring to other material within the requirements specification or legal text provided to the participants; we then verbally discussed those questions and corresponding answers to subsequent groups as part of their training material. For example, a question about the

R-18: Whenever a new employee account is created, iTrust shall create a record of the employee account created and the system administrator who created it in the access control log. [Traces to § 164.312(b)]

Fig. 3. iTrust requirement 18.

R-26: Each time a printable emergency report for a medical record is generated, iTrust shall email a notification to the patient to whom the record pertains and to all of that patient's personal representatives. [Traces to § 164.312(b)]

Fig. 5. iTrust requirement 26.

definition of an invalid employee account was not repeated because it is listed as a term in the glossary provided. However, a question about whether or not an email would be considered part of an access control log (answer: it is not) was included in subsequent training material.

We did not answer substantive questions about legal compliance or interpretation of the software requirements. We instructed the participants that these were the sorts of questions the study was designed for them to consider when making their own assessment of the legal implementation readiness of the requirements. Although such a response provided the participants with some information, namely that the question asked was indeed substantive, we felt our only alternative would have been to forbid any questions about the study materials. Such an approach would not have allowed us to respond to basic clarifying questions about the materials and procedure. Refusing to answer substantive questions about legal compliance or interpretation of the software requirements, and thus indicating that the question was indeed substantive, was an acceptable trade off for us to ensure participants fully understood their task.

4.2.2 Subject Matter Expert Discussion

The first step in the Wideband Delphi consensus process involves each participant individually determining whether each requirement is LIR. These initial determinations are used as the starting point for the first group consensus meeting. However, we also examined the level of agreement in those initial determinations, as described in Section 3.2, and found that our subject matter experts entered the group consensus meeting already having achieved a moderate level of consensus based on their individual responses. However, there were still 12 out of 31 requirements for which the subject matter experts did not achieve universal agreement after the first round of Wideband Delphi.

In some cases, two subject matter experts immediately accepted the rational used by one subject matter expert to denote a non-LIR requirement. Consider Requirement 18, as described in Fig. 3. Note that HIPAA Section 164.312(b) is shown in Fig. 4.

Two of the subject matter experts believed that this requirement met or exceeded its legal obligations. However, one subject matter expert believed that a record should also be created in the access control log for each unsuccessful attempt to create a new employee account as well as the successfully created employee accounts, pursuant to the broad language of HIPAA Section 164.312(b). The other experts agreed with this rationale, and since no other requirement in the requirement set describes this activity, R-18 was found to be in need of further refinement in the canonical requirements set.

In other cases, the subject matter experts took some time to debate whether or not a requirement should be considered LIR. In iTrust, any employee is allowed to generate a printable summary of a patient's medical record for the purposes of handling emergency medical scenarios. Part of this functionality is described by Requirement 26, shown in Fig. 5.

This requirement traces to the same section of HIPAA as our previous example, but the discussion of whether or not it is a LIR requirement is more nuanced. The experts debated several considerations: First, they determined that email notification does not qualify as a record in an access control log. Since R-25 describes the creation of such a record in the exact same printable emergency report scenario, R-26 could not be found in need of this refinement. Second, the experts debated whether or not an email notification qualified as a "procedural mechanism that records or examines activity" as prescribed by HIPAA Section 164.312 (b). On one hand, email notification may allow a patient some oversight through the ability to examine the activity of generating a printable emergency report. On the other hand, email notification does not ensure that a patient will examine the activity. This second position is a broader interpretation of HIPAA Section 164.312(b), and the experts agreed that assuming the broader standard was the safer course of action to ensure legal compliance. Third, the subject matter experts determined that R-26 should not be traced to HIPAA Section 164.312(b) because notification is a separate action from either "recording" or "examining activity." Ultimately, the experts reached consensus that R-26 is LIR because it does not describe any action that violates any legal obligation.

Standard: Audit controls. Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information.

Fig. 4. HIPAA Section 164.312(b).

4.2.3 Data Analysis

We now discuss the results of our first case for each of our research questions identified in Section 3.

Q1. Is there a consensus among subject matter experts on which requirements are LIR?

Measure. Fleiss Kappa statistic for three subject matter experts results requirements.

The Fleiss Kappa statistic for our three experts was $\kappa = 0.517$ ($p < 0.0001$). This result indicates that with a high level of statistical significance, our experts moderately agreed on their first assessment of the 31 requirements

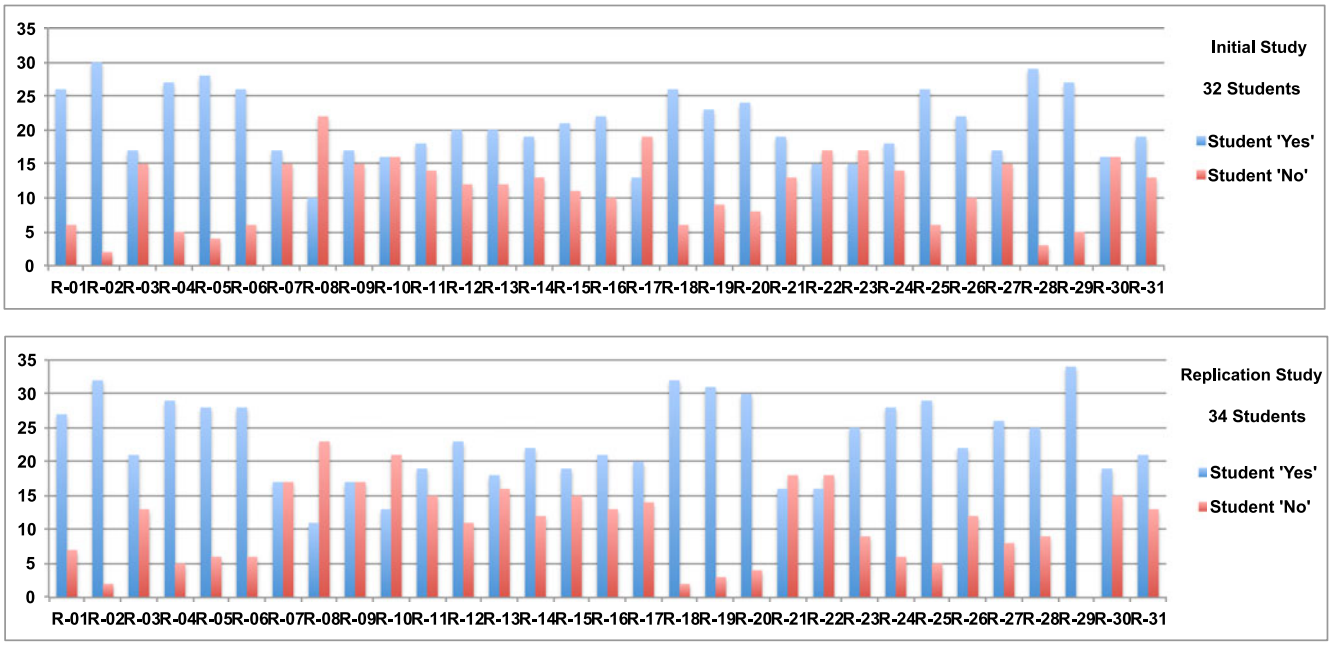

Fig. 6. Raw student responses regarding legal implementation readiness.

before completing the Wideband Delphi session to reach consensus.

Q1 Answer. There is a moderate consensus among experts regarding LIR.

Q2. Is there a consensus among graduate students on which requirements are LIR?

Measure. Fleiss Kappa statistic for 32 graduate students on 31 requirements.

The Fleiss Kappa statistic for our 32 students was $\kappa = 0.0792$ $(p < 0.0001)$. This result indicates that with a high level of statistical significance, the students in our case study had only a slight agreement on their assessment of the 31 requirements. Because Fleiss Kappa accounts for random chance, this level of agreement is only slightly better than what would be expected from a random set of responses. Due to the high level of significance in both Kappa scores for Q1 and Q2, we can say that there was a higher level of consensus among experts than students.

Fig. 6 displays the responses for the students in the LIR Assessment study on the top bar graph, labeled "Initial Study." (The bottom bar graph, labeled "Replication Study," displays the data from our replication study, which is discussed in Section 5.) The vertical axis represents the number of students. Students that indicated a requirements is LIR (i.e., a "Yes" response) with a blue bar or not LIR (i.e., a "No" response) with a red bar. The horizontal axis represents the requirement for which a decision was made. Note that every requirement received some "Yes" responses and some "No" responses. Also, the majority of requirements have more "Yes" responses than "No" responses.

