ABSTRACT

MAXWELL, JEREMY C. Reasoning About Legal Text Evolution for Regulatory Compliance in Software Systems. (Under the direction of Annie I. Antón and Jon Doyle.)

Laws and regulations govern organizations and the software systems that they use. Software engineers must understand these laws and regulations as they are constructing their systems to ensure regulatory compliance in software systems. As laws and regulations evolve over time, engineers must understand the impact on their systems and adapt their software accordingly. Oftentimes, engineers have compressed timelines to address new requirements imposed by regulatory evolution due to compliance deadlines and time-to-market pressures. The approach that we propose herein helps engineers comply with evolving regulations. Our approach consists of three parts: (1) an Adaptability Framework for understanding how and why regulations evolve, and predict which areas of the regulation may change in the future; (2) cross-reference analysis for understanding the impact cross-references have on software, as well as identifying the conflicts cross-references introduce; and (3) a methodology for modeling regulations using production rules for checking existing systems for compliance. We validated our approach using regulations from two domains: healthcare and finance. In addition, we present the findings of an empirical study in which we find that software practitioners are not well-equipped to understand the impact of compliance requirements on software.
Reasoning About Legal Text Evolution for Regulatory Compliance in Software Systems

by
Jeremy C. Maxwell

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina
2013

APPROVED BY:

______________________________  ________________________________
Annie I. Antón                  Jon Doyle
Co-chair of Advisory Committee  Co-chair of Advisory Committee

______________________________  ________________________________
Laurie Williams                Peter Swire

______________________________
Julie Earp
DEDICATION

To Jesus Christ, the Pearl of Great Price, and
To my wife my best friend.
BIOGRAPHY

Jeremy C. Maxwell grew up in Southwest Florida and received his Bachelor’s of Science in Computer Science from Florida Gulf Coast University in 2007. From there, he then enrolled in the Masters of Science in Computer Science program at North Carolina State University (NCSU). All first year graduate students at NCSU have to take a lecture series course. During one of these lectures, Dr. Annie Antón presented on her recent research in compliance, security, and privacy. This piqued Jeremy’s interest in these topics. After enrolling in Dr. Antón’s graduate-level software engineering course in the Spring of 2008, he joined her research group and applied for the Ph.D. program. His research has focused on compliance with healthcare privacy and security regulations, focusing on elements of regulations that evolve or introduce conflicting requirements. He has several publications in the premier conferences and journals in his field.

Jeremy spent the summer of 2008 as a Developer Intern at IBM and completed his Masters degree in 2009. In preparation for the birth of his first child, Jeremy transitioned to working full-time at Allscripts Healthcare Solutions in 2010. As the Application Security Architect, he is responsible for application security for an 80+ product portfolio, including ensuring security, privacy, and legal compliance at all phases of the software development lifecycle. Since 2012, Jeremy has served as Vice Chair of the Security & Privacy Work Group at the HIMSS Electronic Health Record Association (EHRA).

Jeremy is joyfully married to his best friend and has one son. He enjoys spending time with his family and following the Carolina Panthers.

Related Appointments & Employment History

- (2012-present) Vice Chair, Privacy & Security Workgroup, HIMSS Electronic Health Record Association (EHRA)
- (2010-present) Application Security Architect, Allscripts Healthcare Solutions
- (2008-2010) Graduate Research Assistant, North Carolina State University
- (2008) Staff Assistant, Public Policy Task Force, North Carolina State University
- (2007) Teaching Assistant, North Carolina State University

Supporting Publications (Chronologically Ordered)


ACKNOWLEDGEMENTS

When accomplishing an undertaking of this magnitude, there are many that are deserving of gratitude and acknowledgement. In many ways, I am the proverbial dwarf standing on the shoulders of giants. So I would like to give acknowledgement:

• To Jesus Christ, in whom we live, move, and have our being.

• To my beautiful wife, without whose unwavering support I would have thrown in the towel. Thank you for all the dishes and laundry and sleepless nights with the baby that you handled so I could work on school.

• To my father, Donald. Your hard working, even-keeled example taught me to work hard for my family, never complaining, and making the most of the time that we have.

• To my mother, Linda. Whenever I asked you a question as a kid, you always told me to look it up. I usually did. Without nurturing that sense of curiosity and knowledge exploration as a child, I wouldn’t have pursued something that required me to “look up” so many things.

• To my brother, Justin. Your sacrifices as a soldier make mine seem pale in comparison.

• To my advisor, Annie. Six years ago I came into your office and told you that a kid from Podunk, FL didn’t deserve to be there. With your advice and guidance, I suppose I’ve proven myself wrong.

• To my committee members, who guided and challenged me along the way.

• To Seth Cutler, who showed me that academia and industry are not mortal enemies, and who encouraged me to pursue both.

• To ThePrivacyPlace research group, in particular Aaron and Jessica. We began the PhD road together. I was late to the party, but I finally arrived. Also thanks to Ramya, Travis, Paul, Maria, Christopher, Jeffery, and Gurleen.

• To my sources of financial funding, including the NSF (NSF ITR grant #0325269, NSF Cyber Trust grant #0430166, and NSF SoD grant #0725144) and Allscripts Healthcare Solutions.

• To Ingrid Mattson, whose invaluable legal research helped me to understand what lawyers are talking about.
# TABLE OF CONTENTS

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
</tbody>
</table>

## CHAPTER ONE  INTRODUCTION

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Software Requirements within Sociotechnical Systems</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Legal Background</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Definitions</td>
<td>9</td>
</tr>
<tr>
<td>1.4 Legal Texts Analyzed</td>
<td>10</td>
</tr>
<tr>
<td>1.5 Outline of Remaining Chapters</td>
<td>10</td>
</tr>
</tbody>
</table>

## CHAPTER TWO  BACKGROUND

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Legal Compliance in Requirements Engineering</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Software and Compliance Requirements Evolution</td>
<td>16</td>
</tr>
<tr>
<td>2.3 Knowledge Representation and Legal Compliance</td>
<td>18</td>
</tr>
<tr>
<td>2.4 Chapter Summary</td>
<td>20</td>
</tr>
</tbody>
</table>

## CHAPTER THREE  ADDRESSING REGULATORY EVOLUTION

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Adaptability Framework</td>
<td>22</td>
</tr>
<tr>
<td>3.1.1 Adaptability Rationale</td>
<td>22</td>
</tr>
<tr>
<td>3.1.2 Adaptability Taxonomy</td>
<td>27</td>
</tr>
<tr>
<td>3.1.3 Adaptability Heuristics</td>
<td>29</td>
</tr>
<tr>
<td>3.2 Cross-Reference Analysis</td>
<td>31</td>
</tr>
<tr>
<td>3.2.1 Constraint Cross-References</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Exception Cross-References</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 Definition Cross-References</td>
<td>35</td>
</tr>
<tr>
<td>3.2.4 Unrelated Cross-References</td>
<td>36</td>
</tr>
<tr>
<td>3.2.5 Incorrect Cross-References</td>
<td>37</td>
</tr>
<tr>
<td>3.2.6 General Cross-References</td>
<td>37</td>
</tr>
<tr>
<td>3.2.7 Prioritization Cross-References</td>
<td>38</td>
</tr>
<tr>
<td>3.3 Production Rule Modeling</td>
<td>39</td>
</tr>
<tr>
<td>3.3.1 Creating Production Rule Models</td>
<td>39</td>
</tr>
<tr>
<td>3.3.2 Using Production Rule Models</td>
<td>46</td>
</tr>
<tr>
<td>3.4 Chapter Summary</td>
<td>50</td>
</tr>
</tbody>
</table>

## CHAPTER FOUR  RESEARCH DESIGN & VALIDATION

<table>
<thead>
<tr>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Case Selection</td>
<td>51</td>
</tr>
<tr>
<td>4.1.1 Case #1: Adaptability Study</td>
<td>52</td>
</tr>
<tr>
<td>4.1.2 Case #2: Cross-References Study</td>
<td>53</td>
</tr>
<tr>
<td>4.1.3 Case #3: Modeling Study</td>
<td>55</td>
</tr>
<tr>
<td>4.2 Measuring the Cost of Compliance and Noncompliance</td>
<td>56</td>
</tr>
<tr>
<td>4.3 User Study Design</td>
<td>56</td>
</tr>
<tr>
<td>4.3.1 Materials</td>
<td>57</td>
</tr>
</tbody>
</table>
4.3.2 Participants ................................................................. 58
4.3.3 Research Methodology .................................................... 58
4.4 Chapter Summary ............................................................... 59

CHAPTER FIVE FINDINGS ....................................................... 60
5.1 The Effect of Legal Evolution ............................................... 60
5.2 The Effect of Cross-References ......................................... 64
  5.2.1 Identifying Conflicting Requirements .............................. 66
  5.2.2 Resolving Conflicting Requirements .............................. 70
5.3 Modeling Regulatory Texts and Checking Requirements for Compliance 72
  5.3.1 Production Rule Models Developed ................................ 72
  5.3.2 Compliance Requirements Analysis (CRA) Tool ................ 75
  5.3.3 Specifying Canonical Compliance Requirements ................. 77
  5.3.4 Analyzing the iTrust Requirements for Regulatory Compliance .. 78
5.4 The Cost of Compliance ...................................................... 79
5.5 User Study Results ............................................................ 80
  5.5.1 Participant Demographics ............................................ 80
  5.5.2 Participant Performance .............................................. 80
  5.5.3 Observations and Discussion ....................................... 83
5.6 Chapter Summary ............................................................... 85

CHAPTER SIX CONCLUSION .................................................... 86
6.1 Contributions ................................................................. 86
6.2 Threats to Validity ........................................................... 87
6.3 Future Work ................................................................. 88

REFERENCES ................................................................. 91

APPENDICES ................................................................. 101
APPENDIX A PROLOG PRIMER ................................................. 102
APPENDIX B CHANGES PREDICTED IN THE EHR CERTIFICATION RULE 104
APPENDIX C PRODUCTION RULES MODELING 45 CFR 164.510(a)(1)(i)(A-D) 109
APPENDIX D USER STUDY MATERIALS ..................................... 111
APPENDIX E REQ'TS IDENTIFIED IN THE HIPAA SECURITY RULE .... 121
### LIST OF TABLES

Table 3.1  Legal Cross-Reference Taxonomy ............................................. 34  
Table 3.2  Hohfeldian Concepts, Rule Patterns, and Classification Phrases ........ 42  
Table 3.3  Requirement Template Modalities .............................................. 48  
Table 4.1  Cases and Scope ........................................................................ 52  
Table 4.2  Legal Texts Studied .................................................................... 52  
Table 4.3  Number of Cross-References Identified ......................................... 55  
Table 5.1  Cases and Artifacts Discovered in Each ........................................ 61  
Table 5.2  Adaptability Rationale in the Multi Case Study ............................... 61  
Table 5.3  Adaptability Taxonomy Classifications in the Multi Case Study ........ 62  
Table 5.4  Actual Legal Text Changes Identified ........................................... 62  
Table 5.5  Accuracy, Specificity, Precision, and Recall in the Adaptability Study ... 63  
Table 5.6  Results from Applying the Cross-Reference Taxonomy .................... 64  
Table 5.7  Conflicting Requirements ............................................................. 69  
Table 5.8  Resolving Time Period Conflicts .................................................. 70  
Table 5.9  Hohfeldian Concepts in the HIPAA Privacy & Security Rules ............ 73  
Table 5.10 Types of Canonical Compliance Requirements in the the HIPAA Security Rule ............................................................................. 77  
Table 5.11 Person Hours Spent in Multi-Case Study ....................................... 80  
Table 5.12 Participant Demographics ............................................................ 81  
Table 5.13 Possible User Study Outcome for a Single Question ....................... 82  
Table 5.14 Participant Responses by Question, as Percentage of Question Responses . 83  
Table B.1  Changes Predicted in the EHR Certification Rule ............................. 108  
Table E.1  Requirements Identified in HIPAA Security Rule ............................. 128
LIST OF FIGURES

Figure 1.1 Levels of User Interaction .................................................. 5
Figure 3.1 Approach to Addressing Regulatory Evolution .......................... 23
Figure 3.2 Adaptability Framework ......................................................... 24
Figure 3.3 Cross Reference Types .......................................................... 33
Figure 3.4 Production Rule Framework Overview ....................................... 40
Figure 3.5 Production Rule Modeling Methodology Overview ....................... 41
Figure 3.6 §164.510(a)(1)(A-D) of the HIPAA Privacy Rule ......................... 43
Figure 5.1 Unique Pattern-C and Pattern-D Cross-References in the HIPAA Privacy Rule .......................................................... 65
Figure 5.2 Example Structural Ambiguity ................................................ 75
Figure 5.3 Compliance Requirements Analysis (CRA) Tool - Rule Viewer ........ 76
Figure 5.4 Example Misuse Case ............................................................. 78
CHAPTER
ONE

INTRODUCTION

And what he greatly thought, he nobly dared.
—Homer

Requirements engineers need tools and techniques to adapt to regulatory evolution. A regulatory text can change as often as once a year [91], requiring potentially critical modifications to software. Software vendors have strong incentives to be first to market when these changes occur. However, these changes lead to expensive rework of legacy systems that were designed prior to the new legal requirements being announced, and it is well known that software becomes increasingly brittle and the costs of software development increase as software changes and evolves [58, 94, 96]. The goal of this work is to provide a framework that software developers can use to: (a) classify and predict the evolution of regulatory texts, (b) identify conflicting requirements across legal cross-references, and (c) evaluate existing software requirements for compliance against evolving regulatory texts. To support this goal, we ask the following research questions:

$RQ_1$: Can engineers predict which areas of a legal text are likely to evolve?

$RQ_2$: How do cross-references impact software requirements?

$RQ_3$: Can engineers evaluate compliance requirements against evolving legal texts?

$RQ_4$: Do compliance analysis activities add cost to software development that is likely to exceed the cost of noncompliance?

Research question $RQ_1$ helps us evaluate how legal texts evolve and the impact this evolution has on software requirements (part (a) of our goal statement). Because legal evolution has to be tracked across cross-references, $RQ_2$ investigates how cross-references impact software development (part (b) of the goal statement). Engineers must understand how to adapt their systems in respond to legal evolution to remain in compliance, as we examine in $RQ_3$ (part
(c) of the goal statement). Finally, the cost of compliance should be cheaper than the cost of noncompliance, as we examine in RQ4.

Many industries in the U.S. are highly regulated. For example, finance, aerospace, medical devices, and energy are all highly regulated domains. The healthcare information technology (IT) domain is in the process of becoming highly regulated. In the mid-nineties, no regulations directly governed healthcare IT. Beginning with the Healthcare Portability and Accountability Act (HIPAA) of 1996, healthcare IT has since become increasingly regulated. Consider the following history of recent regulations in the healthcare industry.

- **HIPAA (1996)** - Regulates, among other things, how protected health information (PHI) may be used and how it must be protected. Covered entities (most healthcare providers, health plans, and healthcare clearinghouses) and business associates (entities that provide services on behalf of covered entities) must comply with the regulations promulgated pursuant to HIPAA.
- **ANSI 5010** (2009) - Updated the Transactions and Code Set rule in HIPAA, requiring providers, payers, and other covered entities to update the standards used when performing electronic transactions such as claims, payments, eligibility requests, referrals, and health plan enrollments and disenrollments.
- **ICD-10** (2009) - Similar to ANSI 5010, updated the procedure and diagnosis codes that providers and other covered entities must use when filing claims and performing other electronic healthcare transactions.
- **HIPAA Breach Notification Rule** (2009) - Requires that covered entities disclose breaches of unsecured PHI.
- **Medicare and Medicaid EHR Incentive Programs**, a.k.a., Meaningful Use (Stage 1, 2010; Stage 2, 2012; subsequent stages, TBD) - The program established under the Health Information Technology for Economic and Clinical Health (HITECH) Act. HITECH provides incentives to providers that adopt electronic health record (EHR) systems and use them in a meaningful way by demonstrating they can meet clinical quality measures.
- **Electronic Prescription for Controlled Substances** (2010) - Administered by the Drug Enforcement Agency (DEA), specifies controls that must be in place before a healthcare

---

1. 45 CFR Parts 160, 162, and 164
2. 45 CFR 162
3. 45 CFR 162
6. Public Law 111-5, 123 Stat. 115
7. 21 CFR Parts 1300, 1304, 1306, and 1311
provider can electronically prescribe certain controlled substances such as opioid pain relievers.

- **Patient Protection and Affordable Care Act** (2010) - A substantial overhaul of the U.S. healthcare system signed into law by President Barack Obama. This act seeks to move the industry from a “fee for service” model—where healthcare providers are incentivized with each clinical encounter, lab order, and test—to a “pay for performance” model—where providers are incentivized to improve health outcomes.

- **HIPAA Omnibus Rules** (2013) - Expanded the requirements of HIPAA to include business associates and increased the penalties for noncompliance.

Each of these regulations impact healthcare IT, requiring that providers and vendors keep up with this rapid pace of change.

In the United States, regulations are issued by federal agencies that regulate particular domains. For example, the Department of Health and Human Services (HHS) regulates healthcare-related industries. When a regulatory agency seeks to issue a new regulation, the agency will first issue a proposed rule or a notice of proposed rule making (NPRM). Except in emergencies, the public will then be given the opportunity to comment on the proposed rule. The regulatory agency then issues a final rule that is binding on the regulated domain.

Oftentimes, regulatory deadlines are too compressed and do not allow engineers the luxury of waiting for final regulations to be published. For example, as discussed above, the American Recovery and Reinvestment Act of 2009 (ARRA) created the Meaningful Use (MU) program that makes $23 billion in incentives available for healthcare providers that adopt certified EHR technology and use it in a meaningful way. The incentives are paid out over three Stages\(^9\) that require providers to meet increasingly intensified clinical quality criteria. For example, one of the clinical quality criteria concerns patient engagement; for each stage of MU, providers must engage a greater portion of their patients and in more ways. As part of the criteria, EHR technology must be updated during each stage of MU to enable physicians to document, track, and submit the clinical quality criteria. The proposed rule for MU Stage 1 was released on January 13, 2010, whereas the final rule was issued on July 28, 2010. Eligible providers and hospitals could begin applying for Stage One incentives on January 1, 2011. Engineers that waited until the final rule was released were left with less than six months to adapt their EHRs to meet the MU Stage 1 requirements, have their EHR certified, and installed at physician practices and hospitals. EHR vendors have stated that these timeframes are too short [29]. The inability to comply with short compliance timeframes is not limited to EHR vendors. For

---

\(^8\)Pub. Law 111-148

\(^9\)Recently, HHS indicated in the MU Stage 2 NPRM that they are considering additional stages of MU beyond Stage 3.
example, the original GLBA Financial Privacy Rule proposed a compliance date six months after the final rule was published; commenters responded that this timeframe was too short to comply with the rule, and instead asked for a 12 to 24 month compliance timeframe.

Because market forces so often compel software organizations to be the first to market, they must begin complying with regulations before the final rule is published. This introduces significant ambiguity and uncertainty when building software that must comply with these changing legal requirements. Our work seeks to provide engineers with tools and techniques to develop software within changing regulatory environments. Specifically, we: provide a framework for identifying how and why regulations change; heuristics to predict regulatory change; a taxonomy for classifying the impact that legal cross-references have on software requirements, particularly when they introduce conflicts; and a modeling technique for evaluating existing requirements for compliance. In addition, an empirical study provides insights into how well software practitioners understand evolving regulatory requirements.

It is important to note that our work is not intended to replace legal domain experts. Engineers should seek advice from legal domain experts to understand and apply compliance requirements. In addition, not all ambiguity can be removed from regulations or the regulatory process, since legal interpretations will depend on organizational appetite for legal risk.

1.1 Software Requirements within Sociotechnical Systems

Software does not exist in a vacuum. Software exists to meet the needs of stakeholders and provide value to society. When users interact with a software system, they interact on multiple levels (see Figure 1.1). Individuals do not interact directly with software. Software is an abstraction; the user’s mental model of the system, the mapping between actions and results, and feedback from actions taken influence the user’s experience [86]. Through hardware, the user directly interacts with the visible manifestation of software—pixels on screens, input devices, sounds played through speakers, and even larger, more complex hardware such as industrial systems. Behind the user’s interaction with the software is the user’s (sometimes unspoken) needs, wants, desires, fears, beliefs, and personal experiences. This rich personal context is the pretext for user satisfaction, or dissatisfaction, with the immediate human-computer interaction (HCI). Behind the immediate, however, is a broader social context. The user’s perception of the immediate human-computer interaction is influenced by their culture, societal upbringings, scientific knowledge, and religious & national ambitions, dreams, innovations & ideals. In other words, judging whether the software meets its requirements cannot be divorced from the social context in which it is deployed. Software is a sociotechnical system\textsuperscript{10} that has both technical

\textsuperscript{10} Some researchers use the term sociotechnical to indicate the workplace-related collaboration between software developers (e.g., [80]). Our use of the term sociotechnical is not limited to just software developers, but also
and social elements [1, 90, 116, 126].

![Figure 1.1: Levels of User Interaction](image)

Because software is a sociotechnical system, it co-evolves with the social context in which it is developed and deployed. A literature search for “software spontaneous generation” returns zero results, because software does not spontaneously change. Software co-evolves with changes that occur in an interaction level (see Figure 1.1). Software may change: (1) to meet changing software constraints, for example, a commercial off the shelf (COTS) vendor end-of-lifes a component used by the software; (2) to meet changing hardware constraints, for instance, the innovation of new displays led to modern touch-screen mobile device operating systems; (3) to meet changing stakeholder needs arising from immediate HCI with the software (a.k.a., market requirements); or, (4) to meet changing societal needs [60, 95]. In democratic societies, regulations are an expression of the policy of elected leaders, of collective societal needs. By constraining behavior, regulations encourage trust between producers of technology and users of technology [33]. Thus, the focus of this work is to study how software co-evolves with changing societal needs expressed through regulation.

### 1.2 Legal Background

In this section, we provide a legal background for our work. In the United States, the Administrative Procedure Act sets forth the steps in the creation of a federal regulation, also called a rule\(^{11}\). Regulations are binding legal requirements that are issued by a federal agency. Regulations implement a statute that has been passed by Congress. For example, the HIPAA Privacy

---

and Security Rules implement the HIPAA statute passed by Congress. The normal process is that the agency first issues a notice of proposed rulemaking (NPRM)\textsuperscript{12}. The public is then given a period to comment on the rule, with the comment period generally being at least 60 days\textsuperscript{13}. The agency is required to review the public comments and take them into account in drafting a final rule\textsuperscript{14}. The final rule has binding effect on the regulated industry. Both the NPRM and the final rule are accompanied by a preamble, in which the regulatory agency provides background, explanation of the rule, and responds to public comments. These agency interpretations are given “considerable weight” when interpreting sometimes ambiguous statutes [21, 41].

In cases of emergency, an agency can issue a final rule without prior public comment\textsuperscript{15}. In such instances, the agency often solicits public comments after the rule is issued\textsuperscript{16}. Parties affected by a rule in most instances have a right to appeal the rule to federal court\textsuperscript{17}. The court may uphold the rule, or find that the rule is substantively illegal (e.g., the rule did not accurately implement the statute) or procedurally illegal (e.g., the rule and accompanying explanation provided by the agency did not adequately respond to the public comments).

Once a final rule is published, an organization must demonstrate legal compliance with the requirements of the rule. To demonstrate legal compliance, an organization must, at a minimum, demonstrate due diligence (reasonable efforts to satisfy compliance requirements), good faith (acting honestly without an intent to deceive or perpetrate fraud), and a standard of care (reasonable actions to reduce risk) [16]. Even though we focus exclusively on federal regulations in this work, federal regulations often establish a compliance floor [57]. Organizations must consider compliance requirements from other jurisdictions, such as state privacy and security laws or international laws such as the European Union’s Data Protection Directive\textsuperscript{18} [57].

Organizations may fail to comply with a regulation due to: a lack of knowledge of the regulation; a lack of willingness to comply with the regulation; or a lack of ability to comply with the regulation [89]. When organizations fail to demonstrate compliance, they may face an enforcement action by the regulatory agency. The result of enforcement actions may be severe. For example, enforcement actions for HIPAA violations have included:

- $2.25 million fine of CVS, with third party audits every two years, for improper disposal of paper records\textsuperscript{19}
- $1.7 million fine of the Alaska Department of Health and Social Services for a lost thumb

\textsuperscript{12} 5 U.S.C. §533(b) (2006)
\textsuperscript{14} 5 U.S.C. §§603-604
\textsuperscript{15} 5 U.S.C. §608
\textsuperscript{16} 5 U.S.C. §608.
\textsuperscript{17} 5 U.S.C. §553(e).
\textsuperscript{18} http://ec.europa.eu/justice/data-protection/index_en.htm
\textsuperscript{19} http://www.scmagazine.com/cvs-to-pay-225-million-to-settle-hipaa-violation/article/127570/
drive containing patient data

- $20,000 per physician fine of a five physician practice for failing to obtain business associate agreements from contractors

- $1.5 million fine of BlueCross BlueShield of Tennessee (BCBST) for the theft of servers containing patient records

Oftentimes, the fine resulting from the enforcement action is a small percentage of the overall cost of noncompliance. For example, BCBST ended up paying over $17 million dollars in forensics, breach notification, and correction action plans. In addition to action by regulatory agencies, sometimes enforcement authority is delegated. For example, the HITECH Act expanded the enforcement authority under HIPAA, allowing state attorney generals to pursue HIPAA enforcement cases as well. In the first enforcement action by a state attorney general, the Minnesota attorney general fined a business associate for $2.5 million and banned them from doing business in the state of Minnesota for 2 years. If an organization wishes to challenge an enforcement action, it may appeal to a judicial body—a ruling that then becomes a precedent as part of case law.

Legal texts contain cross-references; cross-references are citations from one portion of a legal text to another portion of that text or to another text. The referencing text is the legal text that contains the cross-reference and the referenced text is the legal text that is cited. Cross-references establish relationships and priorities among laws at various levels. These laws are codified in several places at the federal, state, and local levels, but there is no comprehensive legal code. For instance, the complex legal structure of federalism governs when federal or state law takes precedence. At both the federal and state level, the complex legal structure of separation of powers governs how power is allocated among the three branches of government, with statutory law developed by legislative bodies (such as Congress), administrative law (such as regulations) issued by executive agencies, and judicial branch decisions that become case law.

In our studies, we identify references to the U.S. Code, the Code of Federal Regulations, and to Executive Orders issued by the President. The U.S. Code is a compilation of legislative law passed by the U.S. Congress. The U.S. Code is divided by subject into 50 titles. For example,
Title 42 relates to Public Health and Wellness, whereas Title 15 relates to Commerce and Trade. Citations to the U.S. Code are formatted as <Title-Number>U.S.C. <Section Number>. For example, 22 U.S.C. 2709(a)(3) is a citation to Title 22, section 2709, subsection (a), paragraph (3). Sometimes, cross-references will cite a statutory law by its commonly used name, instead of using the U.S. Code title and section number. The U.S. Code contains a table of “Acts Cited by Popular Name” that can be used to determine the title and section numbers for these laws [24]. For example, the HIPAA Privacy Rule cites the Foreign Service Act. Using the “Acts Cited by Popular Name” table, we determine that the Foreign Service Act is codified at 22 U.S.C. 3901 et seq. The Code of Federal Regulations (CFR) is a compilation of regulations published by federal agencies such as the Department of Health and Human Services (HHS) for HIPAA or agencies including the Federal Reserve and the Securities and Exchange Commission for GLBA [24]. The CFR is divided by subject into 50 titles but does not use the same subject divisions as the U.S. Code. The CFR is cited similar to the U.S. Code. For example, 42 CFR 493.3(a)(2) is a citation to Title 42, section 493.3, subsection (a), paragraph (2). An Executive Order is an “exercise of presidential authority related to government business” with sequential numbering in the order they are issued [24]. For example, §164.512(k)(2) of the HIPAA Privacy Rule cites Executive Order 12333, which relates to intelligence activities. A fourth area of law—case law—is created by judicial decisions in response to court cases. Case law provides further explanation and interpretation of statutes and regulations. Herein, we do not study this area of law for two reasons. First, we did not encounter any references to case law in the regulations we studied. Second, case law is expansive, requiring multiple areas of expertise. For instance, an expert in tax law is unlikely to be familiar with employee privacy laws. Software engineers are encouraged to work with legal domain experts to understand the implications of case law in their specific regulated domain.

