UMBC

DISSERTATION APPROVAL SHEET

Title of Dissertation: Regulatory Compliance within the Software Industry: Interpretation of Regulatory Ambiguity as a Compliance Concern

Name of Candidate: Evelyn Kempe
ekempe1@umbc.edu
Doctor of Philosophy, 2024

Graduate Program: Department of Information Systems

Dissertation and Abstract Approved:

DocuSigned by:
Sreedevi Sampath
1EF375110DC9442...

Sreedevi Sampath
sampath@umbc.edu
Associate Professor
Information Systems
7/1/2024 | 11:39 AM EDT

DocuSigned by:
Aaron Massey
8309A5B5A62148E...

Aaron Massey
akmassey@umbc.edu
Research Assistant Professor
Information Systems
7/1/2024 | 1:57 PM EDT

NOTE: *The Approval Sheet with the original signature must accompany the thesis or dissertation. No terminal punctuation is to be used.

Curriculum Vitae

Name: Evelyn Marie Kempe.

Degree and date to be conferred: Ph.D., August 19, 2024
Major: Information Systems

Collegiate institutions attended:

University of Maryland, Baltimore County
Ph.D. Information Systems August 2024

University of Maryland University College
M.S. Information Technology – Telecommunication Management Spec. December 2012
M.S. Cybersecurity Policy May 2015

Georgia State University
B.S. Computer Science August 2002

Military Education:

Joint & Combined Warfighting School, Joint Forces Staff College, Norfolk VA January 2018 – March 2018
Command & General Staff College, Ft Belvoir VA May 2012 – August 2012
Telecommunication Systems Engineer Course, Ft Eisenhower GA April 2009 – November 2009
Signal Basic Officer Leader Course, Ft Eisenhower GA October 2008 – March 2009
Ordnance Officer Basic Course, Aberdeen Proving Grounds MD November 2002- April 2003

Certification and Licenses:

Certified Information Systems Security Professional (CISSP) June 2017 - Present

Professional Publications:

Evelyn Kempe and Aaron K Massey. Regulatory and security standard compliance throughout the software development lifecycle. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021.

Evelyn Kempe and Aaron Massey. Perspectives on regulatory compliance in software engineering. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 46–57. IEEE, 2021.

Evelyn Kempe, Samin Semsar, Aaron Massey, Sreedevi Sampath, and Carolyn Seaman. Modeling, analyzing, and communicating regulatory ambiguity: An empirical study. In

Workshop on Multi-disciplinary, Open, and RElevant Requirements Engineering (MO2RE 2024). ACM, 2024.

Professional Work Experience:

United States Military Academy (USMA), West Point NY July 2022 - Present
Assistant Professor at the within the Electrical Engineering and Computer Science Department teaching students on the core foundations of cyber (to include data mining and cybersecurity) and computer networking.

Joint Spectrum Center, Annapolis MD October 2016 - August 2019
Cyber security Branch Chief for the Joint Spectrum Center (JSC) and Defense Spectrum Organization (DSO). Provided information systems technology support to JSC/DSO to enable net-centric spectrum operations across the globe.

White House Communication Agency (WHCA), Joint Base Anacostia-Bolling DC
November 2014 – October 2016
Chief Network Engineer for WHCA. Responsible for strategic planning and execution for Cyber Protection and future technology initiatives in support of the President of the United States.

U.S. Army Cyber Command (ARCYBER) /Network Enterprise Technology Command (NETCOM), Fort Belvoir VA April 2012 – October 2016
Infrastructure Branch Chief for the U.S. Army's Enterprise Management Division for ARCYBER/NETCOM Current Operations Division. Responsible for the operations, maintenance, and modernization activities within the Enterprise Management Division for the ARCYBER/NETCOM.

3rd Infantry Divisions, Fort Stewart GA January 2010- April 2012
Telecommunication System Engineer for the 3rd Infantry Division (3ID). Setup and maintain tactical computer networks in field and garrison environment to support 3ID tactical mission.

A Co, 299th Brigade Support Battalion (BSB), 1st Infantry Division (1ID), Fort Riley KS
June 2007- October 2008
Company Commander of a Tactical Distribution Company with 299th BSB under the 1st ID. Responsible for the health, safety, and training of 183 soldiers and the Forward Supply Point, charged with pushing tactical supplies to the 2nd Heavy Brigade Combat Team under the 1st ID.

18th Combat Support Battalion, Grafenwoehr, GE August 2005 – June 2007
Adjutant lead (Battalion S-1) for the 18th Corps Support Battalion. Responsible for administrative actions and personnel management for over 800 assigned personnel and in charge of the health and safety of five soldiers within the personnel actions office.

529th Ordnance Company Vilseck, GE May 2003 – July 2005
Platoon leader for an Ammunition supply point located in Vilseck, Germany. Responsible for 44 soldiers, 18 vehicles, and issuing training ammunition for US Army Europe based units within the Grafenwoehr Training Area.

ABSTRACT

Title of dissertation: REGULATORY COMPLIANCE WITHIN THE SOFTWARE INDUSTRY: INTERPRETATION OF REGULATORY AMBIGUITY AS A COMPLIANCE CONCERN

Evelyn Kempe, Doctor of Philosophy, 2024

Dissertation directed by: Professor Sreedevi Sampath and Professor Aaron K. Massey
Department of Information Systems

Software companies must demonstrate and communicate due diligence toward compliance with applicable regulations and laws within their organizational processes and procedures. The ambiguous phrasing within regulations (i.e., regulatory ambiguities), though, can be challenging for a software developer trying to develop and interpret regulatory compliance requirements for software. Because of this challenge, software organizations or development teams need help communicating and documenting their compliance process during software development. Legal consultants, regulating officials, or compliance auditors can have similar challenges trying to interpret the development work of a software organization and determine if they have applied the requisite amount of due diligence toward regulatory compliance.

My dissertation studies a process to assist software developers in interpreting regulatory ambiguities and accomplishes three goals. The first goal is to understand the software industries' perceptions of compliance through an interview study and survey.

The second goal is to observe the reasoning and communication behind interpreting and modeling regulatory ambiguities within a group of software practitioners via a multi-case study. Finally, goal three validates the ambiguity modeling process as useful from an auditor's perspective through a focus group.

The ambiguity modeling process within this work elicits and documents regulatory analysis to support technical compliance decisions and due diligence within a software development process that is reviewable by regulators or external third parties interested in a software organization's compliance procedures. This approach has advantages for various stakeholders. This process allows software developers to communicate specific instances of 'gray areas' in regulation, such as conflicting requirements or unclear terminology, during compliance requirements development so they may receive further guidance and resolution. For auditors assessing organizations for regulatory compliance, ambiguity models demonstrate that software organizations are aware of and discuss their compliance requirements. The models can facilitate meaningful conversations for other stakeholder groups, bridging communication gaps with a software engineering team that can hinder regulatory compliance analysis and development. For the software engineering community, prior research lists challenges with regulatory compliance. My research addresses some of those challenges while promoting compliance communication amongst software stakeholders and assisting in developing a compliance culture within software organizations.

Regulatory Compliance within the Software Industry: Interpretation of Regulatory Ambiguity as a Compliance Concern

by

Evelyn M. Kempe

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland Baltimore County in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Sreedevi Sampath, Chair/Co-Advisor
Professor Aaron K. Massey, Co-Advisor
Professor Carolyn Seaman
Jeff Kosseff, JD
Professor Foad Hamidi

Dedication

To my mom and dad, Patricia and David: Thank you for your love and support. For all the times you came up to watch the kids and allowed me to focus on my schoolwork and research. I could not have done this without your help.

To my kids, Emily and Christopher: You guys are my heart and the reason I work hard every day. Mommy loves you!

Acknowledgments

I would like to take this opportunity to thank the following people:

- Dr. Sreedevi Sampath and Dr. Aaron K. Massey for the countless hours and meetings you spent supporting and mentoring me as my advisors;
- My dissertation committee members (Dr. Carolyn Seaman, Dr. Foad Hamidi, and Jeffrey Kosseff, JD) for your feedback and guidance;
- My fellow graduate student, Samin Semsar, at UMBC who shared her time and talents updating the Ambiguity Heuristic Analysis Builder and co-authoring on my papers;
- Dr. Maria Ebling, my amazing work colleague at the US Military Academy at West Point. Thank you for all your assistance with recruitment for my Focus Group, your review and edits of chapters, and being a sounding board when I needed to talk;
- COL Robert Harrison, my supervisor at the US Military Academy at West Point. Thank you for your sound advice and support;
- The faculty and staff at the US Military Academy at West Point. You covered my classes when I had to go to conferences. You discussed topics with me and gave me insights to consider. You were there when I asked for help. Thank you.
- My professors at UMBC who taught me skills in research and life that I hope to pass on to the next generation of students.

- To the participants in my studies for dedicating your time, energy, and feedback as I pursued my research;
- To my family and friends, whose love and support got me through this stressful time in my life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AHAB	Ambiguity Heuristics Analysis Builder
AO	Authorizing Official
ATC	Authority to Connect
ATO	Authority to Operate
BDD	Behavior Driven Developlment
CCPA	California's Consumer Privacy Act of 2018
EPA	United States Environmental Protection Agency
FAA	Federal Aviation Administration
FTC	Federal Trade Commission
GDPR	General Data Protection Regulation
GRL	Goal-oriented Requirement Language
HIPAA	Health Insurance Portability and Accountability Act
ICCT	International Council on Clean Transportation
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
IT	Information Technology
NICE	National Initiative for Cybersecurity Education
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSF	National Science Foundation
RAM	Regulatory Ambiguity Modeling
RC	Regulatory Compliance
RELAW	International Workshop in Requirements Engineering Law
SDLC	Software Development Lifecycle
SDP	Software Development Process
SSC	Security Standard Compliance
STIG	Security Technical Implementation Guides
UCM	Use Case Maps
UML	Unified Modeling Language
URN	User Requirment Notation

Chapter 1

Introduction

Software practitioners in the industry must implement a growing list of regulatory and security mandates to satisfy societal concerns. However, these laws, regulations, and security standards often contain 'regulatory ambiguities'. A regulatory ambiguity is phrases or statements within a statute or regulation that lack a clear meaning, allowing for multiple or no interpretations. Even though such ambiguities exist for legitimate reasons, they can make demonstrating compliance with a law or regulation challenging. First, they introduce a fog of uncertainty about what compliance with a regulation might look like. Second, they create opportunities for organizations to exploit the gray areas within the law for their benefit without much consideration of the consequences of their actions, as seen in the examples in the next section.

1.1 Background and Motivation

Boeing had a competitive edge on commercial narrow-body airliners until the new Airbus A320 NEO family's fuel-efficient engine came in 2010 [2, 3, 43]. Wanting to regain some of its competitive hold on the airline industry, Boeing designed its fuel-efficient engine for their new 737-MAX aircraft model [43]. The 737-MAX planes are the latest model for Boeing's 737 aircraft series, known as the 'Boeing 737 Next Generation (N.G.) series'. However, the new engine's placement on the 737-MAX differed from the pre-

vious aircraft models in the 737 series due to the engine's larger size. This placement caused issues in the flight trajectory for the operation of the 737-MAX [109]. Without some intermittent system to autocorrect flight trajectory, pilots certified to fly the Boeing 737 series would have had to undergo pilot certification training to operate Boeing's 737-MAX plane, or Boeing would need to redesign the 737-MAX plane structure. Hence, the operations of the aircraft are the same as those of the previous 737 series aircraft. Either scenario would require a full Federal Aviation Administration (FAA) Airworthiness certification inspection for the 737-MAX aircraft and mandatory pilot training for Boeing's customers. Certification and training are costly and time-consuming for Boeing and its customers. Therefore, Boeing decided to rely on a software system known as the 'Maneuvering Characteristics Augmentation System (MCAS)' to automatically mitigate any flight trajectory problems and give the new 737-Max the same operational design specification rather than go through the expense of structural redesign of 737-MAX and pilot training per regulatory certification requirements [109, 61]. By presenting the 737-MAX as an aircraft with almost identical structural and operational design specifications as its predecessor, the Boeing 737 Next Generation (N.G.) series, the FAA allowed Boeing to self-certify [1] the 737-MAX under the Boeing 737 Next Generation (N.G.) series.

[1] Self-certification is the FAA delegating safety analysis of similar series aircraft to airline manufacturers. This delegation allows aircraft manufacturers to update and certify similar aircraft designs under previous version design specifications as long as no significant updates occurred that changed the operations of the aircraft. Self-certification is the FAA delegating the certification process for airworthiness to qualified professionals known as Organization Designation Authorization unit members internal to the airline manufacturers. This delegation allows aircraft manufacturers to update and certify similar aircraft designs under previous version design specifications as long as no significant updates occurred that changed the

The problem was that the aerodynamic operations were autocorrected and masked by the MCAS, so the operational design was not the same. Furthermore, Boeing did not disclose the software-defined system to pilots operating the 737-MAX, thinking that the risk of the MCAS crashing the plane was low or that pilots would be able to intervene in time to prevent a crash [11, 14, 9]. Unfortunately, the decision to avoid regulatory certification requirements and the required certification pilot training by Boeing resulted in two fatal crashes, the loss of 346 lives, a 20-month grounding of all Boeing 737-MAX, and $23.5 billion in fines, compensations, and production to date [75].

Malicious compliance [2] refers to following rules or directives in a way that meets the "letter of the law," sometimes by exploiting regulatory ambiguities that undermine the law's intended spirit. Boeing assessed the MCAS impact on the safe operations of the 737-MAX as low per the FAA's guidance, thinking the MCAS would not pitch the 737-MAX into a dive and pilots would intervene in time to prevent a plane crash. Boeing failed to fully understand and test what the MCAS could and did do within the operations of the 737-MAX aircraft when it received bad data from a sensor to correct the aircraft's plane trajectory [9, 10, 72, 60, 90]. At a minimum, pilots needed to *know* about the MCAS and have additional training to maintain control of the plane and override the MCAS in case of a system failure [9, 10, 72]. Instead, Boeing followed the "letter of the FAA regulation" and assessed the MCAS as a low risk to the operations of the aircraft

operations of the aircraft. Self-certification reduces regulatory overhead and delays to aircraft certification, which would require more resources than the FAA has available for airworthiness audit [11, 10, 13]

[2] There is no universal definition for malicious compliance. In this dissertation, malicious compliance is defined as people conforming to the letter, but not the spirit, of a request.

to get through the certification process as quickly as possible, exploiting the gray area of the FAA's self-certification process rather than following the "spirit" behind FAA's self-certification process. The intent is that internal organizational auditing resources appointed by the FAA [10, 13] involved with designing and testing the aircraft system would have a much better insight into the risk to the aircraft's safe operation and report accordingly to the FAA per their ethical obligations [72, 60, 90] versus an outside team of FAA auditors. However, Boeing was far too concerned with maintaining a production schedule to stop, think, and assess the MCAS's impacts on the 737-MAX operations properly per the intent of the FAA self-certification process, which would have paused production of the 737-MAX [72, 8, 10]!

Enforcement testing and procedures are to confirm compliance with an industry's regulatory standards and requirements for the benefit of society. One problem with enforcement testing and procedures is focusing on outputs versus looking at some steps within a software development process. If all an enforcement agency looks at is the final output, not the process to get that output, the enforcement and auditing process is vulnerable to exploitation. They are vulnerable because companies will manipulate the internal workings of their system to achieve an output without considering the consequences of their actions. Considering the consequence of an action or decision that goes into the design of a software or system signals an intent of compliance or non-compliance. This consideration provides evidence of due diligence (or lack thereof) to exercise care and prudence regarding compliance within the software development process. This type of process improvement is vitally essential to increase industry safety and standards for the better. Also, the intention of compliance holds significant weight in the culpability of

a person or organization when it comes to non-compliance to a regulatory requirement. One example of non-compliance across the industry is using defeat devices in European diesel engines and passenger vehicles within the U.S. from 2009 to 2015, known as Volkswagen's "Diesel Gate" scandal. This scandal involved Volkswagen intentionally using software in their diesel engines to cheat emissions tests, leading to significant environmental and health impacts [7, 28, 48].

Volkswagen used software embedded in over 11 million of their diesel cars' electronic control modules to defeat U.S. emission testing from 2009 to 2015 [7, 28, 48]. The 'switch' software used four analysis factors (i.e., positioning of the steering wheel, speed, duration of the engine's operation, and barometric pressure) to determine when the car was under test conditions for emission. As a result, the emission controls on these vehicles were activated during testing, lowering the cars' emission output to meet the U.S. regulatory emission standards [67]. The International Council of Clean Transportation (ICCT), a clean-air advocacy group, caught Volkswagen through a study in 2013 of their diesel emission system. The three graduate students conducting the test found discrepancies between the lab and road emission tests. Their testing gained the attention of the Environmental Protection Agency (EPA), which repeated the tests in 2014 to determine why Volkswagen diesel cars had different emission results [58, 77]. Finally, in September of 2015, Volkswagen admitted to their deception. Legal prosecution over the "Diesel Gate" scandal ensued over the next three years, costing Volkswagen $2.8 billion USD in criminal fines. The CEO, Winterkorn, was charged with fraud and conspiracy [114, 49]. The "Diesel Gate" scandal is an example of a company that gamed the system and exploited the output-only testing procedures required for its product to make it look like it

was compliant with applicable laws. Why? Volkswagen sought the financial opportunity to "crack the U.S. diesel market and, in the process, become the world's top-selling automaker" while seemingly meeting the EPA's 2004 new emission standards [66].

Volkswagen demonstrated the intent to subvert the U.S. EPA standards to meet the business objective of bringing Volkswagen's diesel vehicle line to the U.S. market. The Volkswagen leadership and engineers that created the defeat device knowing the U.S. EPA's emission standards and the validation procedures. Rather than consider cleaner alternative engines that would have taken time and money to implement, they decided to use a cheat device. Volkswagen's intent and actions to subvert the regulatory standard made them culpable. That signal of intent is critical for auditors when assessing a non-compliant case and deciding on enforcement actions such as violation fines regarding culpability.

Intent is also a critical factor in culpability regarding security requirements. Although not as fatal as Boeing's decision or as blatant as Volkswagen's, Zoom's security and privacy practices impacted millions of users in 2020 during the height of the COVID-19 pandemic [124, 57, 106, 126, 125]. Zoom became massively popular because of the quarantine. One reason was that Zoom advertised HIPAA compliance because of security and privacy features like end-to-end encryption; as it turns out, the end-to-end encryption did not conform to the industry-standard definition.

Each example subverts regulatory expectations, but each does so differently. Boeing misjudged the risk of MCAS failure. Boeing followed the letter of the FAA self-certification of airworthiness program to save time and money in designing, certifying, and training pilots on the 737-MAX. Volkswagen exploited the knowledge of the EPA's

emission testing procedures. Zoom's platform did not provide industry-standard end-to-end encryption, which meant healthcare providers unknowingly violated HIPAA security standards by using Zoom to conduct telemedicine visits during the height of the COVID-19 pandemic. Moreover, these companies got away with it initially because the methods used to verify these products only examined the product superficially, focusing on the tested products' outputs. Within the software industry, there is no standard process to look into a system internally and critique if something is good or not for an auditor or certifying official. There is also no way to communicate any gray areas within a regulation and actions for follow-on to demonstrate due diligence. Instead, any in-depth investigation is done from scratch and relies on the investigators' experience and technical knowledge. Making an investigative process a free-for-all where the minimum standard method is to run the system in a lab environment and only see if the outputs produced are correct. This is similar to grading an exam and only examining the students' final answer without checking how they got it. In most cases, having the correct answer usually means the student took the proper steps to get it. However, sometimes having the correct answer but no context to back it up can also mean that the student cheated to get the answer. In addition, some students did everything correctly except for one misstep, which led to the wrong answer. Therefore, teachers taking a closer look can allow students to demonstrate their knowledge of the tested material, which can get them partial credit.