Q2 Answer. There is little consensus among graduate students regarding LIR.

Q3. Can graduate students correctly assess which requirements are LIR?

Measures. Sensitivity, specificity, percent agreement, and Cohen's Kappa between individual student responses and the consensus expert responses for 31 requirements.

To answer Q3, we examined each individual student response against the consensus expert response. On average, students only agreed with the consensus expert assessment 55.95 percent of the time, with a standard deviation of 7.45 percent. Two students tied for the largest percentage agrement at 67.74 percent. Using Cohen's Kappa, we can characterize their agreement by taking agreement occuring by chance into consideration. Under this measure, the students had an average κ value of 0.110, which is considered to be slight agreement with the consensus expert assessment. At best, a student achieved a κ value of 0.362, which indicates fair agreement with the consensus expert assessment.

For this study, it makes sense to differentiate between false positives from false negatives, as shown in the upper half of Fig. 7 and labeled Initial Study Response Counts. Each individual student response is grouped and labeled from 1 to 32. The consensus expert assessment of the requirements resulted in 15 non-LIR requirements, which are shown on the left side of each student response, and 16 LIR requirements, which are shown on the right side of each student response. Identifying a requirement as LIR when it is not LIR (i.e., a false positive or a Type 1 error), may result in building legally questionable software. These errors are shown in green in the figure. Correctly identified LIR requirements are shown in red. Identifying a requirement as not LIR when it is LIR (i.e., a false negative or a Type 2 error), may result in delays or increased cost of development. These errors are shown in purple in the figure. Correctly identified non-LIR requirements are shown in blue.

Sensitivity measures the ability to predict positives. Specificity measures the ability to predict negatives. Using these measures, the students had an average sensitivity of 0.576 and an average specificity of 0.549. These average values indicate that students were only moderately successful at

Fig. 7. Response counts for both initial and replication studies.

predicting positives and negatives. These results also indicate that the students were more likely to err in saying that a requirement was LIR when it was not than they were to err in identifying a requirement as not LIR when it actually was LIR. This is a serious legal compliance concern; implementing a requirement that has legal compliance concerns may result in expensive efforts to refactor the resulting system or worse, a violation of the law.

We also sought to assess whether we could combine student responses to improve their ability to correctly determine whether requirements were LIR. Our approach was inspired by voting. For example, a consensus could be created by identifying a requirement as LIR if more than half of the students responded that it was. However, because a 50 percent vote is an arbitrary cutoff, we examined every possible vote cutoff and selected the best one as measure against the consensus expert assessment.

Using this approach, the students achieved their best results when 20 or more of the 32 students were required to vote that a requirement was ready for implementation. However, this consensus agreed with the consensus expert assessment only 67.74 percent of the time. Measured using Cohen's Kappa, this agreement is 0.357, which is only considered fair agreement with the consensus expert assessment. Their consensus sensitivity was 0.714 and their consensus specificity was 0.647. These values indicate that even in their best case scenario, the students were still more likely to err in favor of identifying a requirement as LIR when it is actually not LIR. Implementing requirements that need further refinement is the worst possible outcome when making LIR determinations.

Q3 Answer. Individual graduate students cannot accurately assess the LIR status of a requirement and are more likely to miss requirements that are not LIR.

4.3 Discussion of First Case

The results for this case of our study demonstrate that graduate-level computer science students with a background in software engineering are ill-prepared to make legal compliance decisions with any confidence. These students exhibit little consensus regarding whether or not requirements are LIR, and they disagreed with our subject matter experts on almost half of the requirements. If our participant population of computer science graduate students does not differ substantively from the average entry-level software engineer, then this is a crucial finding for organizations that must develop software that complies with laws and regulations.

This finding further illustrates the importance of involving subject matter experts in evaluating requirements for legal compliance. The subject matter experts bested the experiment's participants even before conducting the consensus meeting. We note, however, that 12 out of 31 requirements did not enjoy universal agreement entering the consensus meeting. The experts' initial differences illustrate the specific difficulty of determining whether or not requirements are LIR and the general challenge of dealing with ambiguity-laden legal texts. We suspect that even subject matter experts would benefit from structured support in making their assessments of whether or not a given requirement is LIR.

5 Second Case (Replication Study)

We performed a second case study to replicate our results from the first because we wanted to confirm the finding that graduate students in computer science and software engineering are not able to achieve consensus about whether software requirements are LIR. Although it might have been anticipated that they would not agree with the experts

as to whether the requirements were LIR, it was surprising to find that they did not agree as a group as to whether the requirements were LIR. In fact, their responses were only a marginal improvement over random selection.

In this case, we again employ the GQM approach. Our goal remains the same as our first case study:

> *Goal.* Analyze empirical observations for the purpose of characterizing legal implementation readiness with respect to software requirements from the viewpoint of software engineers in the context of an EHR system that must comply with HIPAA regulations.

However, we focus exclusively on two of our original three research questions outlined in our first case:

- *Q2* Is there a consensus among graduate students on which requirements are LIR?
- *Q3* Can graduate students accurately[7] assess which requirements are LIR?

The written materials for this case were identical to those used in the first case and described in Section 3.1. Although we made a concerted effort to consistently answer participant questions in each case, the questions and our responses to them were not identical in each case. As a part of our effort to maintain consistency, we verbally discussed common questions and their corresponding answers from the first case to participants in the second case as part of their training material. We used the same data collected from our subject matter experts as described in Section 3.2 as a canonical set of correctly assessed requirements.

Our graduate-level software engineering participant population, however, changed entirely from our first case. This case was conducted during the Fall semester of 2011, and we had 34 participants. Our participants were, once again computer science graduate students at North Carolina State University. However, they were all[8] currently taking the graduate-level software engineering course. Furthermore, all of them had received one lecture on requirements engineering and a second lecture on legal compliance in software systems. We provided the exact same 15 minute tutorial to the participants of this case study as in the first.

We now discuss the results of our experiment for both of our research questions.

Q2. Is there a consensus among graduate students on which requirements are LIR?

Measure. Fleiss Kappa statistic for 34 graduate students on 31 requirements.

The Fleiss Kappa statistic for our 34 students was $\kappa = 0.114$ ($p < 0.0001$). This result indicates that with a high level of statistical significance, the students in our case study had only a slight agreement on their assessment of the 31 requirements. This result is marginally higher than that of our first case study ($\kappa = 0.0792$), but it remains only slightly better than the result expected from a random set of responses. Thus, this result confirms the result from our first case study.

Fig. 6 displays the responses for the students in the LIR Assessment study on the bottom bar graph, labeled "Replication Study." The top bar graph displays the data from our previous case study. We have placed them both in proximity so that they may be easily compared. The vertical axis represents the number of students. Students that indicated a requirement is LIR (i.e., a 'Yes' response) are shown with a blue bar, and those that indicated a requirement is not LIR (i.e., a 'No' response) are shown with a red bar. The horizontal axis represents the requirement for which a decision was made. Note the similarities to the LIR Assessment study, which is displayed on the top bar graph. The are some notable differences (e.g., Requirements 17, 21, 23, 24, and 27), but they are generally quite similar.

If the two groups are considered as a single group of 66 participants, which we believe is justifiable because the two groups were created using the same participant selection criteria and provided the same materials and training, then our Fleiss Kappa statistic is $\kappa = 0.0958$ ($p < 0.0001$). Once again, this result indicates slight agreement and is marginally better than random.

Q2 Answer. In both the first case and the second case, we found a higher level of consensus among experts than graduate-level software engineers.

Q3. Can graduate students correctly assess which requirements are LIR?

Measures. Sensitivity, specificity, percent agreement, and Cohen's Kappa between individual student responses and the consensus expert responses for 31 requirements.