The legal texts used herein can be accessed through publicly-available sources. For U.S. Code citations, we used Cornell University Law School’s U.S. Code Collection [25]. We used the Popular Name Tool maintained by the U.S. Office of the Law Revision Counsel [26] to lookup laws in the “Acts Cited by Popular Name” table. For citations to the Code of Federal Regulations, we used the e-CFR [27] maintained by the U.S. Government Printing Office. For Executive Orders, we used two resources: the American Presidency Project [28] hosted by the University of California, Santa Barbara and The Codification of Presidential Proclamations and Executive Orders [29] at the U.S. National Archives. The latter resource only contains Executive Orders issued between April 13, 1945 and January 20, 1989, requiring us to use the American Presidency Project for

---

Executive Orders issued outside this date range.

To model concepts in the law, we employ Hohfeld’s eight legal concepts [49]. These concepts are:

- **right** - A claim an actor makes that places obligations on other actors.
- **obligation** - An action an actor is required, by law, to carry out. Hohfeld calls these duties.
- **privilege** - An action an actor is allowed to perform but not required to perform.
- **no-right** - A right that an actor does not hold.
- **power** - The capability of an actor to change a legal relation and place liabilities on others.
- **liability** - A responsibility to perform some action.
- **immunity** - Just as a privilege signifies that an actor is free from an obligation, an immunity expresses that an actor is free from a legal power. In other words, an immunity is a no-liability.
- **disability** - Whereas a no-right express that another party does not hold a right, a disability expresses that a party does not hold a legal power.

The Hohfeldian concepts have been used in previous requirements engineering research [17, 111].

### 1.3 Definitions

In this work, we use the following definitions:

- **An actor** is a person, organization, or other entity explicitly defined or referenced in a legal text.

- A **compliance requirement** is a software or organizational requirement that enables an organization to comply with a government regulation.

- A **legal statement** is a sentence or sentence fragment in a regulation.

- A **software requirement** is a statement expressing the desires of a stakeholder concerning the software being developed [132].
1.4 Legal Texts Analyzed

In this work, we analyze the following legal texts, and reference them throughout.

- Health Insurance Portability and Accountability Act of 1996 (45 CFR Parts 160, 162, and 164) - Otherwise known as HIPAA, the regulations promulgated pursuant this act regulate how certain organizations—called covered entities—can handle protected health information (PHI). Examples of covered entities include hospitals, physician practices, and health insurance plans. We examine two rules issued by the US Department of Health and Human Services (HHS): the Privacy Rule that regulates how organizations can use health information; and the Security Rule that regulates the controls that must be in place to safeguard electronic health information. The HITECH Act of 2009 expanded the scope of the HIPAA Privacy and Security Rules to include business associates—an entity that performs services on behalf of a covered entity. Examples of business associates include EHR vendors, organizations providing claims processing services, and medical transcriptionists.

- Health Information Technology: Initial Set of Standards, Implementation Specifications, and Certification Criteria for Electronic Health Record Technology (45 CFR 170) - Hereafter, referred to as the EHR Certification Rule, this rule outlines the requirements that EHRs must satisfy in order to be certified under Stage 1 of the Meaningful Use program. The Office of the National Coordinator for Health Information Technology (ONC), a division within HHS, regulates the EHR certification process.

- Gramm-Leach-Bliley Act, Disclosure of Nonpublic Personal Information (Pub.L. 106-102, Title V, Subtitle A; codified at 15 USC, Subchapter I, Sec. 6801-6809) - A portion of the Gramm-Leach-Bliley Act that places requirements on how financial institutions may use and disclose consumer financial information. Hereafter, this act will be referred to as the Gramm-Leach-Bliley (GLB) Act.

- Privacy of Consumer Financial Information (16 CFR Part 313) - One of the rules promulgated pursuant to the Gramm-Leach-Bliley Act by the Federal Trade Commission (FTC), this rule protects consumer financial information held by financial institutions. Referred to hereafter as the GLBA Financial Privacy Rule.

1.5 Outline of Remaining Chapters

The remainder of this work is outlined as follows:
Chapter 2 provides a summary of background and a survey of related work. Related work is categorized into three areas. First, we describe prior work done by computer science researchers exploring how engineers can develop regulatory compliant software. Second, we discuss prior work in regulatory and software evolution. Finally, we present related work in knowledge representation and modeling of regulations.

Chapter 3 introduces our approach for addressing regulatory evolution in software engineering, which consists of an Adaptability Framework, a cross-reference taxonomy, and production rule modeling. The Adaptability Framework contains rationale for why regulations change, classifications for how regulations change, and heuristics to predict that regulations may change. The cross-reference taxonomy helps engineers understand the impact of cross-references on software. Production rule modeling captures compliance requirements as logic rules that can be queried to check existing software requirements for compliance.

To accomplish this research goal outlined in the introduction, Chapter 4 provides an overview of our research methodology. We performed two studies. First, we performed a multi-case study using the five legal texts that are summarized in Section 1.4. We developed our approach for addressing regulatory evolution from our multi-case study. Second, we performed an empirical study of software engineers’ ability to classify cross-references using our cross-reference taxonomy.

Chapter 5 presents the results from our multi-case study and our user study. We discuss the effect of regulatory evolution and the performance of our predictive heuristics in forecasting regulatory change. We also discuss how our cross-reference taxonomy can be used to facilitate the identification of conflicting compliance requirements and describe five instances of conflicting requirements from the HIPAA Privacy Rule. Next, we provide an overview of the production rule models that we developed, a set of tooling for using the models, and our results from using the models to check an open-source EHR for regulatory compliance. Finally, we discuss the findings of our empirical study.

Chapter 6 summarizes the contributions of this work to the scientific body of knowledge. We also discuss threats to the validity of our findings and our strategies for mitigating these threats. Lastly, we discuss future research directions that build upon the work presented herein.
Do you desire to construct a vast and lofty fabric? Think first about the foundations of humility. The higher your structure is to be, the deeper must be its foundation.

—St. Augustine of Hippo

In this chapter, we review prior research in the areas of software engineering & regulatory compliance.

2.1 Legal Compliance in Requirements Engineering

Regulatory compliant software, by definition, meets the constraints placed on it by regulations that govern the target environment. Requirements describe the environment of the system to be built [52], so it is important for requirements engineers to examine the appropriate laws that regulate the target environment to extract compliance requirements.

Several researchers adopt a formal methods approach to legal compliance in requirements engineering. Breaux et al. use the Frame Based Requirements Analysis Method (FBRAM) to extract rights and obligations from regulatory texts [16, 17, 18]. They applied their methodology to the HIPAA Privacy Rule, but focused on access control rules [17], whereas we model both the access control rules as well as the procedural rules placed on covered entities. Our ontology differs from Breaux et al.’s in that we incorporate all eight of the Hohfeldian concepts in our model (see Section 3.3), whereas Breaux et al. solely focus on rights and obligations [16, 18]. As discussed in Section 1.2, the Hohfeldian concepts create a framework for distinguishing between fundamental legal concepts and how these concepts are related to one another to surface policy implications. We previously used the three concepts of rights, obligations, and privileges to model the legal text [69] and subsequently expanded our ontology to include all eight of Hohfeld’s concepts [71]. We modify two techniques from Breaux et al.’s methodology when creating production rule
models [71]. First, in normative phrase analysis, legal text statements are classified based on the natural language phrases in the statement [16, 18]. We expand normative phrase analysis to include all eight of Hohfeld’s concepts (discussed in Section 3.3), not just rights and obligations. Second, Breaux et al. employ rights and obligations balancing to identify implied rights and obligations [18], that is, a right of one group imposes an obligation on others. We expand rights and obligation balancing to accommodate additional legal concepts (see Section 3.3.1). Hassan and Logrippo extract compliance requirements from law using an approach based on the Unified Modeling Language (UML) [47]. Once extracted, they model these requirements in Alloy, a logic-based modeling language. They specify categories of legal statements, such as declarative statements (declaring a fact) and procedural statements (if-then statements) [47]. May et al. formalize a section of HIPAA (§164.506) using Privacy APIs and then use the Spin model checker to find loop holes and perform other queries on the law [76]. They use public comments from an earlier version of HIPAA to test their model [76]. Like production rule models, their model is queryable. However, their model only covers one section of HIPAA, whereas we have modeled the entire HIPAA Privacy and Security Rules. All of these researchers adopt a formal methods approach to addressing regulatory compliance in software. However, requirements engineers oftentimes do not have the luxury of creating formal models due to tight regulatory timelines [29].

There have been two goal-oriented approaches to legal compliance in the requirements engineering literature. The Ghanavati et al. approach [34, 35] is based on the User Requirements Notation (URN), which is composed of use cases and an i*-like goal notation [131]. They use URN to model business practices and legal texts, and compare the two models using traceability links to identify potential areas of noncompliance [34, 35]. In contrast, we go beyond traceability between models by maintaining traceability across all requirements artifacts back to the legal text. The Siena et al. framework, called Nomos [110, 111], is similar to our approach in that they model the law using Hohfeld’s eight legal concepts, formalizing the concepts using deontic logic [111]. They then extract the concepts from the law and model them using a goal-oriented approach based on i*. However, Siena et al. do not analyze the Hohfeldian concepts individually; instead, they combine each of the correlative concepts into a single predicate [111]. As discussed in our previous work [71], failing to analyze all the concepts can overlook compliance requirements. Additionally, Siena et al. do not provide guidance to software engineers about how to extract the concepts from the law, and only demonstrate their methodology on a textbook example. In contrast, we provide prescriptive guidance for extracting legal concepts from the law [71] and have successfully applied our approach to an existing system [70, 72].

Hoffman et al. specify requirements patterns from high level laws with the conjecture that they can be reused for other regulations potentially governing different domains [48]. This is unlike our approach, where we specify canonical compliance requirements—using a template
based approach—that can be reused across organizations that must comply with the same regulated regime. Hoffman et al. focus on specifying general, high level requirements, such as deleting “personal data if they are no longer necessary for system operations.” However, we have determined that regulations differ significantly at these low-level details such as required retention periods on certain types of data [73]. Without capturing the low level details, important compliance requirements may be missed or requirements may conflict [73].

Researchers note that cross-references are particularly challenging for developing compliant software [3, 8, 16, 17, 18, 43, 76, 91]. Specifically, cross-references: can be ambiguous about which legal text takes precedence [16, 43, 76]; decrease understanding of legal texts [8]; add additional priorities and exceptions to compliance requirements [91]; may have a differing context from the citing text [18]; and may cite portions of the legal text out of sequence, causing “engineers to skip around the regulation text” [17].

Massey et al. use cross-references, along with other factors, to prioritize compliance requirements, but do not analyze the cross-referenced texts [65]. They classify cross-references as internal (a reference between different portions of the same legal text) or external (a reference between portions of different legal texts) [65]. However, using this simple distinction introduces ambiguity, as the boundaries between legal texts may not immediately be clear to an engineer. Thus, in Section 3.2, we further classify cross-references into one of four patterns.

To date, requirements engineering research focuses predominately on internal cross-references rather than external cross-references [16, 69, 76]. External cross-references are more challenging to legal compliance than internal cross-references, because different legal texts are likely to have differing context, definitions, and priorities. Breaux uses natural language patterns to identify internal cross-references in the HIPAA Privacy Rule and codify mappings between the respective compliance requirements [16]. He then extracts priorities, exceptions, and refinements to compliance requirements from the identified cross-references [16]. External cross-references are outside the scope of Breaux’s study [16]. May et al. use Promela to express the HIPAA Privacy Rule, including internal cross-references [76]. However, they do not analyze external cross-references [76]; instead, they use environmental flags to signal whether or not an external cross-reference is satisfied [76].

Van Engers and Boekenoogen use scenarios and the Unified Modeling Language (UML) to detect errors in the law and improve legal text quality [120]. They obtain scenarios by interviewing legal domain experts and model sequences of events using decision trees [120]. They do not, however, capture important contextual information about the scenario such as the scenario’s goal, actors involved, and resources needed. After obtaining these scenarios, Van Engers and Boekenoogen analyze a draft of a bill that, at the time, was going through the Dutch law-making process [120]. They identify and describe four inconsistencies in the law, namely, incorrect cross-references, ambiguous references, gaps in the law, and irrelevant portions of the
law [120]. Using our legal cross-references taxonomy, we can identify all of the inconsistencies identified by van Engers and Boekenoogen.

Cholvy checks regulation consistency by modeling the regulation text using a first order language [22]. Using these formal modeling tools, Cholvy identifies potential dilemmas, i.e. a regulation both obligates and forbids an actor from performing an action [22]. Our cross-reference taxonomy can be used to identify conflicts in the law without having to formally model the regulation. In addition, our taxonomy also aids engineers in identifying refinements to existing compliance requirements and areas of the law that are not applicable to software systems, meaning they can be ignored. It is possible that our production rule framework could be used to identify similar dilemmas; exploring the nuances between Cholvy’s approach and ours remains future work.

Hamdaqa and Hamou-Lhadj present a classification scheme for legal cross-references and outline a tool-supported, automated process for extracting cross-references and generating cross-reference graphs [43, 44]. They classify cross-references into two groups, amendments and assertions [43, 44]. Amendments track evolution in the law [43, 44]. Assertion cross-references are further classified using three subtypes: definition cross-references provide a definition; specification cross-references provide more information about the legal text; and compliance cross-references conform the cited text with the citing text [43, 44]. Their definition and specification subtypes align with the definition and constraint classifications in our taxonomy, respectively. Our taxonomy goes beyond their classification scheme by identifying exception, incorrect, unrelated, prioritization, and general cross-references, as well. Hamdaqa and Hamou-Lhadj use their citation analysis technique to examine the HIPAA statute, the GLBA statute, and the Sarbanes-Oxley statute [44]. During their analysis, they point out areas of potential conflict where two statutes amend the same area of the law [44]. However, in our experience, regulations promulgated according to these statutes use prioritization cross-references to establish compliance priority among the regulatory frameworks [75].

Requirements engineers have used compliance and traceability links in compliance research [10, 23, 35]. As previously mentioned, Ghanavati et al. use compliance links to trace goals, softgoals, tasks and actors to the law [35]. Berenbach et al. use just in time tracing (JIT) to identify: (1) regulatory requirements; (2) system requirements that satisfy said requirements; and (3) sections of the law that require further analysis [10]. Cleland-Huang et al. use automated techniques to identify traceability links between HIPAA and 10 sets of requirements specifications [23]. Herein, we present research that can be used to complement existing requirements engineering methods to identify requirements conflicts in cross-references.

Zhang and Koppaka create legal citation networks based on the citations found in case law [133]. Their tool can be used to identify and track legal issues as they evolve [133]. Their tool, however, is designed to be used by legal domain experts and is designed to support evolving
case law [133]. Case law is out of scope for our analysis, as we focus on regulatory evolution.

2.2 Software and Compliance Requirements Evolution

As software evolves and changes, it becomes increasingly brittle and the costs of software development increase [58, 94, 96]. Lehman and Ramil identify two views that must be present in a theory of software evolution: (1) the what and why view that captures the properties of software evolution, and (2) the how view that produces tools and techniques to better equip engineers to address evolution [59]. Our Adaptability Framework discussed in Section 3.1 follows this pattern, with the adaptability rationale and taxonomy providing the first view and the adaptability heuristics providing the second view. Mens et al. identify the need for predictive models of software evolution as an open question in software evolution [82]—a question that we address with our predictive heuristics (see Section 3.1.3).

Prior research has focused on how engineers can evolve software once the changing requirements have already been identified. For example, Bennet et al. put forth a service-oriented architectural model based on ultra late binding that they claim is positioned to better handle software evolution [9]. France and Bieman developed multi-view software evolution (MVSE), a framework for evolving UML models and object-oriented software [32]. Sillito et al. identifies 44 questions that software engineers ask when working on software maintenance and evolution tasks—questions that help to understand the scope and impact of the proposed change [112]. Oreizy et al. examine software architectures that can be used to change the system without having to take it offline, also called zero-downtime upgrades, for mission-critical systems such as air traffic control systems [88]. Whereas this body of literature describes how to evolve the system, our work focuses on identifying new and changed system requirements due to regulatory evolution. Our work is complementary—after evolving compliance requirements have been identified using our techniques, engineers can use the these existing techniques to implement the requirements.

Other research has focused on the phenomenon of software evolution itself by studying how software artifacts such as source code evolve. For instance, Belady and Lehman developed a series of laws of software evolution by studying the growth of the IBM OS/360 system [7]. Many researchers have revisited these laws and examined them within new contexts since Belady and Lehman’s seminal work [58, 61, 103, 104]. Parnas’ seminal work describes software aging—the phenomenon that, over time, software becomes bloated, brittle, more complex, and increasingly fails to meet user needs [94]. Software aging is a side effect of change, including regulatory change. Robbes and Lanza recommend looking at changes that are more granular than source code commits, such as instrumenting integrated development environments (IDEs) to track individual source line changes in between commits [101]. Barry et al. use empirical
analysis to identify patterns of software growth [5] and study how software process automation impacts software evolution [6]. Antón and Potts examine the evolution of telephony features [4]. Mens and Demeyer use software quality metrics to identify: evolution-critical components (components that need to be changed), evolution-prone components (components that are likely to change), and evolution-sensitive components (components that are likely to be impacted by change) [81]. Van Rysselberghe and Demeyer reconstruct the changes that were made to a system by detecting code duplication (“clones”) [122]. Ramil and Lehman examine cost estimates for software evolution [100]. Bohner developed techniques for determining which software artifacts are impacted by changes [15] as part of software maintenance and release management [14]. Others have visualized changes that are made to code [25, 37, 50]. Buckley et al. present a taxonomy that captures the various dimensions of software changes, for example, the change’s scope, whether the change impacts system availability, the rate at which change occurs, and how the change will be implemented in the system [19]. However, all this prior work focuses on software and software artifacts. In contrast, we focus on the upstream compliance requirements that constrain software requirements and other artifacts.

A significant body of work exists in the area of mining software repositories (e.g., [54, 128, 134]). This work is a retroactive look at the changes that occur to source code over time. However, changes that are made to source code may not trace back to compliance requirements. If they do, the source code changes represent one interpretation of the final rule based on the organization’s appetite for risk and understanding of the actions required to demonstrate due diligence; the post hoc analysis used in the mining software repository literature may not apply to engineers examining new regulation, as the previous interpretations and assumptions may not hold.

Some researchers have used formal methods to model software evolution [36, 62, 85, 135]. Ghose developed a formal model for handling requirements evolution, focusing on the inconsistencies that are caused by the interplay of nonfunctional and functional requirements. [36]. Lieberherr and Xiao use propagation patterns that allow engineers to reason about software changes using graph transformations [62]. Zowghi and Offen use a logic-based approach to reason about changing requirements [135]. Nikolić examines large-scale systems, such as how industrial systems impact environmental and economic systems, and provides simulation and modeling for how they evolve [85]. However, in rapidly evolving regulatory environments, engineers may not have time to build formal models to track software evolution. In addition, legal texts contain intentional ambiguity that leaves them open to future interpretation [91]; this ambiguity may make it difficult to track compliance requirements evolution using formal methods.

Requirements engineering researchers have also examined software evolution [20, 53, 87]. Carter et al. develop an evolutionary prototype model that helps engineers address the chal-
lenges associated with requirements creep but is targeted towards smaller development teams [20]. Jones recommends several techniques and processes for making requirements changes less impactful, for example, by creating prototypes or following rapid application development [53]. Nurmuliani et al. use a card sorting technique to explore how software practitioners view and organize requirements changes [87]. Given a set of 52 requirements changes written on index cards, they asked practitioners to group the cards according to their own criteria [87]. Nurmuliani et al. observe that practitioners view and organize requirements change according to their role; for example, a project manager views requirements changes according to the impact upon the project schedule whereas developers are more likely to organize requirements changes according to effort [87].

Other requirements engineering researchers have documented the ways that requirements changes occur [45, 77]. McGee and Greer developed a taxonomy that classifies the source of requirements change [77]. In their taxonomy, the market change domain includes requirements changes due to regulatory changes [77]. Our work is complementary to theirs—they determined that each domain of requirements change should have tailored management techniques [78]. They focus on generic requirements change whereas we focus on compliance requirements change. Harker et al. classify requirements as either stable or changing [45]. They further classify changing requirements into several categories [45]. According to their categories, evolving legal requirements are mutable requirements—requirements that evolve due to environmental change. They also recommend several techniques to manage requirements change [45]. Our work enables engineers to follow the spirit of their recommendations to identify the minimal set of stable requirements, and build for those requirements first [45].

Requirements researchers have examined conflicts in software requirements [12, 28, 30, 102, 118, 121]. Robinson and Fickas describe how to detect and resolve requirements conflicts using a tool-supported approach [102]. Boehm and In use the WinWin model for negotiating resolutions to conflicts among quality attributes [12]. Van Lamsweerde et al. use KAOS to identify and resolve conflicts among software goals [121]. Easterbrook and Nuseibeh use the ViewPoints Framework to handle inconsistencies as a requirements specification evolves [28]. Emmerich et al. examine ISO standards and built a prototype policy checker engine in DOORS [30]. Thurimella and Bruegge examine conflicts among the requirements of various product lines [118]. To the best of our knowledge, we are the first researchers to examine and address conflicts introduced by cross-references in regulations.

## 2.3 Knowledge Representation and Legal Compliance

Knowledge representation techniques proposed for use in requirements engineering include logic, semantic nets, frames, and production rules [27]. Logic programming and the law has been a
popular area of research in the AI and law community [8, 11, 106, 107, 109]. Research to date has failed to provide a repeatable, systematic methodology for translating legal texts into production rules, instead using a trial and error methodology [107], whereas others only present their methodology by example [8, 11, 106]. Our work differs from prior work in logic programming in a few specific ways. First, the nature of the HIPAA legislation is inherently different from that of legislation used in prior work in production rule modeling of legal texts [8, 106, 107, 109]. The legal documents used in these efforts usually seek to answer a single question. For example, in the British Nationality Act effort [107], the query considered is whether or not an individual is a British citizen. In contrast, the HIPAA Privacy Rule does not have one query that unifies the document. Instead, its broad nature allows many possible queries; for example, queries about access control and the right of notice, the interactions between organizations, etc. Moreover, there are 33 different types of stakeholders mentioned in the HIPAA Privacy Rule alone [16]. HIPAA’s broader scope makes it challenging to predict potential queries. We employ Hohfeld’s concepts (a framework for reasoning about the relationships between different types of legal concepts) to capture a broader range of potential queries on the model because all are required in order to provide a holistic view of a compliant software system [66]. Second, the purpose of prior work has not been the creation of legally compliant software. Instead, prior efforts focused on: improving the understanding of law using production rules [11]; knowledge representation research rather than practical uses of production rule models [8, 107]; aiding law makers in drafting legislation [107]; and legal reasoning [11, 106, 109].

Lam et al.’s compliance checker determines whether an email message complies with three HIPAA Privacy Rule sections before it is transmitted [56]. To accomplish this, they translated the three sections into a version of Prolog, focusing on paragraphs that address access to PHI [56]. Based on this analysis, they develop a set of eight message characteristics, used to make compliance decisions [56]. In contrast, we have modeled the entire HIPAA Privacy and Security Rules and specify a variety of legal preconditions beyond Lam et al.’s eight message characteristics [69, 71].

An expert system captures the knowledge of a human expert in a particular domain and makes it accessible for non-experts [42, 63]. Like production rules, expert systems also have a knowledge base and an inference engine to make use of that knowledge [105]. Visser et al. provide a methodology, based on the CommonKADS methodology [105], for constructing legal knowledge-based systems that seek to replicate the legal reasoning performed by a lawyer [123, 124]. Bench-Capon describes expert systems as: interactive; based on one narrow domain of expertise; having the ability to reason under uncertain or incomplete information; and capturing “rules of thumb” used by domain experts. Popple defines a legal expert system as a “system that provides answers to legal questions in a form that one would expect from a lawyer” and that “the output from [a legal expert system] should be usable without further legal analysis”
We do not seek to construct an expert system nor do we seek to replace lawyers because legal interpretation is complex, changes over time, and depends on factors beyond the text of a legal document, such as case law, industry standards and current practice, and administrative clarifications.

2.4 Chapter Summary

In this chapter, we provided background in legal compliance, evolution, and knowledge representation within the context of requirements and software engineering. In the next chapter, we present our framework for addressing compliance requirements evolution.
Ancient wisdom tells us that things in life change. Software engineers find this rule inescapable as they build their software systems. Engineers must adapt their systems as user whims, expectations, and desires change. New opportunities and innovative business models place new demands on software. Sometimes, new technology—such as the Internet or touch interfaces—completely revolutionizes the software industry. These are all the ebb and flow of healthy, free market dynamics, where companies have the opportunity to have their “one more thing”\(^1\) moment. However, regulatory change is qualitatively different. Regulatory change is mandated with deadlines artificially divorced from market forces. Companies do not have the freedom to delay compliance with regulatory changes in the same way they can reprioritize their feature-functionality backlog. Companies can lobby and participate in regulatory comment processes, but sometimes even the most engaged companies cannot influence law makers to draft regulations that are favorable to the company. Software engineers must be equipped to handle this changing regulatory landscape.

In this section, we outline our approach for addressing regulatory change. This approach consists of three components. First, in Section 3.1 we discuss the Adaptability Framework. The framework contains rationale for why regulations change, classifications for how regulations change, and heuristics to predict that regulations change. Next, in Section 3.2, we present a taxonomy for tracking cross-references across regulations. Finally, in Section 3.3, we discuss

\(^1\)A phrase frequently used by Steve Jobs at the end of presentations to announce new products and features.
a production rule modeling methodology for modeling regulations, eliciting new compliance requirements, and checking existing requirements for compliance.

Figure 3.1 provides a graphical representation of our approach. Given the input of a proposed regulation, we analyze the text using our predictive heuristics (part of the Adaptability Framework) and cross-reference analysis. Additional techniques developed by prior researchers, discussed in Section 2.1, can also be used to examine the legal text. This analysis identifies requirements that are likely to change during final rulemaking, and those that are not likely to change—enabling engineers to begin work on the latter. Next, after the final rule is released, we use the Adaptability Framework and cross-reference analysis to examine the changes. Again, existing techniques can also be incorporated in this step. Finally, we model the regulation using production rules. These models enable us to validate existing requirements for legal compliance. The output of our approach is a set of requirements that have been validated for legal compliance; engineers can use these requirements as a starting point for further analysis with legal domain experts.

3.1 Adaptability Framework

The Adaptability Framework identifies three components of each regulatory change (see Figure 3.2): adaptability rationale that capture why regulations change; an adaptability taxonomy that capture how regulations change; and a set of adaptability heuristics that software engineers can use to predict that a regulation will change.

3.1.1 Adaptability Rationale

An adaptability rationale is the stated reason why legal drafters make changes to a proposed rule before issuing the final rule. Rationales are stated in the commentary that accompanies a final ruling. The framework contains a total of thirteen rationales that we identified during our study. As indicated in Figure 3.2, a single change in a regulation may have multiple rationales.

Adaptability Rationale # 1: Ambiguity

Regulations may be amended when public comments reveal that they are ambiguous or unclear. For example, the proposed HIPAA Security Rule requires covered entities to secure their workstation locations, and provide several examples such as “not placing a terminal used to access patient information in any area of a doctor’s office where the screen contents can be viewed from the reception area” (§142.308(b)(5)). Commenters note, however, that the examples are presented in a way that makes them appear to be required. In addition, the proposed rule used the terms “workstation” and “terminal” interchangeably, and commenters said it is ambiguous.
whether covered entities have to secure other types of workstations such as laptops. In the final HIPAA Security Rule, this statement is generalized to require that covered entities “implement physical safeguards for all workstations that access electronic protected health information, to restrict access to authorized users” (§164.310(c)) The examples are removed from the final rule and the term “terminal” is removed.