My dissertation advocates the method of ambiguity modeling to facilitate "a closer look" and communicate "gray areas" when interpreting regulations for compliance. This method serves both organizations wanting to demonstrate and communicate their due diligence toward compliance with an external party and for an external party reviewing the

process and providing accurate compliance assessment or further guidance to a software organization or development team wanting to comply but needing some clarification. For the internal engineering team, it might give them a better understanding of the implications of the software they are designing (i.e., Boeing MCAS design team) and auditors, who have to come in and critique a system or software for certification (like Zoom for HIPAA and telemedicine visits) or recertify an updated system (i.e., Boeing). This process can also highlight an organization's compliance with the law when something happens (i.e., Boeing crashes, Zoom bombing, or the inconsistent data with Volkswagen's lab and road testing), giving auditors an idea of their intention toward regulatory compliance. Even if the results are the same, it can help determine culpability. Did the organization entirely ignore its regulatory requirements, or did they look at the regulations and take one direction that might have seemed reasonable at the time? The idea is that there are multiple reasonable interpretations (i.e., Zoom's interpretation of end-to-end encryption versus the industry's understanding) to comply with the law or regulation, and highlighting these "gray areas" can get people to ask meaningful questions. Moreover, this technique, i.e., ambiguity modeling, is more than pointing out the gray areas within a regulation or law. The ambiguity modeling process captures the heuristics that go into the software development process and links them to actions, updates, or procedures an organization follows as part of the process. This makes the engineering and design team more aware of their regulatory obligations and the regulatory compliance requirements more visible within the software development process. Lastly, ambiguity modeling is evidence that someone stopped and considered what a gray area might mean and took action based on that consideration. In the case of Zoom, if Zoom software engineers had carefully read the HIPAA

regulations, for example, and modeled it, they might have realized that Zoom's end-to-end encryption was not the expected industry standard encryption needed by HIPAA. Then, Zoom would have not been endorsed by FTC or FDA for tele-medicine or Zoom would have upgraded their security protocols to comply with industry security standards much sooner. Therefore, Zoom would have avoided some headaches and their $85 Million class action suit as a result. Their platform would still have increased their user base 300 times, but not in specific regulated industries that had compliance requirements to maintain. For Boeing, had the MCAS design engineers modeled the FAA's regulations for the 737-MAX aircraft, they might have realized the greater risk the MCAS posed to the safe operations of 737-MAX. They could have then created test cases and tested the operation of MCAS under various conditions. A higher level of risk supported by the evidence of the operational test linked to the model, could have either signaled to Boeing's internal audit or the FAA to make necessary design modifications to the MCAS on the Boeing 737-MAX before those crashes had occurred. For "Diesel Gate," the problem was not the regulation but how the EPA verified diesel engine vehicles for passenger cars. When the EPA updated the emission requirements within the "Clean Air Act" in 2007, the emission testing engineers could have modeled their enforcement of emissions with the update of the regulation. The EPA would have seen that there were alternative, available testing procedures to verify emissions, which would have led to an update of emissions testing to include real-world road testing for diesel passenger vehicles using available technology. The EPA would have deterred European automakers from using defeat devices or had caught them much sooner than six years after Volkswagen rolled out their first "cleaner emission" diesel engine in 2009.

1.2 Research Motivation, Purpose, and Questions

1.2.1 Motivations

The traditional viewpoint of software engineers is that ambiguity represents a bug requiring resolution. Even business consultants view ambiguity as problematic:

> "The enemy of accountability is ambiguity."
>
> — Patrick Lencioni, best-known author of the Five Dysfunctions of a Team

However, ambiguity can support accountability efforts in the proper context. It may capture a regulation's spirit or intention more accurately without bogging it down with technical details, such as using an ambiguous phrase like "reasonable security standards" instead of specifying a technical requirement. Using such phrasing in regulations, like in CCPA or GDPR, allows flexibility within laws so that as technology changes and grows, the interpretation and compliance with those laws can grow. Whether or not regulatory ambiguities are helpful or hurtful, they exist and can impact compliance and enforcement of applicable regulations and laws. Given the importance of regulatory compliance to ensure socially acceptable engineering within software development, there needs to be some way for an organization to demonstrate due diligence toward compliance with applicable regulations. Unfortunately, the ambiguous language within a legal text can undermine its purpose and value despite its legitimate uses within legal text and policy [113].

My dissertation aims to study and develop a process for showing how software developers interpret regulatory ambiguities and for communicating and documenting this intermediate step within requirements analysis. The developed approach has advantages

for various stakeholders. The process is meant to clarify ambiguities in regulations for software developers and provides boundaries for requirements development. For policymakers, the process and associated outputs give access to the thinking and reasoning followed by a software development team interpreting a regulation for compliance. For auditors and regulators, a process that interprets legal ambiguities and the developed models allows organizations to demonstrate that the internal development team(s) performed a requisite amount of due diligence toward regulatory compliance. In addition to demonstrating a process involving multiple perspectives, I am striving to include the invaluable industry perspective on compliance and the usefulness of this technique. The insights and experiences from the software industry are crucial for the effectiveness of any compliance methodology or tool developed within academia. The next section summarizes my dissertation, which is broken down into three main parts.

1.2.2 Overview

For part one of my dissertation, I want to understand software organizations' practices and decisions toward regulatory compliance requirements during the SDLC using qualitative research methods. Next, for part two, I take this improved understanding and have software developers as a team review regulations and document ambiguities found within the regulatory text. This method of identifying, classifying, and documenting regulatory ambiguities captures the thinking of a software development team while reviewing regulations for regulatory compliance requirements. This model can be used for internal communication within the team and external communication to those outside the

group, such as leadership or organizational policymakers.

Part three supports the notion that ambiguity modeling helps capture developers' reasoning, and the reasoning can produce meaningful guidance or assist test case development for technical compliance validation. To find this support and validate that ambiguity modeling is useful in demonstrating efforts to comply with regulation, I presented specific data points and key findings from the Multi-Case Study, along with a comprehensive legal analysis of the Virginia Consumer Data Protection Act, to a group of industry professionals with auditing experience. I solicited their feedback on the 'Ambiguity Modeling Process' and whether it helps demonstrate a requisite amount of due diligence or acceptable behavior regarding compliance from an auditor's perspective. From their feedback, I validated that the Ambiguity Modeling Process is useful to a software organization required to demonstrate its regulatory compliance process and have recommendations on how to improve ambiguity modeling and the AHAB tool for future work. The reasoning can produce meaningful guidance or assist test case development for technical compliance validation.

All three parts of my dissertation, encompassing four studies, are designed to promote the ambiguity modeling method to communicate and document regulatory compliance analysis. This method can be used both internally within a software development team and externally with other stakeholders assisting the requirements analysis process. By presenting these innovative techniques, I aim to foster a collaborative environment that encourages other researchers to explore and promote similar methods. Moreover, this work is a stepping stone in creating a comprehensive compliance framework, which requires the collective efforts of academic and industry professionals in software devel-

opment, regulatory compliance, and research.

1.2.3 Research Vision, Goals, and Questions

The high-level vision for my research is to promote the "Ambiguity Modeling Process".This method is designed to document and communicate regulatory compliance analysis within a software development team and to external stakeholders, thereby enhancing the requirements analysis process [94, 95]. Ambiguity modeling can also serve as a valuable tool for auditors, aiding in their review and decision-making processes related to compliance for software-developing organizations.

My high-level vision is further broken down into three specific research goals:

1. Capture the industry perspective and practices regarding regulatory and security standard compliance.

2. Using a modeling technique to identify, classify, and interpret regulatory ambiguities within the legal text with a group of software-developing practitioners as proof of concept.

3. Validate the proof of concept (i.e., ambiguity modeling) as a useful tool for requirements analysis and documentation as evidence of due diligence of regulatory compliance.

These goals translate into my five research questions:

RQ0: What is the state of the art in academic literature on Regulatory and Security Standard Compliance? (Chapter 2)

RQ1: What are the software industries perceptions regarding Regulatory and Security Standard Compliance? (Chapters 3 and 4)

RQ2: How are regulatory ambiguities within legal text interpreted and reasoned by software stakeholders/practitioners individually and as a group? (Chapter 5)

RQ3: Is the ambiguity modeling of a regulation useful for a software organization? (Chapter 6)

RQ4: Does modeling regulatory ambiguities document due diligence toward regulatory compliance from an auditor's perspective? (Chapter 6)

1.3 Approach

1.3.1 Overview of the Study Design

I designed this dissertation into three parts, using four qualitative studies to answer the research questions in the previous sections. Part one is the Interview Study and Survey. These two studies are to gain an initial sense of the software industries' perceptions toward regulatory and security standard compliance (i.e., Chapters 3 and 4). Part two is the Multi-Case Study, which takes a closer look at decision-making and the reasoning behind interpreting regulatory compliance through ambiguity modeling (i.e., Chapter 5). Finally, part three validates the usefulness of the Ambiguity Modeling Process from an auditor's perspective (i.e., Chapter 6) through a focus group study. The following few subsections briefly give an overview of each part of the dissertation.

1.3.2 Part 1: Interview Study and Survey

Part one of my research is to understand the software industries' perceptions of regulatory and security compliance. Before starting on the various studies, I conducted a literature review to understand the state of academic research [85]. The findings motivated me to conduct an interview study and survey. The interview study spoke to 15 software practitioners operating within the software industry [84]. The survey further analyzes the software practitioner's perspective on a larger subset of the software industry to generalize the interview study's results and identify additional findings not uncovered in the interview study.

1.3.3 Part 2: Modeling Regulatory Ambiguities

Part two of my research employs and extends a method for reasoning and decision-making behind documenting regulatory ambiguities within legal text [95, 94, 96] within a group setting. I observed the execution of the methodology via a multi-case study using a browser-based online tool explicitly developed for modeling ambiguities within a legal text and updated for this case study. The multi-case study focused on the ambiguity modeling method and the group discussions on building such a model. I introduce the ambiguity modeling method and online tool to the case study participants in the first of three online sessions. At the end of the first session, each case participant was assigned a legal text and asked to identify, classify, and model the regulatory ambiguities within that text to present and discuss during Sessions two and three within their case group. I observed and recorded the online sessions to capture the reasoning and decision-making

behind documenting regulatory ambiguities within the legal text for each participant and the case group. I analyzed the data from each session over three case groups to formulate my findings regarding the difficulties of ambiguity modeling and the value perceived by the case groups.

1.3.4 Part 3: Validating the Ambiguity Modeling Process

Part three validates the Ambiguity Modeling Process as a useful method for demonstrating regulatory compliance using feedback from two focus groups in software industry professionals with auditing experience. I used the data from the multi-case study and additional data collected from our legal researcher as proof that the Ambiguity Modeling Process is a workable concept for showing requirement analysis of ambiguity and communicating to outside parties like a lawyer for further clarification and guidance. I presented the proof-of-concept data collected to a focus group of industry professionals with background and experience developing and auditing software in a compliance-driven industry (i.e., healthcare or finance). The focus group then gives feedback on the process from a usefulness standpoint and whether applying a technique could demonstrate acceptable behavior or due diligence toward regulatory compliance from an auditor's perspective.

1.4 Contributions

My research makes three contributions. The first contribution is that my research provides better insight into the software practitioners' perspective regarding regulatory

and security standard compliance. As a researcher, I believe building a general technique for industries to demonstrate regulatory and security standard compliance throughout the entire software development lifecycle should elicit actual practitioners' inputs. Practitioner's input and feedback on how they would apply the method highlights usability and application within the software development process. In addition, little academic research is available to represent the software industry's perspective on regulatory and security standard compliance [85]. Therefore, one of my contributions through my dissertation work is helping bridge this gap between "on the books" and "practical" practices.

My second contribution is to propose a method for modeling regulatory ambiguities and documenting decision-making regarding compliance during software development. The first step in demonstrating compliance is to be able to interpret requirements from regulations and law. The presence of regulatory ambiguities makes developing regulatory requirements challenging. Ambiguities can help or hurt requirements development depending on their perspective and context. Therefore, a methodology to identify, classify, and interpret regulatory ambiguities can address the regulatory ambiguity challenge. The analysis from the models can communicate potential "gray areas," so software development teams can receive further guidance. This guidance can translate into documented artifacts or testing use cases to help build a framework or general technique for demonstrable regulatory compliance.

The third contribution investigates whether the ambiguity modeling process is useful for demonstrating due diligence. It involves working with and getting feedback from software industry professionals who have been through or conducted a regulatory audit to gain the auditor's perspective. The goal is to determine whether the Ambiguity

Modeling Process offers insight into the heuristics associated with interpreting regulatory ambiguities that an auditor might want to see when assessing a software organization for regulatory compliance.

Through my dissertation work, I support the statement that the regulatory enforcement process should include techniques like ambiguity modeling to show evidence of due diligence for the software developer. When encountering violations that point to non-compliance, a software development team can demonstrate that they did the requisite work to address regulatory compliance concerns by having a method that documents the ambiguity modeling and regulatory requirements development process. This type of due diligence evidence could reduce fines and further legal action because it can show that the violation was not malicious or negligent but misunderstood.

1.5 Dissertation Structure

The following section outlines the contents of the remaining dissertation's chapters:

Chapter 2 Related Literature - Chapter 2 discusses previous peer-reviewed research in software engineering regarding regulatory and security standard compliance. I also include literature on compliance decision models focusing on software practitioners' compliance heuristics, the associated risks, and the cost and benefits of compliance and other ambiguity modeling techniques.

Chapter 3 Interview Study: Perceptions from the Software Industry - Chapter 3 presents the interview study's methodology, findings, limitations, and lessons learned (i.e., discussion) for academic researchers, practitioners, and requirements engineers to

take into consideration.

Chapter 4 Survey: Perceptions from the Software Industry - Chapter 4 presents the methodology, findings, limitations and discussion of results of a survey that was conducted to allow generalization of the findings from the interview study.

Chapter 5 Multi-Case Study: Modeling Regulatory Ambiguities using GDPR - Chapter 5 is a multi-case study that looks at the reasoning, decision-making, communication, and methods behind interpreting regulatory requirements within a group of software practitioners, specifically derived requirements from regulatory ambiguities found within a chosen legal text. This multi-case study comprises three groups of software development teams, one for each case, interpreting the same regulation. The results for each case are reviewed and generalized using ground theory and comparative analysis.

Chapter 6 Validating the Ambiguity Modeling Techniques - Chapter 6 presents the feedback and results of the case study (i.e., Chapters 5) to Focus Groups of consultants in the field of Software Development who specialize or work within Regulated Domains (i.e., healthcare).

Chapter 7 Study Synthesis - Chapter 7 combines the practices of the industry regarding compliance (i.e., Part 1), the process of the regulatory ambiguity modeling and the creation of guiding documentation (i.e., Part 2), and the validation of the ambiguity modeling process (i.e., Part 3) to form conclusions of the overall dissertation work, the contributions, and the implications software developers, regulatory auditors, other software stakeholders, and the software engineering community.

Chapter 8 Summary and Future Work - Chapter 8 summarizes the dissertation. I also provide context for future work for this research and to further validate the ambi-

guity modeling method into a verifiable framework for compliance throughout the entire software development process. Lastly, I summarize the implications of my research for academic researchers, software practitioners, and regulatory auditors.

Chapter 2

Related Works

2.1 The Challenge with Regulatory and Security Standard Compliance in Software Engineering - Introduction

Society cares about software engineering because of software and technology integration into modern society. Software is all-pervasive. It is in the cars we drive, refrigerators we use to store food, and the planes we fly. use software to process information, manage finances, and socialize, among other things. Furthermore, with data breaches and privacy concerns, society is taking more notice and asking governing agencies to step in and protect our privacy and security.

To regulate the software industry and address societal problems with information technology, the global government passes laws and standards for the software industry to follow. A challenge with regulations and security standards is that they often contain ambiguous statements. These regulatory ambiguities can have multiple valid interpretations, be vague or incomplete (i.e., no definition), or conflict with other legal texts. Regulatory ambiguities make developing and testing for regulatory compliance requirements challenging. I want to help address this challenge. My research asks software practitioners to review regulations for ambiguities. This review identifies, classifies, and reasons why they view this as ambiguous. Through their analysis, I discern some logic in interpre-

tation, decision-making, and communication of regulation from their perspective. Logic that can be documented and linked to artifacts within software development, highlighting why certain decisions were made. This process and links within the software development demonstrate a software design team and organizations' due diligence to regulation and security standards throughout a software development process.

Addressing regulatory ambiguities does not mean eliminating them. While regulatory ambiguities present some obstacles, they can be helpful because they do not tie a particular regulatory requirement to a single technical standard, allowing technical growth and flexibility within the legal interpretation of the law [95]. Also, software organizations can find a standard that meets their unique technical needs, resources, and requirements. For example, the use of "reasonable security procedures and practices appropriate to the nature of the information to protect the personal information" within the California Consumer Privacy Act of 2018 (CCPA) is a regulatory requirement with no specific legal definition[12, 118]. Hence, this statement within CCPA is a regulatory ambiguity. However, it is an implementable regulatory ambiguity because there are security standards publicly available that legal experts would say meet the minimum "reasonable security procedures and practices" to satisfy CCPA. Examples include the NIST Cybersecurity Framework, the 20 controls in the Center for Internet Security's Critical Security Controls (CIS-20), HIPAA's Security Standards, PCI's Data Security Standards, and ISO 27001/27002[118]. The challenge is determining which standard is needed "based on the nature of the information" (i.e., PCI for financial data or HIPAA for health data) and best suits the software organization. Therefore, while my research does explore regulatory ambiguities, the intent is not to eliminate them within regulations or the law but to

examine the interpretation of regulatory ambiguities as part of the requirement analysis process. Then, see what guidance or uses can result from this useful analysis from a software organization and regulatory auditor's perspective.

This section describes prior work in the software industry's perspective on regulatory and security standard compliance (Section 2.2), identifying and classifying ambiguities within regulation (Section 2.3), prior work in modeling regulatory requirements to demonstrate compliance (Section 2.4), and software engineering for critical safety systems and secure development practices (Section 2.5). This framing of the related literature section closely follows my research dissertation's structure. I use the framing to state the gaps within the prior work and relate how my research addresses these gaps and contributes to the body of research.

2.2 The Software Industry perspective on Regulatory and Security Standard Compliance

Compliance with regulations and industry security standards is not a new concern or field of research. Explicit rules for domains like healthcare, children's privacy, and finance have existed for over 20 years. Furthermore, requirement engineering has sought to address regulatory compliance as a first-class concern within software development for more than 15 years [111, 70, 63, 115]. One example is the International Workshop in Requirements Engineering Law (RELAW) conducted from 2008 to 2018 [1]. The workshop brought together practitioners and researchers from industry and academia to discuss chal-

[1] http://gaius.isri.cmu.edu/relaw/

lenges and industry techniques regarding regulatory compliance in software systems [27]. Despite these efforts, there is little research that represents the software industry's perceptions of regulatory and security standard compliance [85]. To understand the state of the art in research is to know the application in practice. Therefore, one goal of my research is to continue to bridge gaps between academia and industry on regulatory and security standard compliance through an interview study and a survey.

Bamberger and Mulligan's study in 2010 [30] is another example with a similar goal of my interview study to try and bridge the gap between privacy law versus "privacy on the books." They interviewed nine Chief Privacy Officers (CPO) from Fortune 1000 companies on privacy practices from five different countries. Bamberger and Mulligan aimed to identify how adopting privacy practices at an organizational level occurred from the CPO's perspective [30]. The reception of the interview study spurred them to conduct further interview studies focusing on CPO's counterparts (i.e., engineers, lawyers, advocates, and regulators) [31]. These follow-up interviews culminated in a book that discusses the critical role of the privacy professional both internally within an organization and externally by helping to shape regulatory privacy and management standards. Both these studies advocate the role of the privacy professional and how they structure and influence an organization's privacy management. In contrast, my interview study and survey do not advocate a particular software development stakeholder's role. Instead, part one of my research examines how software perceives and implements compliance efforts as part of their software development process, including privacy regulations compliance and healthcare and financial regulatory compliance. Knowing compliance with regulation and security standards is a team effort; I intentionally sought different perspectives

from software development stakeholders rather than a specific stakeholder group, such as a compliance officer, privacy engineer, or security manager. The goal was to compare the different perspectives and understand the different viewpoints contributing to the development of software systems and the compliance management process.

Another set of interview studies similar to my research work is Haney and Lutter's [68, 69]. Their two interview studies focused on how cybersecurity advocates promoted security practices within their organizations. Haney and Lutter specifically looked for cybersecurity advocacy skills [68] and how their subjects overcame negative perceptions of cybersecurity at an organizational level [69]. Like Bamberger and Mulligan's organizational focus on privacy practices, Haney and Lutter focused on adopting cybersecurity practices and how organizations influence their people to implement particular cybersecurity practices. My work focuses on how software practitioners interpret, communicate, integrate, and demonstrate regulatory and security standard compliance in their software development process. While Haney and Lutter's work focuses on a single stakeholder group and the skills needed to be a successful cybersecurity advocate, I focused on different stakeholders within the software development process. Also, I was not necessarily concerned with specific skills or what makes a successful compliance program. Instead, I am focusing on the practices and challenges of regulatory and security compliance management processes and whether I can address a specific challenge or concern (i.e., regulatory ambiguities). The idea is that an organization can document decision-making and logic to external parties, like auditors, consumers, and regulators, and demonstrate a reasonable amount of due diligence towards regulatory and security standard compliance throughout the entire software development lifecycle.