To answer Q3, we again examined each individual student response against the consensus expert response. On average, students only agreed with the consensus expert assessment 55.69 percent of the time, with a standard deviation of 9.20 percent. In our first case, the average agreement was 55.95 percent, with a standard deviation of 7.45 percent. The best performance in the this case was a single student with an average agreement of 77.42 percent, which was almost 10 percent better than the best performance in the first study. We believe this performance to be an outlier. The third quartile percentage for both studies was identical: 61.29 percent. The students had an average Cohen's Kappa value of 0.103, which is slightly lower than the 0.110 value found in the first case. These results both indicate that the students, on average, had slight agreement with the expert assessment when taking agreement by chance into consideration. The maximum Cohen's Kappa achieved in this case was achieved by our outlier: 0.545. The third quartile Cohen's Kappa for this study was 0.218 compared to the third quartile of 0.219 from the first study. These numbers indicate that for both studies roughly three quarters of the students had only slight agreement with the consensus expert assessment.

We also sought to differentiate between false positives from false negatives using sensitivity and specificity. The response counts for each student are shown in the lower half of Fig. 7 as the Replication Study Response Counts. Each individual student response is grouped and labeled from 1 to 34. The consensus expert assessment of the requirements resulted in 15 non-LIR requirements, shown

7. See footnote 5.

8. Some of our participants in the first case study had previously completed the course, whereas some of them were currently enrolled in it.

TABLE 1
Comparison of Sensitivity and Specificity from Initial and Replication Cases

Case	Sensitivity	Specificity
First Case Average	0.576	0.549
First Case Voted	0.714	0.647
Second Case Average	0.556	0.508
Second Case Voted	0.667	0.800

as either true negatives or false negatives, and 16 LIR requirements, shown as either true positives or false positives. In this case, the students had an average sensitivity of 0.556, which is slightly lower than the 0.576 achieved in the first case, and an average specificity of 0.508, which is also slightly lower than the 0.549 from the first study. These average values again indicate that students were only moderately successful at predicting positives and negatives. These results also reinforce our finding that the students were more likely to err in saying that a requirement was LIR when it was not than they were to err in identifying a requirement as not LIR when it actually was LIR.

When examining every possible voting cutoff to determine the best student consensus, we found that the group achieved their best results when 19 or more of the 34 students were required to vote that a requirement was ready for implementation. Their statistics also confirm those from our first study: 70.97 percent agreement (compared to 67.74 percent in the first case) and Cohen's Kappa of 0.413 (compared to 0.375 in the first case).

The consensus values for sensitivity and specificity differ both from the first case and from the individual averages. These are the first statistics indicating that students were more likely to err in identifying a requirement as needing further refinement when it is actually LIR than they were to err in identifying a requirement as LIR when it needs further refinement. Table 1 highlights this difference. These results suggest that it may be possible to improve performance by requiring roughly 60 percent of software engineers to vote in favor of considering a requirement to be LIR before implementing the requirement. However, additional research is needed before recommending any cutoff percentage as a software engineering practice.

Q3 Answer: Individual graduate students cannot accurately assess the LIR status of a requirement and are more likely to miss requirements that are not LIR.

Our findings suggest that our first case was correct: the participants were once again unable to achieve consensus on their assessment regarding whether the requirements were LIR.

6 Third Case (Wideband Delphi Study)

We conducted a third case in our study using the Wideband Delphi method to ensure a consensus assessment from graduate-level software engineers regarding whether the requirements from our two previous cases were LIR. Neither our first case nor our second case derived a clearly superior consensus assessment using vote-based methods as to whether the requirements were LIR. We found no cutoff percentage for vote-based consensus generation that would indisputably represent consensus among graduate-level software engineers about whether requirements are LIR. The Wideband Delphi estimation method can be used to derive consensus among smaller groups ranging from three to about a dozen individuals [33]. Therefore, we employed it to derive consensus among 14 graduate-level software engineers about whether the requirements examined in our previous two studies were LIR. Our findings suggest that our participants were able to slightly improve the accuracy of their LIR assessments, but their assessments were much more conservative than the experts, which would likely result in over-engineering of the underlying system.

We once again employ the Goal/Question/Metric (GQM) approach [27], [28] to design our case study. For this case study, our goal changes slightly compared to our previous cases:

> *Goal.* Analyze empirical observations for the purpose of characterizing legal implementation readiness with respect to software requirements from the viewpoint of a **small group of software engineers working together** in the context of an EHR system that must comply with HIPAA regulations.

Note the changes from the goal used in both of our prior case studies are highlighted in bold. This goal prompts the following research questions:

Q4 Can graduate students working together using the Wideband Delphi method accurately assess which requirements are LIR?

Q5 What is the extent of the discussion on requirements during the application of the Wideband Delphi method?

In Section 6.1, we present the methodology and describe the material inputs used. In Section 6.2, we describe our participant population. In Section 6.3, we present the results of our case study. Finally, in Section 6.4, we discuss the implications of this research.

6.1 Wideband Delphi Case Study Methodology and Materials

Because this case was designed to answer questions surfaced in our previous two cases, we retained the written materials described earlier in Section 3.1: the sample legal text, the requirements specification that includes a glossary of terms, and the traceability mapping of the individual requirements to the legal text. We continued our efforts to consistently respond to participant questions by once again verbally discussing common questions and their corresponding answers from the first two cases.

We conducted this case study on three separate days over the course of two weeks. On the first day, we introduced the case study and received completed informed consent forms for those participants interested in taking part in the study. We then provided participants with all of the materials for the case study and walked them through the same tutorial described in Fig. 2. This tutorial introduced some basic concepts in legal compliance for software

requirements and explained the importance of legal compliance in requirements engineering.

On the first day we also provided an overview of our application of the Wideband Delphi method. Wideband Delphi can be used to derive consensus among a group of participants on a set of topics. Topics are the set of things about which the participants seek to achieve consensus. For example, topics may be questions that must be answered, tasks for which effort must be estimated, or requirements that must be disambiguated. In this case, our topics were requirements that must be determined to be either LIR or not.

In Wideband Delphi, each participant is required to attend the initial consensus meeting having already made an initial determination for all the topics about which consensus is to be determined. This background ensures that each participant has considered the topic at hand and made a preliminary determination. In our case study, meeting this aspect of the Wideband Delphi method meant that participants would have to take home the case study materials and determine which requirements were LIR. They were instructed to work alone and not contact other participants, and they were given two days to complete the task.

The second day of the case study began two days after the first one. On this second day, we conducted our first consensus meeting. The meeting began by asking whether the participants' experience in making their initial determinations had prompted any questions. In contrast to our first two case studies in which participants made their determinations in our presence, asking questions as they arose, the participants in this study made their determinations individually. However, during this consensus meeting, we did answer questions that sought to clarify terminology or concepts. As in the previous two cases, we did not answer substantive questions about compliance or content. We discuss this initial question and answer session in Section 4.2.1.

Participants were instructed that our consensus meetings would be time-limited to 75 minutes.[9] Our initial plan was to conduct this as a single block of time, but we were unable to do so given the extent of the questions we received in the initial question and answer session. Therefore, our consensus meetings were divided into two sessions as described in Section 4.2.1.

During each consensus meeting, we moderated the discussion, allowing us to ensure that no one participant took over the meeting by virtue of their official capacity as the moderator. Our role as moderator entailed the following activities:

1) Beginning the discussion of a new requirement with a show-of-hands vote based on both the participants' initial determination made prior to the meeting and their thoughts on the discussion of previous requirements.
2) Answering clarifying questions raised about terminology or concepts, but not compliance or content.
3) After the initial show-of-hands vote, asking for a participant who felt the requirement was not LIR to explain their rationale. We selected a participant who felt the requirement was not LIR because any valid rationale that demonstrates a compliance concern is likely to persuade the other participants that the requirement was not LIR.
4) Allowing the discussion to continue in a free-form fashion until it either achieved consensus or stalled. At that point, we called for another show-of-hands vote. If this vote was unanimously in favor of either considering the requirement under discussion as LIR or not-LIR, we recorded the result and moved on to the next requirement. If this vote was not unanimous, we asked the participants if they wanted to continue discussion or move on to the next requirement. Only requirements that had received a unanimous vote in favor of considering it to be LIR were officially considered LIR at the end of the study.
5) Recording the time taken to discuss each requirement.
6) Recording the votes taken for each requirement.
7) Recording our observations on the level of disagreement for each requirement using a Likert scale as described in Section 6.3.
8) Recording any additional observations made during the consensus meetings, such as the number of requirements for which a participant explicitly referred to the sample legal text as a part of their rationale.

We did not explicitly pressure participants to wrap up their discussion based on the length of time the discussion was taking. However, we did tell participants at the beginning of each session how much time was reserved for the activity.