Adaptability Rationale # 2: Format and Organization

Regulations are subject to format and organizational changes as they are updated. For example, the HIPAA Security Rule was renumbered from 45 CFR 142 to 45 CFR 160-164.
Adaptability Rationale # 3: Technology-Specific Elements

When regulations mandate that certain technologies be adopted, industry innovation is stifled because businesses are disincentivized to improve the state of the art. In addition, government mandated technologies can actually lead to security vulnerabilities in critical systems such as EHRs if security flaws are identified in the mandated technologies. For example, the proposed HIPAA Security Rule mandates that covered entities adopt access controls from a list of technologies that include context-based access control, role-based access control, or user-based access control. However, this discourages innovation of other types of access controls. Thus, the final HIPAA Security Rule removes this list of access control technologies.

Adaptability Rationale # 4: Responses to Questions Posed

Sometimes, a regulatory agency will pose questions to the public. In response to those questions, the agency may update the regulation in final rulemaking. For example, in the proposed GLBA Financial Privacy Rule, the FTC proposes two definitions for “nonpublic personal information” and asked the public to comment on the alternatives. In response to the feedback received, the FTC adopted a hybrid definition combining concepts from both alternatives.

Adaptability Rationale # 5: Inappropriate for Domain

Sometimes, a regulation will specify requirements that are inappropriate for the domain it regulates. For example, the proposed EHR Certification Rule contains requirements that would require EHRs to perform administrative functions such as submitting insurance claims. However, this functionality is typically performed by a practice management system, not by an EHR. Thus, ONC removed these requirements before issuing the final rule.
Adaptability Rationale # 6: Potential Conflict

Regulations can contain requirements that conflict with each other [75]. Notice and comment rulemaking provides legal drafters with an opportunity to identify potential conflicts. For example, the EHR Certification Rule requires that EHRs record patient smoking status using a set of predefined values (i.e., current smoker, former smoker, and never smoked). Commenters determined that the predefined smoking status values in the proposed EHR Certification Rule were inconsistent with the smoking status values used by the Centers for Disease Control (CDC). Thus, ONC adopts the same list of values used by the CDC in the final EHR Certification Rule.

Adaptability Rationale # 7: Resources Lacking to Implement

A regulated industry may lack the resources to implement requirements in a proposed rule. For example, the proposed EHR Certification Rule contains a requirement that certified EHRs record disclosures of PHI made for treatment, payment, or healthcare operations. However, commenters noted that the industry lacked the resources to implement the requirement in time for MU Stage 1. In response to these comments, ONC made the requirement optional for Stage 1.

Adaptability Rationale # 8: Concerns about Over-Regulation

Regulatory agencies may receive comments indicating that certain requirements in a proposed rule are unnecessary and over-regulate the domain. For example, the proposed HIPAA Security Rule would require that covered entities maintain procedures & policies to follow when an employee is terminated. However, commenters point out that this may be unnecessary in certain settings, such as a small rural provider whose only employee is their spouse.

Adaptability Rationale # 9: Concerns about Under-Regulation

Sometimes, commenters will point out that a regulation does not accomplish its stated goal, and needs to be strengthened. For example, in the proposed GLBA Financial Privacy Rule, financial institutions are required to provide consumers with reasonable opportunities to opt out of disclosures of their nonpublic personal information. The proposed rule also provides several examples of ways to provide this opt out. In the provided examples, however, financial institutions only had to provide opt out to consumers with whom they have a customer relationship—an ongoing relationship between the individual and the financial institution. Commenters point out that that this does not provide adequate protections for consumers that, for example, apply for a loan but are not approved. Because the financial institution never developed an ongoing relationship, they would be free to disclose the individual’s nonpublic information without pro-
providing reasonable opt out, under the proposed rule. In reaction to these public comments, the FTC strengthened the rule by expanding the opt out requirements to all consumers, not only those with which the financial institution has an ongoing relationship.

**Adaptability Rationale # 10: Change in Another Regulation**

Regulatory agencies update rules to keep up-to-date with changes in other laws. For instance, as we described in Section 1.4, the EHR Certification Rule places requirements on certified EHRs to allow providers to meet the MU clinical quality criteria. The clinical quality criteria are specified in a separate rule, the Medicare and Medicaid EHR Incentive Program Rule. Before the final EHR Certification Rule was issued, the EHR Incentive Program Rule was updated to include an additional clinical quality criteria: providers must provide patient-specific education resources to patients. This led ONC to add a new requirement in the final EHR Certification Rule that certified EHRs have the capability to provide patient-specific education.

**Adaptability Rationale # 11: Consistency Across Regulations**

Regulations are not developed in a vacuum. Rather, existing statutes, regulations, and case law create a context in which the new regulation will be applied. Sometimes, regulations create conflicting requirements (see Section 5.2); the public comment process helps to mitigate this risk. For example, in the final GLBA Financial Privacy Rule, the FTC adopts a definition of “collect” that is consistent with the Privacy Act of 1974.

**Adaptability Rationale # 12: Redundant**

A proposed rule sometimes contains redundant requirements. For example, the proposed HIPAA Security Rule contains the following two requirements: (1) covered entities must ensure that personnel have the proper authorization to view PHI, and (2) covered entities must maintain personnel clearance procedures. Commenters pointed out that these requirements seemed redundant, and they were combined in the final rule.

**Adaptability Rationale # 13: Unknown**

Some changes that are made to regulations go unexplained in the final rule commentary; the rationale for these changes is unknown. For example, the proposed EHR Certification Rule requires that certified EHRs have the ability to record, retrieve, and transmit immunization information to immunization registries. In the commentary that accompanies the final EHR


\[5\text{ U.S.C. §552a}\]
Certification Rule, HHS agrees with a commenter that recommended that “modify” be included in the list of operations the EHR can perform on immunization information, but neither the commenter nor HHS provide reasoning why this action should be included in the final rule. Thus, the rationale for this change is unknown.

3.1.2 Adaptability Taxonomy

The adaptability taxonomy classifications describe the changes that legal drafters make to proposed rules when they issue final rules. There are nine classifications in the adaptability taxonomy. In the remainder of this section, we describe each classification in detail.

**Taxonomy Classification # 1: Reorganization**

Regulatory texts may be reorganized when they are published as final rules; this reorganization often results in renumbered sections as well. For example, in the proposed HIPAA Security Rule, the security requirements are placed in a single section entitled “Security Standard”. In the final Security Rule, security requirements are reorganized among several sections based on theme, for example, “technical safeguards”, “physical safeguards”, and “administrative safeguards”. These changes do not impact the meaning of the requirements. As we discuss in Section 4.1.1, to maintain traceability, we map requirements in proposed rules to the matching requirements in final rules because they are sometimes reorganized and renumbered.

**Taxonomy Classification # 2: Elaboration**

Legal statements may be updated to provide greater clarity and reduce ambiguity. For example, in the proposed HIPAA Security Rule, covered entities are required to “assess potential risks and vulnerabilities to the individual health data in its possession and develop, implement, and maintain appropriate security measure” (§142.308). In the final HIPAA Security Rule, HHS describes several comments that they received asking to clarify the term “risk assessment”, along with several comments expressing confusion as to how the assessment should be performed. In response to these comments, HHS elaborated this requirement. The new requirement states at §164.308(a)(1) and §164.306(b) that the risk assessment must be used to ensure the confidentiality, integrity, and availability of electronic PHI (ePHI), and that security measures should protect against reasonably anticipated threats based, among other things, on the probability of the threat and the covered entity’s size, complexity, and technical infrastructure.
Taxonomy Classification # 3: Introduction of a Term

Sometimes, legal drafters introduce a term to provide a vocabulary for common concepts. For example, the term “covered entity” is not used in the proposed HIPAA Security Rule. Instead, each organization covered by the proposed Security Rule is defined in §142.302. Throughout the remainder of the proposed Rule, these organizations are referred to as an “entity designated in §142.302”. In the final HIPAA Security Rule, the term “covered entity” is introduced as ‘short hand’ for this language.

Taxonomy Classification # 4: Removal

Legal statements may be removed from a final rule for a variety of reasons. For example, the actions needed to comply with a particular compliance requirement may be too expensive, industry may lack the appropriate infrastructure to support the requirement, or a particular requirement may be out of scope for the domain in question. For example, as we discussed in Section 3.1.1, the proposed EHR Certification Rule contained the requirement that certified EHRs submit insurance claims. However, this functionality is typically performed by practice management systems, not EHRs. As such, ONC removed this requirement before publishing the final EHR Certification Rule.

Taxonomy Classification # 5: Broadening

A broadening occurs when a legal statement’s scope is expanded. For example, §142.308(a)(7)(i) of the proposed HIPAA Security Rule requires that covered entities oversee maintenance workers working in locations that house PHI. In the final HIPAA Security Rule at §164.308(a)(3)(ii)(A), this legal statement is broadened to require covered entities to supervise all members of their workforce that work in locations that house PHI.

Taxonomy Classification # 6: Addition

Additions are legal statements that are added to a final rule. For example, the final HIPAA Security Rule added §164.316(b)(1), which states “Standard: Documentation. (i) Maintain the policies and procedures implemented to comply with this subpart in written (which may be electronic) form; and (ii) If an action, activity or assessment is required by this subpart to be documented, maintain a written (which may be electronic) record of the action, activity, or assessment.” This requirement was not in the proposed HIPAA Security Rule.
Taxonomy Classification # 7: Requirement Made Optional

Regulatory agencies may make a requirement optional based upon feedback they receive from the public. For example, as discussed in Section 3.1.1, ONC made the requirement that certified EHRs record disclosures of PHI made for treatment, payment, or healthcare operations optional for MU Stage 1.

Taxonomy Classification # 8: Introduced Cross-Reference

Cross-references are citations from one legal text to another [75]. Cross-references may add constraints and exceptions to compliance requirements, may be outside of the scope of the software system being developed, or may even introduce conflicting requirements that must be addressed [75]. When issuing final rules, a regulatory agency may introduce a cross-reference that did not previously exist in the proposed rule. For example, the proposed EHR Certification Rule defines what actions are considered a disclosure of PHI. In the final EHR Certification Rule, this definition is replaced with a cross-reference to the definition of disclosure in HIPAA.

Taxonomy Classification # 9: Provided Examples

Some regulations will include examples to elucidate the regulatory text. For example, in the GLBA Financial Privacy Rule, the FTC provides examples of ways to comply with the rule. The rule states that compliance with an example is equivalence to compliance with the rule. In the final GLBA Financial Privacy Rule, several new examples are added that provide further clarity in response to commenters.

3.1.3 Adaptability Heuristics

The adaptability heuristics aid software engineers in identifying which areas of the law are likely to change. Legal drafters may use varying strategies when making changes to the law. For example, ambiguity in a proposed rule may lead drafters to elaborate the rule to resolve the ambiguity; alternatively, they may remove the ambiguous requirement altogether. Thus, a 1-to-1 ratio does not exist between the heuristics and the adaptability taxonomy classifications. The heuristics predict that a legal statement in a proposed rule may change, not how that statement will change.

Not all changes to a proposed rule can be predicted. For example, the adaptability rationale Resources Lacking to Implement is difficult to predict without deep knowledge of the domain, the resources, and associated infrastructure available to organizations. In addition, lobbyists may influence lawmakers to implement changes that are challenging to predict.

We now describe our adaptability heuristics.
**H1:** Ambiguous requirements suggest that the law may be disambiguated and therefore subject to change. We employ the Inquiry-Cycle Model to identify ambiguity [99]. Unanswered Inquiry-Cycle Model questions indicate an ambiguous compliance requirement that needs to be clarified in the final rule, because they represent compliance requirements that cannot be operationalized as software requirements without clarification. For example, consider the proposed EHR Certification Rule, which requires that certified EHRs use a hashing algorithm that is SHA-1 or higher for integrity protection (§170.210(c)). It is not clear who determines that a hashing algorithm is higher than SHA-1 or how they would make such a determination—unanswered who and how-to Inquiry-Cycle Model questions. In the final EHR Certification Rule, ONC revises this requirement to read “A hashing algorithm with a security strength equal to or greater than SHA-1 (Secure Hash Algorithm (SHA-1) as specified by the National Institute of Standards and Technology (NIST) in FIPS PUB 180-3 (October, 2008)) must be used to verify that electronic health information has not been altered” (§170.210(c)). By referring to the FIPS PUB 180-3 document published by NIST, the unanswered who and how-to questions are addressed.

**H2:** A repeated concept suggests that the concept may formally be defined in the final rule. When a concept is repeated in a proposed rule, it may be formally defined in the final rule. In the proposed HIPAA Security Rule, §142.302 describes the entities that must comply with the rule, including health plans, healthcare clearinghouses, and healthcare providers. §142.102 contains similar language. Throughout the remainder of the proposed rule, these entities are referred to as “entities designated in §142.302”. In the final rule, the term “covered entity” is introduced, and the definitions at §142.102 and §142.302 are combined into the definition of a covered entity.

**H3:** Duplicate concepts may be combined or disambiguated. When a proposed rule uses multiple terms for the same concept, they are likely to be combined into one concept or their differences defined. For example, the proposed HIPAA Security Rule uses the terms “health information pertaining to an individual”, “health information”, “data”, and “information” interchangeably. In the final HIPAA Security Rule, these terms are replaced with the term electronic protected health information (ePHI).

**H4:** Technology-specific requirements may be removed. Technology-specific requirements in a proposed rule may be removed in favor of requirements that fosters industry innovation and avoids implementation & design bias. For example the proposed HIPAA Security Rule requires that covered entities adopt an access control mechanism from a list defined in the regulation. However, as commenters point out, defining a list of acceptable access control techniques stifles innovation—if an improved access control technique is developed, it may not meet the requirements of the regulation until the regulation is updated. In the final HIPAA Security Rule, the requirement is restated to require policies and procedures that restrict access to PHI to individuals that have been assigned access rights—without requiring covered entities to adopt specific
access control technologies.

$H_5$: *Technology-specific requirements in examples may not be removed.* The HIPAA Privacy and Security Rules do not use illustrative examples in the text of the regulation. In contrast, the GLBA Financial Privacy Rule provides examples of ways that financial institutions may meet (or be out of compliance with) the regulation. As an exception to the previous heuristic, legal drafters may not remove references to specific technologies from examples. Because examples are only one means to comply with the law, including references to specific technologies does not stifle industry innovation. For example, in §313.4(d)(5)(ii)(B), the proposed GLBA Financial Privacy Rule provides an example of noncompliance, stating that financial institutions may not provide a notice of privacy practices to a consumer via e-mail if the consumer does not agree to receive the notice electronically. The regulation mentions e-mail, a specific technology, but does not mention other technologies that were developed after the rule was released, such as Short Message Service (SMS) text messages. However, because e-mail is referenced in an example, the reference is not removed from final GLBA Financial Privacy Rule.

$H_6$: *Specific requirements subsumed by a broader requirement may be removed.* When requirements are duplicated in a proposed regulation, detailed requirements may be removed in favor of the requirements that can be more broadly applied. In the proposed HIPAA Security Rule, covered entities are required to have: (a) visitor access control procedures, and (b) access control procedures. These requirements express duplication—access control procedures would necessarily have provisions around visitor access control. In the final HIPAA Security Rule, the visitor access control requirement is removed.

$H_7$: *Intentional ambiguity is unlikely to be removed.* As discussed by prior work, regulations contain intentional ambiguity—language included in regulations by law makers that allow for generalization and re-interpretation over time [16, 91]. Words such as “reasonable”, “timely”, and “appropriate” are often used to signify intentional ambiguity. For example, §164.306(a)(2) of the HIPAA Security Rule requires that covered entities must “protect against any reasonably anticipated threats or hazards to the security or integrity of [electronic PHI].” The threats that can be “reasonably anticipated” are subject to change over time and interpretation by regulatory agencies and courts. It is unlikely that such examples of intentional ambiguity will be removed during final rulemaking.

### 3.2 Cross-Reference Analysis

As engineers analyze relevant legal texts, they often encounter cross-references—a reference from a portion of a legal text to another. Engineers must ‘follow’ these cross-references to understand how the referenced text impacts their software. The cross-reference may add additional constraints on the software, introduce exceptions, or even be unrelated to the software being
built. If the referenced text changes through new regulatory or statutory activity, engineers may have to revisit the referenced text to ensure their software is still compliant. In this section, we present a legal cross-reference taxonomy that engineers can use to track cross-references across regulations.

Sometimes, a portion of a legal text will go unanalyzed for various reasons. For example, initially, engineers may have not identified the legal text as relevant to the software being developed. Thus, our selection criteria for examining a cross-reference in our study is:

*Does the cross-reference require engineers to analyze a portion of a legal text that would otherwise remain unanalyzed?*

Figure 3.3 displays the types of cross-references encountered in the our work. The white rectangles are legal texts—a named legal document. Shaded rounded rectangles are portions of legal texts—discrete legal citations—that are under analysis. Circles represent legal statements and arrows represent cross-references. Each cross-reference has been annotated with a number, e.g., (1), and each legal text or portion of a legal text has been annotated with an index, e.g., LT-IV. In Figure 3.3, the Pattern-A cross-reference and the cross-reference from (4) to (5) in Pattern-B are internal cross-references [65]. As discussed in Section 2.1, because prior work has examined internal cross-references [16, 69, 76], we do not examine Pattern-A or the cross-reference from (4) to (5) in Pattern-B in our study. The cross-reference from (3) to LT-VI in Pattern-B is an external cross-reference (specifically, a Pattern-C cross-reference) and is included in our study.

Herein, we examine the Pattern-C and Pattern-D cross-references (see Figure 3.3). Pattern-C represents an external cross-reference—a reference between portions of different legal texts—as classified by Massey et al. [65]. In Pattern-D, the cross-reference points to another legal text portion; in this case, requirements engineers have typically not analyzed the legal statement (see (9) in Figure 3.3). For example, the HIPAA General Administration Requirements contain requirements about the National Provider System and the National Provider Identifier (NPI). Typically, this portion of HIPAA would not be analyzed by a requirements engineering building an electronic health record system, similar to how the developer of a financial system will not typically review the portion of the Social Security Act that establishes Social Security Numbers. However, the HIPAA Privacy Rule cross-references a portion of the General Administration Requirements. The engineer must “follow” the cross-reference and analyze that portion of the General Administration Requirements.

Within the legal texts we studied, each Pattern-C or Pattern-D cross-reference either: (a) introduces a conflicting requirement or definition; (b) refines an existing requirement; or (c) falls outside the software system’s scope. Analyzing these cross-references facilitates refinement early in the software development process by enabling software engineers to address conflicting
requirements that may otherwise thwart legal compliance. This ensures that engineers do not overlook important compliance requirements. As a result of our study, we developed a legal cross-reference classification taxonomy (see Table 3.1). Engineers can use this taxonomy to classify the effect that a legal cross-reference has on existing compliance requirements. The taxonomy was developed in a descriptive fashion, and was proposed as a prescriptive taxonomy (for the legal texts studied) [73]. Our taxonomy complements previous requirements engineering research; before we begin our cross-reference analysis, we assume that compliance requirements have been specified from legal texts using one of the techniques described in Section 2.1 [16, 23, 35, 65, 72, 76, 111, 129].

The seven cross-reference types are: constraint, exception, definition, unrelated, incorrect, general, and prioritization. Constraint cross-references introduce additional constraints on existing compliance requirements. Exception cross-references introduce an exception condition to an existing compliance requirement. Definition cross-references introduce a definition or term. Unrelated cross-references are those in which the referencing or referenced legal texts do not yield requirements for software systems. Incorrect cross-references are references that cite an
Table 3.1: Legal Cross-Reference Taxonomy

<table>
<thead>
<tr>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint</td>
</tr>
<tr>
<td>Exception</td>
</tr>
<tr>
<td>Definition</td>
</tr>
<tr>
<td>Unrelated</td>
</tr>
<tr>
<td>Incorrect</td>
</tr>
<tr>
<td>General</td>
</tr>
<tr>
<td>Prioritization</td>
</tr>
</tbody>
</table>

incorrect portion of a legal text. General cross-references do not cite a specific legal text, rather, they are citations to “applicable law”. Prioritization cross-references are references that position a new legal text with respect to an existing legal text. In the remainder of this subsection, we describe each cross-reference type in detail.

3.2.1 Constraint Cross-References

Requirements are often refined by disambiguating them. In our study, cross-references refine existing requirements by introducing additional constraints. As advocated by Breaux and Antón for internal cross-references [17], we copy constraints from the referenced text into the compliance requirement. For example, §164.512(k)(3) of the HIPAA Privacy Rule states: “A covered entity may disclose PHI to authorized federal officials for the provision of protective services to the President or other persons authorized by 18 U.S.C. 3056.” This paragraph contains a cross-reference to 18 U.S.C. 3056, a law that describes the authority and duties of the U.S. Secret Service. Among other duties, the Secret Service is tasked with protecting individuals such as the President, the Vice-President, their immediate families, former Presidents and their families, visiting heads of state, and Presidential candidates. The cross-reference refines the compliance requirement expressed by §164.512(k)(3). To perform this refinement, we copy the list of people the Secret Service is charged to protect into the requirement expressed by §164.512(k)(3). After refinement, the compliance requirement reads “A covered entity may disclose PHI to authorized federal officials for the provision of protective services to the President, the Vice-President, their immediate families, former Presidents and their families, visiting heads of state, and Presidential candidates.” When refining requirements using constraint cross-references, engineers must be careful to note any definitional conflicts (see 3.2), i.e., a term defined in a legal text may have a different definition in the referenced legal text.

Section 6803(b)(4) of the GLB Act contains an example constraint from the GLB Act study: “The disclosure required by subsection (a) of this section shall include - the disclosures required, if any, under section 1681a(d)(2)(A)(iii) of this title.” This paragraph contains a cross-reference
to section 1681a(d)(2)(A)(iii) of this title, or 15 USC—the title wherein GLBA is codified.
Following the cross-reference to 15 USC 1681a(d)(2)(A)(iii) and copying the constraints, the
refined compliance requirement reads: “The disclosure required by subsection (a) of this section
shall include - communication of other information among persons related by common ownership
or affiliated by corporate control, if it is clearly and conspicuously disclosed to the consumer
that the information may be communicated among such persons and the consumer is given the
opportunity, before the time that the information is initially communicated, to direct that such
information not be communicated among such persons.”

3.2.2 Exception Cross-References

Some cross-references introduce exception conditions. For example, in §164.524(a)(1)(iii)(A) of
the HIPAA Privacy Rule, individuals are given the right to inspect and obtain a copy of their
PHI, except for health information that is covered by the Clinical Laboratory Improvement
Amendments (CLIA) of 1988. When exceptions are encountered, we create a requirement
expressing the exceptional case. In the given example, we create a requirement stating that
a covered entity may withhold information covered by CLIA from the individual.

GLBA Section 6802(e)(5) contains an example cross-reference. This section states: “Sub-
sections (a) and (b) of this section shall not prohibit the disclosure of nonpublic personal
information to the extent specifically permitted or required [. . . ] in accordance with the Right
to Financial Privacy Act of 1978 (12 U.S.C. 3401 et seq.)”. This example from the GLB Act
contains a cross-reference to the Right to Financial Privacy Act (RFPA). We, therefore, create
a compliance requirement stating that a financial institution may release information if it is
allowed to release the information under the RFPA.

3.2.3 Definition Cross-References

Legal texts use cross-references to cite definitions from other laws in much the same way as
a programmer imports object and function definitions from language libraries. For example,
HIPAA does not redefine the definition of “medical care”; instead it cites the medical care
definition used in the Public Health Services Act. Likewise, the GLB Act refers to 12 USC
1843(k) when defining “financial institutions.” When we encounter a definitional cross-reference,
we add the definition to the list of terms defined in the referencing legal text.

Terms spread across multiple legal texts can have differing and sometimes contradictory
definitions. For example, the HIPAA Privacy Rule cross-references the Privacy Act of 1974
at §164.524(a)(2)(iv). In HIPAA, an individual is defined as the “person who is the subject of

\textsuperscript{4}http://www.cms.gov/clia/
\textsuperscript{5}42 U.S.C. 300gg
PHI (§160.103), whereas in the Privacy Act of 1974, an individual is defined as a “citizen of the United States or an alien lawfully admitted for permanent residence” (§522a(a)(2)). These definitions differ; the HIPAA Privacy Rule protects the privacy of groups that the Privacy Act does not, for example, visitors to the U.S. Engineers must resolve these differing definitions or consult with legal domain experts to determine how to proceed.

### 3.2.4 Unrelated Cross-References

Cross-references can introduce referential ambiguity—portions of a cross-referenced text might not be applicable to software systems [16, 91]. In our study, we identify and set aside those cross-references that are unrelated to software systems. To determine which cross-references are unrelated, we ask the questions “Does the referencing legal text paragraphs introduce requirements for software systems?” and “Does the referenced legal text paragraphs introduce requirements for software systems?”

Some cross-references occur in portions of a referencing legal text that are outside the scope of a software system. For example, §164.512(i)(1)(i)(A) of the HIPAA Privacy Rule states:

> A covered entity may use or disclose PHI for research, regardless of the source of funding for that research, provided that the covered entity obtains documentation that an authorization or waiver, in whole or in part, of the individual authorization required by §164.508 for use or disclosure of PHI has been approved by either: an institutional review board (IRB), established in accordance with 7 CFR 1c.107, 10 CFR 745.107, 14 CFR 1230.107, 15 CFR 27.107, 16 CFR 1028.107, 21 CFR 56.107, 22 CFR 225.107, 24 CFR 60.107, 28 CFR 46.107, 32 CFR 219.107, 34 CFR 97.107, 38 CFR 16.107, 40 CFR 26.107, 45 CFR 46.107, 45 CFR 690.107, or 49 CFR 11.107

Although the documentation, authorization, or waiver for a covered entity to use or disclose PHI for research may be tracked by a software system, the subsequent list of 16 cross-references addresses how an institutional review board (IRB) is to be established. This establishment, as prescribed by these cross-references, is clearly outside the scope of software systems. Thus, we perform no further analysis on such cross-references.

In the case of a referenced text, if a legal statement cannot be operationalized as a software requirement, we set it aside and perform no further analysis on it. For example, the HIPAA Privacy Rule, at §164.512(b)(1)(v)(C), cross-references 29 CFR 1904 through 1928. This referenced text regulates safety and health in the workplace. These regulations specify many rules related to various industries, some of which are not related to software systems. For example, 29 CFR 1910.25 regulates the type of portable wooden ladders that can be used in the workplace, whereas 29 CFR 1912a establishes procedures for meetings of the National Advisory Committee
on Occupational Safety and Health. Both of these references are unrelated to software systems governed by HIPAA, thus, we do not analyze such cross-references.

### 3.2.5 Incorrect Cross-References

Cross-references in legal texts may be erroneous. For example, HIPAA §164.512(k)(3) states “A covered entity may disclose PHI to authorized federal officials for the provision of protective services to [...] foreign heads of state or other persons authorized by 22 U.S.C. 2709(a)(3).” This text contains a cross-reference to 22 U.S.C. 2709(a)(3), part of which states: “special agents of the Department of State and the Foreign Service may protect and perform protective functions directly related to maintaining the security and safety of foreign missions (as defined in section 4302(a)(4) of this title).” This paragraph contains another cross-reference to the definition of “foreign missions” at 22 U.S.C. 4302(a)(4). However, the definition at this citation is “real property,” not foreign missions. Foreign missions is defined in 22 U.S.C. 4302(a)(3). This is an obvious error. Although the error is documented via a footnote, and footnotes were considered authoritative in this study, in the U.S. Code at 22 U.S.C. 2709(a)(3), the footnote was not immediately obvious to us and policy makers have yet to correct the legal text.

Sometimes, cross-references are correct when the legal text is drafted, but subsequent amendments to law render cross-references to the amended law incorrect. For example, the GLBA Financial Privacy Rule contains cross-references to the Fair Credit Reporting Act (FCRA). But, section 604(b)(3)(A)(ii) in FCRA contains a cross-reference to 609(c)(3), which no longer exists as a result of an amendment made by Congress to FCRA in the Fair and Accurate Credit Transactions (FACT) Act. This amendment rendered the original cross-reference incorrect because it now points to a non-existent section of law. As in the HIPAA regulations, this is documented in a footnote, not the body of the legal text itself.