Usman et al.'s case study from 2020 is another qualitative work on regulatory compliance in software development. They investigated the "common practices and challenges with checking and analyzing regulatory compliance" with a product development team at the telecommunications company Ericsson AB in 2020 [122]. While not an interview study or a survey, they did use interviews as one of their data collection methods during the workshop proceeding and individually when identifying standard practices and challenges related to regulatory compliance within Ericsson AB telecommunications. They also recruited different system development stakeholders within their study (i.e., a product manager, system manager, product owner, deployment lead, information owner, test manager, and two system architects). This empirical study was a case study, so they were limited to a single IT organization. They also used different qualitative research methods as part of their analysis, including group discussions and document analysis. Overall, these prior works [20, 21, 22, 23, 24, 30, 31, 68, 69, 120, 122] and academic resources[44, 101] have helped guide my work in part one of this dissertation. They have offered guidance in qualitative and quantitative research methods used to analyze my interview study and survey data. In addition, in looking at the previous studies [20, 21, 22, 23, 24, 30, 31, 68, 69, 120, 122], I have focused on gaining software industry insight and perspective as part of my research. By looking into the practices and challenges of regulatory and security compliance management processes, I can address a specific challenge or concern of regulatory ambiguities to demonstrate a software organization's intent to comply (i.e., due diligence toward regulatory compliance) that is useful within an auditing process, which is the focus of part two and three of my dissertation.

2.3 Ambiguities within Regulation

One of the uncertainties found within legal texts is regulatory ambiguities. IEEE defines regulatory ambiguities as a regulatory requirement or specification that does not have a clear, single interpretation [74, 42]. For example, the phrase "commonly used" about data formatting within the California Consumer Privacy Act (CCPA) can refer to multiple types of data formatting for Information Technology (IT) processing. Examples include File-based Data Format, Directory-based Data format, or Data connections [36]. Software organizations intending to comply with regulation can also misinterpret a regulatory requirement, leading to fines, delays in product deployment, and redesign. Because of these issues, regulatory ambiguities are frustrating for software practitioners and enforcement alike, who want a clear interpretation of their legal obligations within software-supported domains. Therefore, the first step to overcoming these regulatory compliance issues within software and requirements engineering is identifying, acknowledging, clarifying, and documenting regulatory ambiguities alongside software requirements and design artifacts.

Research in the classification and interpretation of regulatory ambiguities is familiar and usually in conjunction with research that interprets the legal text to extract legal requirements. Otto and Anton's literature survey is one of the best-cited works investigating past research efforts for interpreting legal text for system development [111]. They analyzed legal text modeling and interpretation research efforts from 1957 to 2007. They identified seven approaches to logically handling legal text for system development. Their survey aims to aid requirements engineers and compliance auditors by assessing these ap-

proaches in systematically handling legal text to specify, monitor, and validate regulatory requirements for software systems. While Otto and Anton cover the challenges with regulatory ambiguities, they also look at other issues with reading and interpreting regulations and legal text that may concern requirements engineers. These challenges include cross-referencing [2] and domain-specific definitions and acronyms. My research focuses on regulatory ambiguities by identifying ambiguities in legal text. Understanding why a software practitioner(s) identifies and reasons a legal word or phrase as ambiguous; whether a team of software professionals can model and come to a shared understanding of regulatory ambiguity interpretation for software requirements analysis and a design artifact in software. Then, they communicate their interpretation of the ambiguities in the legal text to an external stakeholder (i.e., a lawyer) to garner additional guidance and clarification that can translate into actionable software requirements for regulatory compliance. I focus on regulatory ambiguities because they are a leading cause of misinterpreting regulations and laws, leading to poor requirement analysis and software design [94, 95]. More so than cross-referencing and domain-specific definitions and acronyms. Other researchers share this perception and have similar works on identifying regulatory ambiguities for better software requirements analysis [32, 33, 38, 42, 121, 95, 36, 41]. Some examples of detecting, classifying, and removing ambiguities in requirement specifications are from Kamistics Berry's individual and collaborative works in the early 2000s [32, 33, 82, 80, 47, 83, 81, 34]. Much of their work during this time involves developing techniques and tools for identifying, avoiding, and removing ambiguities within requirement specifications for Information Technology (IT) systems. Methods and tools

[2]The frequent referencing within a legal text to different sections and laws [111].

that evolved to automated approaches. Even though tools and techniques, manual or automatic, are helpful, they do not entirely identify or eliminate all ambiguities. Also, tools and methods need to display understanding or reasoning of ambiguities. Therefore, for these tools and techniques to be effective and accurate, they require practical domain knowledge and domain-specific language to understand the exact meaning of an interpretation of a phrase or requirement. Without this expertise, ambiguities are either undetected or unresolved [94].

Other research in ambiguity identification for requirement specification and analysis narrows on specific types of ambiguity. For example, Bhatia et al. investigated vague ambiguities within privacy policies. How vague ambiguities mask an IT or software organization's actual privacy practices regarding consumers' data, putting the data and the consumer at risk [36]. Bhatia et al. later focused on incomplete ambiguities in privacy policies, making similar claims to the 2016 vagueness paper and interpretation misunderstanding that also put the IT or software organization and the consumer at risk [35, 37]. A difference between Bhatia et al.'s work [36, 35, 37] and the studies in my research is that Bhatia et al.'s work targeted a specific type of ambiguity classification in each of those papers where the case studies in this research are interested in modeling all kinds of ambiguities within the legal text. Furthermore, Bhatia et al. focused on privacy policies, whereas I focus on regulation and security standards. Lastly, the case studies observe understanding, reasoning, communication, and decision-making regarding regulatory ambiguities for novice requirement analysis stakeholders versus Bhatia et al.'s risk perceptions and impacts associated with ambiguities [36, 35, 37].

Ferrari et al.'s work on cross-domain or domain-specific ambiguity is another exam-

ple of ambiguity identification for requirement specification and analysis that narrows on specific types of ambiguity [52, 53, 54]. Domain-specific (aka cross-domain) ambiguity, as Ferrari defines it, is when a word has "different vocabulary meanings in different domains" or something different between two people with different domain-specific backgrounds [52]. An example of a domain-specific ambiguity that would require sentence context for clarification that Ferrari uses is the word "interface" [52]. For software, the interface could refer to a software application the user uses to operate a machine or system, such as a Graphical User Interface (GUI). For hardware, the interface would refer to a physical piece of hardware, like a computer mouse or keyboard, used to interact with the machine or system. During the requirements elicitation phase, this difference in interpretation can cause misunderstanding for requirement analysis and software development. Ferrari's referenced works [52, 53, 54] for cross-domain or domain-specific ambiguity, was to present a natural language processing (NLP) approach to detect cross-domain ambiguity. The differences between Ferrari's work and my research are that most of Ferrari's analysis approaches to detecting domain-specific ambiguity are quantitative using automated means. My research approach for data collection and analysis is primarily qualitative (i.e., interviews and case studies), with the survey being quantitative and qualitative. Ferrari and Eusli incorporate a qualitative data collection process through a Likert scale survey. The authors and two human participants manually assess the domain-specific ambiguities across seven scenarios. The analysis is still quantitative [53]. Second, part two of my research focuses on modeling different ambiguities within a given legal text. Individual participants may interpret certain ambiguities differently based on background, job roles, and domain experience, but that ambiguity classification is lexical and is what

I want to observe as part of the individual presentation and later as part of the discussion during the consensus sessions (i.e., Session 3) of the case studies. Lastly, my research focuses on regulation and security requirements and compliance with those requirements. I think Ferrari's other works in analysis involve formal methods for safety-critical systems [55, 56] is comparable to my research in the regulatory and security requirement sense, just not the ambiguity piece for now.

Other works focusing on domain-specific ambiguity created an automated means to detect and handle domain-specific lexicons or corpus [3] to improve the accuracy of ambiguity detection and interpretation. Ezzini et al.'s automated approach analyzed syntactic ambiguity [4] and prepositional phrasing semantic ambiguity [51]. Their approach considers that different stakeholders' perspectives (i.e., different domain knowledge) and interpretations of the same requirement document can lead to unacknowledged ambiguity because there is no single understanding of a requirement. Yet, everyone assumes a shared understanding requirement. Ezzini et al. evaluated six automated methods for identifying another form of semantic ambiguity known as anaphoric ambiguity statement [50]. Both studies used machine learning metrics of accuracy, recall, and precision to measure the average detection outputs. The benefit of an automated approach would be that ambiguity analysis is less time-consuming while bringing forth ambiguities driven by multiple interpretations of domain-specific languages. Their work analyzes ambiguity from a structural perspective. However, it does not communicate a why or deep understanding of the intent

[3] Corpus is a collection of written or spoken material to define terms used with a domain or area. An example is the International Corpus of English (ICE).

[4] "A sequence of words with multiple valid grammatical parsing" [95].

behind the requirement. Critical skills needed for successful requirements specification and ambiguity resolutions within software development. My case study (i.e., part two of my research) focuses on modeling regulatory ambiguities in a group setting. To observe individual and group reasoning, decision-making, and negotiation when interpreting and modeling ambiguities in a given legal text.

Extensive research has identified, classified, and documented regulatory ambiguities within the software for specific regulated domains like healthcare and finance [95, 111]. However, no one has yet created a methodology for modeling regulatory ambiguities within the software and testing such a methodology for general use (i.e., any legal text relevant to the software industry). The closest research I come across is Breaux and Norton's perspective paper [41]. Breaux and Norton's perspective paper promotes legal requirements as a software design activity versus an oversight activity through cross-functional team analysis [41]. They focused their research on data processing, stating:

> *"The diversity and speed of innovation in data processing limits what regulators can accomplish through rulemaking, and thus explains why data processing law includes purposeful ambiguity. This difference requires software designers to bear more of the burden of specifying their own processes in the context of their software. Thus, we believe data processing law presents a starting point where new methods and tools can arise with a better fit to less prescriptive design contexts, while borrowing best practices from domains with more extensive regulation"* [41]

The software industry as a whole is unregulated. The exception is software that operates

in regulated domains, where software organizations that support domains like finance, healthcare, or safety are known to comply with the mature laws established within these fields. However, with other domains like data processing, researchers within software engineering are looking for ways to improve regulatory analysis. Hence, it translates into actionable software requirements [41].

My research with regulatory ambiguities is an extension of Massey et al.'s studies from 2014 and 2015 [95, 96]. These studies investigate the taxonomy to identify and classify regulatory ambiguities within a given legal text. However, my research takes this further by creating a methodology for modeling regulatory ambiguities using the identification and classification taxonomy for software requirement analysis and system design. Other key differences are that the 2014 and 2015 studies focused on HIPAA within their case study; I examined GDPR and Virginia's Consumer Data Protection Act to develop and test the methodology from the case study groups for modeling regulatory ambiguities for general use. Second, the 2014 and 2015 studies examined individual analysis for identifying and classifying regulatory ambiguities. I am looking at individual and group work in analyzing regulatory ambiguities. Software organizations that comply with multiple regulated domains per their business model (i.e., Amazon, Google, Microsoft) could use this holistic methodology for interpreting, documenting, and modeling regulatory ambiguities as part of their software design and development process. They can then adapt the modeling methodology for their specific product for a particular domain rather than use different frameworks or methodologies within their regulatory and security compliance management process. There is a need for this type of research. I intend to contribute through my work by introducing and validating a methodology for regulatory ambiguity

modeling as a workable solution for the software industry.

2.4 Modeling Legal requirements for Compliance

Modeling regulations is not a new field of study. Over the past 20+ years, previous research has presented repeatable, systematic methodologies for translating legal text into requirements through modeling. A common approach to requirements modeling is User Requirement Notation (URN). URN uses models to conceptualize non-functional requirements through goal-oriented requirement language (GRL) or functional requirements using Use Case Maps (UCM) or Unified Modeling Language (UML) [76]. Researchers have extended the URN approach to meet specific requirement elicitation needs. The extension includes Legal-URN to model legal requirements [62, 65, 64] and Textual-URN so URN can use textual and graphical language [89]. Ghanavati et al. developed a method for extracting requirements from regulations using URN's GRL for Legal-URN. This model approach captures stakeholders' goals and specifies requirements to satisfy those goals [62, 65, 64]. Other URN-based techniques use "Use Cases" to logically model the usage of a system without limiting goals. For example, Hassan and Logrippo's modeling technique uses UML to combine formal logic with business processes to extract legal requirements for formal requirements specifications [71].

Like URN, logical modeling is another common approach to translating legal text into requirements through modeling. Researchers have used logic modeling to describe rules that dictate stakeholders' actions specified within policies and regulations. For ex-

ample, Breaux and Anton identify these actions as rights [5] and obligations [6] into requirements for regulatory compliance [39]. Similarly, Maxwell and Anton used production rules to model obligations and rights described by regulation through structured queries and communication between legal experts and requirement engineers. The outcome is software requirements that trace back to a production rule and whether a software requirement meets the regulatory compliance requirement [98].

Amaral et al. use a model-based approach for Compliance checking of Data Processing Agreements against GDPR [26]. They bridge the gap between legal analysis and software engineering by breaking the subcontracted data processing agreements (DPAs) between service providers and parent companies that collect and analyze the data against the listed criteria for such contracts within the European Union's General Data Protection Regulation (GDPR). They used legal experts from Linklaters LLP, a global law firm based in London, England, to help create and test an automated compliance checking tool composed of 14 criteria specifically for DPAs [26]. This type of tool that incorporates legal research applicable to software creation and requirements validation is part of the direction I wish to go with my research. If there is a limitation, it might oversimplify the compliance checking process into a list of criteria that might change. Therefore, the tool might validate DPAs today, but with the changing compliance landscape, such a tool would require updates.

Furthermore, edge case validation is also a consideration. In a specific context, edge cases or deviations within a software development process would be considered

[5] actions permitted by laws

[6] actions required by laws

compliant within these agreements but not within the norm, in which ambiguities within regulation and standard guidance are intentionally placed for this reason. These edge cases and adapting to compliance requirement change require critical thinking and a strategy to capture the critical thinking within the development process. Modeling regulatory ambiguities is a strategy to capture the stakeholders' thinking by getting software stakeholders to review regulation and communicate what is potentially confusing from their perspective modeling regulatory ambiguities. Then, once the regulatory ambiguity model is complete, create supplementary guidance applicable toward complying with the regulation as part of the software requirements analysis and design artifact documentation [94]. The strategy considers regulatory compliance challenges of regulatory ambiguities. It also acknowledges that some ambiguities are intentional to support multiple valid interpretations and that technology evolves. Therefore, the process promotes general use and changes over time.

2.5 Software engineering for safety critical systems and security

My research goal is to examine the challenge of regulatory ambiguity regarding requirement specifications for regulations and security standards. Within this chapter, I explore the legal background and challenges with regulatory compliance, specifically regulatory ambiguities (Section 2.1). I describe prior work in the software industry's perspective on regulatory and security standard compliance (Section 2.2) and related to my work to date. In Section 2.3 and Section 2.4, I describe prior work for identifying and classifying ambiguities within regulation and modeling regulatory requirements to

demonstrate compliance applicable to requirements engineering.

In this section, I explore formal methods of requirements and design analysis for safety-critical systems and secure development practices. For safety-critical systems, I wanted to examine the research regarding regulatory ambiguities with safety-critical systems and see if there is anything I can apply to my research plan or validation. For secure development practices, my research concerns regulatory and security standard compliance as part of the software development practice. Therefore, I would be remiss if I did not explore the security practices and standards that are part of the software development field and the relationship with regulatory compliance.

2.5.1 Safety critical systems

Formal methods are mathematical languages used for hardware and software requirements specification [92, 117]. Some researchers have argued that formal methods are preferred in the requirements and design phase when developing safety-critical systems. The formal methods can reduce ambiguity in design by defining accurate and precise requirements [92, 117]. Lockhart et al. argue this point in their paper by introducing a methodology to add formal methods to translate natural language requirements to the functional specification. This method addresses reliability in software design to reduce abstract requirement specifications for the system or software developer [92]. Singh et al. have done similar work using Unified Modeling Language, specifically Z notation language, with formal methods to verify requirement specification [117]. These works help translate requirements in natural languages to functional requirement specifications

when designing complete end-to-end hardware and software systems where the applicable requirements will likely stay the same. However, these are not holistic solutions; sometimes, the context of the requirements for critical safety systems is a consideration.

Ferrari and Fantechi looked at nine formal methods and tools for railway development in 2020. They found that highly recommended formal methods in railway design, railway companies need more guidance on the appropriate ways or tools for unique needs. Furthermore, different tools are considered better than others given a particular context, such as the background of the typical user, the support of concurrent systems, and the compatibility with other systems that are part of the railway network [56]. Because of these factors, regulatory ambiguities exist and are a valuable tool from a legal sense. Regulatory ambiguities do not hinder the ability to choose the best-suited tool, method, or technical requirement to comply with the intent of a regulation or law.

Ambiguities also account for change or new, possibly better interpretations involving compliance with the law. Going back to the California Consumer Privacy Act of 2018 (CCPA) example in Section 2.1, there are multiple ways to satisfy the"reasonable security procedures and practices" specified in CCPA. That allows software organizations to explore their options, think about what best fits their and their consumers' needs, and take ownership of their actions to comply. That is what regulatory ambiguities facilitate. Flexibility to comply when multiple solutions can satisfy a requirement. That flexibility also accounts for the change and updates within regulation and technology. Given the need for flexibility and adaption to change, some practitioners want to evolve the practices of formal methods and traditional software development approaches when designing and building safety-critical systems [93]. My research aims to address this by testing a

methodology to support the analysis of ambiguities separately from the system or software during development. Then, give further guidance to a software development team from a legal perspective and interpretation.

2.5.2 Secure Development Practices

Modeling compliance requirements is a previously introduced field of study. Using mergeable models in notations or graphical modeling language can express stakeholders' roles, concerns, and activities. It is also a way to identify and remove ambiguities within requirements for expert review over textual descriptions. I explored such techniques to identify and remove ambiguities within regulatory text or requirements, but not so much within secure development or security standard requirements. The reason is that research in secure development practices relating to security compliance or ambiguity is relatively new compared to regulatory compliance within software development(See Table 2.1 [7]). This finding is a little surprising, as some researchers and industry IT leaders have suggested that implementing security and compliance requirements is best done as a "baked-in" solution during software and IT system development since the early 2000s [79, 112, 87, 100]. This finding and some of my other previous research [85] is why I included security standard compliance.

This subsection explores some background and related works on adopting secure development practices regarding compliance and ambiguity. Al-Amin et al.'s case study focused on creating a framework for modeling the adoption of security practices and ways

[7] After doing a Google Scholar search using six different keyword searches, I found that about 40% of Google Scholar results are within the last four years.

to promote adoption amongst developers [25]. The goals behind this case study for secure development practices closely align with the intent of the case study, which is part of my research and intended future work for the NSF grant. Al-Amin et al. looked at developers' reasoning and adoption of certain secure development practices using a framework to document developers' individual preferences. In contrast, my case study aims to observe reasoning, understanding, and decision-making in interpreting and modeling regulatory ambiguities within the legal text. While the plan for future work after my dissertation is to develop a framework, my research group is not yet there

Moyón et al. [104] and Dännart et al.'s [45] works used a graphical modeling language (i.e., Business Process Model and Notation (BPMN) [8]) to model security-standard IEC 62443-4-1 [9] with the Scaled Agile Framework (SAFe) [10] for the industrial control system (ICS). This approach is called Security-standard Compliant Assessment Model (S^2C-AM). It is a methodology to perform security compliance assessment (SCA) with a recommended security standard (i.e., IEC 62443-4-1) as a baseline that adapts to agile development techniques [104, 45]. They then extend their research by developing SCA tools for developers (or non-security experts) to perform self-assessments [103, 105] and to make security standards easier to understand within a continuous or agile software development environment independent of a security experts analysis. The use of modeling

[8]Business Process Model and Notation (BPMN) uses graphical notation and flowcharts to model business processes

[9]Also known as the 4-1 standard, IEC 62443-4-1 is a set of recommended security standards to develop industrial control system (ICS) [104, 45]

[10]Scaled Agile Framework (SAFe) is a framework of principles, processes, and best practices for software organizations to use to adopt agile methodologies in a scalable way [104, 45]

Table 2.1: Google Scholar search on secure development

ID	Keyword search	Number of Articles since 2018	Total Number of Articles
1	"secure development practices" + "security compliance"	12	20
2	"secure development" + "security compliance"	126	225
3	"secure development practices" + compliance	89	232
4	"secure development" compliance	1,110	2640
5	"secure development" + "security standard compliance"	7	8
6	"secure development" + ambiguity	390	1,170
	Total	1734	4295

techniques for security standard compliance and the researcher's intent for deploying the S^2C-AM technique within these four studies are similar to my intent and goals in validating the ambiguity modeling process. In their 2021 case study [105], they even include an International Electrotechnical Commission for the IEC 62443-4-1 standard (i.e., auditor or regulator perspective) to assess the usefulness of their S^2C-AM technique. During

the validation portion of my dissertation, I intend to understand the usefulness of modeling regulatory ambiguities from an auditor's perspective by presenting my the multi-case study data to my focus group for their feedback participants. In addition, the benefits of demonstrating acceptable compliance-based behavior are worth the cost of time to go through such an analysis with a software development process.