6.2 Wideband Delphi Case Study Participants

Fourteen computer science graduate students currently enrolled in a graduate-level Requirements Engineering course at North Carolina State University participated in this study. A graduate-level software engineering course is a pre-requisite for the Requirements Engineering course. Through their coursework, we know that they are familiar with the basic practices and responsibilities of a software engineer working as a part of a larger team. In addition, the study was conducted after a week of lectures on legal compliance concerns in requirements engineering.

These were the most sophisticated, most experienced participants of our three case studies. This was the only case study for which all of our participants were enrolled in a requirements engineering course. In addition, several of them had professional experience as software engineers.

6.3 Wideband Delphi Case Study Results

The first result from our Wideband Delphi case study is the initial starting point for each participant, which represents their individual determination for each requirement prior to any group discussion. This starting point does not represent the initial vote for each requirement. Since we conducted votes by a simultaneous

9. As described in Section 6.2, all of our participants were students in a graduate-level requirements engineering course at North Carolina State University. We chose this time limit because it was the duration of a single class period.

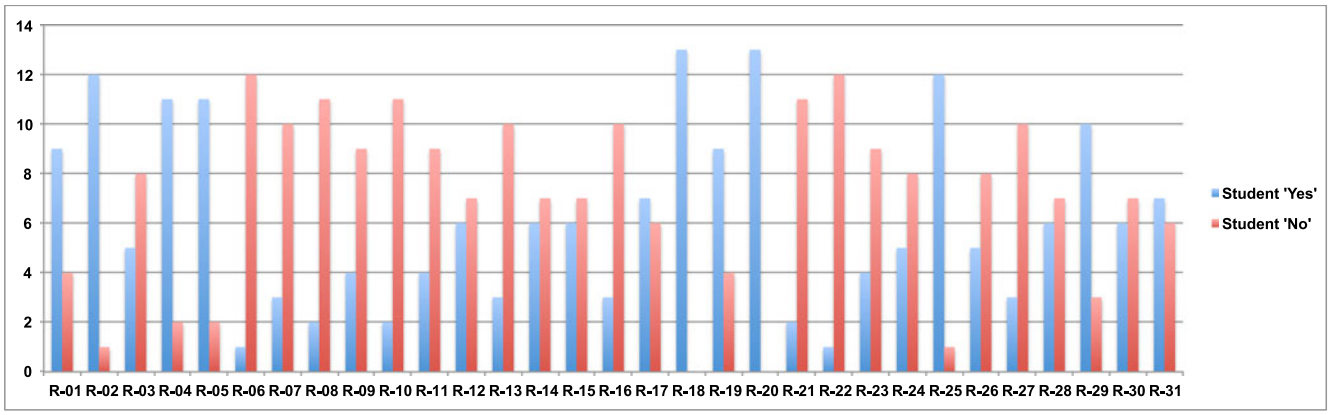

Fig. 8. Assessment of requirements by participants prior to wideband Delphi method.

TABLE 2
Comparison of LIR Assessments

Case	Percent Agreement	Cohen's Kappa	Sensitivity	Specificity
First Case (Average)	55.95%	0.110	0.576	0.548
First Case (Best-case Vote)	67.74%	0.357	0.714	0.647
Second Case (Average)	55.69%	0.103	0.556	0.509
Second Case (Best-case Vote)	70.94%	0.413	0.667	0.800
Third Case (Average)	52.15%	0.044	0.521	0.492
Third Case (Wideband Delphi)	54.84%	0.111	0.625	0.522

TABLE 3
Percentage of Participants Identifying a Requirement as LIR for the Initial, Replication, and Wideband Delphi Cases

C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
I	81	94	53	84	88	81	53	31	53	50	56	63	63	59	66	69	41	81	72	75	59	47	47	56	81	69	53	91	84	50	59
R	79	94	62	85	82	82	50	32	50	38	56	68	53	65	56	62	59	94	91	88	47	47	74	82	85	65	76	74	100	56	62
W	69	92	38	85	85	8	23	15	31	15	31	46	23	46	46	23	54	100	69	100	15	8	31	38	92	38	23	46	77	46	54

show-of-hands, participants could influence one another even for the first vote. Furthermore, discussion from the earlier requirements may influence the voting on later requirements. Fig. 8 displays the initial starting point for the discussions in the Wideband Delphi case study. The vertical axis represents the number of students. Students that indicated a requirement is LIR (i.e., a 'Yes' response) are shown with a blue bar, and students that indicated a requirement is not LIR (i.e., a 'No' response) are shown with a red bar. The horizontal axis represents the requirement for which a decision was made. Note that participants have indicated more requirements are not LIR than the previous case studies shown in Fig. 6, particularly for Requirements 6 through 16. Also, note only 13 students are represented in this figure because one student did not record their individual determinations for each requirement prior to the case study.

For this case study, we seek to examine the accuracy of the assessments and the extent of the consensus within the group. We now discuss results for the accuracy of the assessments made by consensus, which was represented in this study using the following question:

Q4. Can graduate students working together using the Wideband Delphi method accurately assess which requirements are LIR?

Measures. Sensitivity, specificity, percent agreement, and Cohen's Kappa comparing graduate students' consensus assessment to the consensus expert assessment for 31 requirements.

First let us consider the level of agreement found by averaging the individual responses from the Wideband Delphi study participants prior to their consensus meeting. They were a little lower than the averages found in the previous two cases, as shown in Table 2. Their average percent agreement with the consensus expert assessment was 52.15 percent, which is about 3.5 percent worse than the previous two cases, and their average Cohen's Kappa was 0.044, which is close to what would be found with random responses. For sensitivity and specificity, the participants averaged 0.521 and 0.492 respectively. Individual averages were slightly better for both sensitivity and specificity in the first two cases as well.

Table 3 provides another way to compare the individual results from each of our three cases. The column on the left indicates the case with 'I' indicating the Initial case, 'R' indicating the Replication case, and 'W' indicating the Wideband Delphi case. The header indicates each of the requirements examined from 1 to 31. The numbers in each row are the percentage of participants who found the requirement to be LIR. The Initial case had 32 participants and the Replication case

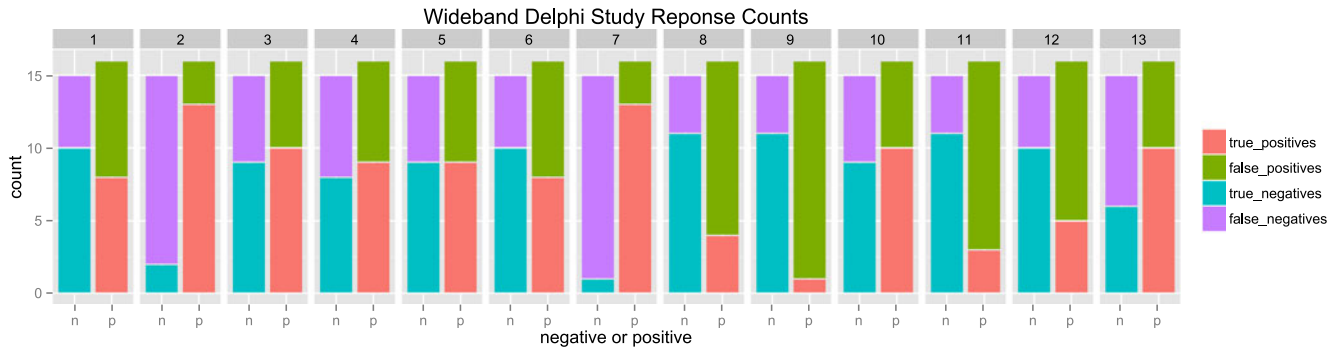

Fig. 9. Raw wideband delphi study response counts.

had 34 participants, and the results of these studies are quite similar. However, the Wideband Delphi case only had 13 participants because the Wideband Delphi technique is designed to work with at most about a dozen participants. The smaller number of participants in the third study may account for some of the discrepancies between it and the previous two studies. In particular, the individual participant responses for requirements 6, 7, 21, and 22 stand out.

Fig. 9 shows the counts for each type of response the individual Wideband Delphi study participants had on their initial evaluation for each requirement. Each individual student response is grouped and labeled from 1 to 13. As with Fig. 7, the 15 non-LIR requirements are shown as true negatives and false negatives, and the 16 LIR requirements are shown as true positives and false positives.