### 3.2.6 General Cross-References

May et al. [76] note that some cross-references do not mention a specific legal text by name. These cross-references are often stated as “other law”, “state law”, or “applicable law.” For example, §164.502(g)(2) in the HIPAA Privacy Rule allows covered entities to treat a parent as a representative of a minor “if, under applicable law, a parent [...] has authority to act on behalf of an individual who is an unemancipated minor.” No law is explicitly stated; instead, it is a general cross-reference to any applicable law. Requirements engineers are likely to require assistance from child law experts in resolving this or similar cross-references. Likewise, GLBA section 6802(e)(5), states that “Subsections (a) and (b) of this section shall not prohibit the disclosure of nonpublic personal information to comply with Federal, State, or local laws, rules,  

---

615 U.S.C. 1681 et seq.
and other applicable legal requirements.” To comply with this legal requirement, engineers must consult legal domain experts to determine which other Federal, State, and local laws are relevant.

### 3.2.7 Prioritization Cross-References

New law, such as the HIPAA or the GLBA Privacy Rules, comes into effect against the backdrop of existing statutes, regulations, judicial decisions, and other law. When legal texts have the same level of formality (e.g., both legal texts are statutes or both are regulations), prioritization cross-references help establish the priority relationship that new law has with respect to existing law. New laws can amend or repeal earlier law. In law, the default priority follows a “last in time” rule [125]—where there is ambiguity, the requirements promulgated later in time are considered authoritative. Legal texts, however, may depart from this default rule by explicitly stating that earlier law should be considered authoritative. Prioritization cross-references toggle priority by stating that an earlier, referenced text applies and therefore takes priority over the new law. Consider the following example of a prioritization cross-reference, from Section 6806 of the GLB Act:

> Except for the amendments made by subsections (a) and (b), nothing in this chapter shall be construed to modify, limit, or supersede the operation of the Fair Credit Reporting Act (15 U.S.C. 1681 et seq.), and no inference shall be drawn on the basis of the provisions of this chapter regarding whether information is transaction or experience information under section 603 of such Act (15 U.S.C. 1681a).

This section contains a cross-reference to the provisions in the FCRA. In Section 6806 of the GLB Act, subsections (a) and (b) amend the FCRA definitions for which federal agencies regulate which financial institutions, and those later-in-time subsections take priority over the earlier FCRA definition[^7]. In contrast, the remainder of FCRA remains unchanged by the GLB Act. By establishing this priority of the earlier law, the drafters of the GLB Act accomplish at least four goals. First, the remaining-in-effect portions of the earlier statute do not need to be explicitly placed into the later statute, e.g., in the above example, the remainder of FCRA does not have to be copied into the GLB Act. Second, and relatedly, having a single referenced text means that only that text needs to be changed if there are future amendments, rather than having to change the referencing law as well. Third, the explicit prioritization reduces legal uncertainty by preserving court decisions, agency guidance, and other legal materials that clarified the meaning of the earlier law. Fourth, private industry and others may have made investments or otherwise relied on interpretation of the earlier law. In the example, companies

[^7]: 12 U.S.C. §1681s

38
had created business models and information systems based on their understanding of what qualified as “transaction or experience information” under FCRA, which receives different legal treatment than other categories of information.

3.3 Production Rule Modeling

In this section, we provide an overview of our production rule framework. Software engineers can use this framework (see Figure 3.4) to analyze existing software requirements for compliance with regulations. The inputs to our framework are a legal text and a legal ontology. We then use our production rule modeling methodology to create a production rule model of the legal text [69, 71]. An expanded version of the modeling methodology is presented in Figure 3.5. Using this model, we develop a canonical legal requirements set that we use to analyze existing software requirements for compliance. In the remainder of this section, we describe the modeling and analysis phases of our framework, illustrating the framework using examples from the HIPAA Security and Privacy Rules.

3.3.1 Creating Production Rule Models

In this section, we provide an overview of the production rule modeling methodology. The methodology consists one preparatory step and two activities. Figure 3.5 displays an overview of the modeling approach. The inputs are a legal text and an ontology of the legal domain. We use the eight Hohfeldian legal concepts as our ontology (see Section 1.2).

In the preparatory step we create patterns to express each concept in the ontology. For example, if a covered entity is obligated by §164.512(f)(2)(ii) of the HIPAA Privacy Rule to not disclose PHI to a third party, we express this obligation as: must(CE, not(discloses(CE,ThirdParty,PHI)), ‘164.512(f)(2)(ii)’). To maintain traceability from the legal text to the production rules, we adopt Sherman’s solution [109]; we include an additional parameter, Source, with each pattern specifying the source of the rule. For instance, the obligation listed above is specified in §164.512(f)(2)(ii) of the HIPAA Privacy Rule. Table 3.2 lists the production rule patterns for each Hohfeldian concept.

Each of the Hohfeldian concepts has a correlative concept [49]—one concept implies another concept. For example, if an individual has a right to be notified of the uses and disclosures of his PHI by a covered entity, the covered entity is obligated to provide such a notice. Table 3.2 lists the concept implied by each concept. We discuss implied rules further in later in this section. Also, each concept has an opposite concept—an actor cannot hold both a concept and its opposite. For example, individuals cannot have both a right to amend PHI about them as well

---

as a no-right to amend PHI about them. Table 3.2 also lists each concept’s opposite concept. We use opposite concepts to express exception cases. To classify sections of the legal text, we use normative phrase analysis [18]; we list the normative phrases in Table 3.2. Immunities lack normative phrases because we did not encounter any immunities expressed in the HIPAA Privacy or Security Rule.
Figure 3.5: Production Rule Modeling Methodology Overview
<table>
<thead>
<tr>
<th>Concept</th>
<th>Production Rule Pattern</th>
<th>Implied Concept</th>
<th>Opposite Concept</th>
<th>Normative Phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>right</td>
<td>right (Actor,Counterparty,Right,Source)</td>
<td>obligation</td>
<td>no-right</td>
<td>has a/the right to retains the right to</td>
</tr>
<tr>
<td>obligation</td>
<td>obligation (Actor,Obligation,Source)</td>
<td>right</td>
<td>privilege</td>
<td>must is required to shall may not is prohibited is subject to</td>
</tr>
<tr>
<td>privilege</td>
<td>privilege (Actor,Privilege,Source)</td>
<td>no-right</td>
<td>obligation</td>
<td>may may elect not to is not required to requirement does not apply is permitted to at the election of is not subject to</td>
</tr>
<tr>
<td>no-right</td>
<td>noRight (Actor,NoRight,Source)</td>
<td>privilege</td>
<td>right</td>
<td>does not have a right to authorize termination of must obtain an authorization may revoke may terminate</td>
</tr>
<tr>
<td>power</td>
<td>power (Actor,LegalRelation,Power,Source)</td>
<td>liability</td>
<td>disability</td>
<td>provide that &lt;actor&gt;will/must obtain assurances that</td>
</tr>
<tr>
<td>liability</td>
<td>liability (Actor,Liability,Source)</td>
<td>power</td>
<td>immunity</td>
<td></td>
</tr>
<tr>
<td>immunity</td>
<td>immunity (Actor,Counterparty,Power,Source)</td>
<td>disability</td>
<td>liability</td>
<td>(none found)</td>
</tr>
<tr>
<td>disability</td>
<td>disability (Actor,Power,Source)</td>
<td>immunity</td>
<td>power</td>
<td>may not authorize</td>
</tr>
</tbody>
</table>
Throughout the remainder of this subsection, we will illustrate our approach using a concrete example from the HIPAA Privacy Rule, §164.510(a)(1)(i)(A-D), displayed in Figure 3.6. In our previous work, we describe the 17 heuristics for creating production rule models [71]. For brevity, below we only present the heuristics that are applicable to our concrete example.

As done in related work [16], we do not include document meta-information such as the table of contents, title pages, appendices, or paragraph and section headings in our analysis. In addition, a legal statement is often broken across a list such as in Figure 3.6. This is called a continuation and results in multiple legal rules [16]. The text in sections (1) and (i) of the legal text are prepended to the text in subsections (A), (B), (C), and (D), resulting in four legal statements.

(1) Except when an objection is expressed in accordance with paragraphs (a)(2) or (3) of this section, a covered health care provider may:

(i) Use the following protected health information to maintain a directory of individuals in its facility:

(A) The individual's name;
(B) The individual's location in the covered health care provider's facility;
(C) The individual's condition described in general terms that does not communicate specific medical information about the individual; and
(D) The individual’s religious affiliation

Figure 3.6: §164.510(a)(1)(i)(A-D) of the HIPAA Privacy Rule

Classify Rule Pattern

We use normative phrase analysis to classify rules according to the natural language phrases used in the legal text [18]. Table 3.2 displays the normative phrases we use to classify rule patterns. For example, we classify the legal statement in Figure 3.6 as a privilege because of the natural language phrase “may”. When multiple normative phrases are identified, we use the most inclusive normative phrase. For instance, §164.502(a) states that “a covered entity may not use or disclose PHI...” We classify this statement as an obligation (using the phrase “may not”) instead of a privilege (using the phrase “may” alone).

After the Classify Rule Pattern step, the production rule expressing section (A) in Figure 3.6 is privilege (_,_,_).
Identify Rule Parameters

In this step, we identify the parameters for the production rule pattern identified in the previous step. For example, in the actor subject to the privilege in section (1)(i)(A) in Figure 3.6 is a covered healthcare provider. Where possible, we identify the most specific actor. HIPAA defines both healthcare providers and covered healthcare providers; we identify the actor in section (1)(i)(A) as a covered healthcare provider because it is the more specific than healthcare provider. The rule action is the portion of the legal statement immediately following the normative phrase identified in the previous step. In section (A), the action the covered healthcare provider is allowed to perform is “use the following PHI to maintain a directory of individuals in its facility: the individual’s name.” The source of each rule is the full section reference. In the case of continuations, the source is the lowest subsection in the legal document hierarchy that is a part of the legal statement. For example, the source for section (A) is ‘164.510(a)(1)(i)(A)’.

After step 2, Identify Rule Parameters, the production rule expressing section (A) in Figure 3.6 is:

\[ \text{privilege}(\text{CHCP}, \, \text{for}(\text{uses}(\text{CHCP}, \text{name}(\text{Individual}))), \text{maintains}(\text{CHCP}, \text{directory})), \text{‘164.510(a)(1)(i)(A)’}). \]

Identify Rule Preconditions

This step entails identifying the legal preconditions that cause a rule to be applicable. Because Prolog is typeless, the first preconditions we add check the types of actors and objects expressed in the regulation. For example, for the privilege in section (A) in Figure 3.6, we express that an organization is a covered healthcare provider (CHCP) using the Prolog predicate isCHCP(CHCP). Next, as mentioned in Section 3.2, cross-references must be carefully followed to obtain additional preconditions. For example, in Figure 3.6, we add the precondition “the objection is in accordance with (a)(2) or (a)(3)” to the rules for (A). The analyst must follow this cross-reference and determine what preconditions are relevant. We next identify other preconditions that follow the phrases “if”, “when”, “whenever”, “that”, “who”, “whose”, “to the extent that”, and “provided that”, as well as temporal phrases such as “after”, “prior”, and “for as long as”. For example, after following the cross-reference, in section (a)(3) of §164.510, a covered healthcare provider may use an individual’s name for their directory, “if the opportunity to object to uses or disclosures to uses or disclosures required by paragraph (a)(2) of this section cannot practicably be provided because of the individual’s incapacity”. We add this precondition to the production rule that expresses (a)(3).

Legal texts express exceptions using the natural language phrases “except”, “is not effective
under”, “other than”, “does not apply to”, “notwithstanding”, and “unless”. For example, the privilege in section (A) of Figure 3.6 does not apply if the individual has expressed an objection according to (a)(2) or (a)(3). We perform two actions when we encounter an exception. First, we add a precondition that is a negation of the exception condition to the production rule. For the example text, the precondition “if the individual did not express an objection according to (a)(2) or (a)(3)” is added to the privilege. Second, we create a new rule expressing the exception (opposite) case. For the example text, a new production rule is created expressing that a covered healthcare provider may not (is obligated to not) use an individual’s name for their directory if the individual has expressed an objection. Each concept’s opposite is listed in Table 3.2.

After step 3, Identify Rule Preconditions Heuristic, the production rules expressing section (A) in Figure 3.6 are:

```prolog
privilege(CHCP, 
  for (uses(CHCP,name(Individual)), 
    maintains(CHCP,directory)),
  '164.510(a)(1)(i)(A)') :-
  isCHCP(CHCP),
  not(s164_510a1_exception(CHCP,Individual)).

obligation(CHCP, 
  not (for (uses(CHCP,name(Individual)), 
    maintains(CHCP,directory)),
  '164.510(a)(1)(i)(A)') :-
  isCHCP(CHCP),
  s164_510a1_exception.
```

Remove Disjunctions

This step entails identifying and removing disjunctions through case splitting. By convention, Prolog rules are expressed using only logical-and. We case split the natural language phrase “or” into multiple rules. For example, in section (A) of Figure 3.6, there two cases: (1) when an objection is expressed according to (a)(2), and (2) when an objection is expressed according to (a)(3). After step 4, Remove Disjunctions, the production rules expressing section (A) are:

```prolog
privilege(CHCP, 
  for (uses(CHCP,name(Individual)), 
    maintains(CHCP,directory)),
  '164.510(a)(1)(i)(A)') :-
  isCHCP(CHCP),
  not(s164_510a2_object). 
privilege(CHCP, 
  for (uses(CHCP,name(Individual)), 
    maintains(CHCP,directory)),
```
Identify Ontological Implications

Depending on the ontology upon which the production rule model is based, the software engineer may infer additional facts to add to the database. We used the Hohfeldian concepts as our ontology, where each concept implies its correlative concept (see Table 3.2). Implied rules are important to obtain, because codifying them increases requirements coverage and provides important traceability information to aid in establishing due diligence [18]. Breaux et al. introduced rights and obligations balancing, a technique to identify implied rights and obligations in HIPAA [18]. We expand rights and obligations balancing to balancing all implied rules and add them during this step. For example, the privilege in section (A) implies a no-right on the individual, whereas the obligation expressed in section (A) implies a right held by the individual.

The production rules after we complete the Specify Production Rules activity that express section §164.510(a)(1)(i)(A-D) are displayed in Appendix C.

3.3.2 Using Production Rule Models

In this subsection, we describe our approach for using production rule models to specify compliance requirements and check existing software requirements for legal compliance. First, software engineers query the model to obtain the rules relevant to the software to be developed. Second, engineers use templates to specify a canonical set of requirements. Finally, using the canonical requirements set, engineers check a set of existing software requirements for compliance. In the remainder of this section, we describe each of these steps in detail.
Query Model

Regulations can impact multiple domains. For example, the HIPAA Privacy Rule impacts domains such as healthcare, insurance, law enforcement, and medical research. Some of the legal rules may not apply to a given organization whose operations do not span all these domains. In this step, we query the model to determine which of the production rules apply to the domain and specific organization in which the software will be deployed. First, we add assert preconditions that model use cases for the software system being developed. For example, asserting the preconditions isHealthInsuranceIssuer(myHealthInsCo), covers(myHealthInsCo, individual), isPHI(phi), electronic(phi), about(phi, individual) is equivalent to the use case where a health insurance issuer possesses electronic PHI about a covered individual. We then query the model to obtain all relevant rules for the given use case. For example, executing the query obligation(Actor,Action, Source) is equivalent to asking the natural language question “What actions are required by the HIPAA Privacy and Security Rules?”

Specifying Compliance Requirements

We now describe the set of heuristics we developed for specifying compliance requirements from production rules, based on the Hohfeldian concept the rule expresses. Using these requirements, we create a canonical set of compliance requirements. A canonical requirements set is a set of requirements for which every organization in a particular domain must satisfy. Depending on the scope of the software system, some of these requirements may be implemented through means other than software, for example, through business processes. In HIPAA, we identify three kinds of canonical compliance requirements: software requirements, business requirements, and software development lifecycle (SDLC) requirements. Software requirements express functionality for a piece of software, or constrain software services [116]. Business requirements constrain procedures or policies of a business. SDLC requirements constrain the process by which the software is developed.

Specifying Requirements from Rights In the HIPAA Privacy and Security Rules, rights are held by individuals. Rights become scenarios that must be supported by other requirements; in particular, the scenarios are supported by the requirements that satisfy the implied obligation for this right. For example, §164.520 of the Privacy Rule states that an individual has the right to be notified of a covered entity’s privacy practices. This right implies that covered entities are obligated to notify the individual about their privacy practices. We create a scenario that the system must support; for example, the system could have a privacy policy that outlines the covered entity’s practices.
Specifying Requirements from the Remaining Hohfeldian Concepts  In the HIPAA Privacy and Security Rules, the other six Hohfeldian concepts—obligations, privileges, no rights, liabilities, powers, and disabilities—are placed upon or held by covered entities (the eighth concept, immunity, is not expressed in the Privacy or Security Rules). We use templates to specify requirements from rules that model these concepts—templates improve the understandability, consistency, and reusability of software requirements [119]. As mentioned above, some production rules express actions that are outside of the scope of a software system. For example, HIPAA provides several scenarios under which a covered entity may terminate a contract with a business associate. The termination of a contract is clearly beyond a software system’s scope.

We specify requirements using the following template:

The <actor> <modality> <action> if/when <preconditions> (<source>).

Below, we describe each component of the template.

Table 3.3: Requirement Template Modalities

<table>
<thead>
<tr>
<th>Concept</th>
<th>Modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>obligation</td>
<td>shall</td>
</tr>
<tr>
<td>privilege</td>
<td>should</td>
</tr>
<tr>
<td>no right</td>
<td>need not</td>
</tr>
<tr>
<td>liability</td>
<td>shall</td>
</tr>
<tr>
<td>power</td>
<td>shall allow &lt;actor&gt;to</td>
</tr>
<tr>
<td>disability</td>
<td>shall not allow &lt;actor&gt;to</td>
</tr>
</tbody>
</table>

• Actor: The actor of the requirement, either “system” for a software requirement, or the name of the business for a business requirement.

• Modality: The modality of the specified requirement depends on the concept the production rule expresses. For instance, obligated actions are required by law, but covered entities may choose to not perform a privileged action. Table 3.3 lists the requirement modality that corresponds with each concept. Note that we do not define a modality for immunities, because we did not encounter any in the Security or Privacy Rules.

• Action: The action is the action allowed, obligated, or forbidden by the regulatory text that is expressed in the production rule pattern.

• Preconditions: The preconditions that cause the production rule to be true.

• Source: The source is the citation to the portion of the legal text to which this requirement traces.
As an example, consider §164.312(e)(1) of the Security Rule: “A covered entity must, in accordance with §164.306, implement technical security measures to guard against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.”

The production rule that models this portion of the legal text is:

\[
\text{obligation (CE,}
\begin{align*}
\text{implements (CE,} & \text{to (technical (SecurityMeasures)}, \\
& \text{from (guards (CE, PHI),} \\
& \text{unauthorized (accesses (CE, PHI)))}, \\
& '164.312(e)(1)' \}) \\
& \text{isCoveredEntity (CE),} \\
& \text{isPHI (PHI),} \\
& \text{electronic (PHI),} \\
& \text{isSecurityMeasure (SecurityMeasures),} \\
& \text{over (transmits (CE, PHI),} \\
& \text{electronic (communicationsNetwork))}. 
\end{align*}
\]

Using our template, we specify the following software requirement (with indexing from our study retained): R27: The system shall safeguard electronic PHI from unauthorized access, when the PHI is being transmitted across an electronic communications network (§164.312(e)(1)).

**Checking Existing Requirements for Compliance**

We use the canonical requirements set developed during the previous step to check existing software requirements for compliance. This enables software engineers to perform a preliminary compliance analysis to identify areas of noncompliance in preparation for meetings with legal domain experts. We are not seeking to replace lawyers; rather, we seek to make meetings with legal domain experts more efficient and subsequently better support software engineers post-meeting compliance efforts, thus reducing the overall cost of legal compliance. To support our framework, we built the Compliance Requirements Analysis (CRA) Tool that aids software developers in using production rule models for compliance analysis.

For each requirement in the canonical requirements set, we scan the list of existing requirements to see if an existing requirement already operationalizes the canonical requirement. The canonical requirement may: (a) be operationalized in the existing set of requirements, (b) be partially operationalized in the existing requirements, requiring the existing requirements to be refined or augmented with the canonical requirement, (c) need to be added to the set of existing requirements if it is not be operationalized, or (d) conflict with existing requirements. In the case of (d), software engineers should discuss the conflicts with system stakeholders to modify the existing requirements so that legal compliance is possible.
When operationalizing canonical requirements, engineers should replace legal text definitions with the appropriate and equivalent definitions used in the existing requirements specification. For example, requirement R27 may be operationalized in the iTrust requirements specification as: R27: iTrust shall safeguard PHI from unauthorized access, when the PHI is being transmitted across an electronic communications network (§164.312(e)(1)). We describe iTrust further in Chapter 5.3.4. Once all requirements in the canonical requirements set have been operationalized in the existing requirements specification, developers can consult with legal domain experts who can provide further compliance validation.

3.4 Chapter Summary

As regulations change, engineers have to adapt their software to keep pace with the changing requirements. In this chapter, we presented our approach for handling regulatory change. This approach includes: a framework for tracking why and how regulations change, and predict which areas of a proposed regulation will change; a taxonomy for understanding how legal cross-references impact software; and method for modeling regulations and specifying legal requirements. In the next chapter, we describe the studies that support the development of our work and validate the approach.
RESEARCH DESIGN & VALIDATION

The most exciting phrase to hear in science, the one that heralds new discoveries, is not “Eureka!” but “That’s funny…”
—Isaac Asimov

In this section, we describe the research methodology we use for evaluating the impact that regulatory evolution has on software requirements. The research methodology consists of two studies. First, we use a multi-case study design, where a case study is defined as “an empirical inquiry that: investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” [127]. We employ this design because it is challenging to separate the phenomenon of legal text evolution from the context of the industry and regulatory regime. The multi-case study is a paradigm that has been previously used in software engineering research [16, 51, 83, 113]. Second, we performed a user study to evaluate software developers’ ability to classify cross-references. Our user study is a first step in evaluating how software engineers understand and validate requirements for legal compliance.

We now describe the design of our multi-case study and user study in detail.

4.1 Case Selection

Table 4.1 summarizes the three cases that comprise our multi-case study. We employ grounded theory analysis [38, 39] to develop the framework we presented in the previous chapter. In grounded theory analysis, theory is developed from the systematic study of a data set [38, 39]. The developed theory is “grounded” in the data, in that it is applicable only to the given data set [38, 39]. Future studies will allow us to make claims about the generalizability of our results. Grounded theory contrasts with the traditional scientific method, where hypotheses
are formulated then tested through experiments. Researchers have previously used grounded theory analysis for requirements engineering research [26, 55] and when analyzing legal and policy requirements [2, 17, 18, 130].

<table>
<thead>
<tr>
<th>Case Name</th>
<th>Research Question(s) Addressed</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability Study</td>
<td>RQ1</td>
<td>Examined legal evolution in healthcare and financial regulations</td>
</tr>
<tr>
<td>Cross-References Study</td>
<td>RQ2</td>
<td>Examined cross-references in healthcare and financial regulations</td>
</tr>
<tr>
<td>Modeling Case Study</td>
<td>RQ3</td>
<td>Modeled the HIPAA Privacy and Security Rules, and verified compliance of iTrust system against HIPAA</td>
</tr>
<tr>
<td>User Study</td>
<td>RQ3</td>
<td>Performed a study measuring the ability of software practitioners to classify the impact of cross-references on software</td>
</tr>
</tbody>
</table>

Recall in Section 1.4, we described the legal texts we analyze in our multi-case study. Table 4.2 displays the texts we analyze, their lengths in number of words, and the regulatory agency responsible for promulgating the rules.

We describe each case in our multi-case study in detail below.

<table>
<thead>
<tr>
<th>Legal Text</th>
<th>Length (in # of words)</th>
<th>Regulatory Agency</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIPAA Privacy Rule</td>
<td>29,659</td>
<td>HHS</td>
</tr>
<tr>
<td>HIPAA Security Rule</td>
<td>5,563</td>
<td>HHS</td>
</tr>
<tr>
<td>EHR Certification Rule</td>
<td>4,736</td>
<td>ONC (within HHS)</td>
</tr>
<tr>
<td>GLB Act</td>
<td>5,037</td>
<td>statutory</td>
</tr>
<tr>
<td>GLBA Financial Privacy Rule</td>
<td>12,716</td>
<td>FTC</td>
</tr>
</tbody>
</table>

4.1.1 Case #1: Adaptability Study

In this subsection, we describe the design of our Adaptability Study. We examined three texts in this case study: the HIPAA Security Rule, the EHR Certification Rule, and the GLBA Financial
While examining the HIPAA Security Rule, we developed the Adaptability Framework (see Section 3.1) with grounded theory analysis. Due to renumbering and reorganization changes, the topic of a particular section in a proposed rule may differ from the topic in the matching section number of a final rule. Thus, we first perform a topical mapping from the proposed HIPAA Security Rule to the final HIPAA Security Rule. For example, §142.308(a)(3) in the proposed HIPAA Security Rule describes business continuity and disaster recovery plans that a covered entity must maintain. However, in the final HIPAA Security Rule, these plans are discussed in §164.308(a)(7)(i). Where a topic that appeared in one section in the proposed rule is discussed in multiple sections in the final rule, we add a mapping for each section, and visa versa. As we map each section, we examine the text of the proposed and final Security Rule, documenting changes to the regulation using a spreadsheet. Our spreadsheet contained columns to document the original text, the modified text, the classification, and the rationale from the final rule preamble. For each identified change, we examine HHS’s commentary that accompanies the final Security Rule to reason about the stated rationale for why a change was made. Upon completing our analysis of the Rules, we formulated the Adaptability Rationale and Adaptability Taxonomy. We then developed heuristics based on the rationale and taxonomy. The heuristics help requirements engineers predict that changes will occur in a proposed regulation as it is updated during final rulemaking.

For the other two texts we examined in this case study, the EHR Certification Rule and the GLBA Financial Privacy Rule, we used the same protocol. First, we used the Adaptability Heuristics (see Section 3.1.3) to predict the areas of the proposed EHR Certification Rule that were likely to be changed during final rulemaking. Second, to validate our predictions, we analyzed the rules using the same grounded theory analysis on the EHR Certification Rule that we performed when examining the HIPAA Security Rule.

Prior to a proposed regulation being issued, pundits, legal domain experts and industry experts may identify the need for regulatory action and speculate on what form future rules may take. Examining proposed regulations is a special case of this general case where the proposed text is a speculative, albeit more concrete, commentary of future regulatory state. Additional predictive heuristics could be developed when examining public discourse.

4.1.2 Case #2: Cross-References Study

In this subsection, we describe the design of our Cross-References Study. Initially, we planned to examine three texts in this case study: the HIPAA Security Rule, the HIPAA Privacy Rule, and the GLBA Financial Privacy Rule. However, we later discovered that we had analyzed the GLB Act instead of the GLBA Financial Privacy Rule. This anecdotally affirms the challenges
for engineers in writing software for regulatory compliance—Otto and Antón have previously pointed out that identifying relevant legal texts is a challenge for regulatory compliance in software systems [91]. Despite this mistake, analyzing the GLB Act has merit—Congress, in the GLB Act, provided considerable detail in the statute itself about cross-references. The GLBA Financial Privacy Rule went into additional detail, but much of that detail already existed in the statute. In contrast, the HIPAA statute only gave broad guidance to the Department of Health and Human Services (HHS) to write a regulation if Congress did not enact new legislation by 1999. When Congress did not act, HHS was required to create most of the details of the HIPAA Privacy Rule from scratch, including the cross-references. Thus, we did not review the HIPAA statute as part of this case study.