One difference between my work and what is in the S^2C-AM technique case studies [103, 105] is that they used a semi-structured interview as their survey instrument for both case studies. My first case study uses observations, group interviews, and individual participant surveys, and the second will use **interviews** for data collection. Another difference in the S^2C-AM case studies is the researchers targeting developers and security experts instead of seeking other stakeholders' perspectives, such as the project manager. Researchers in the second case study noted issues with model interpretation amongst four out of their 16 participants and would want to address [105]. My multi-case study's three case groups are recruiting developers, project managers, and privacy or security experts. Anyone reasonably involved in the software development process. Lastly, S^2C-AM case studies focused on IEC 62443-4-1 security standard. I am testing the modeling technique for regulatory ambiguities on legal text taken from the European Union's General Data Protection Regulation. However, given the intent and goals, the collective works presented here are close to what I envision my case study would look like from a design perspective text focusing on regulatory versus security standards.

Over the last twenty years, research has highlighted the need for improvement in process development to translate regulation into requirements. This starts by understanding the software industry's perspective on regulatory compliance practices. Chapters 2,

the Interview Study, and Chapter 3, the Survey, capture that insight, thus contributing to the research area [85, 122] and helping identify gaps where my later studies can help address the challenges still in the software industry [20, 122].

Chapter 3

Understanding the software industry's perceptions on regulatory compliance: An Interview Study

The first part of my research is an exploratory mixed-method research design using qualitative and quantitative research methods to gain some perspective of the software industry's take on regulatory and security standard compliance as part of a software development process (RQ1) [101]. The first phase of this exploratory mixed-method research design was an interview study of 15 software developers, managers, and directors from 13 different organizations currently working in the software industry from October 2020 to January 2021. I chose an interview study because it is a well-established research technique in software engineering [78, 91, 73] and has been used to identify potential gaps between research and practice [30, 31]. This interview study is meant to capture the industry perspective and practices regarding regulatory and security standard compliance by answering the following research questions:

RQ1: Where are regulatory and security requirements assessed and addressed in the SDLC?

Finding 1: The software release process is the phase of the SDLC where software companies most consistently and regularly examine regulatory and security standard compliance for their products.

RQ2: How do non-requirements engineers perceive the regulatory and security compliance process? **Finding 2:** Software developers view compliance checks and pro-

cess redundancy related to regulatory compliance and security as freeing rather than burdensome.

RQ3: How burdensome are regulatory and security requirements for businesses to implement? **Finding 3:** Participants believed their organizations viewed compliance not as an externally imposed necessity but as a competitive advantage with some financial rewards in the marketplace.

RQ4: When the Regulatory compliance changes, how is the software engineering process affected?

- **Finding 4:** Additional requirement analysis to respond to new regulatory requirements.
- **Finding 5:** Inspection and possible updates to integrated third-party dependencies and libraries to respond to new regulatory requirements.

RQ5: When the Regulatory compliance changes, how is the business model affected?

- **Finding 6:** They assess and de-conflict new compliance requirements against existing requirements and their organization's business model.
- **Finding 7:** They assess the cost and benefits of compliance against their organization's business model.
- **Finding 8:** They assess the compliance responsibility and its requirements. If the business's model cannot support the regulatory compliance requirements, they then shift the compliance responsibility to other stakeholders, like the customer, to fulfill the regulatory compliance requirements .

RQ6: What steps are required to ensure and demonstrate Regulatory compliance within a software organization?

- **Finding 9:** Having a strategized response toward regulatory compliance requirements is required to ensure and demonstrate regulatory compliance.
- **Finding 10:** Communicating the regulatory compliance process internally is required so everyone is aware of the regulatory requirements.

3.1 Methodology

This study took an empirical approach to understand the software practitioner's perspective on addressing and managing the regulatory and security standard compliance landscape. I used a semi-structured interview method during the interview study because of some flexibility to ask follow-up questions based on participants' responses and examples during the interview [26]. The execution of the interview study consisted of three phases: the pilot study, the main study, and the analysis phase. In this section, I discuss the design of the interview study (Section3.1.1: The Interview Study Design) and the methods used to collect, transcribe, code, and analyze the data (Section 3.1.2: Data Collection and Analysis).

3.1.1 The Interview Study Design

I conducted the interviews in two stages, beginning with a pilot study to assess the quality of our interview protocol and assess the soundness of our assumptions. Based

on the feedback from our six pilot study participants, I created two interview protocols [1]. One interview protocol was tailored for participants who had technical roles in the SDLC, i.e., people in the roles of software developers, security engineers, product managers. The second interview protocol was tailored for participants who were in the role of regulators or lawyers. The goal being to keep all participants, from the different stakeholder groups (i.e., software developers, project managers, technical directors, security engineers, data or privacy engineers, lawyers, policy managers, and business managers or directors) engaged throughout the interview. Once I incorporated the feedback, I began recruitment for the main interview study in September 2020 using my dissertation committee and personal contacts to members in the software industry.

3.1.2 Data Collection and Analysis

I conducted 15 semi-structured interviews using two different interview protocols that averaged an hour [2] [84]. Due to COVID restrictions and geographic locations of the participants, all interviews were completed and recorded online using Google Meets. Prior to the interview, all participants were given an overview of the study describing its purpose and goals, the IRB consent form for review and signature, and the interview protocol questions for them to review and refer to during the interview. At the start of each interview, the interviewer reviewed the consent form, informed the participant of the intent to record the interview, and if they had any concerns before starting the interview.

[1] Interview protocols are in Appendix A or at the following DOI: `https://doi.org/10.6084/m9.figshare.14842242.v1`

[2] Ibid.

Again, the average time for the interviews was about an hour. At the end of the interview, the participants were thanked for their input and that any follow-up would be to inform them of publications with a copy of the article. All video recordings were transcribed using Otter AI [3] and copies of the video recording and transcripts are stored in my Google Drive and offline on a SD card coded under the participants identifiers. For data analysis, I used Straussian grounded theory as outlined in Corbin and Strauss's book [44]. I did not form explicit research questions (RQs) and hypotheses beforehand because I wanted to reflect the perceptions of the industry software practitioners within the study [44].

I organized the analysis in two stages. The first stage was the generation of our coding and heuristics, and it was where we performed our first findings formulation (or preliminary hypotheses formulation). This stage was conducted from December 2020 to January 2021. The first step in this stage was to investigate the themes that were reported in the data within the interviews through code analysis and provide initial impressions within the interview context. I reviewed the interviews, individually coding the data through the use of a structured coding scheme, and recorded coding patterns, themes, and ideas that started to noticeably repeat within interviews. I then referenced similar ideas within other interviews to construct a preliminary findings (or hypotheses) list of 15 high level topics.

The second stage was the application of our coding to the interview data. In this stage, we applied memoing, or constant comparison of the coding and data, [44, 101, 116]. The second stage was conducted in February 2021. Once we formulated our findings, we needed to confirm them. T Therefore, we built and organized our field notes into

[3]Otter AI is a natural language speech to text transcriber.

separate memos in support of a particular finding, not to prove it, but to provide weighted evidence for our findings across the interviews. Thus, the transition from rudimentary thoughts (i.e., field notes [116]) to growth, clarity, accurate representation of the data, and analysis of the data (i.e., memoing [44]). Once we built and reviewed the memos, we had ten findings with more than 50% supporting comments from the interview study within the memos of our data analysis and coding.

3.2 Recruitment and Participants' Demographics

We conducted interviews with 15 participants (14 new candidates and one re-interview from the pilot study) from a recruitment list of 29 participants contacted for the main study. They varied in background, domain, and levels of experience in their current role. We used the technically oriented protocol for 14 of the interviews and the law-and-policy protocol for one interview. We recruited 17 participants (pilot and main) from professional contacts, and three of the participants (main) were follow-on recommendations from three previous interviews. The participants operate within different domains of the software industry, which are categorized as follows: technology [4], healthcare [5], finance [6], government [7], and other [8]. The participants were also categorized into three groups based

[4]Technology: Companies with an advertising-oriented business model requiring them to market, store,or analyze potentially sensitive data.

[5]Healthcare: Companies that manage, insure, or provide services like billing for the healthcare industry.

[6]Finance: Companies regulated by GLBA or a similar financial regulation.

[7]Government: Agencies in the U.S. Federal government.

[8]Other: Companies that do deal with compliance, but do not fit within the domains of the four other categories.

on their roles and self-reported titles. Those categories are data or privacy engineer [9], software developers [10], and manager or director [11]. The participants company size was categorized into small [12], medium [13], and large [14] based on structural development teams and resources available for software development. Lastly, the participants' years of experience range from two to 30 years and refers only to work experience in a related role. Table 3.1 represents the pertinent demographic of varying backgrounds, roles, and company sizes for the main study participants.

3.3 Findings

This section covers my findings in more detail, focusing on our 10 high-level findings, to answer my six research questions. Within each subsection, I provide data and example quotes from the participants' interviews using a narrative format. Our goal through the interviews was to gather insight, so we did not impose a viewpoint during the interview and encouraged elaboration of their organizational and personal processes

[9]Data/Privacy Engineers (D/PE): Engineers who interpret requirements and give guidance for technical implementation.

[10]Software Developers (SD): Developers that build and maintain software products.

[11]Manager/Directors (M/D): People who direct, coordinate, or set developmental priorities within the software development process.

[12]Small: A single development team and no internal compliance resources, like a separate quality assurance, security, or testing team, available to them.

[13]Medium: One to three development teams for different products with an internal security/compliance team.

[14]Large: Multiple development teams for a single product (i.e., a team dedicated to UX design, another to chat messaging, and so on) and separate internal compliance resources (i.e., governance, risk, and compliance team; security team; and testing team).

Table 3.1: Interview Participant's Demographic

ID	Years of Exp	Industry	Org size
D/PE1	less then 10	Technology	Large
D/PE2	less then 10	Technology	Small
D/PE3	less then 10	Technology	Large
D/PE4	**less then 10** [15]	Other	Large
D/PE5	less then 10	Technology	Medium
D/PE6	less then 10	Other	Medium
SD1	less then 10	Healthcare	Large
SD2	10-20	Technology	Medium
SD3	10-20	Healthcare	Medium
SD4	less then 10	Finance	Large
M/D1	20+	Healthcare	Large
M/D2	20+	Other	Small
M/D3	20+	Healthcare	Small
M/D4	10-20	Technology	Large
M/D5	10-20	Government	Small

and views. This led to participants responding, at times, in a manner that we read as a response to a hypothetical regulator asking about compliance processes. We have not removed this perspective in transcribing their statements below, but we have edited their transcripts for clarity.

3.3.1 RQ1: The Software Release Process

Throughout the interview study, the participants commented on software release processes within an organization. Our participants from large companies said that separate product teams were not required to use a specific software development process and could choose a development model that fit their skills and experiences with the exception of the final software release process which was standardized to address compliance and quality assurance. Participant D/PE3 expressed it this way:

> **D/PE3:**"They'll [software development teams] use standard stuff, like that agile method or whatever. But that sort of happens separately from the larger software release process, which does have defined steps that are followed across Organization 1. [You have] standardized code review processes no matter what sort of process you use within the cycle. You have standard release documentation. You have standard people who have to sign off and things like that."

SD2 described their company's development and release process as follows:

> **SD2:**"We're not formal agile, but we kind of do pull bits and pieces. For compliance reasons, everything has to be code reviewed and approved by one developer. But I mean that we just do that we do that anyway, at least the team I'm on do that matter what compliance require...And basically, you can't, you can't go, you can't release the product without kind of these best practices. So we have, we have something called release review before we release something new. So it's a new service, where you're adding something

> big. You know, there's a big audit of basically, all these crazy practices with architects and our chief architect. So things like security, encryption, encryption, at rest, encryption and transit, are focused on before we even go live. So I think that's prevent a lot of the issues."

SD4 had this to say:

> **SD4:**"anytime we develop a product, we have to get it scanned and approved. And there's one section, especially since most of what we're doing is in [Amazon Web Services] AWS. Yeah, they've got we've got a team that looks at everything we're going to put into AWS and make sure it in matches with all the regulations that they're aware of. And make sure that only the people with or only the people that are supposed to have access, have access to it"

Thirteen out of 15 interviews commented that their organization's internal release process not only ensures due diligence towards compliance but also catches mistakes or known security vulnerabilities that could be highly embarrassing or have significant consequences if released. Thus, the software release process was the primary means to ensure that any product released, either internally to the organization or externally to the public, meets with their own governing policy and regulatory requirements. Out of the two participants that did not explicitly mention this, one was a Government IT Director, whose organization collaborates with industry to help define standards for regulatory and security compliance rather than produce software.

3.3.2 RQ2: Compliance-oriented Processes is Freeing

Regulators want to see compliance-oriented processes in software development and evidence that they are adhered to by employees. This goal is often thwarted when compliance is viewed as inconvenient or burdensome. Evidence and documentation are crucial because the default position must be that regulators assume non-compliance without evidence demonstrating compliance.

To some, the documentation and checks within a software development process might seem burdensome. Another step in an already lengthy process. However, our participants found compliance checks and redundancies to be freeing. Developers are aware, generally, that regulatory and security compliance are important, but they may not know specifically what's required. Based on our interviews, when developers and managers are aware of the compliance concerns, whether regulatory or security-related, and they understand why these concerns must be positively documented, then they are freed from the fear of not knowing whether or not the system will be found to be compliant.

The finding here should not be taken to mean that a software release process that includes compliance checking is not burdensome. On the contrary, compliance requires the time and expertise of people who are proficient in both technology and policy. Interpreters who understand both technology and policy are critical for clear communication of compliance requirements. Despite this, all of the participants understood why compliance checking was built into the software release process and some explicitly appreciated it as a benefit. Participant SD4 may have summarized this view most succinctly.

The participating software developers in the interview study view compliance checks

and process redundancy related to regulatory compliance and security as freeing.

The finding here should not be taken to mean that a software release process that includes compliance checking is not burdensome. On the contrary, compliance requires the time and expertise of people who are proficient in both technology and policy. Interpreters who understand both technology and policy are critical for clear communication of compliance requirements. Despite this, all of the participants understood why compliance checking was built into the software release process and some explicitly appreciated it as a benefit. Participant SD4 may have summarized this view most succinctly.

> **SD4:** "So far, I think it's all been valuable to a company perspective. Because there are times when someone will push a bad update that screws up the login for customers. And then, like Newsweek, or something like that will run an article saying, 'Oh no, Organization 2 has been hacked.' And then our profits are hurt. And if that goes on too badly, there's all sorts of financial decisions that have to happen. So the extra push for security and regulation and all that [helps] calm a lot of that down. We're a lot better at catching most issues before they happen now that we have lots of environments for testing and more eyes on the resulting product."

Similarly, participant D/PE1 expressed the critical gate keeping role the software release process provided as an explanation regarding why compliance checking was focused on that aspect of the SDLC.

> **D/PE1:**"So it's not just one developer saying, 'Oh, yes, this is a cool feature. Let me add it to search.' And they just push it to production. So we have these

multiple levels, all the way from a privacy, security, [to] everything you can think of. And we also [do this when we] add anything new. [It] has to—even if it's like a color change, like we have emails, and we have our inbox and there has to be a slight color change—even that goes through all the levels of reviews. So yeah, I think that's a good way of catching anything, even if we didn't catch it now. Every increment goes through all levels of reviews. So I think that's a pretty robust way of catching anything that could potentially go wrong in the future."

Not all of our participants held this opinion. Even though they all understood why these processes were put in place, one software developer and two IT managers or directors shared mixed opinions about how regulatory, and security standard compliance translated to actual compliance within an organization. Participant M/D2 said, "Compliance is necessary but not sufficient." Similarly, participant M/D4 explained it this way:

M/D4: "[It's] valuable for building trust and making sure that the actual things that we're doing help the people that we're intending to help. But as with anything, scope creep kind of gets in the way sometimes. And I have definitely seen a list of about 300 different criteria that we have to meet in order to be certified as a particular thing, and just looked at it and been like, 'Wow, that's just overkill.' You know? And maybe, if it was written in a little bit more plain English, it could be a little bit more understandable. But I mean, that's regulatory compliance in general."

Opinions like this were not as prevalent as those expressing understanding or relief

that compliance was both present and defined. Four participants pointed out that compliance cannot cover everything and more could be done to assist industry to establish more enforceable standards.

3.3.3 RQ3: More Compliance = More Customers = More Money

Just as regulators are interested in software developers actively leaning into compliance efforts, regulators are interested in ensuring that organizations are also leaning into regulation. When software organizations hold a synergistic "more compliance, more customers, more money" perspective, then they view compliance as an investment and respond accordingly. Several participants identified their organization as internalizing and communicating about regulatory compliance from this perspective. For example, participant D/PE2 said:

> **D/PE2:** "We look at what would potentially be a competitive advantage, right? I mean, we do privacy and civil liberties, because we think it's the right thing to do. But there's no reason to also not make it a business edge as well."

Seven of our 15 participants commented on how certification and compliance with regulations and standards like HIPAA, GDPR, and PCI was simply mandated to participate in the market. As participant D/PE5 put it:

> **D/P5:** "[anyone] doing healthcare in the United States needs their technology providers to be HIPAA compliant. So that's kind of easy, just from a numbers standpoint, to be able to say, well, we think that this market is worth, you

know, so many hundreds of millions of dollars, and we can't access it at all, if we're not HIPAA compliant."

Others saw compliance as part of their customers' requirements rather than an external mandate. Thus, compliance to a particular set of regulations and standards becomes a contractual requirement and a means of developing trust with their customers. As participant D/PE4 said:

D/P5:"Due diligence, not with just regulations, but also with the respect that their customers want to have for their privacy, that builds trust with our customers. And that allows them to build trust with their customers by not being spammy or scammy, or anything like that. We don't want anything like that."

One participant that held this view also expressed that it prevented their organization from providing the customer "what they need" or not operating in certain regulated fields. Thus, when working with customers with some regulatory concerns, their organization would provide software tools to allow their customers to demonstrate compliance, but actually using these tools to do that would be left up to the customer. This approach puts all the responsibility on the customer, with the compliance boundary being defined either by the product itself or through a contractual agreement. For example, an organization could use a cloud-based data storage service in whatever regulated field they want, but compliance with retention regulations and accepted security practices would be their responsibility.

Informal communication of organizational compliance occurred as well. One participant identified the regulator's job as being there to "help industry help itself" and indi-

cated that informal means of compliance communication helped move adoption of a regulation along faster than formal communications. They indicated that informal compliance communication "normalize [compliance] and help it become scalable." They pointed out that this not only directly improves compliance for customers, but also that when "the top 30 or 40% of the population are doing it, well, then you're not going to get this massive political pushback." Ultimately, regulators just want industry to do and be better in protecting their assets and customers' assets, whether that requires informal communication or formal regulatory action.

3.3.4 RQ4: Regulatory change affects to the Software Engineering Process

With change to regulatory requirements comes change to a software organization's processes. The engineering process is definitely impacted according to the interview study participants. First, new regulatory requirements require **additional requirements analysis**. Second, additional requirements might mean technical change, therefore **integrated third-party systems and libraries require inspections and compliance validation** in response to any new regulatory compliance requirements. The extent to which these changes impact the engineering process varies from one organization to another, but the changes themselves are common for organizations operating within the software industry, according to the interview study participants.

3.3.4.1 Additional Requirements Analysis

Our participants commented that new regulatory requirements require additional requirements analysis. Nine participants (8 out of 13 organizations) talked about how any response to a compliance change starts within the requirements phase of their engineering process. First, they need to assess, interpret, and understand the new regulation requirements to determine the best course of action for a software engineering change. For example, consider PE5's comment made in their December 2019 interview on their organization's initial response to European Union's General Data Protection Regulation (i.e., GDPR) in 2018 when it was first enacted:

> **PE5:** "So as an organization, we looked at what we thought were the main requirements of GDPR. And how we felt like we, you know, kind of a gap analysis like, how do we comply with GDPR? And what do we need to do to what do we need to change to comply with GDPR. And those changes that the gaps in compliance with that regulation for the market that we wanted to serve? They basically got translated into a big project that we use."