Now let us consider the level of agreement between the graduate students' Wideband Delphi consensus assessment using the experts' consensus assessment. In this case, the students achieved 54.84 percent agreement with the consensus expert assessment, an improvement of 2.69 percent over their individual average. This result is similar to the average agreement from the first two cases, which were 55.95 and 55.69 percent respectively. When measured using Cohen's Kappa, the students demonstrated slight agreement with a value of 0.111, which is an improvement over their individual average as well. This result is also similar to the averages from the first two studies, which were 0.110 and 0.103 respectively. The similarity of these numbers overall indicates that students working together to make LIR determinations using Wideband Delphi performed roughly as well as the average individual student.

The students' Wideband Delphi consensus assessment also improved for both sensitivity and specificity. The largest improvement was for sensitivity, where the students went from 0.521 to 0.625. For specificity, the students only went from 0.492 to 0.522. Although an ideal evaluation of requirements would have both high sensitivity and high specificity, if forced to pick between the two, we would prefer to improve specificity because we would rather see improvement in the ability of the group to correctly exclude a requirement as definitively not LIR. It is safer to be more cautious and avoid mistakes when identifying a requirement as needing further refinement.

We can also compare our results from the real consensus derived using Wideband Delphi to the hypothetical consensus graduate student assessments from the first two cases. These results are displayed in Table 2. In a Wideband Delphi consensus scenario, students performed worse than the hypothetical, best case votes from both the initial study and the replication study. The sensitivity measure was the largest improvement between the Wideband Delphi consensus assessment and the average assessment from the first two cases. This matches the improvement found when the Wideband Delphi consensus result is compared to the average sensitivity for the students prior to their consensus session. This indicates that the Wideband Delphi consensus technique slightly improved the group's ability to correctly identify a requirement as legally implementation ready.

Q4 Answer. The similarity of the Wideband Delphi consensus results found in this case indicate that graduate students in software engineering remain ill-prepared to make legal implementation assessments of software requirements even when using a methodology designed to facilitate discussion among participants and achieve unanimous consensus.

Q5. What is the extent of the discussion on requirements during the application of the Wideband Delphi method?

Measures. Length of discussion in minutes and observation of the level of disagreement on a Likert scale from 1 (very little disagreement) to 5 (very high disagreement).

As discussed in Section 6.1, this case study was conducted with a planned 75-minute window for discussion. To ensure that each requirement was discussed, the participants agreed to end discussion without achieving unanimous consensus seven times. The participants also agreed that those requirements where unanimous consensus could not be reached would be considered non-LIR. Had the participants only required 50 percent of the participants to achieve consensus, only one of these requirements (R-27) would have been considered LIR; the others (R-11, R-14, R-15, R-28, R-30, and R-31) would have all remained non-LIR.

The length of discussion for each requirement varied from unanimous agreement at the outset to roughly 8 minutes of discussion. For example, Requirement 11, displayed in Fig. 10, was discussed for 6 minutes and 18 seconds. The participants were split between two beliefs about the account structure described by the iTrust system. One group believed that employee accounts were distinct from patient accounts; the other group believed that employees with accounts must use the same account when they are

R-11: When an employee record is updated, iTrust shall send an email to an employee's email address according to the stored preferences regarding email for that employee. [Traces to § 164.312(b)]

Fig. 10. iTrust requirement 11.

considered patients in the system. Ultimately, the participants decided that they would not be able to come to a unanimous consensus for this requirement, which meant that it is considered not LIR. Fig. 11 displays the duration of the discussion for each of the requirements.

In addition to recording the duration of discussion for each requirement, we observed and recorded the level of disagreement for each requirement using a Likert scale. For a discussion that was unanimous or nearly unanimous, we recorded a 1, indicating very little disagreement. If a participant raised a point for the sake of argument but didn't press the point when challenged, we recorded a 2, indicating little disagreement. If participants discussed two or more alternative interpretations and weighed them seriously before making a decision, we recorded a 3, indicating moderate disagreement. If participants discussed two or more alternative interpretations with a standard bearer from each alternative actively campaigning for their choice, we recorded a 4, indicating a high level of disagreement. If participants discussed two or more alternatives and either appeared to be unable to come to consensus or actually were unable to come to a consensus as a result of the discussion, we recorded a 5, indicating a high level of disagreement.

We also noted the content of the discussion for those requirements with a non-trivial amount of discussion (i.e., those requirements that were not unanimously or nearly unanimously decided). For example, we noted the specific words or phrases that were interpreted differently by the various participants. We also noted whether the participants referenced the glossary, the traceability matrix, or the legal text during these disputes.

6.4 Wideband Delphi Case Study Discussion

We now discuss the results of and observations from our third case, which used the Wideband Delphi method to achieve consensus among graduate students in computer science with a background in software engineering. Our discussion begins with the first consensus meeting for our third case, which was after the participants had made their individual determinations for each requirement.

6.4.1 Question and Answer Session

The session began with a brief question and answer period. Recall that after providing our participants with our tutorial on legal implementation readiness, they had two days prior to our first consensus session to make their individual determinations about the requirements we would be discussing in our consensus session. They came back with two basic types of questions about the study, as we now discuss.

The first type of question involved clarifying questions about the concepts (e.g., legal implementation readiness or refinement of requirements) from the study. We responded to these as directly and succinctly as possible. For example, one participant wanted to know whether the requirements that were determined by consensus to be LIR would be implemented immediately after the decision was made. We explained that additional design and engineering concerns may require further refinement of the requirement, but that it would not be refined further to address legal concerns prior to being implemented.

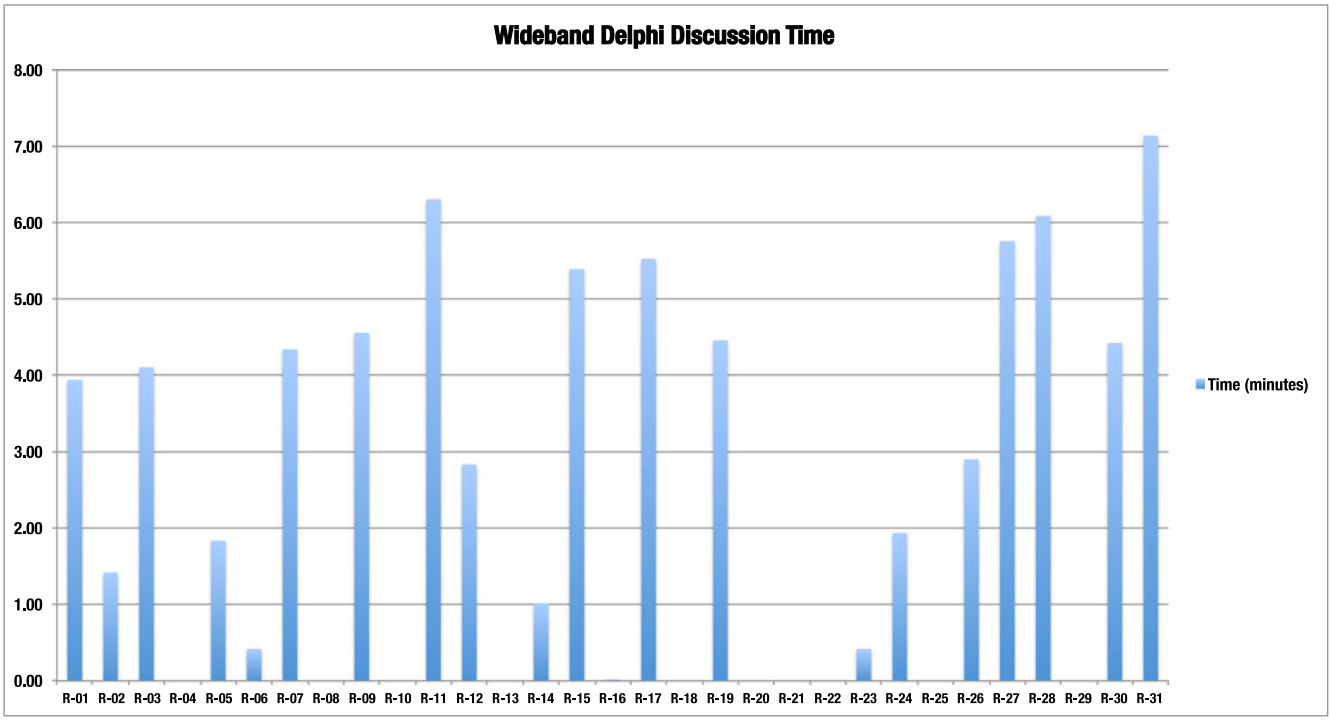

Fig. 11. Duration of discussion for requirements in the wideband delphi case study.