Units of Analysis in the Cross-Reference Study

We use cross-references as the unit of analysis in our multi case study. When specifying compliance requirements for software systems, engineers must begin with relevant legal texts [91]. For example, HIPAA governs the use and dissemination of healthcare information by covered entities whereas the GLBA Financial Privacy Rule governs the collection and disclosure of customers’ personal financial information by financial institutions.

Recall that we defined the different cross-reference patterns in Section 3.2. In our study, we include Pattern-C and Pattern-D cross-references (see Figure 3.3). Table 4.3 displays the number of cross-references we identified in each of the legal texts we studied. For each legal text that we examined, we first scanned the legal texts in their entirety to identify the Pattern-C and Pattern-D cross-references within them—108 in the HIPAA Privacy Rule, 0 in the HIPAA Security Rule, 32 in the GLB Act, and 49 in the GLBA Financial Privacy Rule. For each identified cross-reference, we analyzed the text indicated by the reference to identify new cross-references within those additionally referenced texts. Due to time constraints, we limited our analysis to those cross-references that represent a “distance” of no more than two cross-references away from each of the original source texts. Within these referenced texts, we identified an additional 69 Pattern-C and Pattern-D cross-references in the texts referenced by HIPAA, 328 in the texts referenced by the GLB Act, and 318 in the texts referenced by the GLBA Financial Privacy Rule. This resulted in 177 total examined cross-references from HIPAA, 360 total cross-references examined from the GLB Act, and 367 total cross-references examined from the GLBA Financial Privacy Rule. Table 4.3 summarizes the number of cross-references we identified in each legal text.
Table 4.3: Number of Cross-References Identified

<table>
<thead>
<tr>
<th>Legal Text</th>
<th># of Cross-Refs Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIPAA Privacy Rule</td>
<td>108</td>
</tr>
<tr>
<td>In Texts Referenced in HIPAA Privacy Rule</td>
<td>69</td>
</tr>
<tr>
<td>HIPAA Security Rule</td>
<td>0</td>
</tr>
<tr>
<td>GLB Act</td>
<td>32</td>
</tr>
<tr>
<td>In Texts Referenced in GLB Act</td>
<td>328</td>
</tr>
<tr>
<td>GLBA Fin. Pri. Rule</td>
<td>49</td>
</tr>
<tr>
<td>In Texts Referenced in the GLBA Fin. Pri. Rule</td>
<td>318</td>
</tr>
</tbody>
</table>

**Cross-Reference Study Research Protocol**

We performed two passes through each original source text. In the first pass, we identified the Pattern-C and Pattern-D cross-references. During this pass, we downloaded PDF versions of each regulation, and annotated the document by highlighting each cross-reference. In the second pass, we used open coding—tagging each unit of analysis with a descriptive categorization—to classify each cross-reference’s effect on compliance requirements. We documented each cross-reference in a spreadsheet with columns for the location of the referencing text, the location of the referenced text, the paragraph that contains the cross-reference, and the descriptive categorization. After performing these two passes, we then followed each cross-reference and performed the same procedure on the referenced legal text. Upon classifying each cross-reference, we compiled the classifications into broader categories to form the taxonomy presented in Figure 3.1.

**4.1.3 Case #3: Modeling Study**

In our Modeling Study, we modeled the HIPAA Privacy and Security Rules using production rules over three phases. First, we developed our production rule modeling methodology through a small pilot study on four sections of the HIPAA Privacy Rule—§164.520-164.526—using three Hohfeldian concepts [69] (see Section 3.3 for a description of the methodology). After this, we examined the entire HIPAA Privacy and Security Rules using all eight of the Hohfeldian concepts [71, 72]. In our third and final phase, we used the Security Rule model to check the requirements of iTrust for compliance [72]. iTrust, our test bed, is an open-source EHR used as an instructional tool at North Carolina State University, developed as part of undergraduate and graduate courses\(^1\). For our case study, we use the 73 requirements developed by Massey et al. [66]. Of these requirements, 63 of them are functional requirements, and 10 of them are non-functional. In addition, we analyze the 12 requirements we developed during our previous iTrust

\(^1\)http://agile.csc.ncsu.edu/iTrust/wiki/doku.php
case study [70], providing us with 85 total requirements to validate for regulatory compliance.

4.2 Measuring the Cost of Compliance and Noncompliance

Ideally, the cost of noncompliance should outweigh the cost of compliance to encourage companies to comply. Noncompliance carries various costs, including regulatory fines, remediation efforts, legal fees to defend against lawsuits, erosion of consumer confidence, and brand damage [92]. Certain costs such as brand damage and lost consumer confidence may be challenging to quantify, and will vary by industry and circumstance. Regulatory fines, however, are publicly known and represent a minimum cost for noncompliance. HIPAA establishes a maximum penalty of $1.5 million “for all violations of an identical provision.”\(^2\) We use this amount as our threshold for determining what is the cost of noncompliance for research question RQ\(_4\). Compliance cost models must also take into account the chance that noncompliance will be detected [97], as well as organizational appetite for risk [108]. In our work, we conservatively adopt a risk-averse approach; we assume that all instances of noncompliance will be discovered and given the maximum fine.

We track the person hours we invest in developing and using our approach in the multi-case study (see Section 5.4). Using the median salary of $92,648 for software engineers in the U.S.\(^3\), we calculate a cost of $46.32 per person hour\(^4\). This allows us to determine the cost of our analysis activities based on the person hours we spend. However, simply analyzing compliance requirements does not produce compliant software; designs have to be developed, code written, and software tested. Assuming that six percent of project resources should be spent on requirements activities\(^5\), we conclude that our approach does not add cost to software development that is more than the cost of noncompliance if it is less than $90,000—or six percent of the maximum fine allowed under HIPAA.

4.3 User Study Design

In this subsection, we describe the design of our user study. Our user study seeks to test the ability of software engineers, legal domain experts, and healthcare professionals to correctly classify cross-references using the cross-reference taxonomy we developed in our multi-case study (see Section 3.2). This study lays the groundwork for future studies further examining how well healthcare IT professionals understand regulatory requirements. We compare participant

\(^2\)http://www.hhs.gov/ocr/privacy/hipaa/administrative/enforcementrule/hitechenforcementifr.html


\(^4\)Assuming a 40 hour work week and 52 paid weeks a year.

\(^5\)Based on a COCOMO model estimate of a system with 1000 function points written in Java. http://csse.usc.edu/tools/COCOMOII.php
responses to classifications made by a group of experts including the author, a law school professor, and a software privacy professor. We measure precision to test participants’ ability to make classifications. Precision measures consistency, or whether the classifications made by participants are the same classifications made by the experts. Recall is often measured with precision (e.g., [16]). Recall measures completeness, or whether participants identify all cross-references in the healthcare regulation. We do not measure recall in our user study. To measure recall, we could have provided participants with large paragraphs of legal text and asked them to identify cross-references in the text (to measure recall) then asked them to classify each (to measure precision). However, we determined that this exercise would have taken participants too much time and would have likely led to reduced participation by our target population (software and healthcare practitioners). Thus, we decided to ask participants to just do classification tasks and left identification tasks to future studies. To test precision, we use a set of cross-reference classifications that experts have identified and classified in healthcare and financial regulations.

Our study has the following null hypothesis:

\[ H_1: \text{Individuals from the participant group have equivalent or greater precision than the expert classifications when classifying cross-references using the taxonomy.} \]

4.3.1 Materials

The study materials consisted of an informed consent form, a demographics survey, a tutorial, and a 10 question survey. The demographics survey captured participant work experience, education, and comfort level with making security, privacy, and legal decisions. The tutorial explained our cross-reference taxonomy with examples of each classification. The survey consisted of 10 legal statements drawn from the healthcare and financial regulations we studied (covered in Section 1.4). We chose to use only 10 legal statements to ensure that our study was short; a longer study may have discouraged some participants from taking the survey. When selecting the 10 legal statements to use for our study, we preferred legal statements that were shorter in length and ensured that each classification was exhibited by at least one cross-reference. The only classification that does not appear in our study is the incorrect classification. We did not include an incorrect cross-reference because they require in-depth analysis to identify; we identify incorrect cross-references by observing footnotes in the legal text or analyzing the referenced legal text—neither of which we provided in our study.

We employed the Qualtrics survey software to deliver our survey. We reproduce our study materials in Appendix D.

\[\text{https://www.qualtrics.com/}\]
4.3.2 Participants

The participants in our study are software engineers, healthcare professionals, legal domain experts, requirements engineers, software architects, development managers, individuals working in support and professional services roles, and other individuals that work in the field of healthcare IT. We targeted two organizations to recruit participants. First, we recruited participants from a trade association of 41 EHR vendors, including 9 of the top 10 EHR vendors based on number of providers attesting for MU Stage 1 incentives\(^7\). Second, we targeted a nonprofit consortium of 220 organizations that includes healthcare systems, healthcare IT vendors, academic medical centers, government agencies, and other technology vendors. Participants were recruited through email to various work groups and primary contacts at member organizations, and offered the chance to enter a random drawing to win a small gift card to a popular online book seller.

The expert group consisted of the author, a requirements engineering professor who is an expert in compliance requirements, a law professor who was a senior manager in the drafting of the HIPAA Privacy and Security Rules and the GLBA Financial Privacy Rule, and two senior PhD students who are familiar with compliance requirements research.

4.3.3 Research Methodology

We performed a pilot before running our full study. Current and former members of the Realsearch\(^8\) and ThePrivacyPlace\(^9\) research groups were recruited to participate in the study. The pilot ran for one week; after the pilot closed, we performed initial analysis on the results and collected verbal feedback from several pilot participants. Based on this analysis, we made no substantive changes to the survey instrument before running the full study.

We ran the full study by targeting the participant groups discussed in Section 4.3.2. The survey was open for 30 days; after the survey completed, we gathered the results from the Qualtrics software for comparison to the classification performed by the experts (mentioned in Section 4.3.2). Consensus was achieved on the expert results through discussion during a reading group meeting lasting two hours, involving the author, a requirements engineering professor, and two PhD students. When the expert group could not reach consensus, we set the cross-reference aside and consulted the law professor expert during a later call. The final expert consensus serves as the oracle against which we compare participant responses.

\(^7\)https://www.thehitcommunity.org/2012/05/hitc-data-watch-top-10-ehr-vendors-in-the-cms-ehr-incentive-program/
\(^8\)http://www.realsearchgroup.org/realsearch/
\(^9\)http://theprivacyplace.org/
4.4 Chapter Summary

In this chapter, we discussed the research design of our multi-case study and our user study. The multi-case study comprises three case studies on legal text evolution, cross-references, and production rule modeling. We produced the framework we presented in Chapter 3 from the grounded theory analysis we performed in our multi-case study. We also discussed the design for our user study. In the next chapter, we describe the results of these studies.
CHAPTER
FIVE

FINDINGS

Any change, even a change for the better, is always accompanied by drawbacks and discomforts.
—Arnold Bennett

We developed our approach to addressing regulatory evolution through a multi-case study and ran an empirical study to measure the ability of software engineers to evaluate software requirements for regulatory compliance. Table 5.1 displays the artifacts we developed from each case in our multiple case study. In addition to our approach discussed in Chapter 3, we surfaced several results and findings not discussed by prior researchers. We discuss these results in the remainder of this chapter. First, in Sections 5.1 and 5.2, we discuss how legal evolution and cross-references impact software requirements, respectively. In Section 5.3, we present the results of our production rule modeling. Section 5.4 outlines the cost of compliance. Finally, we present the findings of our user study in Section 5.5.

5.1 The Effect of Legal Evolution

Our study reveals that legal evolution can have significant impact on software requirements. Requirements can be added, elaborated, or removed due to legal evolution. Table 5.2 and Table 5.3 display the adaptability rationale and taxonomy classifications, respectively, that we identified in our study.

As discussed in Section 4.1.1, we used our adaptability heuristics to predict changes in the EHR Certification Rule and the GLBA Financial Privacy Rule. We predicted 14 areas in the proposed EHR Certification Rule that would change, and 7 areas in the GLBA Financial Privacy Rule that would change. Appendix B lists these predictions. We then analyzed the final EHR Certification Rule and GLBA Financial Privacy Rule to validate our predictions and
Table 5.1: Cases and Artifacts Discovered in Each Case

<table>
<thead>
<tr>
<th>Case Name</th>
<th>Artifacts Discovered</th>
<th>Supporting Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability Study</td>
<td>Developed the Adaptability Framework</td>
<td>[74]</td>
</tr>
<tr>
<td>Cross-References Study</td>
<td>Developed the cross-reference taxonomy and identified 5 conflicts that could lead to noncompliance</td>
<td>[73, 75]</td>
</tr>
<tr>
<td>Modeling Study</td>
<td>Developed the Production Rule Framework</td>
<td>[69, 70, 71, 72]</td>
</tr>
<tr>
<td>User Study</td>
<td>Determined that software practitioners are not well equipped to understand the impact of cross-references on software requirements</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Adaptability Rationale in the Multi Case Study

<table>
<thead>
<tr>
<th>Rationale</th>
<th>HIPAA Sec. Rule</th>
<th>EHR Cert. Rule</th>
<th>GLBA Fin. Priv.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguity</td>
<td>14</td>
<td>14</td>
<td>23</td>
<td>51</td>
</tr>
<tr>
<td>Format and Organization</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>Technology Specific Elements</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Responses to Questions Posed</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Inappropriate for Domain</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>Potential Conflict</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Resources Lacking to Implement</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Concerns about Over-Regulation</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>Concerns about Under-Regulation</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Change in Another Regulation</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Consistency Across Regulations</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Redundant</td>
<td>11</td>
<td>0</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Unknown</td>
<td>17</td>
<td>5</td>
<td>24</td>
<td>46</td>
</tr>
</tbody>
</table>
Table 5.3: Adaptability Taxonomy Classifications in the Multi Case Study

<table>
<thead>
<tr>
<th>Taxonomy Classification</th>
<th>HIPAA Sec. Rule</th>
<th>EHR Cert. Rule</th>
<th>GLBA Fin. Priv.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorganization</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Elaboration</td>
<td>11</td>
<td>5</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>Introduction of a Term</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Removal</td>
<td>21</td>
<td>21</td>
<td>11</td>
<td>53</td>
</tr>
<tr>
<td>Broadening</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Addition</td>
<td>12</td>
<td>8</td>
<td>25</td>
<td>45</td>
</tr>
<tr>
<td>Requirement Made Optional</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Introduced Cross-Reference</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Provided Examples</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

identify unpredicted changes. Table 5.4 displays the results of this analysis in terms of false and true positives and negatives, where:

- A true positive is a change that we accurately predicted ($t_p$),
- A false positive is a change we predicted that was not accurate ($f_p$),
- A true negative is a legal statement for which we predicted no change and for which no change occurred ($t_n$), and
- A false negative is a legal statement for which we predicted no change and which changed in the final rule ($f_n$).

Table 5.4: Actual Legal Text Changes Identified

<table>
<thead>
<tr>
<th>EHR Cert. Rule</th>
<th>Accurate</th>
<th>Inaccurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>11 (true positives)</td>
<td>5 (false positives)</td>
</tr>
<tr>
<td>Not Predicted</td>
<td>104 (true negatives)</td>
<td>33 (false negatives)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GLBA Fin. Priv. Rule</th>
<th>Accurate</th>
<th>Inaccurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>2 (true positives)</td>
<td>5 (false positives)</td>
</tr>
<tr>
<td>Not Predicted</td>
<td>177 (true negatives)</td>
<td>71 (false negatives)</td>
</tr>
</tbody>
</table>

Table 5.5 displays the accuracy, specificity, precision, recall of our study. Accuracy is the ratio of predictions that were correct and is defined as $\frac{t_p + t_n}{t_p + t_n + f_p + f_n}$. Specificity is the ratio of legal statements that we correctly predicted would remain unchanged and is defined as $\frac{t_n}{t_n + f_p}$. Precision is the ratio of predictions that were accurate and is defined as $\frac{t_p}{t_p + f_p}$. Recall is the ratio of the regulatory changes we identified and is defined as $\frac{t_p}{t_p + f_n}$. Because we developed
Table 5.5: Accuracy, Specificity, Precision, and Recall in the Adaptability Study

<table>
<thead>
<tr>
<th>Regulatory Text</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHR Certification Rule</td>
<td>0.75</td>
<td>0.95</td>
<td>0.64</td>
<td>0.21</td>
</tr>
<tr>
<td>GLBA Financial Privacy Rule</td>
<td>0.70</td>
<td>0.97</td>
<td>0.29</td>
<td>0.03</td>
</tr>
</tbody>
</table>

rather than applied the heuristics when examining the HIPAA Security Rule, we do not include the accuracy, specificity, precision, and recall for the HIPAA Security Rule in Table 5.5.

The impact of an error, either a false positive or a false negative, depends on the specific compliance requirement that was missed, the change that occurred, and the particular system being developed. For example, consider false negative errors—we predict that no change will occur for a legal text statement, but it changes in final rulemaking. If the legal statement expressed a constraint that was removed in final rulemaking, then the software may be overly constrained but still in compliance. Alternatively, if the legal statement expressed an exception that was removed in final rulemaking, then the software may be out of compliance because the exception is no longer allowed. Thus, we cannot calculate the impact of errors a priori.

We sought to address research question **RQ** in our Adaptability Study: can engineers predict which areas of a legal text are likely to evolve? As previously discussed, not all changes that are made to a proposed rule can be predicted. For example, our results indicated that it is particularly challenging to predict how legal drafters may reorganize and renumber legal statements within a rule. Even with the understanding that not all changes can be predicted, we identified only 21% and 3% of the actual regulatory changes in the EHR Certification Rule and the GLBA Financial Privacy Rule, respectively. Our study’s low recall highlights the challenges that face requirements engineers developing software for regulated domains. Thus, the current set of adaptability heuristics do not fully address research question **RQ**. We position our adaptability framework as a preliminary effort to address the challenges introduced by evolving compliance requirements. To date, the literature has lacked tools for software engineers to identify and prepare for changes in legal rules. Our work is the first to attempt to predict what areas of a proposed legal rule will change in the final rule. Progress in this task will assist in legal compliance, reduce the costs of adapting legacy systems to changes in legal requirements, and assist a company to be first to market with compliant software.

We only predicted changes for three adaptability rationales: ambiguity, redundant, and technology-specific requirements. We did not predict any changes for other rationale such as Inappropriate for Domain or Resources Lacking to Implement. To identify these changes likely requires deep domain knowledge. To improve the heuristic performance, knowledge external to the regulation must be modeled. In our future studies, we plan expand the heuristics by analyzing public comments that are submitted for a proposed rule. Comments are publicly
available online\(^1\). By analyzing these comments we seek to determine if they are more efficient than the adaptability heuristics to identify potential changes that will be made to proposed rules before they are finalized.

## 5.2 The Effect of Cross-References

Regulatory texts contain a high number of cross-references; as analysts examine each additionally cited text, complexity and effort exponentially increase. Figure 5.1 displays a graphical representation of the cross-references we identified in the HIPAA Privacy Rule. Rounded boxes represent legal texts; the directional arrows reflect that at least one cross-reference exists from the referencing legal text to the referenced legal text. We have not included a similar graphical representation of the cross-references in the GLB Act or the GLBA Financial Privacy Rule, because there are too many cross-references in those texts to produce a meaningful graph that will fit on a single sheet of paper.

Table 5.6 displays the number of Pattern-C and Pattern-D cross-references by type that we identified in our Cross-References Study. The table includes both the cross-references from the source text, as well as the cross-references found in those referenced texts (see Section 4.1.2). Because we did not identify any Pattern-C or Pattern-D cross-references in the HIPAA Security Rule, we do not include it in Table 5.6. When examining HIPAA, we identified six types of cross-references. When examining the GLB Act and the GLBA Financial Privacy Rule, we identified these same six types of cross-references, in addition to a seventh type of cross-reference that we did not identify when examining HIPAA.

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>HIPAA Priv. Rule</th>
<th>GLBA</th>
<th>Fin. Priv. Rule</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraint</td>
<td>51</td>
<td>21</td>
<td>40</td>
<td>112</td>
</tr>
<tr>
<td>Exception</td>
<td>18</td>
<td>40</td>
<td>31</td>
<td>89</td>
</tr>
<tr>
<td>Definition</td>
<td>30</td>
<td>59</td>
<td>56</td>
<td>145</td>
</tr>
<tr>
<td>Unrelated</td>
<td>58</td>
<td>175</td>
<td>154</td>
<td>387</td>
</tr>
<tr>
<td>Incorrect</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>General</td>
<td>18</td>
<td>18</td>
<td>2</td>
<td>38</td>
</tr>
<tr>
<td>Prioritization</td>
<td>0</td>
<td>36</td>
<td>41</td>
<td>77</td>
</tr>
</tbody>
</table>

We did not identify any prioritization cross-references in HIPAA or the texts referenced by HIPAA [73]. However, we identified 36 prioritization cross-references in the GLB Act and 41

\(^1\)http://www.regulations.gov
Figure 5.1: Unique Pattern-C and Pattern-D Cross-References in the HIPAA Privacy Rule
prioritization cross-references in the GLBA Financial Privacy Rule [75]. Privacy law in the U.S. is sectoral, meaning a regulation in one industry sector (e.g., finance) does not apply to another industry sector (e.g., healthcare). Because HIPAA led to the first national privacy and security regulations in the healthcare sector, there was no existing body of law against which to establish priorities. In contrast, GLBA is a financial law and there were many existing national laws in the financial sector at the time GLBA was enacted in 1999. Our identification of the prioritization cross-references reflects the fact that the drafters of GLBA had to establish priority among existing laws.

In Figure 5.1, we see from the cross-reference graph that the HIPAA Privacy Rule appears to be more connected than other legal texts. We believe this is due to the fact that medical privacy cuts across many domains, such as records maintained by a healthcare provider, veterans' records, employee records, etc. This large scope of medical privacy means that HIPAA has many inter-connections among various legal texts. In contrast, other legal texts quite possibly do not have the same degree of interaction with multiple other domains.

5.2.1 Identifying Conflicting Requirements

As previously discussed, cross-references introduce challenges to regulatory compliance [8, 16, 17, 18, 43, 76, 91], but researchers have yet to examine cross-references that introduce conflicting compliance requirements. Our HIPAA analysis surfaced five critical conflicts that if not resolved would lead to non-compliance. For example in the HIPAA Privacy Rule, PHI must be retained by a covered entity for six years from the date when it was last in effect (§164.530(j)(2)), whereas in the Privacy Act of 1974, the information must be retained for five years or the life of the record, whichever is longer (5 USC §552a(c)(2)). Covered entities that must comply with both of these regulations, for example, a U.S. Department of Veteran’s Affairs hospital, may be noncompliant if they focus on the five year minimum in the Privacy Act rather than the six year minimum under HIPAA. Analyzing such cross-references helps requirements engineers identify conflicting compliance requirements.

Although five conflicts may not seem like a significant number, the conflicts are critical because they can lead to non-compliance. A method for identifying conflicts also can avoid the cost of building software systems that must later be re-engineered once a conflict is discovered, at greater expense. If requirements engineers limit their efforts to only examining a single regulation for legal compliance, the requirements they specify may be noncompliant with other laws, as in the case of the Privacy Act of 1974 and HIPAA example above.

In implementing GLBA, there was the potential for a substantial number of conflicts due to the allocation of authority to draft the regulation. The following eight separate federal agen-
cies were each given authority to write their own regulations to apply to financial institutions under their jurisdiction: the Federal Deposit Insurance Corporation (state-chartered banks not members of the Federal Reserve); the Federal Reserve (financial holding companies and state-chartered banks that are members of the Federal Reserve); the Office of Comptroller of the Currency (banks chartered by the federal government); the Office of Thrift Supervision (thrifts); the National Credit Union Administration (credit unions); the Securities and Exchange Commission (broker-dealers and others); and the Federal Trade Commission (financial institutions not otherwise regulated). These agencies avoided conflicts, however, by instituting a consensus process that resulted in identical requirements being imposed on all financial institutions in the GLBA Financial Privacy Rule formally issued separately by each agency.

Several factors in the financial services sector tend to reduce the number of conflicting requirements. First, in the United States, financial services have long been intensively regulated at the federal level, such as through the Glass-Steagall Act of 1933\(^3\), which created national rules for commercial and investment banks. Regulation in the financial sector is accompanied by strong enforcement mechanisms, such as when bank supervisors go onsite to examine a bank’s records. The existence of detailed, enforced legal requirements mean that industry experts are knowledgeable about the state of the law at the time of proposed amendments, and likely to be aware of any existing conflicting requirements. Second, the financial sector participates actively in the legislative and regulatory process, with the highest lobbying expenses of any industry sector\(^4\). This intensive activity by the financial sector means that possible conflicts are likely to be quickly identified, clarified, and addressed.

Otto and Antón note that definitions may conflict in regulations [91]. In our study, legal texts cite definitions from other legal texts 145 times (see Table 5.6). Among these 145 cross-references, we only identified one conflicting definition in our analysis—a conflict between the definitions of “individual” in the HIPAA Privacy Rule and the Privacy Act of 1974 as discussed in Section 3.2.3. In explaining the low number of conflicting definitions, we offer three observations. First, our experience suggests that definitional cross-references within a domain serve to clarify and reduce conflicts in legal texts by providing a common terminology, rather than introduce conflicts as previously thought [91]. Second, the coordinated promulgation of the GLBA Privacy Rule across eight agencies is evidence of efforts by drafters of legal rules to avoid conflicting definitions. Third, heavy lobbying and other political activity by regulated entities may help identify conflicting definitions, either avoiding their promulgation or clarifying them promptly. For the one conflicting definition we identified, we note that the Privacy Act of 1974 and HIPAA apply to generally distinct domains—federal agencies and healthcare, respectively. In addition, the Privacy Act of 1974 does not apply to private entities, but only

\(^3\)Pub. L. No. 73-66, 48 Stat. 162 (1933) (repealed 1999)

\(^4\)http://www.opensecrets.org/lobby/top.php?indexType=c.
to federal agencies prohibited by law from lobbying.

Lamsweerde et. al outline a set of heuristics for identifying goal conflicts [121]. We employ these heuristics to identify conflicting legal requirements during our cross-references case study. Table 5.7 outlines the conflicts we identified. These heuristics are summarized below:

- Safety goals may prevent satisfaction goals from being accomplished [121]. For example, in a healthcare software system, the goals \textit{Avoid[PatientChartEditByUnauthorizedParties]} may conflict with the goal \textit{Achieve[EmergencyAccessToPatientCharts]}.

- Goals that state information must remain confidential may conflict with goals that state the information should be shared [121]. For example, goals requiring the disclosure of certain infectious diseases to public health agencies may conflict with a patient’s desire to restrict disclosure. This heuristic helped us identify Conflict #4 in Table 5.7.

- Goals that optimize a value can conflict [121]. For instance, a conflict can be introduced by one goal that states “at least X” and another goal that states “no more than Y”, if $X > Y$. This heuristic identifies Conflicts #1 and #2 in Table 5.7.

- A goal that can have multiple instances can conflict by introducing competing goals among agents [121]. For example, in the classic meeting scheduler problem, a conflict can be introduced by multiple instances of \textit{Achieve[ScheduleMeeting]} if individuals try to schedule meetings at the same time [121].

- Goals can conflict that have overlapping achieve and avoid constraints [121]. We employ this heuristic to identify Conflicts #3 and #5 in Table 5.7. For example, the goal \textit{Achieve[ChargeForCopiesOfPHI]} conflicts with the goal \textit{Avoid[ChargingForFirstCopyOfPHI]}. 