Their organization assessed the requirements of GDPR when it first came out and responded by forming a project team that created an Application Programming Interface (API) to handle GDPR requests. Other participants commented that they took a similar approach by having legal experts assess the requirements of GDPR and engage the product teams directly to determine whether they were GDPR compliant, as seen by PE4's comment below:

> **PE4:**"When GDPR came about, the legal teams engaged the various product

teams. They started asking a lot of serious questions about how the product is used. Just really exploring anything that could be a GDPR problem for us or our customers."

Even with minor compliance changes, some requirements assessment happens to ensure an engineering change is required and to avoid implementing unnecessary changes to the software (i.e., false starts). For example, consider Microsoft's decision to phase out password-based basic authentication and implement a token-based form of authentication (i.e., OAuth 2.0) to access Microsoft Cloud service resources such as Exchange Online mailboxes in 2020 [16]. AAll of Microsoft's customers had to comply with this new security standard. Therefore, any product that interfaced with Microsoft Cloud Service also had to update authentication to Microsoft Cloud; however, not everyone fully understood the new security requirement initially. One of our participants, M/D3, described this misunderstanding as follow:

M/D3: "And we had to go through a bunch of channels to finally end up talking to their Microsoft architect to find out that nobody understood the requirement, or at least nobody passed it to us correctly; the requirement was to go to OAuth 2 and get rid of basic [password] authentication. And just the telephone game. They [M/D3's customer] just heard that was you have to get away from IMAP... the request that we got was pretty unambiguous. Drop IMAP and implement AWS. It was when we found out that we couldn't authenticate that account using their systems. When what worked for us [the

[16] https://www.nylas.com/blog/microsoft-basic-auth-vs-microsoft-oauth/

> authentication] and our test environment but didn't work in their testing environment, we realized that the security that they had set up on the account was different. And that was fun, a bunch of emails, finally, a meeting with the [Microsoft] architects who actually understood what was going on."

The initial change request from M/D3's customer was to get rid of IMAP and migrate to an Application Programming Interface (API) to access Microsoft Cloud service resources such as Exchange Online mailboxes. However, the actual requirement was to migrate from basic password authentication to a token-based OAuth 2.0 authentication. Once their tool could authenticate using OAuth 2.0, their tool can continue to intake their customer's emails hosted by Microsoft's Cloud exchange web service. According to M/D3, it took testing and many emails between them and the customer's representatives to find the right person to explain the requirements of OAuth 2.0 authentication. Finally, a meeting with their customer's Microsoft architect is what correctly vetted the new security requirement before M/D3 software engineering team took the next step in engineering a solution within their software to meet the OAuth 2.0 authentication requirement. M/D3 pointed out the coding to implement OAuth 2.0 took only 40 hours.

These are just a few examples from our participants commenting on how the requirement analysis is emphasized and given much more time within the software engineering process (i.e., Software Development Lifecycle). The additional time spent on requirement analysis is meant to thoroughly vet a new requirement, make sure the rationale for the requirement is correct, and avoid false starts during the design and implementation of the engineering process. Additional requirements analysis may also include inspecting

and updating third-party dependencies and libraries for any engineering or compliance issues in response to a new compliance requirement.

3.3.4.2 Integrated Third-party Systems and Libraries

Using third-party code, libraries, and software is common in building a software or system platform within software development. Therefore, any engineering change requires further inspection and rework of the integrated third-party dependencies and libraries, especially for compliance. In addition, the longer software or systems are in use, the more likely other systems or platforms depend on them. Therefore, any engineering change would affect those integrated systems or platforms. One of our participants, M/D2, gives an example of reworking software authentication due to a change affecting other dependents and integrated systems.

> **M/D2:** "Oh, yeah, I'm sure that has happened in the many years that we've had things that were approved on or a previous ATO [Authorization to Operate] the standards or the guidelines were updated, and therefore they were no longer acceptable. So yes, I'm sure that I've had that happen on several occasions... It caused rework. In some cases, it might be a significant rework. Depending on where the functionality and compliance of the regulation impacted if it were a foundational building block, it could impact not only the stuff that we were developing but also everybody who was working on that project. So, everybody who was integrating with a core capability from security, as they were investigating different methods of authentication,

> different methods for you know, there, there were false starts, you would integrate with one thing, and then they say, Nope, that's not adequate. Now, we need to move to something different for authentication. And so you'd have to rework that integration effort."

Also, consider the previous quote, earlier in this section, from M/D3. M/D3's engineering rework was because of a security change decision made by Microsoft to phase out their password authentication protocol within their email cloud platform in favor of a more secure, token-based authentication.

> **M/D3**: "Is that a regulatory change? Probably not strictly speaking, but it is Microsoft basically decided based on Microsoft's reviews of security incidents involving their cloud platform. And they basically said 90% of the vulnerabilities have to do with basic [password] authentication. So, as a company, we're simply going to phase it out. It's [password authentication] no longer going to be supported... Interestingly enough, that was not even our customer's decision. So because they, [M/D3's customers], decided to go with Microsoft, as a cloud vendor, they were at the mercy of whatever Microsoft decided"

As our participant explained, customer and provider systems integrated with Microsoft's cloud platform had to respond to their new security update by abandoning password authentication in favor of token-based authentication, creating a new security standard for compliance. Though necessary and meant to resolve problems within an industry, both regulatory and security changes can become increasingly complicated to fix when one

considers the prevalent use of integrated, third-party software and libraries within today's developed software.

3.3.5 RQ5:Regulatory change affects on the Business Model

The engineering process is not the only process affected by Regulatory change. An organization's business model can be affected with regulatory compliance change. According to the participants, businesses respond to changing regulatory requirements. These responses include **de-conflicting new compliance requirements** with other compliance requirements or business priorities and **weighing the cost and benefits of the new compliance requirement**. After assessing these updates to regulatory requirements, a final decision has to be made. Is adhering to new compliance requirements worth the complication and associated cost? If the answer is no, some software organizations might make changes on the business side to include **shifting compliance responsibilities** onto other ecosystems or stakeholders to avoid the complication and cost of compliance. Much like the engineering process, the extent of compliance changes affecting business models varies between our participants' organizations, but these three findings emerged within the transcribed interviews with our participants.

3.3.5.1 De-conflicting Compliance requirements

The compliance landscape is complex, and according to our participants, changes within the compliance landscape affects the business models of their organization, creating conflict. These conflicts with compliance requirements arise for different reasons.

First, a new compliance requirement may conflict with an existing compliance requirement. Second, new compliance requirements require assessment against an organization's business model to align with the business goals and requirements. Third, while assessing new compliance requirements and how an organization might have to change to comply, gaps within an organization's business model arise. Thus, the business model must also update to address these gaps to adhere to the new compliance requirements. To the first point, the conflict between compliance requirements is not new; other papers have commented on the conflict of regulations that overlap or contradict themselves [111, 99]. Our participants echo some of the conclusions within these papers when asked about their own experience in applying and evaluating compliance requirements. Consider PE4's comment about conflicts between the financial compliance requirements to retain data and the General Data Protection Regulation (GDPR) compliance requirements to minimize data.

> **PE4**:"the compliance landscape is complicated, and a tremendous amount of resources at [Organization 5] are dedicated to compliance. So it takes a lot of time and a lot of money. And sometimes, it's difficult to comply because financial players, for example, have requirements to retain data, whereas GDPR has requirements to minimize data. And those two things are in tension. So I've been in conversations before where I've had to listen to our customers in that industry, talk about the different tensions that they've had within it."

Many software organizations dedicate time and money to interpret and document the baseline compliance requirements according to their business requirements to address

these conflicts. Some organizations with a large footprint that expands across several industries need to further document and trace industry-specific regulations to ensure compliance for all their products. Consider for example an international organization deconflicting the privacy requirements (e.g., the European Union's General Data Protection Regulation (GDPR) or California's Consumer Protection Act (CCPA)), technical and operational requirements to secure and protect credit card data (e.g., PCI), and security and privacy requirements (e.g., HIPAA) to secure and protect personal health information, as PE3 explains.

> **PE3**:"So we have a pretty large set of internal policies and standards that we've built at the company. And that's sort of like the baseline compliance there, right, and all of that stuff is internal. But if you sort of backtrack and figure out why those standards were created internally and what they're based on, right. It's usually these external floors, things like CCPA and GDPR. And the privacy side, right, are kind of incorporated by tacit assumption into our internal policies, and then things in specific domains. . . Right, there might be additional compliance burdens if you're on the health side with HIPAA. For my colleagues who work with financial data, there are some ISO and PCI privacy standards that I don't know about that they have to comply with. And there are these other industry specific ones. But generally, most privacy and, honestly, security standards are internally written, with the policy floor being some sort of external regulation. But we usually try and sort of exceed those [external regulations] with our internal policies. . . . So I think just because

> of [Organization 1's] size and scope, right, that's kind of necessary. You need that additional layer of abstraction to make things work smoothly. But I think that's like a relatively unique way of working."

In the above quote, PE3 explains, as part of a large, international technology organization, the requirement for internal policies and standards is needed "to make things work smoothly." To serve as a baseline for compliance but also to resolve any conflicts is supporting multiple regulated domains such as GDPR, CCPA, PCI, and HIPAA. Compliance requirements and business decisions require documentation within an organization's business model to resolve compliance requirements conflicts and gaps. The documentation also helps software development stakeholders, internal and external to the organization, understand what the business is doing and how best to support them. M/D3 explains that business and software stakeholders' understanding of that business model must be mutual. A failure to have a mutual agreement or gaps within the business model can lead to conflicts or missed fulfillment of the different stakeholder requirements and compliance requirements.

> **M/D3**:"I found by really drilling down on the front end [requirements phase of software development], I found that a lot of things that they thought they knew the answer to, they [the customers] sort of thought they had the business process behind it. When I started asking the important questions of how this is going to work, I discovered they really hadn't thought about that before. And so, in my mind, I look at these things as well. Business process engineering has gotten a bad connotation to it. But I still think you must understand [what]

> the business is doing and how it works to drive the requirements, right? Then you can figure out what the automation is supposed to do to support those business decisions. And so, as is typical, I'm in there pushing them to make these business decisions to build out their business model that they really had not built out before. Because again, this is a new venture for them. It's not like they, [the customers], have a working system that they're replacing or enhancing. This is a new venture for them as well."

M/D3's comments on business model assessment as part of a compliance requirements assessment are common throughout the data of other participants' interviews. Eight out of our 15 participants commented on the amount of time spent interpreting and assessing the requirements to compliance and de-conflicting the compliance requirements against other requirements. They also commented on prioritizing and weighing each requirement carefully against the business model to ensure the requirements align with the business's goals and services. Should a conflict arise between compliance requirements and the business model, both are carefully assessed. Then either the business model is changed or updated, or the cost and benefits of the compliance requirement are further analyzed. This cost-benefit analysis of the compliance requirement is our second finding to RQ5.

3.3.5.2 Weighing the Cost and Benefits of New Compliance Requirements

Cost-benefit analysis (CBA) is a fundamental business and economic principle that organizations within the software industry apply when starting a new project, developing new software or developing a new software feature. When **weighing the cost and benefits of compliance**, our participants commented on the business's requirements and the customer's requirements. The business requirements are tracing directly to the business model, and the customer is their primary source of revenue. Sometimes, compliance requirements align with business and customer requirements; other times, compliance requirements only align with one. Another factor was weighing the risk of non-compliance and the impacts of prioritizing compliance over other requirements.

Sometimes weighing the cost and benefits of compliance requires taking a close look at the business and the services it provides. Then, an organization may choose not to compete in specific fields because those customers' requirements do not align with the business's requirements. Consider PE4's comment about how their organization does not provide regulated services to meet healthcare or finance customers' needs when the researcher asked them if they had "any customers in highly regulated fields?"

> **PE4**:"I'm going to say no. And the reason is because the customers [in health and finance] are highly regulated... we do not allow them to put regulated data into our product. If they do, it's their own bad. So our product does not work within those regulated fields, because we cannot give them what they need to comply with those [requirements]."

PE4's product is specifically for marketing purposes, so while they must comply with data retention and privacy laws like GDPR, Privacy Shield, or similar, they are not in the business of supporting healthcare or finance customers' data and security requirements. Therefore, there is no benefit to adhering to healthcare or finance compliance requirements when considering the overall business model of PE4's organization.

However, we did interview participants that operated in those fields of finance and healthcare. When I asked them about their background with compliance requirements and regulations, M/D1 responded with this:

> **M/D1**:"One of the most important things that I do is ensuring the secure and safe travel of clinical information. So, I have to do that both internally when we're talking systems internally, as well as when we're communicating with systems externally. . . HIPAA, so those provide sort of guidelines and regulations for the safeguarding of patient, clinical and financial information. So, there are regulations and guidelines in there that we have to adhere to when we are transmitting or managing, internally patient information."

M/D1's organization runs a healthcare network, and compliance to laws like HIPAA are compliance requirements that align with both their business and customer's requirements. Other participants commented on how the compliance requirements, such as the General Data Protection Regulation (GDPR), were more of a customer requirement than their business requirements. However, because it is a customer requirement, businesses still respond and take the time to build tools to become GDPR compliant to meet their customers' needs, as SD2 explains:

SD2:"We have it in contracts, and we have various different compliance requirements for contracts. So I guess my first real exposure in this company, when I started five years ago, is with GDPR compliance. It was before GDPR was out. But we had to rapidly get GDPR compliant because obviously nobody in Europe will use your call center software to talk to their customers unless you're GDPR compliant.. . . .our customers are the ones who have to be GDPR compliant. So we get GDPR requests on behalf of other people through our customers. So our requirements are different for GDPR. So we have to enable our customers to both, you know, get all their data for a particular user, and also deleted."

SD2 describes how their company had to become GDPR compliant because it was their customers' requirement; therefore, to ensure the use of SD2's organization's services in European markets, their customer's GDPR compliance requirement became their new business requirement. This new business requirement is outlined and communicated through contractual agreements between SD2's organization and their customers to adhere to the GDPR compliance requirement. Similar comments from other participants explain how their organization's business model directly connects to helping their customers meet their compliance requirements for their industry.

Consider PE2's comment on creating tools or products to help customers build "an efficient and effective GDPR compliance program".

PE2:"From a regulatory compliance perspective, we also are thinking that what kinds of capabilities do organizations need to have to comply with

> things like GDPR with the various security and data governance standards that are out there?...we say [we have] the tools that you need to have an efficient and effective GDPR compliance program. [By] thinking about things like how do you build better access controls, managing deletion, building audit logs and data governance, analysis capabilities to actually help you provide effective oversight of your products. Those are the kinds of things that our teams are responsible for thinking about and building into the product."

Although PE2's perspective is to help customers with their compliance requirements, they are also careful to communicate where their responsibilities lie with GDPR compliance while helping to meet their need, which is our third finding.

3.3.5.3 Shifting the compliance responsibility

Seven out of our 15 participants commented that their organizations view compliance as an investment and respond accordingly with time, staffing, and money. Other participants commented that while compliance is an investment, the cost to comply can have drawbacks. Though they might provide tools to assist with compliance, they have contracts in place to either define their compliance responsibilities or avoid compliance requirements on their end. Thus, avoiding the complications and the cost of compliance.

The business reasoning behind side-stepping compliance responsibility is relatively straightforward. First, they do not have the resources to shoulder the cost associated with compliance. They must be careful in what compliance requirements they can take on as a software provider or practitioner. Second, while a service provider or software developer

might provide a capability for compliance, the customer's use of the tool or service plays a role in how a compliance requirement is met and maintained. Third, the compliance requirement is a customer's requirement and ultimately their responsibility to meet.

Our first point to the shifting of compliance responsibility comes down to cost. While adhering to compliance requirements is necessary to operate within specific industries, it does not negate that using resources to manage and track compliance is costly. Although some businesses have more than enough resources to shoulder the cost of compliance, small or start-up companies must be careful about what they are willing to take on. Consider M/D3's comment about how his organization has to be very strategic about what compliance requirements they can take on.

> **M/D3**: "Being a small company, I don't want to say that we were trying to avoid the security requirement, but because it is so onerous, um, we would be very strategic in, you know, what we took on so, you know, putting up a website, you know, you know, a web-based tool that can be accessed, like, Oh, my God, right, what you have to do to secure that and keep it operationally secure. Uh, you know, that probably could be equal to the rest of our development budget, you know, if you needed to do that, so we would have avoided doing things like that where we could, or we would try and make sure that what we're going to set this up, but we're going to contractually arrange it. So you're responsible for the operational security will meet what we need to provide in the software design and development. But, you know, we're not in a position is, you know, even at a very small company, to take on all of that."

As M/D3 points out in their comments, because of the size of their company and the resources available to them, they must make clear, through contractual agreements, where their compliance responsibilities lie. They ensure they do not take on more than they can handle and that their customers manage specific compliance requirements, like operational security. PE2 makes a similar comment about compliance responsibilities and how they are careful to communicate to their customers what their responsibilities are.

> **PE2**: "So we are very careful not to be legal advisers, we don't want to tell them how to be GDPR compliant. But basically, we will say to them. You can use these capabilities to meet your goals, right, whatever your compliance goals are. So yeah, I'm helping them as a consultant, like, here's how you might think about using these capabilities to meet whatever needs you have. And they're a lot of different ways to configure these things for different workflows, different purposes and stuff."

PE2 supports our first point of how some organizations are careful on what compliance requirements they will take on, but they also support the second point about customers' use of the tool. Other participants made similar comments that meeting and maintaining compliance is more than providing a capability. Also, how a tool is used is important. Consider PE6's comment about applying basic operational security:

> **PE6**: "It does come down to what you do with it. We have a system that allows you to encrypt all data in your database or encrypt all your files. But if I don't turn on that encryption option, it's on me. Theoretically and mathematically, encryption will work. But if I don't apply it correctly, if I don't change

my passwords, if I have like a simple password, it's not going to help. I think they, [Amazon], do all they can but it's [up] to the user. If a lot of the breaches are basically someone having like, admin username, admin password, right? . . .Amazon is good. They cover all the things [in documentation and guidance], but it's up to the users, how they use it [Amazon's services]. So if we don't use it correctly, we will be in trouble."

This finding at first glance may lead to an assumption that providers are just passing the buck to the consumers when it comes to compliance requirements—stating that it is not our problem. However, the shifting compliance responsibilities are more about defining the compliance responsibilities because compliance is a shared responsibility between provider and consumer. As we saw with previous comments within these findings and other findings, the shared responsibility of compliance is unavoidable. It also must be defined within the business model of a software organization. Failure to do so creates problems on both the business and engineering sides of software development. To avoid these problems, software organizations must think beyond today's current applicable regulations and security standards. They do so by strategizing compliance and response to change and internalizing compliance communication within their organization's culture.

3.3.6 RQ6: Strategies to responding and ensuring Regulatory Compliance

"If you fail to plan, your are planning to fail" –Benjamin Franklin

Organizations must have a **strategy to respond to changes regulatory compliance re-**

quirements. Regulatory compliance requirements change all the time. The first step in ensuring and demonstrating regulatory compliance is to have a plan to comply and respond to updates when the compliance requirements change. This first step allows them to assess what their requirements are and plan a response or needed actions to go forward. The second step is**internal compliance communication** to employees and applicable stakeholders. According to our participants, this communication is seen through employee compliance training, documented compliance checking procedures, and access to internal subject matter experts that are available to answer or communicate compliance concerns. Although compliance response strategy and internal communication do vary amongst organizations, these two findings are were the most consistent planning steps commented by the interview participants.

3.3.6.1 Strategic Compliance Response Plan

Ten out of our 15 participants commented about their organization's overall strategy to change within the compliance landscape. Some participants detailed that their organization strategy is to invest time, money, and staffing in monitoring the compliance landscape. Once a change occurs, their organizations will bring in experts to interpret and outline what needs to change, then respond accordingly, sometimes going "all-out" to ensure compliance with a particular regulation. Some commented on how they create tools and applications to monitor landscape change, and once it occurs, make the necessary updates for compliance. Others talked about a "wait and see approach," where their organization's resources are limited. Therefore, customers need to come to them with

the requirement first, sometimes specifying a "fee for service" within a contract before they make the changes to comply. Even though strategies varied between organizations amongst our participants and some of those strategies are resource-driven, our participants provided an overview of their organization's strategy in responding to regulatory change and why those strategies are in place. For example, consider SD1's description of how their organization monitors changes to the healthcare regulatory landscape for long-term assisted-care facilities.

> **SD1**: "The first step, of course, is getting notified that this change was coming. We have some kind of tracker that just watches a couple of CMS [Center of Medicaid Services] pages and just sends us an email anytime anything changes. We also have a couple of people that [watch] the Federal Registrar where documentation [of] impending changes to existing legislation exists, is published basically when it goes into effect... And so, the first step is getting learned. The second step is actually reading through what all the changes required."