Standard: Access control. Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in § 164.308(a)(4).

Fig. 12. HIPAA Section 164.312(a)(1).

Several students wanted to know whether the iterative cycles in the development process used for these requirements were days, weeks, or months long. We explained that LIR decisions could be made in short or long development cycles, and that the more important concern was that any requirement determined to be LIR would not be further examined as a part of the legal requirements compliance process.

The second type of question the participants asked related to the content of the study. We responded to these questions by reinforcing that they were ultimately the sorts of things we wanted the participants themselves to consider as they determined whether the requirements in the study were LIR. For example, some students asked what weight they should give to the cross-reference in the sample legal text. Fig. 12 displays the cross-reference they were concerned about, which we mentioned in Section 3.1.

We explained that participants could determine for themselves whether or not they believed this cross-reference was important from a legal compliance standpoint. We also told the participants that it may or may not contain information relevant to the various requirements that traced to Section 164.312(a)(1), and each participant could decide whether they wanted to concern themselves with this sort of risk.

Participants also asked several questions about the glossary terms. For example, one participant wanted to know if there was any overlap between the iTrust roles defined in the glossary provided. We explained that anything not explicitly excluded by the role could be a valid concern to anyone in a legal compliance decision-making process.

The third type of question participants asked was procedural. For example, the students wanted to know more about the role of the moderator. We explained that the moderator's purpose was to facilitate the consensus sessions, including prompting participants to vote, observing and recording aspects of the discussion, and ensuring the group proceeded from one requirement to another if the vote resulted in consensus or if it was clear consensus could not be achieved in a timely fashion.

6.4.2 First Consensus Session

Once participant questions had been adequately addressed, we began the first discussion session. This session took 55 minutes and covered the first 26 requirements. Our stated goal was to take no more than 75 minutes to discuss the requirements. Although time limitations are not typically a part of the Wideband Delphi process, they are a real world constraint. Fig. 11 details how long participants discussed each requirement.

R-1: iTrust shall support user authentication by requiring a user to input their user ID and password. [Traces to § 164.312(a)(1), § 164.312(a)(2)(i), and § 164.312(d)]

Fig. 13. iTrust requirement 1.

The discussion was conducted one requirement at a time starting with R-1 and ending with R-31. For each requirement, the discussion started with an initial vote, conducted by a show-of-hands. For the first several requirements, these initial votes were identical to the participants' individual determinations made in the previous 48 hours. However, as the votes progressed, some participants altered their decisions for their initial public vote based on the discussion of previous requirements.

For many of the early requirements, the participants focused the discussion on terminology. In particular, the first requirement, R-1 shown in Fig. 13, spawned an extended discussion about the uniqueness of user IDs.

Some participants asked questions about account hierarchies based on the definitions provided in the glossary. These participants were concerned that employees who were also patients in the healthcare facility would be able to violate access control rules if they had two separate accounts. The participants spent some time discussing this before determining that many of the authentication requirements were LIR based on the definitions provided in the glossary. Although it was not the purpose of this study, it is interesting to note that the actor hierarchies discussed in our prior work [31] may have alleviated some of this discussion had they been available.

6.4.3 Second Consensus Session

We were unable to achieve consensus on every requirement before our scheduled session ended. Therefore, we completed the consensus building in a second session. The second consensus session took about 20 minutes, covered the final five requirements, and consisted of 12 participants. In addition, two of the 13 participants from the first session were unable to attend the second. One participant who was unable to attend the first session was present for the second.

We began the second session with a brief overview of the previous session. When asked, the participants unanimously decided that their determinations for each of the previous 26 requirements remained correct and did not require revision. We then proceeded to examine the final five requirements, which proved to be some of the most contentious requirements in the study. The participants in the study agreed that they would be unable to achieve a unanimous consensus for four of these final five requirements. For example, the discussion for Requirement 27, displayed in Fig. 14, focused on whether or not a cryptographic hash would provide any level of security. Some participants felt that it was intended to do so; others felt that the hash was meant simply as a means of identification. The participants felt that it was clear these two groups would not agree on the final decision.

At the end of the session, we asked the participants if they felt their determinations for any of the requirements

R-27: Each time a printable emergency report for a medical record is generated, iTrust shall include a cryptographic hash of the authenticated employee user ID and the patient user ID at the end of the report. [Traces to § 164.312(b)]

Fig. 14. iTrust requirement 27.

had changed. The participants unanimously agreed that their earlier determinations remained accurate even after subsequent discussion.

7 Threats to Validity

Internal validity refers to the ability to establish causal relationships based on recorded observations [37]. The case studies described in this paper are descriptive in nature. We only attempt to describe the relationship that exists between the structure of a legal text and the software requirements that can be traced to them. We have not attempted to establish a causal relationship (e.g., we show that our participants were ill-prepared to make LIR decisions, but we do not explain why they were ill-prepared). Therefore, we do not need to consider internal validity as a threat.

Construct validity refers to the appropriateness and accuracy of the measures and metrics used for the concepts studied [37]. The primary measure we use for all three cases in our study is the canonical set of LIR requirements generated by the subject matter experts using a Wideband Delphi procedure, which was constructed as described in Section 3.2. The use of a canonical set of LIR requirements to test other requirements for legal compliance is an accepted practice [32]. In addition, Wideband Delphi is an accepted practice for achieving consensus in software engineering [33]. However, the three subject matter experts worked together previously on the concept of LIR requirements. As a result, they are not entirely independent individuals despite conducting the first phase of our Wideband Delphi consensus procedure independently from one another. In addition, our subject matter experts have not developed a commercial EHR system.

In Section 3.1, we describe our process for creating the materials used throughout this case study. Although we adhered as closely as possible to the iterative techniques described in our prior work [31], we did include some requirements that we knew were not legally implementation ready. The 31 requirements used were designed to be realistic, albeit not fully refined, and the traceability mapping was designed to reflect the correct relationship between the requirements and the relevant areas of legal text. It is possible that the design of these materials introduced some bias, but based on our extensive work with healthcare requirements, we believe any resulting bias to be minimal.

Our selection of Section 164.312 as the sole legal text under examination may also limit the significance of our findings. To ensure statistically significant results and establish a baseline against which tools, techniques, and processes for LIR decision-making could be compared, we sought to focus our examination on a single section of HIPAA. If we had been able to survey more than 32 participants in the initial study, then we may have been able to expand the selection of text examined while maintaining reasonable confidence that our results would hold statistical significance. Depending on the number of participants, we may have been able to group participants and have them evaluate requirements in different sections of HIPAA to determine whether and how participant performance was affected by the legal text under examination. This is an area for potential future work.

The materials for this study were designed so that the study could be conducted in a classroom session on printed paper. For each of the first two cases, the participants were monitored as if they were sitting for a proctored examination, and a result sheet containing the LIR or not-LIR status for each of the requirements was collected. Because participants could make determinations for requirements in an unknown order, we are not able to determine whether there was a learning effect in participant performance or whether participants identified possible connections from one part of the study to another. Studies examining whether and how order affects LIR decision-making are an area for potential future work.

In Section 5, we performed a statistical calculation on the combined results from both the original case and the replication case. Although these studies were conducted with the same materials, training, and participant selection criteria, they were not conducted at the same time, and they were conducted on different groups. It is possible that differences in their respective backgrounds are not appropriately reflected in these results.

In Section 6.4.3, we discuss our Wideband Delphi consensus process, which spanned two separate sessions. Conducting multiple consensus sessions may have introduced some bias that differentiates the analysis the group used on requirements discussed in the first session as compared to the second session. However, real world compliance analysis for systems intended for deployment are unlikely to be conducted in a single session, so any bias introduced may better reflect real world situations.