<table>
<thead>
<tr>
<th>Index</th>
<th>Conflicting Legal Texts</th>
<th>Summary of Conflict</th>
<th>Applicable Resolution Strategies</th>
</tr>
</thead>
</table>
| 1     | • HIPAA §164.530(j)(2)  
|       | • Privacy Act of 1974\(^1\)  
|       | • 29 CFR 1910.1020\(^2\) | Length of data retention: 
|       |                           | • HIPAA: at least 5 years 
|       |                           | • Privacy Act: at least 6 years or the life of the record, whichever is longer 
|       |                           | • 29 CFR 1910.1020: at least 30 years if the employee worked for longer than a year | • Follow the most restrictive law  
|       |                           |                     | • Keep data separate |
| 2     | • HIPAA §164.524(b)(2)  
|       | • Privacy Act of 1974  
|       | • 29 CFR 1910.1020 | Length of time an organization has to respond to a request for access to data: 
|       |                           | • HIPAA: in fewer than 30 days 
|       |                           | • Privacy Act: in fewer than 10 days 
|       |                           | • 29 CFR 1910.1020: in fewer than 15 working days | • Follow the most restrictive law  
|       |                           |                     | • Keep data separate |
| 3     | • HIPAA §164.524(c)(4)  
|       | • 29 CFR 1910.1020 | Under HIPAA, a covered entity may charge a reasonable, cost-based fee when providing copies of PHI to an individual, whereas in 29 CFR 1910.1020, employers must provide the first copy of an employee’s medical record free of charge | • Obligations supersede legal privileges  
|       |                           |                     | • Keep data separate |
| 4     | • HIPAA §164.524(c)(4)  
|       | • 29 CFR 1910.1020 | HIPAA and 29 CFR 1910.1020 contain different conditions that prevent the release of protected information to individuals. Even if an organization can withhold information under one law, they must release it under the other law. | Consult legal domain expert |
| 5     | • HIPAA §164.524(c)(4)  
|       | • 29 CFR 1910.1020 | Under HIPAA, covered entities must de-identify health information before they release it, but under 29 CFR 1910.1020, they may release data to employees if personal identifiers cannot be removed. | • Do not exercise legal privileges  
|       |                           |                     | • Keep data separate |

\(^1\)Cited at §164.524(a)(2)(iv))  
\(^2\)Cited at §164.512(b)(1)(v)(C)
5.2.2 Resolving Conflicting Requirements

In this section, we provide guidance to engineers for resolving conflicting compliance requirements. The strategies are descriptive in that they were developed based on our experiences in addressing the conflicts in Table 5.7. Thus, the following strategies are based on our analysis to date and should be considered valid for the data set with which we worked [39]. Using the strategies described below, we were able to resolve four conflicts in Table 5.7. The remaining conflict, #4, requires consultation with legal domain experts to resolve.

Multiple strategies may be used to resolve a given conflict. Engineers should select a conflict resolution strategy in conjunction with system stakeholders because the selected strategy may place additional requirements on the system or impact existing non-legal software requirements.

We now discuss the conflict resolution strategies.

Follow the Most Restrictive Law

Multiple regulations contain compliance requirements that govern the same kinds of software systems (e.g. EHR systems). When a compliance requirement expressed in one legal text is more restrictive than the corresponding compliance requirement expressed in another legal text, requirements engineers should choose to follow the more restrictive of the two; in complying with the more restrictive text, the system complies with both.

To determine which legal text is more restrictive, each legal statement is classified as either:

- a *ceiling* rule, where the constraint is in the form “at least x”, or
- a *floor* rule, where the constraint is in the form “no more than y”

Once each legal statement is classified, a simple matrix determines which legal statement is more restrictive (see Table 5.8). For example, consider Conflict #1 in Table 5.7; each legal statement is a ceiling rule, therefore, the more restrictive option is to retain data for 30 years. Similarly, consider Conflict #2; each legal statement is a floor rule, thus, it can be resolved by responding to all requests within 10 days.

<table>
<thead>
<tr>
<th>Legal Text 1</th>
<th>Legal Text 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceiling rule</td>
<td>Comply with longer time period</td>
</tr>
<tr>
<td>Floor rule</td>
<td>Consult legal domain expert if not overlapping intervals</td>
</tr>
</tbody>
</table>

Table 5.8: Resolving Time Period Conflicts
Store Data Separately

Different laws apply to different sets of records. For instance, consider Conflict #1 in Table 5.7. HIPAA applies to records held by covered entities, whereas the Privacy Act applies to personally identifiable information (PII) held by agencies of the federal government, and 29 CFR 1910 applies to employee records. Recognizing the different scope of laws can permit compliance by holding records separately. For instance, employee records can be maintained in a separate database and retained for 30 years independently of the database with HIPAA records, which must be retained for 6 years. Likewise, consider Conflict #3 in Table 5.7. If HIPAA data and employee data are kept in separate data stores, then differing policies can be adopted for each data store. For example, the policy for HIPAA data can be to charge a reasonable cost-based fee, whereas the policy for employee data is to only charge for copies beyond the first.

Alternatively, the different kinds of data can be tagged using a markup; business rules can be developed for retaining the data elements for different time periods. If individual data elements are covered by conflicting legal requirements, this strategy should not be used. Rather, requirements engineers should employ other strategies outlined in this section.

Obligations Supersede Legal Privileges

An obligation is an action that an actor is required by law to perform, whereas a privilege is an action an actor may perform but is not obligated to perform [49]. Legal texts denote obligations using natural language phrases such as “must” and denote privileges using phrases such as “may” [69]. Conflicts between obligations and privileges can be resolved by not exercising legal privileges. The conflict is resolved by performing the obligated action instead of the privileged action because an obligation trumps a privilege due its priority. Consider Conflict #3 in Table 5.7, under HIPAA, covered entities have the privilege to charge a fee for copies of PHI, whereas under 29 CFR 1910.1020, employers are obligated to not charge for the first copy of an employee’s medical record. This conflict can be resolved by specifying requirements to not charge for the first copy of PHI—complying with the obligation and not exercising the privilege. Likewise, engineers can resolve Conflict #5 in Table 5.7 by not releasing data that has not been de-identified.

Consult Legal Domain Experts

Some conflicts cannot be resolved with our current set of conflict resolution strategies. For example, consider Conflict #4 in Table 5.7. Both the HIPAA Privacy Rule and 29 CFR 1910.1020 mandate that individuals and employees have access to their health information, respectively.
Both regulations also provide mutually exclusive conditions under which a covered entity or employer can withhold information from an individual or employee. For example, HIPAA allows covered entities to withhold psychotherapy notes from an individual. An employer may deny employees’ direct access to their medical records, if the medical record contains a diagnosis of a terminal or psychiatric illness (but they may be required to release the information to a third party such as the employee’s primary physician). Thus, even if an organization has the privilege to withhold health information under one law, they are obligated under the other law to release it.

Using the **Obligations Supersede Legal Privileges** strategy, Conflict #4 could be resolved, in theory, by always releasing the information or logging the release and performing *a posteriori* auditing. However, releasing the information could be unethical or encourage healthcare professionals to violate professional codes of conduct if they believe the release will bring harm on someone. In this case, the **Obligations Supersede Legal Privileges** strategy does not adequately resolve the conflict, and requirements engineers should seek legal domain experts to assist in determining the priority between these conflicting compliance requirements. Engineers may have to consult multiple subject area experts, for instance, a tax law expert may be unable to address questions about Social Security law.

5.3 Modeling Regulatory Texts and Checking Requirements for Compliance

In this section, we discuss our experiences in modeling the HIPAA Privacy and Security Rules and using those models to validate software requirements for regulatory compliance.

5.3.1 Production Rule Models Developed

Recall that we modeled the HIPAA Privacy and Security Rules as part of our multi-case study. The Privacy Rule comprises 45 CFR Part 160 and Part 164, Subparts A and E. Our analysis focuses on Subpart E, which describes privacy requirements for covered entities. The only portion of Part 160 that we model is §160.103, which contains defines key terms used throughout the Privacy Rule. We do not model the remainder of Part 160, which describes enforcement, penalties, and other requirements for governmental agencies, or Part 164, Subpart A, which describes applicability and covered entities’ legal options under HIPAA, because these sections do not typically apply to healthcare IT systems. The Security Rule is codified at 45 CFR 160 and Subparts A and C of 45 CFR 164. We include §160.103 from our Privacy Rule study, since that section contains definitions common to both rules. In addition, we model Subpart C of 45 CFR 164, namely, §164.302-164.316. We did not model §164.318 of the Security Rule or §164.534
of the Privacy Rule, as these sections relate to compliance deadlines for the rules—all of the compliance deadlines have passed. The portions of the HIPAA Privacy Rule that we modeled comprise a total of 37 printed pages whereas the Security Rule comprises seven printed pages. Our model contains 2,258 and 355 rules and took approximately 294 and 56 person hours to construct for the Privacy Rule and Security Rules, respectively.

Table 5.9 displays the number of production rules expressed using each of the concept patterns. Disjunction splitting and refactoring impacts the number of production rules that express each of the concepts. The majority of the concepts identified in our case study were privileges and obligations. In the Privacy Rule, the majority of these privileges and obligations are placed upon covered entities, whereas, individuals hold rights and the majority of the implied no-rights. We did not encounter any immunities expressed directly in the Privacy or Security Rules; instead, they are implied by disabilities. In the Security Rule, we primarily identify obligations, because the Security Rule focuses on actions that covered entities and business associates must take to protect electronic PHI. We found that liabilities are placed on third parties, e.g., business associates of covered entities. For instance, §164.504(e)(2)(ii)(B) states that business associates is liable to safeguard the information entrusted to them. It is important to model these liabilities, because many organizations are classified as business associates throughout the healthcare value chain, such as legal firms, IT subcontractors, and property management firms.

<table>
<thead>
<tr>
<th>Concept</th>
<th>HIPAA Privacy Rule</th>
<th>HIPAA Security Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rights</td>
<td>Explicit</td>
<td>Implied</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>258</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>88</td>
</tr>
<tr>
<td>Obligations</td>
<td>258</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>88</td>
<td>0</td>
</tr>
<tr>
<td>Privileges</td>
<td>177</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>No-Rights</td>
<td>2</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Powers</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Liabilities</td>
<td>29</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Immunities</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disabilities</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Using all eight Hohfeldian concepts allow us to identify compliance requirements that may have been missed by researchers that only use some of the concepts [16, 18, 69, 111]. For example, §164.520 of the HIPAA Privacy Rule regulates a covered entity’s notice of privacy practices. Section 164.520(a)(3) contains two legal statements: 1) “An inmate does not have a right to notice under this section,” and 2) “the requirements of this section do not apply to a
correctional institution that is a covered entity." Statement one is a no-right, indicated by the phrase “does not have a right to.” The second statement is a privilege, indicated by the phrase “requirements [. . . ] do not apply.” In our early work, we failed to classify the first statement as a no-right, instead classifying all of §164.520(a)(3) as a privilege, because our methodology did not include all the Hohfeldian concepts at that point [69].

We classify and model all of the Hohfeldian concepts individually. Identifying the Hohfeldian concepts individually is important for software engineers to capture for three reasons. First, they establish exceptions and priorities between legal requirements, which Breaux and Antón found to add additional constraints and requirements to legal rules [17]. Second, capturing the concepts implied by the additional Hohfeldian concepts is important for legal compliance. For example, no-rights imply privileges on the counterparty. In the no-right from §164.520(a)(3), the implied privilege is “a correctional institution is not required to provide a notice of privacy practices to inmates.” Identifying this privilege could save a correctional institution the expense of notifying their inmates and allow software developers to not have to build this functionality into their systems. On the other hand, software developers may have to build this functionality into their EHRs if they are developing systems that will be deployed in other types of covered entities in addition to correctional institutions. Third, engineers can identify inconsistencies in the law using the opposite concepts. For example, the law is inconsistent if an individual can hold both a no-right and the opposite right.

Breaux identifies various kinds of ambiguity in legal texts, such as logical ambiguity, attributive ambiguity, referential ambiguity, and under-specification [16]. During our case study, we identified an additional kind of ambiguity: structural ambiguity. Structural ambiguity occurs when different levels of a hierarchical document use the same symbols to denote headings. This causes ambiguity, because the nesting level of a heading may not be clear when another heading at a different level shares the same hierarchical symbol. For example, Figure 5.2 displays an example of structural ambiguity. There are two top-level sections in Figure 5.2, labeled (a) and (b). There are two subsections in (b), labeled (1) and (2). Section (2) also has subsections, labeled (a) and (b). Section (c), bolded in the figure, represents an instance of structural ambiguity. Heading (c) could be interpreted as a top-level section or as a subsection of (b)(2). Structural ambiguity exists in the Privacy Rule because the Privacy Rule reuses numerals at different levels of the hierarchy, as well as the symbol “i” (the lower case letter) and “i” (the first Roman numeral). For example, §164.514(e)(4)(ii)(C) of the Privacy Rule contains subsections labeled (1)-(4). The next section is labeled (5) and could be interpreted as §164.514(e)(4)(ii)(C)(5) or §164.514(e)(5). To resolve this ambiguity, software engineers must use other context clues, such as continuations or surrounding section headings, to determine the correct interpretation. In the Privacy Rule example, section (5) is a level 5 heading, because the next section heading is (iii). If section (5) was a level 2 heading, the next section would have been (i). This finding is
an important consideration for future attempts at automated tool support or natural language processing of regulatory texts.

As a result of our analysis, we discovered that privileges are more complex than obligations. For example, in the HIPAA Privacy Rule, we classified 74 fewer privileges than obligations. Despite of this, privileges have the same number of type-checking preconditions as obligations, 65 more preconditions identified using precondition keywords, only 25 fewer cross-reference preconditions, 32 more disjunctions, and one more disjunction masquerading as a logical-and. This suggests that the question “what must an organization do to comply with the law?” is less complex, though still challenging, than the question “what is an organization allowed to do under the law?”

5.3.2 Compliance Requirements Analysis (CRA) Tool

In this section, we describe the CRA Tool that aids in specifying compliance requirements and validating existing requirements for legal compliance. This tool aids software engineers in the analysis of production rule models to obtain legal requirements. Engineers can use the tool during the Query Model step (see Section 3.3.2). The three features of the tool are: a manual query interface, the precondition viewer, and the rule viewer.

- **Manual Query Interface** - This interface provides a mechanism to manually query the production rule model. This improves upon the command line interface that we originally used to query the model [69]. The interface includes a text area to list all preconditions associated with the query; all query responses are displayed in a table.

- **Precondition Viewer** - The precondition viewer lists all preconditions used in the model. The tool allows engineers to select those preconditions that are relevant to the software being developed, and will automatically copy them into the manual query interface for further analysis.
• **Rule Viewer** - The rule viewer uses a tree structure to display each of the rules sorted according to their ontological concept. Software developers can expand each of the entries to view the preconditions for that rule. Figure 5.3 displays the rule viewer with several rules loaded, with an obligation expanded.

![Rule Viewer](image)

**Figure 5.3:** Compliance Requirements Analysis (CRA) Tool - Rule Viewer

In the next section, we describe our use of the CRA Tool to validate existing software requirements for regulatory compliance.
5.3.3 Specifying Canonical Compliance Requirements

In this section, we provide an overview of our experiences with specifying canonical software requirements from the HIPAA Security Rule. We used our model of the HIPAA Security Rule to create a canonical list of compliance requirements, that we then used to check an existing set of software requirements for regulatory compliance. Recall that a canonical requirements set is a set of requirements for which every organization in a particular domain must satisfy.

We employed the CRA Tool’s rule viewer to specify a canonical list of compliance requirements. Using the rule viewer is equivalent to selecting all preconditions then querying the model. We identified three kinds of canonical requirements: software requirements, business requirements, and software development lifecycle (SDLC) requirements—requirements that regulate how the software is to be developed. Table 5.10 displays the number of each requirement we specified. Recall that legal rights create scenarios that the software must support (see Section 3.3.2); we did not specify any scenarios because we did not identify any rights in the HIPAA Security Rule.

<table>
<thead>
<tr>
<th>Requirement Type</th>
<th>Number Specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Requirement</td>
<td>29</td>
</tr>
<tr>
<td>Business Requirement</td>
<td>46</td>
</tr>
<tr>
<td>SDLC Requirement</td>
<td>4</td>
</tr>
</tbody>
</table>

In addition to the requirements types listed in Table 5.10, we specified five misuse cases to aid software engineers in disambiguating and operationalizing requirements. Misuse cases express functionality that should not exist in the system [114] and have been used to elicit security requirements [79, 114]. We use misuse cases to model legal statements such as §164.312(e)(1):

A covered entity must, in accordance with §164.306, implement technical security measures to guard against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.

We specify a misuse case to model this security-related compliance requirement, displayed in Figure 5.4. Software engineers can use the misuse case to brainstorm about new security requirements to mitigate attacks upon the system. For example, to mitigate the attack “Break encryption” in Figure 5.4, we can specify the requirement: “The system shall use industry-standard strong encryption.”

Prior researchers have identified that the law contains software requirements and business requirements [16, 34]; in our work, we identify a third class of requirements, SDLC requirements,
that researchers have not previously specified. SDLC requirements are requirements placed by the law on the covered entity, but are most appropriate to implement while the software is being constructed. For example, §164.308(a)(1)(ii)(A), listed below:

A covered entity must, in accordance with §164.306 conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity.

This legal text requires covered entities to perform a risk and vulnerability analysis of their systems. However, some covered entities, such as rural doctor offices, are not equipped to perform this sort of analysis. In addition, software security risk analysis should be integrated early in the development lifecycle [79]. Thus, we specify SDLCR to satisfy §164.308(a)(1)(ii)(A):

Software developers shall conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the system (§164.308(a)(1)(ii)(A)).

5.3.4 Analyzing the iTrust Requirements for Regulatory Compliance

In this section, we present our experiences with evaluating iTrust for compliance with the HIPAA Security Rule. For each of the 29 canonical software requirements in the Security Rule, we scan the list of 85 iTrust requirements to see if the canonical requirement is satisfied by the iTrust requirements. We specify 18 new software requirements to add to the iTrust requirements specification—13 functional requirements and five non-functional requirements. In other words, the original iTrust specification missed 65% of the requirements required for regulatory compliance. This is not unexpected, as iTrust is being developed as an instruction tool and test bed, not a production EHR. We do not refine the non-functional requirements, as previous
researchers have developed techniques for refining non-functional requirements as functional requirements (for example, see [84]).

During our study, we determined that important contextual information can be lost when a continuation is split following previous recommendations [16, 71]. A continuation occurs when a legal statement is broken across multiple lines of a legal text [16]. For example, the following legal text contains a continuation:

§164.504(a)(1) A covered entity is permitted to use or disclose protected health information as follows: (i) To the individual; (ii) For treatment, payment, or health care operations, as permitted by and in compliance with §164.506;

Prior researchers have recommended to prepend the portion of the continuation before the colon ((a)(1) in the example above) to the statements following the colon, creating multiple statements [16]. We previously recommended splitting continuations [71]. However, when examining the HIPAA Security Rule, we discovered that not all continuations can be split because important contextual information can be lost. For example, §164.306(a)(1)-(2) is listed below:

Covered entities must do the following: (1) Ensure the confidentiality, integrity, and availability of all electronic protected health information the covered entity creates, receives, maintains, or transmits. (2) Protect against any reasonably anticipated threats or hazards to the security or integrity of such information.

The antecedent of the phrase “of such information” in paragraph (2) is the electronic PHI in paragraph (1). We would introduce ambiguity if we split this continuation, because the statement corresponding to paragraph (2) would not have the antecedent. Thus, engineers should replace the pronoun with the antecedent before splitting the continuation in such cases.

5.4 The Cost of Compliance

As discussed in Section 4.2, we tracked the number of person hours we spent to calculate the cost of compliance for using our approach. If the cost is less than $90,000, then we conclude that our approach does not add cost to software development that is more than the cost of noncompliance. Table 5.11 lists the person hours spent in each case in our multi-case study. Multiplying the 513 total number of hours spent by the $46.32 national median hourly salary for a software engineer, we calculate that the cost of our approach is $23,762.16. This is below our threshold cost of $90,000, so we conclude that our approach does not add cost to software development that is likely to exceed the cost of noncompliance, addressing research question RQ4.
Table 5.11: Person Hours Spent in Multi-Case Study

<table>
<thead>
<tr>
<th>Case</th>
<th>Person Hours Spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability Study</td>
<td>35</td>
</tr>
<tr>
<td>Cross-References Study</td>
<td>128</td>
</tr>
<tr>
<td>Modeling Study</td>
<td>350</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>513</strong></td>
</tr>
</tbody>
</table>

5.5 User Study Results

In this section, we discuss participant demographics and performance and make observations based on our user study results.

5.5.1 Participant Demographics

As discussed in Chapter 4, we performed a pilot survey as part of our user study. We had 11 participants begin the pilot study and seven complete the pilot, that is, they completed every question. For the user study, we had 56 participants begin the survey and 33 complete it. We summarize the demographics of our user study in Table 5.12. Note that the numbers may not add up to the number of participants, because participants may select multiple answers. For example, a participant may be both a healthcare practitioner as well as a business analyst.

Our survey included a free-form text box where participants could enter other roles they are currently in or held previously. In the pilot study, participants entered “software engineering researcher” and “software educator” for other roles. In the full study, participants entered “development manager”, “program manager” (twice), “vice president product strategy”, “infrastructure architect”, “director of product management”, and “quality manager” for other roles.

5.5.2 Participant Performance

Recall that participants were asked to classify ten legal statements using our cross-reference taxonomy. Cross-references can be classified with multiple classifications. For the ten legal statements used in our study, the experts classified each cross-reference with either one or two classifications. We say a selection matches between a participant and the expert group when the participant selects the same classification as the expert group. For example, if the experts classified a cross-reference as a constraint, a participant’s selection would match if he or she also classified the cross-reference as a constraint. We say a participant has a correct answer when he or she makes the same number of selections as the experts and all of their selections match the expert selections. A participant response is partially correct if: (1) he or she selects
Table 5.12: Participant Demographics

<table>
<thead>
<tr>
<th>Current Role / Previous Experience</th>
<th>Pilot Study</th>
<th>User Study</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(# in role / median years of experience)</td>
<td></td>
</tr>
<tr>
<td>Business Analyst / Requirements Engineer</td>
<td>1 / 8</td>
<td>4 / 12.5</td>
</tr>
<tr>
<td>Software Architect / Developer</td>
<td>3 / 3.75</td>
<td>17 / 15</td>
</tr>
<tr>
<td>Quality Engineer / Tester</td>
<td>1 / 0</td>
<td>5 / 9.5</td>
</tr>
<tr>
<td>Implementations / Support / Services</td>
<td>0 / 0</td>
<td>1 / 7</td>
</tr>
<tr>
<td>Network Engineer / IT</td>
<td>0 / 0</td>
<td>1 / 3</td>
</tr>
<tr>
<td>Compliance / Legal</td>
<td>1 / 0</td>
<td>3 / 3</td>
</tr>
<tr>
<td>Healthcare Practitioner</td>
<td>0 / 0</td>
<td>2 / 7.5</td>
</tr>
<tr>
<td>Other - S/W Eng. Researcher</td>
<td>1 / 3</td>
<td>0 / 0</td>
</tr>
<tr>
<td>Other - S/W Educator</td>
<td>1 / 7</td>
<td>0 / 0</td>
</tr>
<tr>
<td>Other - Dev. Manager</td>
<td>0 / 0</td>
<td>1 / 15</td>
</tr>
<tr>
<td>Other - Program Manager</td>
<td>0 / 0</td>
<td>2 / 3</td>
</tr>
<tr>
<td>Other - VP Product Strategy</td>
<td>0 / 0</td>
<td>1 / 12</td>
</tr>
<tr>
<td>Other - Infrastructure Architect</td>
<td>0 / 0</td>
<td>1 / 3</td>
</tr>
<tr>
<td>Other - Dir. of Product Management</td>
<td>0 / 0</td>
<td>1 / 2.5</td>
</tr>
<tr>
<td>Other - Quality Manager</td>
<td>0 / 0</td>
<td>1 / 10</td>
</tr>
<tr>
<td>Other - S/W Support Services Manager</td>
<td>0 / 0</td>
<td>1 / 15</td>
</tr>
<tr>
<td>Other - Policy Analyst</td>
<td>0 / 0</td>
<td>1 / 10</td>
</tr>
<tr>
<td>Median Years Experience Working a Highly Regulated Domain</td>
<td>1.75</td>
<td>12.5</td>
</tr>
</tbody>
</table>

Highest Education Level
- High School Diploma / GED | 0 | 3 |
- Associates | 0 | 2 |
- Bachelors | 1 | 21 |
- Masters | 1 | 5 |
- PhD | 5 | 0 |
- Professional Degree (M.D., J.D., etc.) | 0 | 2 |

fewer or more classifications than the expert group and one or more of their selections match the experts, or (2) he or she selects the same number of classifications as the expert group, and one selection matches and others do not. A participant response is incorrect if he or she...
selects no classifications that match the expert classifications. Table 5.13 displays the possible outcomes based on the number of classifications that are made by a participant and the expert group. We give correct answers one point, partially correct answers half a point, and incorrect answers zero points.

Table 5.13: Possible User Study Outcome for a Single Question

<table>
<thead>
<tr>
<th>Participant Selected</th>
<th>Experts Selected</th>
<th>One Classification</th>
<th>Two Classifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Classification</td>
<td>1 match = correct</td>
<td>1 match = partially correct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 matches = incorrect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two Classifications</td>
<td>1 match = partially correct</td>
<td>2 matches = correct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 matches = incorrect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As displayed in Table 5.13, there are different types of partially correct answers. We do not differentiate between these different types of partially correct answers in our scoring. Instead, we give each partially correct answer half a point. We selected this scoring rubric because there is no partially correct answer that is more compliant than another partially correct answer. In fact, the impact of a partially correct or incorrect classification on compliance depends on the context of the cross-reference. For example, if a cross-reference is actually unrelated but an engineer misclassifies it as a constraint, this error may lead to a more conservative interpretation because it over-constrains the software. However, if an engineer misclassifies the cross-reference as an exception, this error may lead to less compliant software.

Before we calculated participant performance, we set aside the participants who did not complete the entire exercise. We then totaled each participant’s score based on our scoring rubric. The highest possible score for the exercise was ten points. We did not have enough participants for our data to be normal, so we use non-parametric statistical tests, specifically, we use Mann-Whitney U to test our hypothesis. The median pilot participant score was 7 out of a maximum of 10 points, while the median participant score for the full study was 5.5 out of a maximum of 10 points. Pilot participants spent a median time of 10 minutes, 53 seconds taking the survey, whereas participants taking the full study spent a median of 8 minutes, 7 seconds taking the survey.

As done in previous studies [67], we use a consensus approach to identify trends in participant
responses; based on the degree of consensus among participant responses, we can measure which questions participants performed well on and which questions they struggled with. Table 5.14 displays the participant responses by question, listed as a percentage of the total responses to the question. For each column in Table 5.14, we bold the classification that was made by the expert group. The percentages may not total 100, due to rounding. If participants achieved perfectly correct consensus on a question with a single classification, the expected outcome would be that 100 percent of the participant classifications match the expert classification for that question. For questions with two classifications, the expected outcome would be 50 percent of the participant classifications match the first expert classification and 50 percent match the second classification.