SD1 describes how their organization monitors the Center of Medicaid Services websites and the Federal Registrar for any forecast regulatory changes. Once they know a change is coming, they review the new regulatory requirements and, together, determine what needs to change to comply using their internal compliance to track the changes. Using tools to stay up to date on compliance landscape changes then relying on a compliance team to track and document the required software updates is one example of a strategized response to compliance landscape change. Another example is to bring to-

gether a team of experts to review a product line for an upcoming new regulation before its effect. For example, consider PE1's comment. They explained how their organization formed a cross-functional team of subject matter experts responding to GDPR and CCPA to go through the organization's entire product line to ensure compliance.

> **PE1**: "So all our products had products because most everything that(Organization 1) owns is, I think, except for a few products, everything is on the App Store. So we had to comply with it. So, we had task forces responsible [for assessing] bottoms up to ensure all our products were compliant. They listed out steps that the product teams needed to go through to do what was expected. They [the cross-functional team] were people who studied this in-depth knew how to do this. And they were approvers, which all the [development] teams had to go through. So yeah, there is an actual streamlined process to make sure that we are compliant with the changing, especially in privacy, [where] I think things change so quickly."

This response was like the other participants from large companies' reactions to GDPR and CCPA. Other participants made similar comments about how they have dedicated staffing to help translate new compliance mandates from governing standard bodies such as the Federal Trade Commission into engineering requirements as M/D4 comments below.

> **M/D4**: "So we have particular groups of engineers, program managers, and product managers who have these existing relationships with both standard bodies and the regulated industries that actually do these certifications. And

so we have experts that we've hired to be those translators for us that actually helped turn those into the requirements for engineers to follow."

Teams of experts, engineering, compliance, and legal, coming together to interpret and assess compliance requirements and translate them into engineering requirements is a strategy that offers a competitive advantage when it comes to demonstrating compliance. However, it is a very resource-intensive strategy, and not everyone can respond in such a way when a compliance landscape changes, as PE3 points out:

PE3: "When it comes to scale, [if] you have enough people working on the problem, where you have people working full time on compliance infrastructure, it's not an afterthought. It is a fully staffed function. So efforts, where you figure out what data project uses by automatically interrogating all these different storage systems. That's the kind of thing [is] a full-time project. And if you had someone who's only working on compliance, like 50%, [like] at a small start-up, you just couldn't do that."

SD2 makes a similar point about how much resources it took for their organization to become GDPR compliant. They also point out how smaller companies would be at a disadvantage because they cannot dedicate such resources, especially for a fluctuating regulation like GDPR.

SD2: "As far as development-wise, it's mainly kind of been a hurdle. You know, GDPR. It seems like it would hurt, hurt smaller companies. We spent months doing this. It was very expensive for a company to do GDPR because

of the amount of effort it took to do and be compliant, and it seems that it seems difficult for maybe smaller companies to be able to do so. So now everybody who wants to compete in Europe has to be GDPR compliant if they want to compete with us. And we've already kind of done that. So, it's kind of a high barrier of entry."

Therefore, smaller companies must adjust and deploy different strategies when it comes to compliance. For example, a "wait and see" approach, as regulation becomes more defined, new industry standards become known, and case law offers a better interpretation of the regulation, smaller companies can respond to new requirements either dictated by the industry or given to them by the customers. Recall M/D3's previous quote explaining how their available resources are the primary consideration in choosing this approach.

M/D3: "Being a small company, I don't want to say that we were trying to avoid the security requirement, but because it is so onerous, we would be very strategic in what we took on. So, putting up a website [and] a web-based tool that can be accessed [There is a lot of security requirements] you have to do to secure that [website] and keep it operationally secure. That probably could be equal to the rest of our development budget, if we needed to do that, so we would have avoided doing things like that where we could, or we would try and make sure that what we're going to set this up, but we're going to arrange it contractually. So [the customer is] responsible for the operational security will meet what we need to provide in the software design and development. [Because] we're not in a position is, [as] a very small company, to take on all

of that."

This strategy is not about delaying adherence to compliance but, as M/D3 put it, "be very strategic" in what a smaller organization takes on based on available resources. Regardless of what strategy is used, most of our participants agree that an approach is required, especially to support highly regulated industries like healthcare or finance. Furthermore, this strategy must be communicated within an organization for effectiveness.

3.3.6.2 Internal Organizational Communications about Compliance

The second step to ensuring and demonstrating regulatory compliance is communication. Ten out of 15 participants described both formal and informal types of internal compliance communication within their organization. Formal compliance communication included the policy and procedures regarding compliance, required employee training, and contractual agreements with third-party vendors. Informal internal compliance communication had compliance conversations amongst the employees and the overall compliance culture promoted within the organization. For example, consider M/D3's description of their organization's compliance culture and the internal compliance communication that occurs weekly.

> **M/D1**: "It can withstand scrutiny, they[auditors] would come in and do an audit . . . [Organization 3's] is very transparent. We're one, I hate to say, one big family. We really help one another deliver those services, whether it's a doctor giving it directly to the patient or us providing services to the doctor so they can take care of the patient. We all know the importance of adhering

to regulation, compliance, security, and privacy. It's all sort of ingrained in us as a value system. . .we always start each meeting with a safe moment, like how we safeguarded the patient. And it ranges from clinical safeguarding to data processing safeguarding, so it's ingrained in our culture."

As previously described, M/D1's organization is a healthcare network; therefore, compliance to HIPAA is a requirement tied to their business model. However, M/D1 comments tell just how ingrained compliance is within their organization's culture and how compliance is a weekly conversation amongst M/D1's teams. Therefore, awareness of compliance requirements remains high. Other participants talked about compliance awareness and response to compliance changes through more formal means of compliance communication such as required compliance training. Consider SD3's comment about the organization's response to compliance changes:

SD3: "If they have changes, then I guess I have to retrain the development teams on those changes to the regulations. So we've been trained on regulations and things like HIPAA. I think everyone in the company has to go through HIPAA training. But if there are changes to HIPAA or changes to regulatory processes, then we would have to be retrained and change our process."

Employee training was a standard answer amongst our participants when asked how they identify regulatory or security standard requirements or when the conversation talked about compliance awareness. Consider SD4's comments on their organization's compliance training for the employees and how it developed over time.

SD4: "When I first started, it was mostly just project work. And the manager would tell us, "Hey, keep an eye out for this". A few years ago, part of the education team started pushing out mandatory training every year. We have some [training] on software development life cycles, we have some [training] on keeping applications secure, and in various other things like that. The company tries to push it more to keep it in our minds every year. And if there's any updates that they need to add, they throw that in there as well, so it's more in the front of our minds when we're developing now."

For some participants, like SD4, the formal compliance communication of mandatory employee training was not initially pushed. However, over time, compliance training started becoming mandatory for every employee. For SD4, it is now an annual requirement that helps raise compliance awareness, placing it at the forefront of their minds when developing software.

Mandatory compliance training was not the only formal means of compliance communication from our participants. As seen from previous quoted participants' comments, contractual agreements on compliance were also a common answer. Maintaining and demonstrating compliance is a shared responsibility. Therefore, contractual agreements outlining what that shared responsibility entails and the expectation of fulfilling those responsibilities is essential to maintain that relationship. PE6 describes this contractual relationship regarding GDPR and CCPA compliance and how the agreements between the different businesses work to maintain their compliance with GDPR and CCPA.

PE6: "And so the contract between the companies basically says that they are

> CCPA compliant, or they are GDPR compliant, and we are GDPR compliant. That's how we continue the relationship, which means they will store the data, and if there is a breach, they'll let us know. They will comply with best practices. They will encrypt it at rest, and so forth. As long as we know, they are GDPR [compliant], then, according to GDPR, we can work with them, and we are GDPR compliant because the part of data that is outside of the org is also under the same standards."

Contractual agreements may not necessarily be "internalized compliance communication" since they usually include external entities associated with a software organization. However, contractual agreements help to interpret and clarify compliance requirements for an organization and document those exact requirements, which can later be pointed to when needed. Therefore, while not "internalized compliance communication", they affect internal compliance communication and are essential for ensuring and demonstrating adherence to compliance.

3.4 Discussion and Lessons Learned

I was encouraged by most of these findings because software developers are interested in implementing compliance requirements effectively. Also, the participants were able to elaborate details of their Regulatory Change Process within their Software Engineering/Development Process. Therefore, we may be past the point where regulatory and security compliance requires a priori justification. Indeed, our general takeaway was that engineers are comforted when they know their software process includes strategies and redundancies to address regulatory and security standards compliance. This is not to say that the industry as a whole is in universal agreement, but our findings in this area are encouraging. In this section, we discuss seven potential lessons for researchers, software practitioners, and regulators to consider structured around our three high-level findings.

Our participants reported, almost to a person, that the software release process was the focal point for compliance during development. Certainly, it should be a focal point, if only because it's the last chance to catch a problem before the customer does. Perhaps more importantly, knowing that this is a common way companies address compliance affords regulators an opportunity to develop release standards and practices along with processes for verifying that they are being used. Perhaps stronger requirements engineering practices can be bootstrapped once compliance is a firmly established part of the release process, as intimated by one of our participants arguing that a critical mass of 30–40% could tip the scales without pushback. Hence, **Lesson 1:** Target and test compliance requirements within the software release process.

Not everything about this finding is encouraging. If the software release process

becomes the only place where compliance is positively affirmed, then organizations will be fundamentally inefficient in building compliant software. Security researchers and engineers have been arguing for years that security must be a "baked-in" from the start of any software engineering effort to ensure that it is done and incorporated correctly. Regulatory compliance should be viewed similarly. Only examining compliance concerns in the release process creates a single point of failure and could turn the release process into an arbitrary list of "do this" or "do not do that" with no real understanding of how regulatory compliance fits into software development. It also might be too late to fix before a release. Therefore, software might take on "technical debt" as a result.

When compliance is "baked-in", the software development process becomes a more synergistic process as some of our participants explained. Four of our participants from larger companies explained that compliance is part of every step of the development process, including requirements, design, implementation, and change management within the maintenance phase. Larger organizations also have resources to ensure that a security developer can work with the development team so that the final release process is more of a verification of preexisting requirements than an imposition of a new requirement.

> **D/PE4:**"The product security organization will from the beginning, from the design phase of that work item, look at the design of the work item, and they'll give their input into design. They'll work kind of hand in hand with the developer to help the developer think about the security requirements of that. And then as the code is actually written, the security organization will also side by side with the development team, review the code and help them find

> issues with the with any potential issues. And then after that, we have various kinds of analysis, static and dynamic. And it goes on to penetration testing and bug bounty and all of that once we get into the production process."

Security-focused developers as recommended in the NIST Cybersecurity Framework is a resource companies should have. However, regulatory compliance focused developers are not mentioned within popular industry cyber frameworks like NIST. Whenever a big change within the regulatory compliance landscape occurs, such as the GDPR, organizations seeking to comply will review all of their products for compliance as D/PE1 previously quoted Section 3.4.6.1.

> **D/PE1:**"We had task forces that were responsible to make sure all our products are compliant [with GDPR]. They listed out steps that the product teams need to go through to do you know, what was expected. There were people who studied this in depth [and] knew how to do this. And were approvers through which all the teams had to go through. So, there is an actual streamlined process to make sure that we are compliant with the changing [requirements.] Especially in privacy, I think things change so quickly."

Therefore, our second lesson. **Lesson 2:** Software organizations, developers, and managers must incorporate and account for the costs of compliance throughout the SDLC when planning software systems with compliance concerns. Compliance requires resource commitments in funding, time, and staffing. Planning for these resources is a part of a regulatory compliance plan or strategy whether it is initial certification to operate within a regulatory domain or responding to new regulatory requirements. Unfortunately,

planning for resources for regulatory compliance is not immediately prioritized at the beginning of the software development process since compliance could get incorporated only after a near miss occurs. Incorporating compliance use cases may happen only after a near miss or when a big mistake points to obvious defects made in the absence of or with an inadequate release process [110]. As one participant put it:

> **SD1:**"And there was no rigorous process for identifying things like that. And that did lead to quite a few mistakes. We had a fairly big miss, you know, actually one that I caused. I made a change to how something works. . . [Redacted details of mistake.] So there wasn't. . . Yeah, I don't know of any specific process. After that massive regulatory change, we actually made a couple of changes on team, we created a compliance team to kind of formalize the process of tracking those software updates, tracking the registrar tracking, the XYZ website, and kind of being the ones who identify any changes. . ."

Software organizations might not be as fortunate as SD1 to catch a mistake before it has dire consequences. Thus, **Lesson 3:** Learning from organizations that have failed to achieve compliance and incorporate those lessons learned into organizational practices.

Learning from other's mistakes is something that certain industries seem to be really bad at, thus requiring regulators to step in to force the issue. Consider the Volkswagen "Diesel Gate" example from Chapter 1. Volkswagen was not the only automaker to have cheat devices. They were just the first caught. It turns out that nearly every European and some Asian automakers used cheat devices to circumvent diesel emission testing [6].

In the wake of this industry scandal and the global climate change, countries globally are imposing regulatory changes within the car industry to meet climate goals within the next 10 years according to the Paris Climate Accord [6]. These changes have supported manufacturing and selling of electric vehicles to lower emissions. So, while "Diesel Gate" might have been a scandal of massive proportions, it might have been the "scandal" to force needed change.

The problem is that figuring out that change needs to occur after someone dies or a company is fined billions of dollars, should not be the answer. Also, for the software industry, the release process alone may be incapable of verifying compliance in some circumstances. For example, new efforts to define concepts like "fairness" in algorithms [102] and determine how to implement them does not necessarily include demonstrating that they were implemented correctly and are working properly. It may be the case that evaluating fairness for information systems using inputs and outputs alone is either inefficient or ineffective relative to evaluation of requirements and design artifacts. The relationship between fairness, accountability, and transparency is not currently well understood. For it to be well understood will take more effort. Effort that can signal to regulators that the software industry is capable of taking these steps and figure out what are the next steps to a more robust process.

Redundancy and compliance checks during development free developers because they allow developers and designers to focus on implementation and design without having to obsess over perfect compliance. Compliance checks that catch mistakes allow innovation to move forward at a faster pace. Participants, especially the software developers, commented that the compliance integrated into the release process, the development of

compliance requirements, and the use of training and organizational policy all raise security awareness in a way that is not typically found in entry-level software developers. The fact that developers recognize the benefits both by affirming that a compliance-focused environment is valuable and by worrying over it when compliance checking is not present confirms that integrating compliance into the SDLC is necessary. Requirements engineering educators should incorporate communicating compliance concerns into their curriculum.

From the regulatory perspective, the fact that software developers are leaning into compliance checks and using the release process to catch mistakes is great news. Participant SD2 expressed the sentiment this way:

> **SD2:**"I would say that the biggest benefit is [that] the baseline for all developers is there for security reasons. . . Whereas in the past, without all of this auditing, the baseline developer didn't know as much about security, I would say, because it wasn't taken seriously, either because of compliance or regulatory reasons, or for contractual reasons. I feel like overall, people are much more aware of what the right thing to do [is] and the right way to do security is. And so, I think that's probably the biggest benefit otherwise."

A drawback to separate compliance checks might be a lack of ownership or responsibility for software quality when developers rely too much on the release process and compliance checking to catch mistakes. Some of our developers and engineers commented on how little they are involved in the security and compliance process. They know it is there, but they do not have an active role within the process:

> **SD4:**"So sometimes it just depends on the decision, whatever. It's like regulation stuff. I'm happy to let them deal with that and just tell me what to do. Because there's so much that I don't understand. And I really don't want to get into looking at laws and figuring out the best way to deal with this. So I'm more than happy to allow them to say, 'Hey, this needs to be done.' And I mean, they pay me to do what they tell me to do. So I'm more than happy to deal with that."

This sentiment, though understandable, is deeply problematic. Complex compliance concerns are probably not what drew participant SD4 to the profession, but engineers have a moral obligation know what their regulatory responsibilities are and why. The ACM's Code of Ethics [107] explicitly requires engineers to know and respect laws and regulations that pertain to professional work and responsibilities. This notion is taken so seriously in the Code of Ethics that engineers must recognize when there is "a compelling ethical justification" for not following local laws and regulations. This is included in the Code of Ethics because laws and regulations may have an "inadequate moral basis or cause recognizable harm." Relying extensively on process triggered compliance checks may encourage engineers to shirk their professional responsibilities in this area.

Expecting process-oriented compliance checking to catch all possible mistakes that can result in complex code is a recipe for failure. Compliance has to be more than just a checklist or a process, even if those things are both an important piece of the puzzle. **Lesson 4:** We must account for the organizational and cultural environment in addition to our own professional ethical responsibilities towards compliance. Building compliant

software requires a holistic organizational commitment that is more than the sum of the individual ethical decisions made by engineers and managers. Compliance must be both a part of the process and a value within a software development organization and the software industry as a whole.

Our participants identified a clear market incentive for organizations to achieve demonstrable compliance. **Lesson 5:** Pitching compliance to business analysts as an investment they can advertise to customers. Our participants believe that compliance establishes trust both within the organization and externally with the organization's customers. The goal of regulated economies is to incentivize and reward actions and behaviors perceived to be beneficial. Seeing evidence of this working for software developing organizations is reassuring. Six of the participants seemed to agree that visible evidence of compliance is valuable. They affirmed transparency as a means of achieving this by sharing statements about transparency with customers or with the industry as a whole. Some participants shared optative statements about how transparent they believe their organization should be.

> **D/PE3**"I think, whether it's Organization 1 or some of some of our peers, [the] big tech sector should be less afraid of talking about how we do things in the privacy and security space internally. I think that would do a lot for the media narrative for public trust, etc. There's like a lot of cool stuff that we do internally at Organization 1, that I often wish I could go talk to people and say, 'Hey, don't worry about that. We actually do this, this, and this, but we can't talk about it because of regulatory risk or legal questions or whatever.'

> But I think, you know, more transparency on our part would be good, because I think we're doing a lot of the things already that some people are wishing for. We just can't necessarily say it outright."

Transparency is beneficial. It can make compliance more obvious to the customer and make enforcement easier. Recall an earlier comment from a participant claiming there is little push-back for formal regulation 30–40% of the industry is already complying as an industry-standard. The benefits of transparency are not limited to more sensible regulation. One participant also connected transparency with their organization's ethical value system:

> **M/D1:**"Organization 3 is very transparent. We're one, I hate to say, one big family, but we really help one another, deliver those services, whether it's a doctor giving it directly to the patient, or us providing services to the doctor so they can take care of the patient. We all know the importance of adhering to regulation, compliance, security and privacy. It's all sort of ingrained in us as a value system."

M/D1's comments reflect our sixth lesson. **Lesson 6:** Software practitioners feel more comfortable in an environment that appreciates and incorporates compliance. Working with lawyers to get the requirements right is not enough. Compliance requirements must be communicated to software practitioners explicitly as part of the organization's compliance effort. M/D1's comments also reaffirm the earlier lesson of ethical values as part of organizational culture and the benefits of demonstrating a strong commitment to compliance. Aside from the regulatory and ethical benefits, transparency in practice

would make requirements engineering in these environments easier if only because requirements engineers could learn from public failures.

Transparency isn't a panacea. Many modern algorithms are so complex and adaptive, that complete transparency of all the data and processes involved could be totally overwhelming, particularly for regulatory agencies with few technical staff on the payroll. Worse, companies actively seeking to take unethical shortcuts will not be stopped by transparency and may even find it a useful smokescreen for their illicit efforts. To be clear, none of our participants even hinted at something like this, but we know that this happens in the real world. Volkswagen spent $14.7billion in the U.S. alone to settle their "Diesel Gate" scandal where they directed engineers to build a defeat device to bypass mandated emissions testing [7]. How much easier would it be to build a defeat device in other software system versus comply with annoying regulation? Therefore, there is a need for transparency, but scale it so it is relevant.

One way to achieve transparency at scale would be to commoditize components used in software systems. The potential benefits of transparency disappear as software becomes more complex and requires more dependencies. This effect could be mitigated if software dependencies were more clearly delineated and separated from the final, delivered product. Consider this comment from M/D5:

> **M/D5:**"The work that I do is focused on software supply chain transparency [aka Software Bill of Material (SBOM)], which is to say, all software's built on other software. How do we create both a good market expectation that people will track this information and will share it down the supply chain?

And what are the technical requirements that we need to do this from everything from data standards to how we share the data and execution side of things?"