The answers we provided to participants' questions during the experiment are another source of potential construct validity concerns. For example, one participant asked about the difference between Required and Addressable as used in the legal text. We explained that "Required" indicates a legal obligation to implement all elements described in that section as closely as possible, whereas "Addressable" indicates that the general concept must be present but does leave some implementation details up to the specific software engineer. Our explanation for "Addressable", however, was an inadvertent misrepresentation of the term's precise legal meaning, provided in HIPAA Section 164.306(d)(3). "Addressable" sections allow entities to assess whether the section's requirements are "reasonable and appropriate" for their systems; if not, then entities may instead implement equivalent alternative measures that are in fact "reasonable and appropriate" for their system, or explain why no such measures are reasonable and appropriate. Our explanation implied that all "Addressable" sections had been predetermined to be "reasonable and appropriate" for implementation in iTrust. Since we used this explanation in response

to a question for the first case, we continued using it in the remaining cases to maintain consistency.

External validity refers to the ability to generalize findings and results to other domains [37]. By selecting a highly technical section of HIPAA that should have played to the technical background of our participants, we believe that we have established a valid baseline from which we can generalize to other sections of HIPAA or other pieces of legislation. Had we chosen a less technical, more legally oriented section of HIPAA, we believe our results would be less generalizable to other domains and more variable over repeated experiments. We believe it is likely that our participants would exhibit even less consensus on a more legally oriented piece of legislation because they lacked legal domain knowledge. However, it is also possible that their lack of domain knowledge would increase consensus simply because the participants would not even know what questions to ask about a more legally oriented piece of legislation. In either case, their lack of legal domain knowledge would lead to a less generalizable, and perhaps less repeatable, result had we not chosen a highly technical section of HIPAA.

Another external validity concern is that graduate students may perform differently from practitioners in a genuine legal compliance scenario. For example, we had to limit the time and resources that our participants were allowed to use while reviewing the requirements for legal compliance. Graduate students in computer science are a population of convenience. It is extremely challenging to get practitioners whose professional reputations depend on their ability to build software compliant with laws to take part in research that may reveal their inability to do so effectively. However, graduate students may be better qualified than practitioners to assess requirements for legal compliance [1]. To our knowledge, no research has been conducted that demonstrates otherwise. Maxwell has shown that practitioners are ill-equipped to perform a similar task when evaluating legal cross references [1]. Many practitioners have never received formal training in software engineering or requirements engineering, either at the graduate or undergraduate level. Also, in each of the cases for this study, several of our participants had several years of professional experience as software engineers.

Reliability refers to the ability of other researchers to repeat an experiment [37]. We assiduously documented our process and procedures while conducting our experiment. In addition, we have made available[10] copies of all materials used in our experiment for other researchers interested in repeating it.

Statistical conclusion validity is the degree to which relationships found using statistical methods are valid [38]. A type 1 error occurs when a correlation is found when that correlation does not actually exist [38]. A type 2 error occurs when no correlation is found and a correlation actually does exist [38]. Numerous methodological mistakes could have resulted in both types of errors. When selecting between multiple statistical techniques, we chose the most accepted, widely used, and reliable technique to mitigate this possibility. It is possible that our restriction of responses to a simple LIR or not-LIR could result in a statistically underpowered survey that may be mitigated by accepting a probability of LIR as a response from participants and using alternative statistical measures. In making determinations of this nature, we have opted to adhere as closely as possible to real-world engineering scenarios. For example, software engineers must determine whether or not to implement a requirement rather than assess the probability that a requirement is implementable.

8 Summary

There is a clear need to better understand how we can support software engineers in developing legally compliant software systems. Understanding the law is an ethical obligation for software engineers. The Association for Computing Machinery Code of Ethics[11] states that computing professionals must "know and respect existing laws pertaining to professional work." This is increasingly important in an increasingly regulated environment. Software engineers must manage compliance with laws and regulations during development. Legal compliance may ultimately become the single most important non-functional requirement for a large number of software systems. In this paper, we discuss a multi-case study examining legal implementation readiness decision-making with three cases:

- *First case.* An initial examination of how graduate-level software engineers assess software requirements for legal implementation readiness. The findings from this study indicate that they are ill-prepared to make these determinations.
- *Second case (replication study).* Confirms our findings from the First Case; graduate-level software engineers are ill-prepared to make LIR assessments of software requirements.
- *Third case (wideband delphi study).* Indicates that graduate students in software engineering remain ill-prepared to make legal implementation assessments of software requirements even when using a methodology designed to facilitate discussion among participants and achieve unanimous consensus. This study also found that the Wideband Delphi consensus technique slightly improved LIR assessment accuracy. However, both our quantitative findings and our qualitative findings suggest that graduate-level software engineers using the Wideband Delphi method to achieve consensus about software requirements are overly cautious in their assessments.

Each of these studies is designed to examine the nature of assessing software requirements for legal compliance. This research supports the goal of understanding legal compliance in software systems by establishing an empirical baseline for the accuracy of legal compliance decisions made by graduate-level computer science students with a background in software engineering individually or in groups. As we have shown in this paper, these engineers are ill-prepared to make legal compliance decisions with any confidence. They exhibit little consensus

10. http://www.cc.gatech.edu/ akmassey/documents/LIRstudy.pdf

11. http://www.acm.org/about/code-of-ethics

regarding whether or not requirements are LIR, and they disagreed with our subject matter experts on most decisions. If our participant population of computer science graduate students does not differ substantively from the average entry-level software engineer, then this is a crucial finding for organizations that must develop software that complies with laws and regulations.

Our research further illustrates the value of involving subject matter experts in evaluating whether or not requirements are LIR. The subject matter experts did not display universal agreement on 12 of the 31 requirements prior to conducting the consensus meeting. These differences highlight the general challenge of dealing with ambiguity-laden legal texts. Since this research was conducted, we have begun a detailed examination of ambiguity in legal texts [39] with the eventual goal of providing structured support to both experts and novices assessing legal implementation readiness for requirements.

Herein, we present empirical evidence that Wideband Delphi alone does not provide enough improvement to ensure accurate LIR decision-making. We also present empirical evidence demonstrating that graduate students need tools and methods to help guide their interpretations of the law. This support must serve two purposes. First, it must provide insight that engineers can use to make engineering decisions. Although tool support cannot replace lawyers, it may provide useful, actionable guidance to engineers. Second, it must enable productive exchanges with lawyers. Tools and methods that provide insight to engineers regarding legal concerns reduce compliance costs and improve communication. In our future work, we plan to continue to develop tools and methods, such as legal requirements metrics [40] or models of policy documents [41], which support engineers building software systems that verifiably comply with laws and regulations.

Acknowledgments

This work was partially supported by NSF ITR Grant #522931 and NSF Cyber Trust Grant #0430166. The authors would like to thank the members of The Privacy Place for their feedback on early drafts of this paper. A. K. Massey is the corresponding author.

References

[1] J. C. Maxwell, "Reasoning about legal text evolution for regulatory compliance in software systems," Ph.D. dissertation, Department of Computer Science, North Carolina State University, Raleigh, NC, 2013.

[2] P. N. Otto and A. I. Antón, "Addressing legal requirements in requirements engineering," in *Proc. 15th IEEE Int. Requirements Eng. Conf.*, 15–19 Oct. 2007, pp. 5–14.

[3] K. Beaver and R. Herold, *The Practical Guide to HIPAA Privacy and Security Compliance*. New York, NY, USA: Auerbach, 2004.

[4] Q. Yin, N. Madhavji, and M. Pattani, "Eros: An approach for ensuring regulatory compliance of process outcomes," in *Proc. 6th Int. Workshop Requirements Eng. Law*, Jul. 2013, pp. 21–24.

[5] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum, "Privacy and contextual integrity: Framework and applications," in *Proc. 2006 IEEE Symp. Security Privacy*, 2006, pp. 184–198.

[6] M. J. May, C. A. Gunter, and I. Lee, "Privacy APIs: Access control techniques to analyze and verify legal privacy policies," in *Proc. Comput. Secur. Found. Workshop*, 2006, pp. 85–97.

[7] F. Massacci, M. Prest, and N. Zannone, "Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation," *Comput. Standards Interfaces*, vol. 27, no. 5, pp. 445–455, 2005.

[8] I. Jureta, T. Breaux, A. Siena, and D. Gordon, "Toward benchmarks to assess advancement in legal requirements modeling," in *Proc. Sixth Int. Workshop Requirements Eng. Law*, Jul. 2013, pp. 25–33.