Table 5.14: Participant Responses by Question, as Percentage of Question Responses

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>Q10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pilot Study</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraint</td>
<td>71.4</td>
<td>12.5</td>
<td>12.5</td>
<td>0.0</td>
<td>28.6</td>
<td>11.1</td>
<td>0.0</td>
<td>0.0</td>
<td>75.0</td>
<td>14.3</td>
</tr>
<tr>
<td>Exception</td>
<td>0.0</td>
<td>0.0</td>
<td>87.5</td>
<td>0.0</td>
<td>0.0</td>
<td>11.1</td>
<td>50.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Definition</td>
<td>28.6</td>
<td>0.0</td>
<td>0.0</td>
<td>87.5</td>
<td>0.0</td>
<td>0.0</td>
<td>50.0</td>
<td>87.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Unrelated</td>
<td>0.0</td>
<td>12.5</td>
<td>0.0</td>
<td>12.5</td>
<td>42.9</td>
<td>0.0</td>
<td>0.0</td>
<td>12.5</td>
<td>12.5</td>
<td>42.9</td>
</tr>
<tr>
<td>Incorrect</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>General</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>28.6</td>
<td>66.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>14.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Prioritization</td>
<td>0.0</td>
<td>75.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.1</td>
<td>0.0</td>
<td>0.0</td>
<td>12.5</td>
<td>28.6</td>
</tr>
<tr>
<td><strong>Full Study</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraint</td>
<td>22.2</td>
<td>5.9</td>
<td>7.9</td>
<td>3.0</td>
<td>38.9</td>
<td>15.4</td>
<td>2.2</td>
<td>5.4</td>
<td>48.6</td>
<td>17.1</td>
</tr>
<tr>
<td>Exception</td>
<td>19.4</td>
<td>8.8</td>
<td>76.3</td>
<td>0.0</td>
<td>0.0</td>
<td>10.3</td>
<td>50.0</td>
<td>0.0</td>
<td>8.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Definition</td>
<td>11.1</td>
<td>11.8</td>
<td>2.6</td>
<td>84.8</td>
<td>16.7</td>
<td>7.7</td>
<td>30.4</td>
<td>81.1</td>
<td>16.2</td>
<td>5.7</td>
</tr>
<tr>
<td>Unrelated</td>
<td>36.1</td>
<td>5.9</td>
<td>0.0</td>
<td>6.1</td>
<td>25.0</td>
<td>5.1</td>
<td>8.7</td>
<td>2.7</td>
<td>13.5</td>
<td>40.0</td>
</tr>
<tr>
<td>Incorrect</td>
<td>0.0</td>
<td>2.9</td>
<td>2.6</td>
<td>0.0</td>
<td>2.8</td>
<td>0.0</td>
<td>2.2</td>
<td>0.0</td>
<td>2.7</td>
<td>0.0</td>
</tr>
<tr>
<td>General</td>
<td>5.6</td>
<td>5.9</td>
<td>7.9</td>
<td>3.0</td>
<td>11.1</td>
<td>59.0</td>
<td>2.2</td>
<td>10.8</td>
<td>8.1</td>
<td>22.9</td>
</tr>
<tr>
<td>Prioritization</td>
<td>5.6</td>
<td>58.8</td>
<td>2.6</td>
<td>3.0</td>
<td>5.6</td>
<td>2.6</td>
<td>4.3</td>
<td>0.0</td>
<td>2.7</td>
<td>14.3</td>
</tr>
</tbody>
</table>

5.5.3 Observations and Discussion

In this section, we discuss participant performance and make some observations about our study. Examining participant performance in Table 5.14, we see that participants performed well on three questions (Q3, Q4, and Q8), moderately well on two questions (Q2, Q7), and poorly on five questions (Q1, Q5, Q6, Q10).

Participants performed most poorly on question one, which states: “45 CFR 164.512(k)(3): A covered entity may disclose protected health information to authorized federal officials for the provision of protective services to the President or other persons authorized by 18 U.S.C. 3056
The expert group classified this as a constraint cross-reference but 39.4% of participants classified this as an unrelated cross-reference. We hypothesize that the participants and the experts had different assumptions about what requirements are unrelated to software. To test this hypothesis, we are planning on re-running our study with modifications. We will include a software specification in the participant materials; participants will be asked to classify a cross-reference as unrelated if it is unrelated to the software specification provided.

Participants appeared to perform better classifying definition cross-references than other types of cross-references. They also appeared to perform poorly on questions with multiple classifications, only making one correct classification. As discussed in Section 4.3.1, we chose to use only 10 legal statements in our study to keep it short and encourage participants to take the survey. However, this small number of legal statements means that participants only saw one or two cross-references for each classification. Thus, we lacked the statistical power to test our observations above. In our future study, we plan on using a greater number of cross-references to gain statistical power to validate classification trends.

In the remainder of this section, we discuss our statistically validated observations.

Observation #1: Software Engineers are Not Well Equipped to Understand the Impact of Cross-References on Software Requirements

In Section 4.3, we introduced the null hypothesis for our user study:

\[ H_1: \text{Individuals from the participant group have equivalent or greater precision than the expert classifications when classifying cross-references using the taxonomy.} \]

In our study, we found that software practitioners had a median score of 5.5 for the exercise. Recall that the highest possible score for our exercise was 10 (the performance of the experts). Based on the Mann Whitney U test, we find that software practitioners cannot classify cross-references with the same precision as the expert group \((p = 0.0002)\). Thus, we reject our null hypothesis, and conclude that software practitioners are not able to classify cross-references with equivalent or greater precision than the experts using our taxonomy. As discussed in Figure 3.1, cross-reference analysis is a subprocess of other compliance analysis. If software practitioners are not well equipped to perform cross-reference analysis, they are likely not well equipped to perform other compliance analysis activities. Massey finds that software engineering graduate students are “ill-prepared to identify legally compliant software requirements with any confidence” [64]. We suggest that this is not unique to students but that software practitioners are ill-prepared as well.

Prior work has stated that software engineers need tools and techniques to help them address regulatory requirements [16, 35, 64, 69, 130]. Our work suggests that software engineers are
likewise ill-equipped to use these tools and techniques even when they are provided to them. Previous studies have found that software engineers perform better on compliance tasks when provided tools and techniques (e.g., [16, 64, 130]) but to our knowledge we are the first computer science researchers to empirically study how well software practitioners perform compliance tasks when compared to experts.

In the next subsection, we compare the best and worst performers in the full study.

**Observation #2: Participants with More Experience in Regulatory Domains Perform Better**

Recall that, as part of our study, we collected various demographic characteristics for each participant, such as current role, past experience, education, and the participant’s comfort level with reading and understanding software requirements and legal texts. To analyze the best performing and worst performing participants, we first sorted participants according to their performance. Once sorted, we examined the top-performing quartile and the bottom-performing quartile, comparing each of the demographic characteristics. We only determined one weak statistically significant trend among demographic characteristics that explain participant performance—participants with more experience in regulatory domains performed better than those with less regulatory experience ($p = 0.0548$).

**Observation #3: Pilot Participants Performed Better than Software Practitioners**

The pilot participants performed better than full study participants with ($p = 0.0374$) with median scores of 7 and 5.5 respectively. We hypothesize two possible reasons for this observation. First, the pilot participants were better educated than participants in the full study; in the pilot, 85.7% of the participants had an advanced degree (a masters, PhD, or professional degree), whereas in the full study, only 21% of the participants had an advanced degree. Second, the pilot participants came from ThePrivacyPlace and Realsearch research groups. These groups have a wide range of experience with compliance, security, and privacy requirements [17, 67, 75, 91, 115, 130]. This familiarity with the research field may have better equipped them to correctly classify cross-references.

### 5.6 Chapter Summary

In this chapter, we outlined the findings of our multi-case study and user study. We discussed: the impact of regulatory evolution and cross-reference on software requirements; developing and using our production rule models; the cost of compliance, and; our user study. We conclude our work in the next section.
A conclusion is the place where you got tired thinking.
—Martin H. Fischer

Software developers must adapt their software to meet changing stakeholder demands. Changing stakeholder expectations, technology, company policy, and regulations can force software updates. In this work, we examine the latter, focusing on how regulatory change impacts software. Because the cost to change software functionality becomes exponentially more expensive in later stages of the software development lifecycle [13], our work focuses on techniques that engineers can use to identify and predict change, enabling them to concentrate on building software for the stable sections of the law, while waiting for the unstable sections to be clarified in final rulemaking. This dissertation shows overall progress towards this goal. In the remainder of this chapter, we outline our contributions (Section 6.1), discuss threats to the validity of our findings (Section 6.2), and outline a roadmap for future work (Section 6.3).

6.1 Contributions

This work provides several contributions to the field of regulatory compliance in software and requirements engineering:

- an approach to addressing regulatory evolution that incorporates both new techniques developed herein and prior art (see Figure 3.1);

- a framework to address how and why regulations evolve, and heuristics to predict which areas of a proposed regulation are subject to change (see Section 3.1);
• a taxonomy of the impacts that cross-references have on compliance requirements (see Section 3.2);

• concrete instances of conflicting compliance requirements along with conflict resolution strategies (see Section 5.2.2);

• a methodology for modeling regulations using production rules (see Section 3.3);

• a tool supported, template-based approach to validating existing requirements for regulatory compliance (see Sections 3.3.2 and 5.3.2);

• specified a set of canonical compliance requirements for the HIPAA Security Rule, using them to validate the iTrust requirements specification for compliance and identify 18 new software requirements to add to the specification to meet compliance obligations (see Sections 5.3.4 and 5.3.3);

• a new type of ambiguity—structural ambiguity—that occurs when legal texts reuse section headings at multiple levels (see Section 5.3.1);

• an improvement to the continuation splitting technique used by Breaux and other researchers [16, 130], to avoid losing important contextual information (see Section 5.3.4);

• an empirically validated finding that software practitioners are not well equipped to understand the impact of cross-references on software requirements (see Section 5.5.3); and

• an additional empirically validated finding that individuals with more experience in regulatory environments are better at classifying the impact of cross-references on software requirements than those with less experience (see Section 5.5.3).

6.2 Threats to Validity

When designing any study, care should be taken to mitigate threats to validity. There are four types of validity that should be maintained: construct validity, internal validity, external validity, and reliability [31, 127]. Construct validity addresses the degree to which a case study is in accordance with the theoretical concepts used [127]. Internal validity measures the validity of cause-effect or causal relationships identified in a study [127]. External validity is the ability of a case study’s findings to generalize to broader populations [127]. Finally, reliability is the ability to repeat a study and observe similar results [127].

Three ways to reinforce construct validity are: use multiple sources of reliable evidence; establish a chain of evidence; and have key informants review draft case study reports [127]. We employ multiple sources of evidence by selecting multiple regulations from two domains.
To establish a chain of evidence, we maintained careful documentation when performing our analyses; this documentation became the adaptability framework, cross-reference taxonomy, and production rule modeling described in Chapter 3. Finally, our draft case study reports were reviewed by several members at ThePrivacyPlace.org as well as by a law professor who was a senior manager in drafting both the HIPAA Privacy Rule and the GLBA Financial Privacy Rule.

The studies that comprise our multi-case study herein are exploratory in nature; we do not make causal inferences nor identify cause-effect relationships in our work. The adaptability rationales capture the motivation for a regulatory change. However, these are not causal in nature, because we identify them by examining the final rule’s preamble and documenting the regulatory agency’s stated reason for making a change. We do not infer these rationales ourselves. Because we make no causal inferences in our study, internal validity is not a concern [127].

We employ grounded theory analysis; thus, our results are currently applicable to the healthcare and financial regulations that we examined. Our future work will examine regulations and statutes in other domains to continue to reinforce our external validity. However, previous researchers outline anecdotal evidence that the HIPAA Privacy and Security Rule are similar to other regulatory texts [64]. Our study strengthens this anecdotal evidence, by demonstrating that techniques developed using healthcare regulations (not just the regulations promulgated pursuant to HIPAA) can be applied to other domains (namely, finance).

To reinforce our study’s reliability, we carefully document each code and classification using our grounded theory approach. Moreover, by adopting a multi-case study approach, we relied on multiple sources of evidence; herein, our GLB Act study and GLBA Financial Privacy Rule study benefited from the prior development of our previous work during our HIPAA studies. Our GLBA studies yielded further, albeit anecdotal, evidence of reliability via the addition of two additional researchers who were successfully able to apply the methodology described in Section 4.1.1 with minimal training and guidance.

### 6.3 Future Work

We have several areas of study for future work. First, we will continue to refine our approach as new regulations are released. For example, as of the time of writing, HHS just released a new update to the HIPAA Rules¹. We will examine these rules using the analysis techniques described herein. We also seek to explore how software systems can be better equipped to handle regulatory evolution. In particular, we propose as a best practice that functionality enabling regulatory compliance should be separated from other functionality in the system.

When regulations evolve, the compliance functionality can be updated independently from the rest of the system. To help accomplish this, we are designing Compliance Decision Support, a proof-of-concept system that will enable software engineers to incorporate regulatory compliance logic in their systems. This system lets engineers mark up their code to indicate compliance events, such as disclosing PHI to a third party. When a compliance event occurs, a listener runs it through a ruleset to determine the compliance implications and notify stakeholders appropriately.

To overcome the limited expressiveness of Prolog [93], we plan to explore other logics to expand the reasoning power of our logical models. Preliminary analysis indicates that our production rule framework presented in Section 3.3 can be extended to use other reasoning logics. For example, the following legal statement taken from the HIPAA excerpt in Figure 3.6 “[...] a covered health care provider may: Use the following protected health information to maintain a directory of individuals in its facility: The individual’s name”. We can model this legal statement in RuleML\(^2\) with the following rules:

\[\text{RuleML}^2\text{http://ruleml.org/}\]
The empirical user study performed in this work is foundational for future studies examining software engineers’ ability to understand and apply compliance requirements. Participants in our study spent a median of 8 minutes, 7 seconds taking the survey, meaning they spent no more than a few minutes reading the tutorial provided with the survey. We hypothesize that participants will do better with more training and feedback. We plan to conduct a follow-up study to test this hypothesis. In this future study, participants will again be provided a tutorial and a set of questions. After completing this set of questions, participants will be given feedback on their responses, then given a second set of questions. In addition, as discussed in Section 5.5.3, we will also provide participants with a software specification, to reduce ambiguity around what legal requirements may be unrelated to software systems. We will compare the students’ performance with the practitioners’ performance in the study described herein.

Finally, in future research, we hope to extend our analysis to conflicting definitions and requirements across jurisdictional boundaries, such as among the 50 U.S. states or among different nations. The number and severity of conflicts may be substantially greater across jurisdictional boundaries than within a single jurisdiction, such as the U.S. national law of the HIPAA and GLBA Privacy Rules [40]. First, it is much more difficult to develop common terminology across multiple jurisdictions, which have different legal systems that vary in numerous respects. Second, coordination is simpler within one jurisdiction than across multiple jurisdictions. For example, the federal agencies that promulgated the GLBA Financial Privacy Rule work repeatedly with each other on numerous issues, and would be subject to criticism if they simultaneously had issued varying privacy rules for different sectors of the financial services industry. In contrast, legislators in two U.S. states or two nations lack repeated interactions, and decision makers in each jurisdiction have relatively little reason to make concessions to the views of decision makers in other jurisdictions. Third, the balance of lobbying and other political forces varies across jurisdiction, so a particular industry sector may be more or less effective in achieving its goals as legal requirements are crafted in different states or nations.
REFERENCES


PROLOG PRIMER

We employ Prolog to encode production rules because of its relatively straightforward design and its prior use in the area of legal knowledge representation [8, 11, 106, 107, 109]. The syntax of a Prolog rule is:

\[
\text{result} : - \text{precondition1}, \text{precondition2}, \ldots, \text{preconditionN}.
\]

Where the symbol : - is interpreted as the if conditional, the comma symbol is interpreted as logical-and, and the period symbol is interpreted as a full stop (the end of a rule). The result is evaluated to true only if the preconditions \{precondition1, precondition2, ..., preconditionN\} are evaluated to true. The Prolog rule \(\text{father}(X,Y) : - \text{male}(X), \text{child}(Y,X)\) is read “X is the father of Y if X is male and Y is the child of X.”

In Prolog, an atom is a quoted string, name, or a sequence of special characters (: - is one example). A variable signifies a single yet unspecified quantity and begins with a capital letter. A predicate is a relationship between atoms [117]. The production rule model makes use of two built-in Prolog predicates. The \text{assert}(\text{NewFact})\) procedure adds a new fact to the knowledge base. Similarly, the \text{retract}(\text{Fact})\) procedure removes the first occurrence of the specified fact from the rules base [117].

The strength of Prolog to answer queries comes from two concepts: unification and backtracking [93]. Unification occurs when the inference engine attempts to find a single value to bind to multiple occurrences of a variable. For example, unification would occur if a single value is found for the variable Org in the predicates \text{coveredEntity}(\text{Org})\) and \text{healthPlan}(\text{Org})\). The inference engine uses backtracking to determine the result of a query. The initial query is treated as a top-level goal, then the engine searches the rules to determine the goal’s value through trial
and error.
## CHANGES PREDICTED IN THE EHR CERTIFICATION RULE

<table>
<thead>
<tr>
<th>Index</th>
<th>Proposed Text</th>
<th>Final Text</th>
<th>Heuristic</th>
<th>Accurate?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>170.202(a-b): The Secretary adopts the following standards [...] (a) Standard. The Organization for the Advancement of Structured Information Standards (OASIS) Simple Object Access Protocol (SOAP) [...] (b) Alternative Standard. A stateless, client-server, cacheable communications protocol that adheres to the principles of Representational State Transfer (REST) must be used.</td>
<td>&lt;removed&gt;</td>
<td>H4</td>
<td>Yes</td>
</tr>
<tr>
<td>Index</td>
<td>Proposed Text</td>
<td>Final Text</td>
<td>Heuristic</td>
<td>Accurate?</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-----------</td>
</tr>
<tr>
<td>2</td>
<td>170.102: Certified EHR Technology means a Complete EHR or a combination of EHR Modules, each of which: (1) Meets the requirements included in the definition of a Qualified EHR; and (2) Has been tested and certified in accordance with the certification program established by the National Coordinator as having met all applicable certification criteria adopted by the Secretary.</td>
<td>170.102: Certified EHR Technology means: (1) A Complete EHR that meets the requirements included in the definition of a Qualified EHR and has been tested and certified in accordance with the certification program established by the National Coordinator as having met all applicable certification criteria adopted by the Secretary; or (2) A combination of EHR Modules in which each constituent EHR Module of the combination has been tested and certified in accordance with the certification program established by the National Coordinator as having met all applicable certification criteria adopted by the Secretary, and the resultant combination also meets the requirements included in the definition of a Qualified EHR.</td>
<td>H1- unresolved what-is question</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>170.205: &lt;requires that certified EHRs adopt various standards such as LOINC and SNOWMED&gt;</td>
<td>170.205-207: &lt;requires that certified EHRs adopt various standards such as LOINC and SNOWMED&gt;</td>
<td>H4</td>
<td>No</td>
</tr>
<tr>
<td>Index</td>
<td>Proposed Text</td>
<td>Final Text</td>
<td>Heuristic</td>
<td>Accurate?</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>4</td>
<td>170.210(a)(1): The Secretary adopts the following standards to protect electronic health information created, maintained, and exchanged: (a) Encryption and decryption of electronic health information. (1) General. A symmetric 128 bit fixed-block cipher algorithm capable of using a 128, 192, or 256 bit encryption key must be used.</td>
<td>170.210(a)(1): The Secretary adopts the following standards to protect electronic health information created, maintained, and exchanged: (a) Encryption and decryption of electronic health information. (1) General. Any encryption algorithm identified by the National Institute of Standards and Technology (NIST) as an approved security function in Annex A of the Federal Information Processing Standards (FIPS) Publication 140-2 (incorporated by reference in §170.299).</td>
<td>H4</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>170.210(b): The date, time, patient identification, and user identification must be recorded when electronic health information is created, modified, deleted, or printed; and an indication of which action(s) occurred must also be recorded.</td>
<td>170.210(b) The date, time, patient identification, and user identification must be recorded when electronic health information is created, modified, accessed, or deleted; and an indication of which action(s) occurred and by whom must also be recorded.</td>
<td>H1- unresolved what-if question</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>170.210(c): A secure hashing algorithm must be used to verify that electronic health information has not been altered in transit. The secure hash algorithm (SHA) used must be SHA-1 or higher.</td>
<td>170.210(c): A hashing algorithm with a security strength equal to or greater than SHA-1 (Secure Hash Algorithm (SHA-1) as specified by the National Institute of Standards and Technology (NIST) in FIPS PUB 180-3 (October, 2008)) must be used to verify that electronic health information has not been altered.</td>
<td>H4</td>
<td>No</td>
</tr>
<tr>
<td>Index</td>
<td>Proposed Text</td>
<td>Final Text</td>
<td>Heuristic</td>
<td>Accurate?</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>7</td>
<td>170.210(c): &lt;see prediction #6&gt;</td>
<td>170.210(c): &lt;see prediction #6&gt;</td>
<td>H1 - unresolved ( \text{who} ) and ( \text{how-to} ) questions</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>170.302(g)(1): Electronically receive clinical laboratory test results in a structured format and display such results in human readable format.</td>
<td>170.302(h)(1): Electronically receive clinical laboratory test results in a structured format and display such results in human readable format.</td>
<td>H1 - unresolved ( \text{what-is} ) question</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>170.302(h): Enable a user to electronically select, sort, retrieve, and output a list of patients and patients’ clinical information, based on user-defined demographic data, medication list, and specific conditions.</td>
<td>170.302(i): Enable a user to electronically select, sort, retrieve, and generate lists of patients according to, at a minimum, the data elements included in: (1) Problem list; (2) Medication list; (3) Demographics; and (4) Laboratory test results.</td>
<td>H1 - unresolved ( \text{what-is} ) question</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>170.302(h)(i)(1): Calculate and electronically display quality measures as specified by CMS or states.</td>
<td>170.302(n): For each meaningful use objective with a percentage-based measure, electronically record the numerator and denominator and generate a report including the numerator, denominator, and resulting percentage associated with each applicable meaningful use measure.</td>
<td>H1 - unresolved ( \text{what-is} ) question</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>170.302(m)(2): Electronically record, retrieve, and transmit immunization information to immunization registries in accordance with the applicable state-designated standard format.</td>
<td>&lt;removed&gt;</td>
<td>H1 - unresolved ( \text{what-is} ) question</td>
<td>Yes</td>
</tr>
<tr>
<td>Index</td>
<td>Proposed Text</td>
<td>Final Text</td>
<td>Heuristic</td>
<td>Accurate?</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>12</td>
<td>170.302(r)(1): Record actions related to electronic health information in accordance with the standard specified in §170.210(b). (2) Provide alerts based on user-defined events.</td>
<td>170.302(r)(1): (r) Audit log. (1)–Record actions related to electronic health information in accordance with the standard specified in §170.210(b).</td>
<td>H1 - unresolved what-is question</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>170.302(u)(1): Encrypt and decrypt electronic health information according to user-defined preferences in accordance with the standard specified in §170.210(a)(1).</td>
<td>170.302(u)(1): Encrypt and decrypt electronic health information in accordance with the standard specified in §170.210(a)(1), unless the Secretary determines that the use of such algorithm would pose a significant security risk for Certified EHR Technology.</td>
<td>H1- unresolved what-if question</td>
<td>Yes</td>
</tr>
<tr>
<td>14</td>
<td>170.304(d): Electronically generate, upon request, a patient reminder list for preventive or follow-up care according to patient preferences based on demographic data, specific conditions, and/or medication list.</td>
<td>170.304(d): Enable a user to electronically generate a patient reminder list for preventive or follow-up care according to patient preferences based on, at a minimum, the data elements included in: (1) Problem list; (2) Medication list; (3) Medication allergy list; (4) Demographics; and (5) Laboratory test results.</td>
<td>H1- unresolved how-to question</td>
<td>No</td>
</tr>
</tbody>
</table>

Table B.1: Changes Predicted in the EHR Certification Rule
APPENDIX


\[
\begin{align*}
s_{164\_510\_a1i}(\text{CHCP}, \text{Individual}, \text{PHI}) & : - \\
& \text{about(\text{PHI}, \text{Individual})}, \\
& \text{isCHCP(\text{CHCP})}, \\
& s_{164\_510\_a1\_info}(\text{Individual}, \text{PHI}), \\
& \text{not}(s_{164\_510\_a1\_exception}(\text{CHCP}, \text{Individual}, \text{PHI})), \\
& \text{add_to_wm}(s_{164\_510\_a1i}(\text{CHCP}, \text{Individual}, \text{PHI})). \\
s_{164\_510\_a1\_info}(X, \text{name}(X)). \\
s_{164\_510\_a1\_info}(X, \text{location}(X)). \\
s_{164\_510\_a1\_info}(X, \text{generalCondition}(X)). \\
s_{164\_510\_a1\_info}(X, \text{religiousAffiliation}(X)).
\end{align*}
\]
\[
\begin{align*}
\text{privilege}(\text{CHCP}, \\
& \text{for(\text{uses(\text{CHCP}, \text{PHI}), maintains(\text{CHCP}, \text{directory}))}, \\
& '164.510(a)(1)i')) : - \\
& \text{wm}(s_{164\_510\_a1i}(\text{CHCP}, \text{Individual}, \text{PHI})).
\end{align*}
\]
\[
\begin{align*}
\text{implied}(\text{noRight}(\text{Individual}, \\
& \text{right(\text{Individual}, \text{not(\text{for(\text{uses(\text{CHCP}, \text{PHI}), maintains(\text{CHCP}, \text{directory}))}),)}), -)), \\
& '164.510(a)(1)i')) : - \\
& \text{wm}(s_{164\_510\_a1i}(\text{CHCP}, \text{Individual}, \text{PHI})).
\end{align*}
\]
\[
\begin{align*}
s_{164\_510\_a1\_onlyConditions}(\text{CHCP}, \text{Individual}, \text{PHI}) & : - \\
& \text{about(\text{PHI}, \text{Individual})}, \\
& \text{isCHCP(\text{CHCP})}, \\
& s_{164\_510\_a1\_info}(\text{Individual}, \text{PHI}), \\
& s_{164\_510\_a1\_exception}(\text{CHCP}, \text{Individual}, \text{PHI}), \\
& \text{add_to_wm}(s_{164\_510\_a1\_onlyConditions}(\text{CHCP}, \text{Individual}, \text{PHI})).
\end{align*}
\]
\[
\begin{align*}
\text{obligation}(\text{CHCP}, \\
& \text{not(\text{for(\text{uses(\text{CHCP}, \text{PHI}), maintains(\text{CHCP}, \text{directory}))}),} \\
& '164.510(a)(1)i')) : - \\
& \text{wm}(s_{164\_510\_a1\_onlyConditions}(\text{CHCP}, \text{Individual}, \text{PHI})).
\end{align*}
\]
\[
\begin{align*}
\text{implied}(\text{right}(\text{Individual},
\end{align*}
\]
not(for(use(CHCP, PHI), maintain(CHCP, directory))),
'164.510(a)(1)(i)') :-
wm(s164_510a1i_onlyConditions(CHCP, Individual, PHI)).
What are some general things you should know about research studies?

You are being asked to take part in a research study. Your participation in this study is voluntary. You have the right to be a part of this study, to choose not to participate, or to stop participating at any time without penalty. The purpose of research studies is to gain a better understanding of a certain topic or issue. You are not guaranteed any personal benefits from being in a study. Research studies also may pose risks to those that participate. In this consent form you will find specific details about the research in which you are being asked to participate. If you do not understand something in this form it is your right to ask the researcher for clarification or more information. A copy of this consent form will be provided to you. If at any time you have questions about your participation, do not hesitate to contact the researcher(s) named above.

What is the purpose of this study? The purpose of the study is to determine the ability of software engineers to use a taxonomy to classify the impact cross-references in legal texts have on software requirements.

What will happen if you take part in the study? If you agree to participate in this study, you will be asked to complete an exercise. The exercise will take approximately 20 minutes.

Risks There are no foreseeable risks to participating in the study.
**Benefits** There will be no other direct benefit to the subject or to the subject’s company. The body of knowledge will benefit from the examination of the effectiveness of requirements engineers to classify the impact that cross-references have on software.

**Confidentiality** The information in the study records will be kept confidential to the full extent allowed by law. The exercise will be using the Qualtrics survey software. No data will be collected that could link you to your responses. No reference will be made in oral or written reports that could link you to the study.

**Compensation** If you participate in the study, you will be entered into a drawing for one of 2 $20 Amazon gift cards.

**What if you have questions about this study?** If you have questions at any time about the study or the procedures, you may contact the researcher, Jeremy C. Maxwell, at jmmaxwe3@ncsu.edu.

**What if you have questions about your rights as a research participant?** If you feel you have not been treated according to the descriptions in this form, or your rights as a participant in research have been violated during the course of this project, you may contact Deb Paxton, Regulatory Compliance Administrator, Box 7514, NCSU Campus (919/515-4514).

**Consent To Participate** “I have read and understand the above information. I have received a copy of this form. I agree to participate in this study with the understanding that I may choose not to participate or to stop participating at any time without penalty or loss of benefits to which I am otherwise entitled.”

I consent □

I do not consent □
DEMographics Survey

Directions: Please answer each question to the best of your ability.