Compliance could be greatly simplified if engineering components in the supply chain could be evaluated once, found to be compliant, and then used by consumer products. Imagine what might have happened if Zoom could have just used a known-good end-to-end encryption component that was already available in the supply chain. By normalizing and making supply chain transparency standard we can make regulatory and security standard compliance more effective and efficient:

M/D5:"So this is an unsolved problem still, is this idea of a software bill of materials. You can think of it as a list of ingredients for software. Right? So the concern isn't that the code that I'm giving you is bad, right? Because you know, there's some code from Organization 4. And then there's CVE[Common Vulnerabilities and Exposures number [17]] against it, we all know that. But if you're just buying my software [like] I'm selling 50,000 units of software to banks, well, then the concern is going to [just] be my software, but it's also going to be, 'Am I using a third-party library [from Organization 4] that has a known vulnerability?"

[17]Common Vulnerabilities and Exposures number is a uniquely identifying security number. The CVE repository of known cybersecurity vulnerabilities is managed through a partnership between the U.S. Department of Homeland Security, Cybersecurity and Infrastructure Security Agency, and the MITRE Corporation found at `https://cve.mitre.org/` or `https://www.cve.org/`.

M/D5 comments is our seventh lesson. **Lesson 7:** Fulfillment of compliance by third party libraries may require more formal inspection than is currently available. This concern is not just theoretical. Kula et al. recently found in a study examining 4,600 GitHub software projects and 2,700 library dependencies that 81.5% of these projects do not update dependencies with known security vulnerabilities in them [88]. Similarly, Zimmerman et al. found that poor maintenance causes developers to depend on vulnerable dependencies for years, even well after these vulnerabilities are made public [127]. Incorporating this sort of review alone into a compliance process would likely prove fruitful.

Fundamentally, being able to reason at an industrial level about standard, commoditized components is part of what separates an engineering discipline from personal tradecraft [123]. Someone building a treehouse in their backyard does not need the same level of regulatory scrutiny as someone building a twelve-story apartment complex. Analogous situations in software are not easily differentiated. A 17-line third-party library known as "left-pad" that was part of an open-source package formally known as "kik" was made available through NPM [59, 5]. Using the "left-pad" code may have been perfectly acceptable for personal projects. However, "left-pad" was incorporated into hundreds of professional projects through the javascript compiler Babel [59, 5]. When the developer who maintained the open source "kik" package deleted "kik" from NPM, they ultimately "broke the Internet" because of the "left-pad" dependency [59]. The removal of the "left-pad" is, to say, nothing of actively malicious packages [111] or fake GitHub repositories [16] designed to look legit and tempt coders to unknowingly download malware into their software, which is far more problematic. Software organizations could address

both problems more effectively if compliance processes required examining third-party library software usage.

As aspects of the regulatory compliance landscape become more defined, so does its relevance to the business processes for organizations operating within a regulated domain. Developing techniques and strategies to track and maintain regulatory compliance is becoming essential for organizations to attain their overall business goals. However, to help define and build the software industry around compliance, regulators and policymakers must clarify what compliance is to help meet the intent. This help can be handled in many different ways. One way is through normalizing regulated industry's best practices. Another is working with regulators to help build and model regulations.

Within academia, researchers can research regulation, see the gaps, and build tools or processes to fill in those gaps. Whether done individually or in combination, this assistance presents an opportunity (i.e., revenue) within a regulated domain for businesses and researchers alike. Building tools to assist with compliance or helping consumers build a program for compliance, raises the bar for compliance and addresses societal concerns. Business associates operating within the regulated domain have built businesses around this concept. These businesses are created as consulting firms to help understand the intent and the natural evolution of regulation and its requirements for compliance, assisting smaller companies that can't afford internal resources.

However, more can be done. Even though there are companies that make it a business to assist with compliance, these companies also try to avoid the requirements of compliance by shifting the responsibility onto other stakeholders, namely the customers. This compliance avoidance strategy is seen within examples of the participants' com-

ments as they describe their strategies for tracking and managing the regulatory compliance landscape within their companies. There is a drawback to this response, though. As we pointed out earlier, regulations are still in flux and evolving. This response, in its nature, is proactive and requires in-depth knowledge and foresight of the natural evolution of a particular regulation. In many ways, you are building tools and techniques for future requirements, which is considered a bleeding edge. These kinds of concepts and tools often require time and money to be assembled, and there is no guarantee that there will be a need for such devices or techniques. No guarantees or demand make developing these devices or techniques a high risk. These companies guess how regulation will change and evolve, creating a supply for a potential market. This risk or lack of guarantee is why smaller companies will "wait and see" how enforcement vets regulatory requirements. This approach, while understandable, is reactive and might be counterproductive in some respects because small businesses will either have to go through the expense of compliance so they can compete, define the line of responsibility so they do not have the cost or requirement of compliance, risk non-compliance (cut corners) (operate within the domain until the company get caught or are no longer in business because the company was acquired or stamped out for one reason or another) or combination of all—the inequalities in regulatory compliance burden small businesses trying to operate in regulated domains. Regulators see these burdens and do not want to stamp out the prospective companies, so they write flexibility into the laws and regulations to strike a balance. However, the gray area this flexibility creates still needs further investigation and guidance to navigate.

Governance and a framework to help guide and balance regulatory compliance inequalities while promoting better compliance practices are needed. Researchers are in a

position to assist. The later chapters of my dissertation are about promoting a tool and method to ensure and demonstrate regulatory compliance within a software organization.

3.5 Threats to Validity

Little academic research is currently available that represents the software industry perspective on regulatory and security standard compliance [85]. This interview study represents an opportunity to address this gap and help bridge "on the books" practices versus "practical on the job" practices within software engineering and development. Although interview studies are a well- established research technique in software engineering [78, 91, 73] and have been used to identify potential gaps between research and practice [30, 31], they are, however, not without limitations. This section briefly discusses this interview study's threats to validity and what steps I have taken to address them.

3.5.1 Threat to Internal Validity

One of the main concerns of the interview study is the application of experimenter bias in the findings. I used a grounded theory approach to mitigate experimenter bias in analyzing the interview study's data. I wanted the findings to reflect my participants' background and experience within the software industry, not mine. To ensure this, I piloted the interview protocol three times. I reported my pilot study findings to my advisor and peers for feedback and updates. While conducting the interviews, my advisor closely moderated me in all but one interview. I did have two different interview protocols, but I only used regulation and policy-oriented interview protocol in one interview because

of the participant's unique background in the software industry. The data analysis took several sessions, where I sought feedback from other committee members in formulating my findings.

Overall, I tried to ensure that this study focused on capturing an industry perspective on compliance and the findings reflect my participants' insight and not my own experiences.

3.5.2 Threat to External Validity

A threat to external validity is the population size. Fifteen software practitioners are not representative of the entire software industry. Compared to the number of subjects interviewed in similar interview studies from our related work, this study is on the low end by comparison. This study has 15 participants whereas similar research (i.e., 11 in Abdullah et al. [20], 15—this study, 28 in Haney and Lutters [68], 53 in Bamberger and Mulligan [30]) has double to triple the number of participants. I addressed this issue by taking the findings of the interview study and conducting further analysis on a larger subset of the software development community, using an online survey. Even with the survey, my work should not be read as validating or defining practices throughout the entire industry (cf., external validity).

Another external validity threat is that I am only providing a limited set of perspectives; therefore, there is a potential for selection bias. Most of our participants were either engineers or product managers. Therefore, I was able to get strong data regarding technical measures or organizational processes related to compliance. However, I was not able

to interview a comparable number of regulators or lawyers despite our protocol design and recruitment efforts. This is another limitation I wanted to address with the survey, because their perspective could prove crucial to better understanding the trade-offs being made at software organizations seeking to build compliance into their software systems and processes. Gathering more software stakeholders perspectives from industry may also help researchers better understand the regulatory enforcement or auditing process.

Another potential selection bias that could contribute to external threat is geographical location. All the participants for the interview study were geographically located in the United States. Two of the interview participants did indicate industry-related focus outside of the U.S. through descriptions of their job roles and experiences. Nevertheless, there is a potential for either cultural bias or industry-related focus for the compliance practices for only the U.S.

Finally, the education level of our participants. All but two of the interview participants held graduate level degrees. According to the U.S. Bureau of Labor Statistics, software developers need a bachelor's degree in computer, information technology, or a related field [108]. However, college degrees are not a requirement. Therefore, the reported level of education is a potential bias of the findings, based on the assumption that most of the software industry does not have an educational level past an undergrad degree from an accredited university. This limitation is something that the survey can potentially address by capturing respondents with different levels of education and comparing answers to the same questions.

3.5.3 Threat to Reliability Validity

This type of study is not novel. However, as pointed out in my literature review [61], there is little research available that focuses on the software industry's perspective on regulatory and security standard compliance and practices. Part of my contribution is adding to those studies that are available in the research field through the interview study and the survey (Chapter 4). In addition, the interview protocols are available at `https://doi.org/10.6084/m9.figshare.14842242.v1`, for anyone wanting to replicate this study.

3.5.4 Threat to Construct Validity

The threat to construct validity within this study is whether or not we had enough evidence to support the interview study findings. I covered 10 findings within this interview study and set a bar of 50% or more participants commenting on a theme within the context of the interviews. However, a larger sample size or different participants would provide more evidence that could contradict or support my claims from the interview study. Hence, the follow-on survey study is my approach to address this and other validity threats from the interview study.

3.6 Summary

In this interview study, we examine how regulatory, and security standard requirements are addressed in the software development process, how these techniques and procedures are perceived by engineers, managers, and directors, and how impactful regula-

tory change requirements can be. We interviewed 15 software engineering practitioners with different roles in the software engineering process across several industry domains. Our findings suggest that participants perceive the software release process to be the ultimate focus for regulatory compliance and security standard reviews. Most participants suggested that having a defined process for addressing regulatory and security requirements was freeing rather than burdensome. Participants generally saw these requirements as a valuable investment for both their organizations and their customers. However, our participants also pointed out that, whether valuable or not, changing regulatory requirements can be impactful for an organization's processes whether they are engineering or business processes. A regulatory compliance strategy to respond to these changes and internal communication of that strategy is essential for ensuring an organization remains compliant.

Some of these findings may seem counter intuitive at first glance. Why would an externally imposed regulatory requirement be "freeing" rather than restrictive? However, based on our participants' perspectives, companies operating within regulated environments need to monitor changes in the compliance landscape and have processes to track and manage regulatory and security standard compliance before release. An organizational release process can and does allow confidence and trust in the quality of large companies' products with regulators and the consumer. Organizations like Zoom may simply be outliers. After all, without the COVID-19 pandemic, their systemic failure to meaningfully address security, privacy, and regulatory requirements may have ultimately doomed them in the marketplace. Consumer trust is not easily rebuilt. Requirements engineers may take several lessons from this study. First, we should consider targeting

compliance requirements for the release process. Second, requirements engineers seeking to address compliance concerns must account for resource commitments in funding, time, and staffing. Third, we should learn from organizations that have failed to achieve compliance rather than waiting for a near miss in our own organization to take compliance requirements seriously. Fourth, requirements engineers must position compliance as an ethical value that must be an affirmed, supported part of organizational culture. Fifth, requirements engineers should pitch compliance to business analysts as an investment they can advertise to customers. Sixth, requirements engineers should communicate compliance concerns to practitioners because they feel more comfortable in an environment that appreciates and incorporates compliance. Seventh, fulfillment of software requirements by third party libraries may require more formal inspection than is currently conducted in practice.

Chapter 4

A Survey on the Perceptions on Regulatory Compliance in the Software Development Industry

The survey is the second part of my exploratory mixed method research design [101]. The survey examines the same problems, themes, and challenges as the data collected from the interview study on a larger subset of the software development community; however, I used the survey to anonymously collect data. The data analysis uses quantitative methods as opposed to the qualitative approach used in the interview study. It also addresses some threats to validity indicated in the interview study. Combining the data of these two studies (i.e., data triangulation), serves to validate my findings on the perceptions and practices of the software industry on regulatory and security standard compliance and answer the following sub-research questions (SQ) regarding the dissertation research question (i.e.,RQ1): What are the software industries perceptions regarding Regulatory and Security Standard Compliance?

SQ1: Who is responsible for the regulatory and security requirements in the software development process?

SQ2: When is compliance assessed within the software development phases?

SQ3: What factors give practitioners a confident perception of the compliance process in their organizations?

SQ4: What are the perceived difficulties in achieving regulatory compliance?

The motivation behind these questions came from the interview study's six sub-research questions (See Chapter 3). The survey's SQ 2-4 is a modification of the interview study's SQ 1-3. We did ask specific questions about regulatory compliance management, communication, and strategy to correspond to the interview study's SQ4-6 questions and findings. However, those questions were a combination of Open-ended and Likert Scale questions with no pattern or trend. Therefore, I decided to use the modified interview study's SQ1-3, with another add-on of SQ1.

The following subsection further discusses the methodology (Section 4.1) to include survey design (Section 4.1.1) and data collection and analysis (Section 4.1.2), recruitment and participant's demographics (Section 4.2), the four findings answering the sub-research questions (Section 4.3), the discussion of three takeaways derived from the findings (Section 4.4), and the four threats to validity (Section 4.5) for the survey.

4.1 Methodology

In this section, I discuss designing the survey (Section 4.1.1) and the methods used to organize and analyze the data (Section 4.1.2).

4.1.1 Survey Design

The ten findings from my previous interview study [84] produced seven lessons learned (See Section 3.4) for software organizations wanting to demonstrate efforts toward regulatory compliance. However, the sample size of the interview study was small.

Therefore, this limitation motivated me to test the findings of the interview study on a larger subset of the software industry through this survey study. I developed a comprehensive survey that included questions on a wide variety of issues raised in previous work. Through pilot testing of that initial version of the survey with 17 subjects, I found out quickly that this expanded version would be too long and would fail to attract a meaningful number of respondents. Therefore, I decided to focus on five areas:

- The Survey Participant's consent (i.e., Q1-3)
- The Survey Participant's personal and organizational demographics (i.e., Q4-17)
- Software Development Phases and Compliance Efforts (i.e., Q18 -25)
- Perceptions of the Organization's Compliance Practices (i.e., Q26-62; Q68-72), encompassing Compliance Culture, Internal Compliance Program, Communications, and Strategies in dealing with Compliance
- Regulatory Compliance Governance (i.e., Q63-67)

I focused on these areas based on the comments from the interview study. Specifically:

- Questions 1-3 captured survey respondents' consent per the approved IRB protocol.
- Questions 4-17 were based on the demographics data we collected from the Interview Study participants.
- Questions 18-25 to answer where compliance is assessed and addressed within the Software Development Phases.

- Questions 26-72 are based on the intent of the study regarding the perceptions of compliance; however, I scoped it to organizational compliance to give the study focus.

- Q63-67, because we got good feedback on Regulatory Governance from the interview study, so I wanted to explore it more from a software practitioners perspective. Also, it connects Part one of the dissertation with Part two and three.

The listing of the survey question is in in Appendix B. The final version of the survey is available in the artifact repository at `https://doi.org/10.6084/m9.figshare.25078061`

4.1.2 Data Collection and Analysis

I hosted the survey on the Qualtrics XM survey platform [1]. To facilitate anonymity, I used an URL link that only participants can click on to access the survey. I collected responses from January 20 to December 25, 2022, recruiting participants through online professional and social media platforms. Although 110 people tapped the link to the survey, only 42 people hit the Submit button, and one of those respondents marked "No" for the survey consent. According to the approved IRB protocol, I could only analyze the 41 survey respondents who consented and submitted the survey.

To guide my analysis, I relied on our four research questions to focus our analysis. For Sub-research Question 1 (that is, "Who is responsible, according to our survey respon-

[1] Qualtric XM is a survey platform used to host and collect survey data to aid in academic research https://www.qualtrics.com/

dents?"), I focused on Question 26 within the survey, "Who do you think is responsible for ensuring compliance with regulatory and security standard compliance within your organization?" I used frequent item sets to measure which groups (see Figure 4.4) had the highest count and to discover if there was a grouping pattern within the survey responses.

For the other two sub-research questions, I wanted to see what survey items impacted whether a respondent had a confident or not confident opinion about their organization's compliance. First, I classify my pool of respondents into those with a generally confident view of their organization's compliance practices and those without such a confident view. To do this, I used "K-means clustering" with RapidMiner. K-means clustering is an unsupervised machine-learning technique that groups similar answers into a predetermined number of clusters [15, 18]. The K-means algorithm randomly chooses a center or centroid for each cluster. Then, K-means assigns every data point in the data set to the nearest centroid. Once all data points are assigned, K-means calculates the average for all data points in that cluster. The average becomes the new centroid of that cluster, and the K-means algorithm will reassign the data points to the closest centroid. K-means clustering will continue to loop through calculating new centroids and reassigning data points until the centroids no longer move or the predefined maximum repetitions of the center reassignment have been reached [18]. In RapidMiner, the default is 100 repetitions [15].

I used the K-means clustering algorithm to define two groups of respondents. I grouped the respondents as having a 'confident' or 'not confident' perception of their organization's compliance practices and processes. I used the survey respondents' answers to the survey items shown in Table 4.2 to form the clusters. I used these items for the

grouping for the following reasons:

- They were indicative of the respondents' perceptions of their organization's compliance programs and policies (as opposed to other outside factors).
- The tone of the statement was unambiguously, either confident or not confident.
- These statements had the least amount of missing data that required replacement.

I define a "confident" perception as a respondent that rates high (i.e., a rating of 3.4 or more overall) to the following attributes:

- A respondent viewed their organization as prioritizing compliance within software development (i.e., Q33-34 in Table 4.2).
- A respondent viewed their organization's compliance process and practices as an investment that benefits the organization (i.e., Q48 - Q51 in Table 4.2)
- A respondent rated high (i.e., 4 or 5) regarding confidence in the quality of their products because of their compliance process and practices (i.e., Q50 in Table 4.2).

These attributes also align with some of the findings and discussions from the interview study, where most participants reported high confidence in their organization's practices and procedures. Conversely, I define 'non confident' perception as one where the respondents gave a rating of below 3.4 to the questions noted for the above attributes.

Before clustering, I needed to clean the survey data. First, I converted the Likert scale responses to a numerical scale from 1 to 5, with one associated with Strongly disagree and five being Strongly agree. Second, I did not require survey respondents to

answer every question. They had the option not to answer any of the compliance questions, which left a few responses missing. I could have excluded these answers, but that would have eliminated over a third of my participants' responses, further reducing my small survey sample size. Therefore, I opted to fill in missing responses with the median of other responses for a statement or question asked in the survey. Lastly, most questions or statements had a generally confident tone, but Q35 had an opposite tone. (Q35: When resources are tight (i.e., limited staffing, time, or money), the compliance assessment process is the first thing to change.). Therefore, I reversed the scaling of the responses to that item before analysis so that a rating of 5 corresponded to Strongly Disagree and 1 to Strongly Agree.

After I ran the survey responses through Rapidminer using the K-Means Clustering algorithm, I had twenty-six respondents (63.42%) classified in the confident group and 15 (38.58%) in the not confident group. The K-mean clustering analysis revealed that 3.4 was the dividing line between confident and not confident clusters based on a respondent's overall average response to the survey items in Table 4.2. I then compared open-ended responses and overall average numerical scores to verify consistency in the grouping of respondents concerning their confident or not confident perspective on their organization's compliance program or process. Once grouped into confident and not confident perception clusters, I use two ways to analyze the data, given the small sample size. The goal is to determine which other survey responses are impactful in creating a confident or not confident perception for respondents concerning their organization's compliance practices. One is to use logistic regression with responses to the survey items as attributes and focus on those attributes with p-values below 0.05 and with high coef-

ficients compared to the other survey items. The other is to compare the means of the responses to each survey item among respondents with overall confident and overall not confident perceptions. I performed both of these analyses and compared the results for two reasons. First, comparing and validating findings using two different methods (i.e., findings triangulation) can increase the confidence of findings. Second, different methods have their strengths and weaknesses. Therefore, using two different methods can uncover patterns that might be missed by only using one method of data analysis [101].