[9] M. El Kharbili, Q. Ma, P. Kelsen, and E. Pulvermueller, "Corel: Policy-based and model-driven regulatory compliance management," in *Proc. 15th IEEE Int. Enterprise Distrib. Object Comput. Conf.*, Aug. 2011, pp. 247–256.

[10] M. El Kharbili, Q. Ma, P. Kelsen, and E. Pulvermueller, "Enterprise regulatory compliance modeling using Corel: An illustrative example," in *Proc. IEEE 13th Conf. Commerce Enterprise Comput.*, Sep. 2011, pp. 185–190.

[11] W. Hassan and L. Logrippo, "Requirements and compliance in legal systems: A logic approach," in *Proc. Requirements Eng. Law*, Sep. 2008, pp. 40–44.

[12] W. Hassan and L. Logrippo, "Governance requirements extraction model for legal compliance validation," in *Proc. 2nd Int. Workshop Requirements Eng. Law*, Sep. 2009, pp. 7–12.

[13] W. Hassan and L. Logrippo, "Towards a process for legally compliant software," in *Proc. 6th Int. Workshop Requirements Eng. Law*, Jul. 2013, pp. 44–52.

[14] J. Maxwell and A. Antón, "The production rule framework: Developing a canonical set of software requirements for compliance with law," in *Proc. 1st ACM Int. Health Informat. Symp.*, 2010, pp. 629–636.

[15] S. Ghanavati, D. Amyot, and L. Peyton, "Compliance analysis based on a goal-oriented requirement language evaluation methodology," in *Proc. 17th IEEE Int. Requirements Eng. Conf.*, Aug. 2009, pp. 133–142.

[16] S. Ghanavati, D. Amyot, and L. Peyton, "A systematic review of goal-oriented requirements management frameworks for business process compliance," in *Proc. 4th Int. Workshop Requirements Eng. Law*, 2011, pp. 25–34.

[17] S. Ghanavati, L. Humphreys, G. Boella, L. Di Caro, L. Robaldo, and L. van der Torre, "Compliance with multiple regulations," in *Proc. 33rd Int. Conf. Conceptual Model.*, 2014, pp. 415–422.

[18] S. Ghanavati, D. Amyot, A. Rifaut, and E. Dubois, "Goal-oriented compliance with multiple regulations," in *Proc. 22nd IEEE Int. Requirements Eng. Conf.*, 2014, pp. 73–82.

[19] A. Rifaut and S. Ghanavati, "Measurement-oriented comparison of multiple regulations with grl," in *Proc. 5th Int. Workshop Requirements Eng. Law (RELAW)*, Sep. 2012, pp. 7–16.

[20] D. Gordon and T. Breaux, "Reconciling multi-jurisdictional legal requirements: A case study in requirements water marking," in *Proc. 20th IEEE Int. Requirements Eng. Conf.*, Sep. 2012, pp. 91–100.

[21] D. Gordon, "The regulatory world and the machine: Harmonizing legal requirements and the systems they affect," in *Proc. 21st IEEE Int. Requirements Eng. Conf.*, Jul. 2013, pp. 381–384.

[22] S. Ingolfo and V. Silva Souza, "Law and adaptivity in requirements engineering," in *Proc. ICSE Workshop Softw. Eng. Adaptive Self-Manag. Syst.*, May 2013, pp. 163–168.

[23] B. Berenbach, D. Grusemann, and J. Cleland-Huang, "The application of just in time tracing to regulatory codes and standards," in *Proc. 8th Conf. Syst. Eng. Res.*, 2010.

[24] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory requirements codes to product specific requirements," in *Proc. 32nd Int. Conf. Softw. Eng.*, May 2–8, 2010.

[25] T. Breaux, "Exercising due diligence in legal requirements acquisition: A tool-supported, frame-based approach," in *Proc. 17th IEEE Int. Requirements Eng. Conf.*, 31—Sep. 4 2009, pp. 225–230.

[26] T. Breaux and D. Gordon, "Preserving traceability and encoding meaning in legal requirements extraction," in *Proc. 6th Int. Workshop Requirements Eng. Law*, Jul. 2013, pp. 57–60.

[27] V. R. Basili, "Software modeling and measurement: The goal/question/metric paradigm," University of Maryland at College Park, College Park, MD, USA, Tech. Rep. UMIACS TR-92-96, 1992.

[28] V. R. Basili, *Applying the goal/question/metric paradigm in the experience factory*, International Thomson Computer Press, 1995, pp. 21–44.

[29] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Trans. Softw. Eng.*, vol. 34, no. 1, pp. 5–20, Jan. 2008.

[30] J. C. Maxwell and A. I. Antón, "Discovering conflicting software requirements by analyzing legal cross-references," in *Proc. 19th IEEE Int. Requirements Eng. Conf.*, Sep. 2011, pp. 197–206.
[31] A. Massey, P. Otto, L. Hayward, and A. Antón, "Evaluating existing security and privacy requirements for legal compliance," *Requirements Eng.*, vol. 15, pp. 119–137, 2010.
[32] A. K. Massey and A. I. Antón, "Triage for legal requirements," North Carolina State University, NC, USA, Tech. Rep. TR-2010-22, 2010.
[33] B. W. Boehm, *Software Engineering Economics*. Upper Saddle River, NJ, USA: Prentice-Hall, 1981.
[34] D. Olson and D. Delen, *Advanced Data Mining Techniques*. New York, NY, USA: Springer, 2008.
[35] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," *Computational Linguistics*, vol. 22, no. 2, pp. 249–254, 1996.
[36] J. L. Fleiss and J. Cohen, "The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability," *Educ. Psychol. Meas.*, vol. 33, pp. 613–619, 1973.
[37] R. K. Yin, *Case Study Research: Design and Methods*, 3rd ed. Newbury Park, CA, USA: Sage, 2003, vol. 5.
[38] P. C. Cozby, *Methods in Behavioral Research*, 10th ed. New York, NY, USA: McGraw-Hill, 2009.
[39] A. K. Massey, R. L. Rutledge, A. I. Antón, and P. P. Swire, "Identifying and classifying ambiguity for regulatory requirements," in *Proc. 22nd IEEE Int. Requirements Eng. Conf.*, 2014, pp. 83–92.
[40] A. Massey, B. Smith, P. Otto, and A. Anton, "Assessing the accuracy of legal implementation readiness decisions," in *Proc. 19th IEEE Int. Requirements Eng. Conf.*, pp. 207–216, Sep. 2011.
[41] A. K. Massey, J. Eisenstein, A. I. Antón, and P. Swire, "Automated text mining for requirements analysis of policy documents," in *Proc. 21st Int. Conf. Requirements Eng.*, 2013, pp. 4–13.

Aaron K. Massey received the BS in computer engineering from Purdue University, the MS and PhD degrees in computer science from North Carolina State University. He is a postdoctoral fellow at the Georgia Tech's School of Interactive Computing and the associate director of ThePrivacyPlace.org. His research interests include computer security, privacy, and regulatory compliance software engineering. He was the recipient of a Google policy fellowship and the Walter H. Wilkinson graduate research ethics fellowship. He is a member of the ACM, IEEE, IAPP, and the USACM public policy council. He is a member of the IEEE.

Paul N. Otto received the BS in computer engineering from the University of Virginia, the MS degree in computer science from North Carolina State University, and the JD degree from Duke Law School. He has worked as an attorney for the Utah State Courts, and he was a Google policy fellow for the Center for Democracy and Technology. His research interests include general security and privacy issues, privacy policies, data breaches, security and privacy requirements, and legal compliance. He is a member of the IEEE.

Annie I. Antón is a professor and chair of the School of Interactive Computing at Georgia Tech. Her research focuses on methods and tools to support the specification of complete, correct behavior of software systems used in environments that pose risks of loss as a consequence of failures and misuse. This includes web-based and healthcare systems in which the security of personal and private information is particularly vulnerable. Current extensions to this work include the analysis of security and privacy policies, regulations and compliance practices. Her professional activities include a notable combination of multi-disciplinary research and education. She is a co-founder of the Symposium on Requirements Engineering for Information Security (SREIS) and the Annual Requirements Engineering and the Law Workshop (RELAW). She is a former associate editor for the *IEEE Transactions on Software Engineering*, former cognitive issues subject area editor for the *Requirements Engineering Journal*, and currently a member of the International Board of Referees for Computers & Security. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.