1. Which one best describes your current role (check all that apply):
   - Business Analyst / Requirements Engineer
   - Software Architect / Software Developer
   - Quality Engineer / Tester
   - Implementations / Support / Services
   - Network Engineer / IT
   - Compliance / Legal
   - Healthcare Practitioner
   - Other: ____________________

   How many years have you been in this role? ________________ Years

2. Which roles have you had worked in the past, and for how long (check all that apply)?

   - Business Analyst / Requirements Engineer ________________ Years
   - Software Architect / Developer ________________ Years
   - Quality Engineer / Tester ________________ Years
   - Implementations / Support / Services ________________ Years
   - Network Engineer / IT ________________ Years
   - Compliance / Legal ________________ Years
   - Healthcare Practitioner ________________ Years
   - Other: ________________ Years

3. How much time have you worked in a highly regulated domain such as healthcare or finance?
   Years ________________

4. To what extent do you agree or disagree with the following statements?
<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly Disagree</th>
<th>Somewhat Disagree</th>
<th>Neither Agree Nor Disagree</th>
<th>Somewhat Agree</th>
<th>Strongly Agree</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am comfortable interpreting, analyzing and specifying software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I am comfortable reading and interpreting policies, laws, and regulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I am comfortable making legal decisions on behalf of my company</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I understand what is required for software to be secure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I understand what is required for software to protect individuals privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I understand and follow current trends in the healthcare industry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I understand and follow current trends in the finance industry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. What is your highest level of education?
   □ High School Diploma / GED
   □ Associates
   □ Bachelors
   □ Masters
   □ PhD
   □ Professional degree (M.D., J.D., etc.)
TUTORIAL

Legal texts often contain cross-references from one text to another. Cross-references can reference several kinds of legal texts, including named laws (e.g., the Fair Credit Reporting Act), portions of the US Code of Federal Regulations (e.g., 16 CFR 313.3(g)), portions of the US Code (e.g., 15 USC 6801), Executive Orders (e.g., Executive Order 12333), or a general set of laws (e.g., State law). When the legal text governs software systems, these cross-references may impact software requirements. In this study, you will be asked to classify cross-references using the following taxonomy in Figure 1.

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Exception</th>
<th>Definition</th>
<th>Unrelated</th>
<th>Incorrect</th>
<th>General</th>
<th>Prioritization</th>
</tr>
</thead>
</table>

A cross-reference may have multiple classifications.

Below, we provide a description of each category in the taxonomy followed by an example. In this tutorial, examples taken from legal texts appear in a grey-shaded box:

**Constraint Cross-References** Cross-references refine existing software requirements by introducing additional constraints are called constraint cross-references. Here is an example constraint cross-reference from the Gramm-Leach-Bliley (GLBA) Financial Privacy Rule:

16 CFR 313.3(f): Consumer reporting agency has the same meaning as in section 603(f) of the Fair Credit Reporting Act (15 U.S.C. 1681a(f))

Where 16 CFR 313.3(f) is the location in the law from where the example is taken. For emphasis, we also italicize cross-references in the examples that follow.

**Exception Cross-References** Exception cross-references add exception conditions to software requirements. Here is an example exception cross-reference from the GLBA Financial Privacy Rule:

16 CFR 313.6(a)(7): The initial, annual, and revised privacy notices that you must provide […] must include each of the following items of information: […] any disclosures that you make under section 603(d)(2)(A)(iii) of the Fair Credit Reporting Act (15 U.S.C. 1681a(d)(2)(A)(iii)) (that is, notices regarding the ability to opt out of disclosures of information among affiliates)

Exception Cross-References Exception cross-references add exception conditions to software requirements. Here is an example exception cross-reference from the GLBA Financial Privacy Rule:
16 CFR 313.15(a)(5)(i): The requirements for initial notice do not apply when you disclose nonpublic personal information to a consumer reporting agency in accordance with the Fair Credit Reporting Act (15 U.S.C. 1681 et seq.)

**Definition Cross-References** Legal texts use cross-references to cite definitions from other laws in much the same way as a programmer imports object and function definitions from language libraries. Here is an example definitional cross-reference in the GLBA Financial Privacy Rule:

16 CFR 313.3(f): Consumer reporting agency has the same meaning as in section 603(f) of the Fair Credit Reporting Act (15 U.S.C. 1681a(f)).

**Unrelated Cross-References** Unrelated cross-references are not relevant to software systems, because they do not introduce software requirements for the system. For example, the following cross-reference from the HIPAA Privacy Rule is not relevant to software systems, because the way an institutional research board is formed is beyond the scope of a software system.

45 CFR 164.512(i)(a)(i)(A): A covered entity may use or disclose PHI for research, regardless of the source of funding for that research, provided that the covered entity obtains documentation that an authorization or waiver, in whole or in part, of the individual authorization required by §164.508 for use or disclosure of PHI has been approved by either: an institutional review board (IRB), established in accordance with 7 CFR 1c.107 [...]

**Incorrect Cross-References** Sometimes, updates to law make a cross-reference point to an incorrect portion of the law. For example, section 604(b)(3)(A)(ii) of the Fair Credit Reporting Act contains a cross-reference to 609(c)(3) of the same Act. However, this reference no longer exists after Congress updated the Fair Credit Reporting Act. This fact is documented in a footnote in the legal text.

FCRA 604(b)(3)(A)(ii): Except as provided in subparagraph (B), in using a consumer report for employment purposes, before taking any adverse action based in whole or in part on the report, the person intending to take such adverse action shall provide to the consumer to whom the report relates—[...] a description in writing of the rights of the consumer under this title, as prescribed by the Federal Trade Commission under section 609(c)(3)2.

2 The references in Sections 604(b)(3)(A) and 604(b)(3)(B) should be to Section 609(c)(1), not (c)(3) that no longer exists as the result of Congress’ re-organization of Section 609(c) in 2003 (FACT Act).

**General Cross-References** General cross-references are cross-references that do not reference a specific law. Here is an example general cross-reference in the GLBA Financial Privacy Rule:

16 CFR 313.17: This part shall not be construed as superseding, altering, or affecting any statute, regulation, order, or interpretation in effect in any State [...]

Often, general cross-references have multiple classifications. In the example above, the cross-reference is both a general and prioritization cross-reference.
Prioritization Cross-References Prioritization cross-references establish the priority between new law and existing law. Here is an example prioritization cross-reference in the GLBA Financial Privacy Rule:

16 CFR 313.16: Nothing in this part shall be construed to modify, limit, or supersede the operation of the Fair Credit Reporting Act (15 U.S.C. 1681 et seq.)
Directions: For the following legal statements, using the definitions on the previous pages for reference, please classify the cross-reference using the categories from the cross-reference taxonomy.

1. 45 CFR 164.512(k)(3): A covered entity may disclose protected health information to authorized federal officials for the provision of protective services to the President or other persons authorized by 18 U.S.C. 3056 […]

2. 16 CFR 313.1(b): Nothing in this part modifies, limits, or supersedes the standards governing individually identifiable health information promulgated by the Secretary of Health and Human Services under the authority of sections 262 and 264 of the Health Insurance Portability and Accountability Act of 1996, 42 U.S.C. 1320d–1320d–8

3. 45 CFR 164.524(a)(1)(iii)(B): an individual has a right of access to inspect and obtain a copy of protected health information about the individual in a designated record set, for as long as the protected health information is maintained in the designated record set, except for:
   […]
   (iii) Protected health information maintained by a covered entity that is:
       […]
       (B) Exempt from the Clinical Laboratory Improvements Amendments of 1988, pursuant to 42 CFR 493.3(a)(2).

4. 45 CFR 160.103: Health plan means an individual or group plan that provides, or pays the cost of, medical care (as defined in section 2791(a)(2) of the [Public Health Services] Act, 42 U.S.C. 300gg91(a)(2)).

5. 15 USC 6805(a)(3): This subchapter and the regulations prescribed thereunder shall be enforced by the Federal functional regulators, the State insurance authorities, and the Federal Trade Commission with respect to financial institutions and other persons subject to their jurisdiction under applicable law, as follows:
   […]

6. 45 CFR 164(e)(4)(ii)(C)(1): A data use agreement between the covered entity and the limited data set recipient must provide that the limited data set recipient will not use or further disclose the information other than as permitted by the data use agreement or as otherwise required by law

7. 45 CFR 160.103: Protected health information means individually identifiable health information: (1) Except as provided in paragraph (2) of this definition, that is: (i) Transmitted by electronic media; (ii) Maintained in electronic media; or (iii) Transmitted or maintained in any other form or medium.
(2) Protected health information excludes individually identifiable health information in: (i) Education records covered by the Family Educational Rights and Privacy Act, as amended, 20 U.S.C. 1232g;

8. 16 CFR 313.3(k)(1): Financial institution means any institution the business of which is engaging in financial activities as described in section 4(k) of the Bank Holding Company Act of 1956 (12 U.S.C. 1843(k)). An institution that is significantly engaged in financial activities is a financial institution.

9. FCRA 604(a)(5): Any consumer reporting agency may furnish a consumer report under the following circumstances and no other: To an agency administering a State plan under Section 454 of the Social Security Act (42 U.S.C. §654) for use to set an initial or modified child support award.

10. FCRA 621(a)(1): Compliance with the requirements imposed under this title shall be enforced under the Federal Trade Commission Act [15 U.S.C. §§41 et seq.] by the Federal Trade Commission with respect to consumer reporting agencies and all other persons subject thereto
# Requirements Identified in the HIPAA Security Rule

<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>Requirement</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1</td>
<td>Functional Requirement</td>
<td>The system shall maintain the confidentiality, availability, and integrity of patient records.</td>
<td>164.306(a)(1)</td>
</tr>
<tr>
<td>FR2</td>
<td>Functional Requirement</td>
<td>The system shall protect patient records from reasonably anticipated threats to the confidentiality or integrity of those records.</td>
<td>164.306(a)(2)</td>
</tr>
<tr>
<td>FR3</td>
<td>Functional Requirement</td>
<td>The system shall protect patient records from unauthorized uses or disclosures of patient records.</td>
<td>164.306(a)(3)</td>
</tr>
<tr>
<td>FR4</td>
<td>Functional Requirement</td>
<td>The system security measures shall reduce risks and vulnerabilities to a reasonable and appropriate level.</td>
<td>164.308(a)(1)(ii)(B)</td>
</tr>
<tr>
<td>FR5</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall determine whether an employee is authorized to access PHI.</td>
<td>164.308(a)(3)(ii)(A)</td>
</tr>
<tr>
<td>FR6</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall log employee operations on PHI.</td>
<td>164.308(a)(3)(ii)(A)</td>
</tr>
<tr>
<td>FR7</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall determine whether an employee’s access to PHI is appropriate.</td>
<td>164.308(a)(3)(ii)(B)</td>
</tr>
<tr>
<td>FR8</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall terminate employees’ access to PHI when their employment ends.</td>
<td>164.308(a)(3)(ii)(C)</td>
</tr>
<tr>
<td>FR9</td>
<td>Functional Requirement</td>
<td>The system shall determine whether a person’s access to PHI is consistent with the HIPAA Privacy Rule.</td>
<td>164.308(a)(4)(i), 164.312(a)(1)</td>
</tr>
<tr>
<td>FR10</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall grant access to PHI in accordance with the covered entity’s access authorization policy.</td>
<td>164.308(a)(4)(ii)(B), 164.312(a)(1)</td>
</tr>
<tr>
<td>Index</td>
<td>Type</td>
<td>Requirement</td>
<td>Source</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>FR11</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall allow system administrators the ability to establish, document, review, and modify a user’s right to access PHI.</td>
<td>164.308(a)(4)(ii)(C)</td>
</tr>
<tr>
<td>FR12</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall support periodic security updates.</td>
<td>164.308(a)(5)(ii)(A)</td>
</tr>
<tr>
<td>FR13</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall monitor login attempts and report discrepancies in login attempts.</td>
<td>164.308(a)(5)(ii)(C)</td>
</tr>
<tr>
<td>FR14</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall support password management, including creating, changing, and safeguarding passwords.</td>
<td>164.308(a)(5)(ii)(D)</td>
</tr>
<tr>
<td>FR15</td>
<td>Functional Requirement</td>
<td>The system shall respond to, document, and mitigate harmful effects from security incidents.</td>
<td>164.308(a)(6)(ii)</td>
</tr>
<tr>
<td>FR16</td>
<td>Functional Requirement</td>
<td>The system shall support creation, maintaining, and retrieving of exact backups of PHI.</td>
<td>164.308(a)(7)(ii)(A-B)</td>
</tr>
<tr>
<td>FR17</td>
<td>Functional Requirement</td>
<td>If a business associate agreement is in place, the system shall allow business associates to create, update, and access PHI.</td>
<td>164.308(b)(1)</td>
</tr>
<tr>
<td>FR18</td>
<td>Functional Requirement</td>
<td>The system shall maintain an electronic copy of the business associate agreement.</td>
<td>164.308(b)(4)</td>
</tr>
<tr>
<td>FR19</td>
<td>Functional Requirement</td>
<td>The system shall address the final disposition of PHI.</td>
<td>164.310(d)(2)(i)</td>
</tr>
<tr>
<td>FR20</td>
<td>Functional Requirement</td>
<td>The system shall assign each user a unique username.</td>
<td>164.312(a)(2)(i)</td>
</tr>
<tr>
<td>FR21</td>
<td>Functional Requirement</td>
<td>The system shall allow medical professionals access to PHI in emergency situations.</td>
<td>164.312(a)(2)(ii)</td>
</tr>
<tr>
<td>FR22</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall automatically end a user session (log a user out) after a predetermined period of time.</td>
<td>164.312(a)(2)(iii)</td>
</tr>
<tr>
<td>FR23</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall have a means to encrypt and decrypt PHI.</td>
<td>164.312(a)(2)(iv)</td>
</tr>
<tr>
<td>FR24</td>
<td>Functional Requirement</td>
<td>The system shall log user operations on PHI.</td>
<td>164.312(b)</td>
</tr>
<tr>
<td>FR25</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall contain a means of verifying that PHI has not been altered or deleted by an unauthorized user.</td>
<td>164.312(c)(2)</td>
</tr>
<tr>
<td>FR26</td>
<td>Functional Requirement</td>
<td>The system shall have a means to identify users.</td>
<td>164.312(d)</td>
</tr>
<tr>
<td>FR27</td>
<td>Functional Requirement</td>
<td>The system shall safeguard PHI that is in transit from unauthorized access.</td>
<td>164.312(e)(1)</td>
</tr>
<tr>
<td>Index</td>
<td>Type</td>
<td>Requirement</td>
<td>Source</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>FR28</td>
<td>Functional Requirement</td>
<td>The system shall detect improper modifications to PHI.</td>
<td>164.312(e)(2)(i)</td>
</tr>
<tr>
<td>FR29</td>
<td>Functional Requirement</td>
<td>(addressable) The system shall encrypt PHI whenever deemed appropriate.</td>
<td>164.312(e)(2)(ii)</td>
</tr>
<tr>
<td>FR30</td>
<td>Functional Requirement</td>
<td>The system shall use industry-standard strong encryption.</td>
<td>Misuse case</td>
</tr>
<tr>
<td>SDLC1</td>
<td>SDLC Requirement</td>
<td>Software developers shall conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity.</td>
<td>164.308(a)(1)(ii)(A)</td>
</tr>
<tr>
<td>SDLC2</td>
<td>SDLC Requirement</td>
<td>Software developers shall assess the criticality of applications or data in support of emergency contingency plan components.</td>
<td>164.308(a)(7)(ii)(E)</td>
</tr>
<tr>
<td>SDLC3</td>
<td>SDLC Requirement</td>
<td>Software developers shall perform a periodic technical and nontechnical evaluation–based initially upon the standards of the Security Rule and subsequently, in response to environmental or operational changes affecting the security of electronic PHI—that establishes the extent to which an entity’s security policies and procedures meet the requirements of HIPAA.</td>
<td>164.308(a)(8)</td>
</tr>
<tr>
<td>SDLC4</td>
<td>SDLC Requirement</td>
<td>Software developers shall take into account the following factors when deciding to use a particular security measure: the covered entity’s size, complexity, capabilities, technical infrastructure, hardware security capabilities, software security capabilities, and cost.</td>
<td>164.306(b)(2)</td>
</tr>
<tr>
<td>BR1</td>
<td>Business Requirement</td>
<td>A covered entity shall ensure their employees comply with the HIPAA Security Rule.</td>
<td>164.306(a)(4)</td>
</tr>
<tr>
<td>BR2</td>
<td>Business Requirement</td>
<td>A covered entity shall periodically review and update security controls to address changing risks.</td>
<td>164.306(e)</td>
</tr>
<tr>
<td>BR3</td>
<td>Business Requirement</td>
<td>A covered entity shall meet each addressable requirement or implement a compensating control.</td>
<td>164.306(d)</td>
</tr>
<tr>
<td>BR4</td>
<td>Business Requirement</td>
<td>A covered entity shall implement policies and procedures to prevent, detect, contain, and correct security violations.</td>
<td>164.308(a)(1)(i)</td>
</tr>
<tr>
<td>Index</td>
<td>Type</td>
<td>Requirement</td>
<td>Source</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>BR5</td>
<td>Business Requirement</td>
<td>A covered entity shall conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity.</td>
<td>164.308(a)(1)(ii)(A)</td>
</tr>
<tr>
<td>BR6</td>
<td>Business Requirement</td>
<td>A covered entity shall apply appropriate sanctions against workforce members who fail to comply with its security policies and procedures.</td>
<td>164.308(a)(1)(ii)(C)</td>
</tr>
<tr>
<td>BR7</td>
<td>Business Requirement</td>
<td>A covered entity shall regularly review records of information system activity, such as audit logs, access reports, and security incident tracking reports.</td>
<td>164.308(a)(1)(ii)(D)</td>
</tr>
<tr>
<td>BR8</td>
<td>Business Requirement</td>
<td>A covered entity shall identify the security official who is responsible for the development and implementation of security policies and procedures.</td>
<td>164.308(a)(2)</td>
</tr>
<tr>
<td>BR9</td>
<td>Business Requirement</td>
<td>A covered entity shall implement policies and procedures to ensure that all members of its workforce have appropriate access to electronic protected health information.</td>
<td>164.308(a)(3)(ii)(A)</td>
</tr>
<tr>
<td>BR10</td>
<td>Business Requirement</td>
<td>Health care clearinghouse that are a part of a larger organization shall implement protect the ePHI from unauthorized access by the larger organization.</td>
<td>164.308(a)(4)(ii)(A)</td>
</tr>
<tr>
<td>BR11</td>
<td>Business Requirement</td>
<td>&quot;The covered entity shall implement a security awareness and training program for all members of its workforce.&quot;</td>
<td>164.308(a)(5)(i)</td>
</tr>
<tr>
<td>BR12</td>
<td>Business Requirement</td>
<td>The covered entity shall guard against, detect, and report malicious software.</td>
<td>164.308(a)(5)(ii)(B)</td>
</tr>
<tr>
<td>BR13</td>
<td>Business Requirement</td>
<td>The covered entity shall establish an emergency response plan (for example, fire, vandalism, system failure, and natural disaster) that damages systems that contain ePHI.</td>
<td>164.308(a)(7)(i)</td>
</tr>
<tr>
<td>BR14</td>
<td>Business Requirement</td>
<td>The covered entity shall establish a disaster recovery plan.</td>
<td>164.308(a)(7)(ii)(C)</td>
</tr>
<tr>
<td>BR15</td>
<td>Business Requirement</td>
<td>The covered entity shall test DR/BC plans.</td>
<td>164.308(a)(7)(ii)(D)</td>
</tr>
<tr>
<td>BR16</td>
<td>Business Requirement</td>
<td>The covered entity shall perform a periodic evaluation of its security controls for compliance with HIPAA.</td>
<td>164.308(a)(8)</td>
</tr>
<tr>
<td>Index</td>
<td>Type</td>
<td>Requirement</td>
<td>Source</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>BR17</td>
<td>Business Requirement</td>
<td>The covered entity shall limit physical access to its ePHI and the facility(ies) in which they are housed, while ensuring that properly authorized access is allowed.</td>
<td>164.310(a)(1)</td>
</tr>
<tr>
<td>BR18</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity’s DR/BCP shall allow facility access in support of restoration of lost data under the disaster recovery plan and emergency mode operations plan in the event of an emergency.</td>
<td>164.310(a)(2)(i)</td>
</tr>
<tr>
<td>BR19</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity shall establish a facilities security plan for safeguard the facility and the equipment therein from unauthorized physical access, tampering, and theft.</td>
<td>164.310(a)(2)(ii)</td>
</tr>
<tr>
<td>BR20</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity shall control and validate a person’s access to facilities based on their role or function, including visitor control, and control of access to software programs for testing and revision.</td>
<td>164.310(a)(2)(iii)</td>
</tr>
<tr>
<td>BR21</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity shall document repairs and modifications to the physical components of a facility which are related to security (for example, hardware, walls, doors, and locks).</td>
<td>164.310(a)(2)(iv)</td>
</tr>
<tr>
<td>BR22</td>
<td>Business Requirement</td>
<td>The covered entity shall specify the proper functions to be performed, the manner in which those functions are to be performed, and the physical attributes of the surroundings of a specific workstation or class of workstation that can access ePHI.</td>
<td>164.310(b)</td>
</tr>
<tr>
<td>BR23</td>
<td>Business Requirement</td>
<td>The covered entity shall implement physical controls for all systems containing electronic PHI, to prevent unauthorized access.</td>
<td>164.310(c)</td>
</tr>
<tr>
<td>BR24</td>
<td>Business Requirement</td>
<td>The covered entity shall implement policies and procedures that govern the receipt and removal of hardware and electronic media that contain electronic protected health information into and out of a facility, and the movement of these items within the facility.</td>
<td>164.310(d)(1)</td>
</tr>
<tr>
<td>BR25</td>
<td>Business Requirement</td>
<td>The covered entity shall address the final disposition of electronic protected health information, and/or the hardware or electronic media on which it is stored.</td>
<td>164.310(d)(2)(i)</td>
</tr>
</tbody>
</table>

125
<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>Requirement</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR26</td>
<td>Business Requirement</td>
<td>The covered entity shall implement procedures for removal of electronic protected health information from electronic media before the media are made available for re-use.</td>
<td>164.310(d)(2)(ii)</td>
</tr>
<tr>
<td>BR27</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity shall maintain a record of the movements of hardware and electronic media and any person responsible therefore.</td>
<td>164.310(d)(2)(iii)</td>
</tr>
<tr>
<td>BR28</td>
<td>Business Requirement</td>
<td>(addressable) The covered entity shall create a retrievable, exact copy of electronic protected health information, when needed, before movement of equipment.</td>
<td>164.310(d)(iv)</td>
</tr>
<tr>
<td>BR29</td>
<td>Business Requirement</td>
<td>Business associates of the covered entity shall implement administrative, physical, and technical safeguards that reasonably and appropriately protect the confidentiality, integrity, and availability of the electronic protected health information that it creates, receives, maintains, or transmits on behalf of the covered entity.</td>
<td>164.314(a)(2)(i)(A)</td>
</tr>
<tr>
<td>BR30</td>
<td>Business Requirement</td>
<td>Business associates of the covered entity shall ensure that any agent, including a subcontractor, to whom it provides such information agrees to implement reasonable and appropriate safeguards to protect it.</td>
<td>164.314(a)(2)(i)(B)</td>
</tr>
<tr>
<td>BR31</td>
<td>Business Requirement</td>
<td>Business associates of the covered entity shall report to the covered entity any security incident of which it becomes aware;</td>
<td>164.314(a)(2)(i)(C)</td>
</tr>
<tr>
<td>BR32</td>
<td>Business Requirement</td>
<td>When a business associate and a covered entity are both government entities, the covered entity may rely on a memorandum of understanding with the business associate that contains terms that safeguard ePHI.</td>
<td>164.314(a)(2)(ii)(A)(1)</td>
</tr>
<tr>
<td>BR33</td>
<td>Business Requirement</td>
<td>When a business associate and a covered entity are both government entities, the covered entity may rely on other laws that require the business associate to safeguard ePHI.</td>
<td>164.314(a)(2)(ii)(A)(1)</td>
</tr>
<tr>
<td>BR34</td>
<td>Business Requirement</td>
<td>A covered entity may permit access to ePHI by a business associate if the business associate is required to access the ePHI by law and the covered entity could not obtain assurances that the business associate will safeguard the information.</td>
<td>164.314(a)(2)(ii)(B)</td>
</tr>
<tr>
<td>Index</td>
<td>Type</td>
<td>Requirement</td>
<td>Source</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>BR35</td>
<td>Business Requirement</td>
<td>A covered entity may omit contract termination clauses from its business associate agreements if the clauses would be inconsistent with other laws.</td>
<td>164.314(a)(2)(ii)(C)</td>
</tr>
<tr>
<td>BR36</td>
<td>Business Requirement</td>
<td>A group health plan shall ensure that its plan documents provide that the plan sponsor will reasonably and appropriately safeguard electronic protected health information created, received, maintained, or transmitted to or by the plan sponsor on behalf of the group health plan.</td>
<td>164.314(b)(1)</td>
</tr>
<tr>
<td>BR37</td>
<td>Business Requirement</td>
<td>Plan sponsors of group health plans shall implement administrative, physical, and technical safeguards that reasonably and appropriately protect the confidentiality, integrity, and availability of the ePHI that it creates, receives, maintains, or transmits on behalf of the group health plan.</td>
<td>164.314(b)(2)(i)</td>
</tr>
<tr>
<td>BR38</td>
<td>Business Requirement</td>
<td>Plan sponsors of group health plans shall ensure that security controls are in place to provide separation from the plan sponsor and the group health plans.</td>
<td>164.314(b)(2)(ii)</td>
</tr>
<tr>
<td>BR39</td>
<td>Business Requirement</td>
<td>Plan sponsors of group health plans shall ensure that any agent, including a subcontractor, to whom it provides ePHI agrees to implement reasonable and appropriate security measures to protect the information.</td>
<td>164.314(b)(2)(iii)</td>
</tr>
<tr>
<td>BR40</td>
<td>Business Requirement</td>
<td>Plan sponsors of group health plans shall report to the group health plan any security incident of which it becomes aware.</td>
<td>164.314(b)(2)(iv)</td>
</tr>
<tr>
<td>BR41</td>
<td>Business Requirement</td>
<td>A covered entity may change its policies and procedures at any time, provided that the changes are documented and comply with HIPAA.</td>
<td>164.316(a)</td>
</tr>
<tr>
<td>BR42</td>
<td>Business Requirement</td>
<td>A covered entity shall maintain the policies and procedures implemented to comply with HIPAA in written (which may be electronic) form.</td>
<td>164.316(b)(1)(i)</td>
</tr>
<tr>
<td>BR43</td>
<td>Business Requirement</td>
<td>A covered entity shall maintain documentation in a written (which may be electronic) record of the action, activity, or assessment.</td>
<td>164.316(b)(1)(ii)</td>
</tr>
<tr>
<td>BR44</td>
<td>Business Requirement</td>
<td>A covered entity shall maintain documentation required by HIPAA for 6 years from the date of its creation or the date when it last was in effect, whichever is later.</td>
<td>164.316(b)(2)(i)</td>
</tr>
</tbody>
</table>
A covered entity shall make documentation available to those persons responsible for implementing the procedures to which the documentation pertains.

A covered entity shall review documentation periodically, and update as needed, in response to environmental or operational changes affecting the security of the ePHI.

<table>
<thead>
<tr>
<th>Index</th>
<th>Type</th>
<th>Requirement</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR45</td>
<td>Business Require</td>
<td>A covered entity shall make documentation available to those persons responsible for implementing the procedures to which the documentation pertains.</td>
<td>164.316(b)(2)(ii)</td>
</tr>
<tr>
<td>BR46</td>
<td>Business Require</td>
<td>A covered entity shall review documentation periodically, and update as needed, in response to environmental or operational changes affecting the security of the ePHI.</td>
<td>164.316(b)(2)(iii)</td>
</tr>
</tbody>
</table>

Table E.1: Requirements Identified in HIPAA Security Rule