I was unable to use all the questions within the survey for logistic regression analysis and statistical mean comparison. As explained previously, some of the questions or statements in the survey had been used for the K-means clustering to create the confident or not confident classification of the respondents' perception regarding their organizations compliance status. Fourteen questions were open-ended questions (See Table B.1 in Appendix B); therefore, they could not be used within a quantitative analysis such as logistic regression or statistical means. A third or more of the respondents did not answer certain survey items [2]; therefore, replacing answers with median or statistical mean (i.e., average) might present a research bias, despite the common practice of imputation [130]. Finally, the format of Multiple Choice (MC), Ordered, or Mark if Applied survey item could not be analyzed using logistic regression. For these reasons, I excluded answers to those survey items. Survey Items not used in the analysis are marked No in Table B.1 in Appendix B; survey items used in the analysis are marked Yes, FI, or cluster. Open-ended questions were qualitatively analyzed to support and provide depth to the discussion of the results. For the quantitative analysis, I focused on scaled questions for the logistic regression and

[2]Figure 4.1, Survey Items Answered, depicts the survey items answered by the respondents

the means analysis where I asked the respondents to rate:

- Q19-25:Their organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspects of software development (See Table 4.1 and Figure 4.5).
- Q56-61: Their agreement with statements on their organization's communication and management of Regulatory and Security Standard compliance (see Table 4.3).
- Q68-72:Their agreement with statements on their organization's strategy to regulatory requirements and change (see Table 4.4).
- Q63-67:Their agreement with statements on their perceptions of compliance governance (see Table 4.5).

I use the 'confident' or 'not confident' group labeling as the outcome variable to produce metrics to assess the responses of the respondents. For logistic regression, the metrics analyzed were the p-value and the coefficients of the survey items. Attributes (that is, survey items) that had low p-values (that is, below 0.05) and a high coefficient (anything above 1.0) were of interest. I use Data Tab to produce the logistic regression results.

The second analysis involved calculating arithmetic means on the answers to any of the survey's items by their K-means cluster grouping and determining the relative distance between the average answer by cluster groups for any of the survey items. Algorithm 1 outlines the steps on how I calculated statistical means for each survey item (i.e., Step 1-7) [3] and the difference between the clusters (i.e., Step 8).

[3]Steps 1-7 in Algorithm 1 is the equivalent of MS Excels AVERAGEIFS function, where the range is the survey respondent's answers to a survey item. Criteria 1 is the Survey Item; Criteria 2 is K-means Cluster

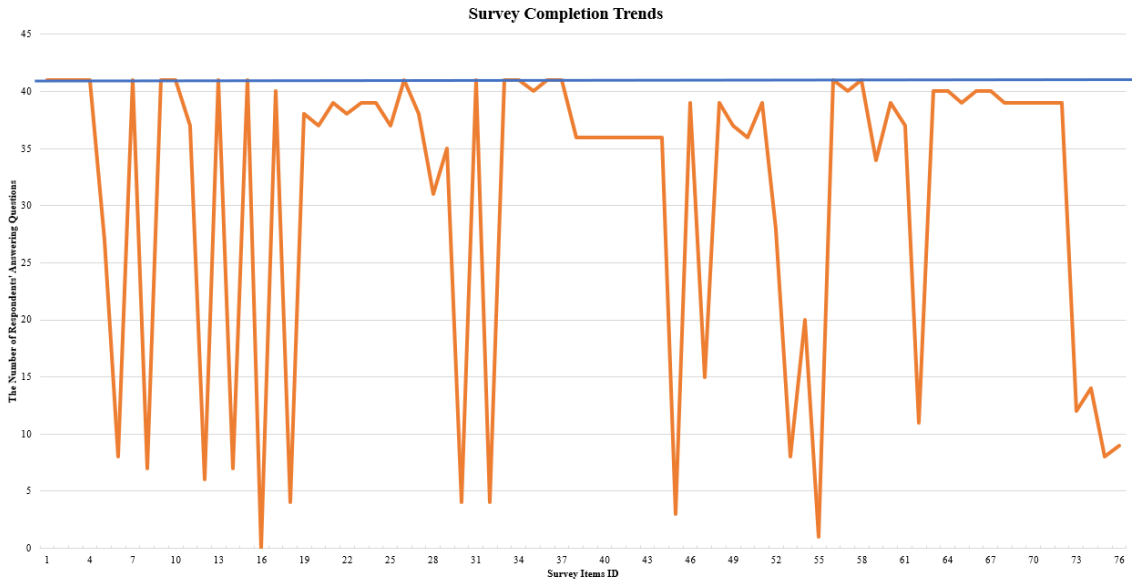

Figure 4.1: Number of Respondents that answered a Survey Item

I examined the difference between the means for each survey item to see how close or far apart the cluster centers were by survey item. I excluded survey items labeled Q33-37 and Q48-51, because these items were used to create the cluster groups and they were needed to define the threshold of difference for the arithmetic mean analysis.

To set a threshold to define meaningful large differences, I calculated the average rating of the survey items used to form the cluster groups (i.e., Q33-37 and Q48-51) by their grouping label of confident (4.80) and not confident (3.13). The difference between these averages is 1.67. I considered any survey item with a difference (See Step 8 in Algorithm 1) of 1.67 or greater a good separation. To set a threshold to define meaningful small differences (i.e., agreement amongst all the respondents), I subtracted the respondent who had the lowest average rating (i.e., The average rating for this survey respondent is 3.44) over all survey items used in the cluster grouping (i.e., Q33-37 and Q48-51) and

grouping of the Survey Respondent (i.e. Confident or Not Confident Cluster Group)

Algorithm 1: Calculate Mean for Survey Item by K-mean Cluster Group

```
1  SurveyItem (any survey item with a numerical rating of 1 to 5) begin
2      if the Answer to the SurveyItem's Cluster Group is "Confident" then
3          place it in a list Q[SurveyItem].Confident[ ];
4      else
5          place it in a list Q[SurveyItem].NotConfident[ ];
6  Confident Mean = Q[SurveyItem].Confident[ ].mean();
7  NotConfident Mean = Q[SurveyItem].NotConfident[ ].mean();
8  Difference = abs(Confident Mean  - NotConfident Mean);
```

Figure 4.2: Calculating the Arithmetic Means

the respondent with the highest average rating within the not confident group (i.e., 3.33). The difference is 0.11. Therefore, I considered anything with a difference of 0.11 or lower a small meaningful separation.

By reviewing the coefficient and p-value results from the logistic regression and the difference in the two groups' arithmetic means any survey item, I gleaned some evidence of what factors influence software practitioners' perceptions regarding compliance, on whether it was confident or a not confident perception.

4.2 Recruitment and Participant's Demographics

This section covers the recruitment procedures and the reported survey respondents' demographics.

4.2.1 Survey Recruitment

The participants of the survey were recruited over the course of 11 months (i.e., 20 January — 25 December 2022) through personal contacts, specialty mailing list, social media platforms, and online meeting forums. I targeted platforms where the audience is software developers or project managers currently operating in the software industry, privacy or security engineers managing compliance issues, or legal representatives that consult on matters of regulatory compliance. Due to COVID-19 restrictions, I was limited to online forums for recruitment. Although 110 people tapped the link to the survey, I could only analyze the 41 survey respondents based on consent per the IRB guidance (See Figure 4.1).

4.2.2 Survey Respondent's reported Personal and Organizational Demographics

To capture the respondents' personal demographics, I presented multiple choice questions for years of experience in industry and in their current job, level of education, and current and past work roles (see list below). An overview of the participant's and their organization's demographics can be seen in Figure 4.3. For job roles, I gave the respondents six options:

- **Privacy Manager or Engineers:** People who interpret privacy requirements for technical implementation.
- **Software Developers:** Developers that build and maintain software products.

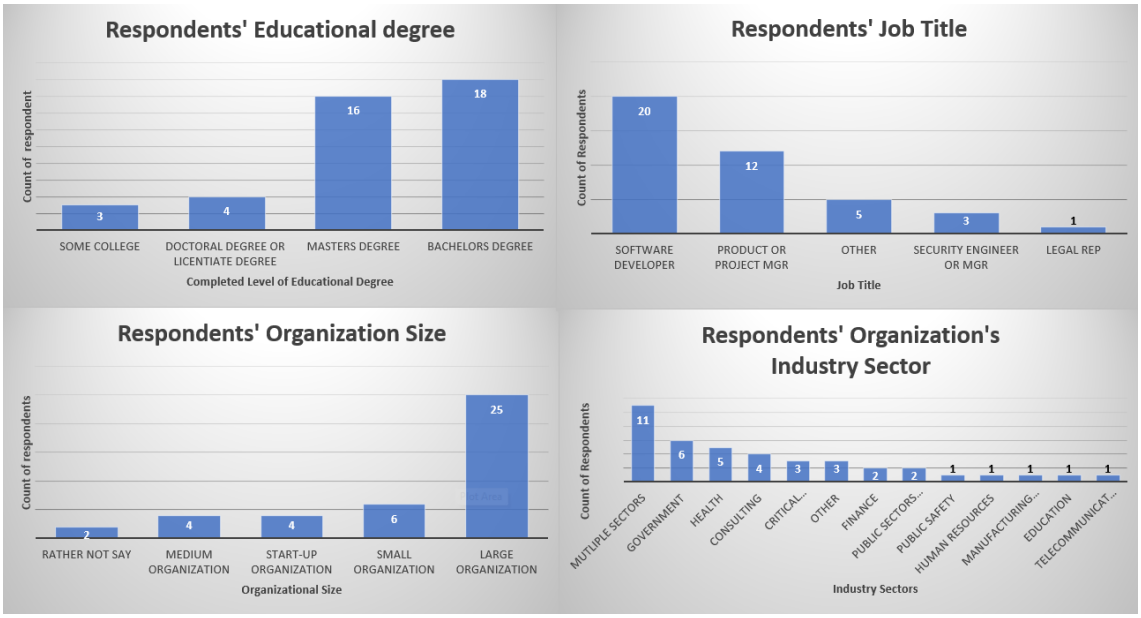

Figure 4.3: Survey Respondent's reported demographics

- **Project or Product Manager:** People who work with software developers and direct, coordinate, or track software development requirements.

- **Security Manager or Engineer:** People who interpret security requirements for technical implementations.

- **Legal representative (e.g. lawyers):** people who interpret legal requirements for their organization to fulfill technical regulatory requirements.

- **Other Software-Related Job:** People who are in software-related jobs not listed in the survey.

For organizational demographics, I wanted to know what industry sectors their organization operated in and the size of the organization. I presented 13 categories for industry sectors: Finance, Public Sector, Healthcare, Government, Construction, Public

Safety, Critical Infrastructure, Education, Manufacturing and Development, Human Resources, Telecommunications, Consulting, and Other.

For organizational size, I describe the size based on the structure of development teams and resources available to the software development process as also defined in the interview study.

- **Start-up:** New or recently created organization with development activity focused on a single activity, product, or service. This organization has limited internal compliance resources.
- **Small:** A single development team and no internal compliance resources available, such as a separate quality assurance, security, or testing team.
- **Medium:** One to three development teams for different products with an internal security/compliance team.
- **Large:** Multiple development teams for a single product (that is, a team dedicated to GUI, another to chat messaging, and so on) and separate internal compliance resources (i.e., governance, risk, and compliance team; security team; and testing team).

In general, industry experience ranged from 1 to 50 years with a mean of 18.02 years (median of 20 years). For jobs, 20 respondents identified themselves as software developers, 12 as product or project managers, three as security engineers or managers, one legal person, and five others (i.e., CTO, Owner, Research Lead, Data Officer, and one left blank) (See Figure 1: Respondents' Job Titles). The Privacy Manager role was selected

four times, but in conjunction to other job titles. In other words, the respondent selected one of the five job titles seen in Figure 1 plus Privacy Manager in four cases (e.g., Security Manager/Engineer and Privacy Manager/Engineer or Developer, Product Manager, and Privacy Manager/Engineer).

For organizational demographics, respondents identified their organization in the following sectors: Consulting - 4; Critical Infrastructure -3, Finance - 2, Government - 6, Health - 5, Public Safety - 1, Human Resources - 1, Manufacturing and Development - 1, Education - 1, Public Sectors (non-govt or non-profit) - 2, Telecommunication -1 and Other -3 (that is, Insurance, IT, Software Services). Eleven respondents listed multiple sectors (See Figure 4.3: Respondents' Organization's Industry Sectors).

Regarding size, 25 respondents were from large organizations, four from Medium, six from Small, four start-ups, and two marked "Rather not say"(See Figure 4.3: Respondents' Organization Size). In general, the respondents' varied in background, domain, and level of experience, giving a small but diverse selection of survey respondents.

4.3 Findings

This section presents the findings from the survey analysis using the methods outlined in Section 4.1 to answer the four sub-research questions.

SQ1: Who is responsible for the regulatory and security requirements in the software development process?

SQ2: When is compliance assessed within the software development phases?

SQ3: What factors give practitioners a 'confident' perception of the compliance process

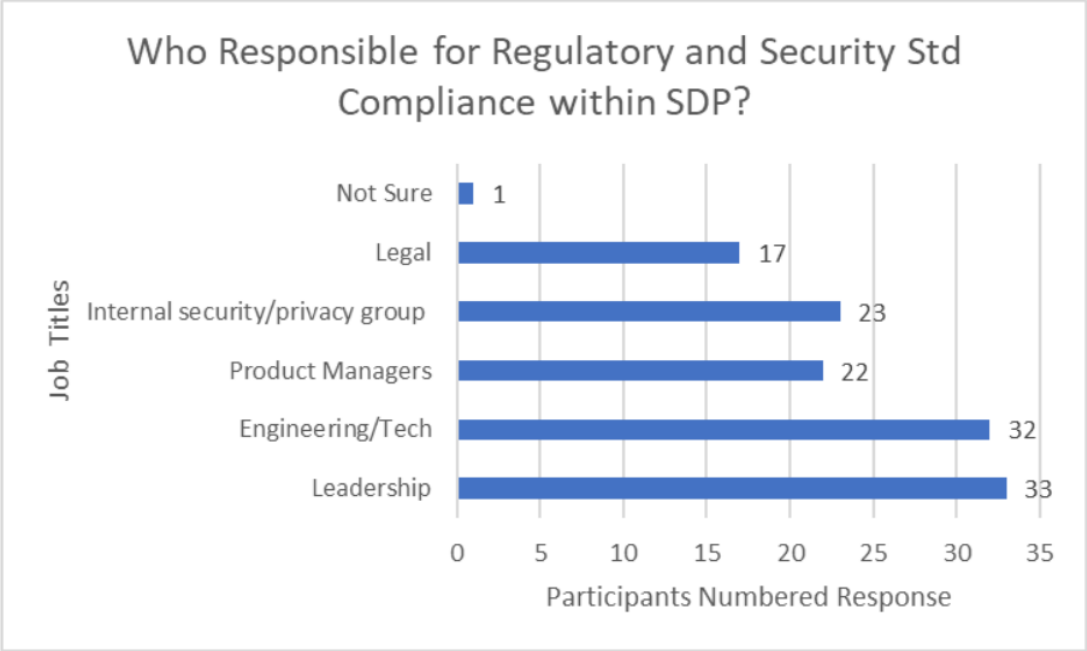

Figure 4.4: Responsibility By Count

in their organizations?

SQ4: What are the perceived difficulties in achieving regulatory compliance?

4.3.1 Compliance Responsibility

SQ1: Who is responsible for the regulatory and security requirements in the software development process? **Finding 1:** It is a shared responsibility amongst multiple stakeholder groups within the software development process, according to a majority of survey respondents. The top two groups by count were Organizational Leadership and Engineers & Technology people.

I asked the survey respondents: Who do you think is responsible for ensuring adherence to both regulatory and security standard compliance within your organization? I

gave the survey respondents six options and allowed them to select multiple options. The number one group selected was the organization's leadership; Engineers and Technology was the second highest (see Figure 4.4). However, only 10 respondents singled out one group. Thirty-one of the respondents chose multiple groups, 14 selecting everyone, and 3 selecting everyone but Legal. Seventy-five percent of the respondents believe that regulatory and security compliance is a shared responsibility with leadership. 65% indicating a shared responsibility with Engineers and Technology people.

4.3.2 Compliance throughout the Software Development Process

SQ2: When is compliance assessed and applied within the software development phases? **Finding 2:** When applied, efforts to comply to regulation is seen throughout the entire software development process.

I asked the respondents to rate their organization's efforts regarding regulatory and security standard compliance specifically in the seven-phase software development process (SDP) of Planning, Requirement, Design, Implementation, Testing, Deployment/Release of Software, and Maintenance. Using these phase-specific responses as attributes, a logistic regression model showed that no one phase was statistically significant, meaning that no particular phase was especially impactful in predicting whether a respondent's perceptions were confident or not confident in general about their organization's compliance efforts (see Table 4.1). I saw that the confident group reported an average higher than 4 or "A Lot of Effort" regarding compliance in all phases of software development. The not confident group averaged within the "Moderate Effort" range (i.e., 2.5 to 3.06: See

Table 4.1: Logistic Regression & Mean for SDP compliance efforts

ID	SDP	Coeff	Std Er	P	Mean (CI)	Mean (NCI)	Diff
	Intercept	7.25	2.77	0.009			
Q19	Planning	-0.14	0.66	0.827	4.08	2.53	1.55
Q20	Requirements	-0.02	0.84	0.983	4.23	2.73	1.50
Q21	Design	-0.79	0.65	0.219	4.08	2.73	1.35
Q22	Implementation	-0.35	0.67	0.601	4.35	3.07	1.28
Q23	Testing	-0.24	0.82	0.770	4.27	2.73	1.54
Q24	Deploy/Re	-0.16	0.74	0.828	4.15	2.67	1.48
Q25	Maintenance	-0.50	0.77	0.519	4.08	2.93	1.15

Figure 4.5). I took this analysis one step further. I analyzed each phase within its own logistic regression model to see if it predicted the respondents' 'confident' or 'not confident' perception regarding their organization's compliance. Every phase of the software development process when reviewed by itself (i.e., One phases responses), gave a p-value less than 0.05. It indicates that the efforts of compliance when assessed individually by software development phase has an impact on the respondents' perceptions.

I then examined the mean differences between responses for each phase in the confident and not confident clusters and none had a difference greater than 1.67, the threshold identified in Section 4.1.2. But, when I looked at the averages, I saw a uniform application of efforts of compliance within the confident and not confident groups (see Table 4.1 and Figure 4.5).

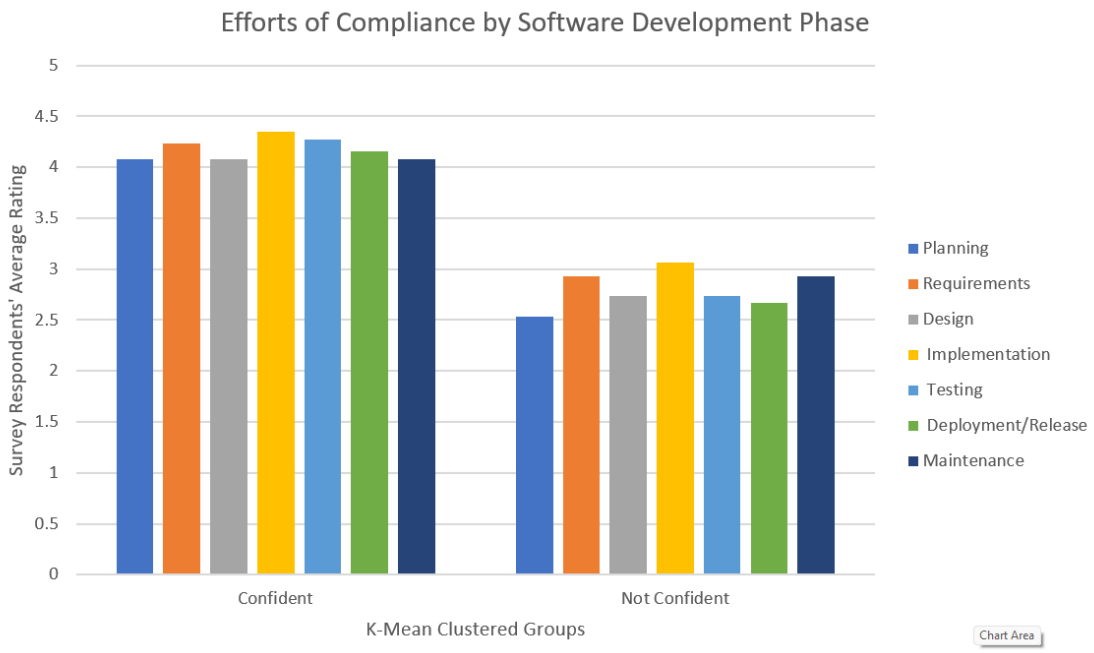

Figure 4.5: Survey respondents' average ratings of Compliance Efforts by Software Development Phase grouped by Perceptions

This analysis shows that software professionals who view their organization's compliance in a 'confident' light, do not see it in just a portion of their software development phases, but emphasized across all phases of software development. Even the 'not confident' grouping sees some efforts of compliance throughout the entire software development process. Compliance efforts appear to be at their lowest at the beginning of the planning phase, where four of the 15 respondents rated the planning as one or "without effort". Only two respondents rated ones across all the software development phases. Thus, the 11 respondents within this group were much more likely to rate a two (i.e., A little effort) or a three (i.e., Moderate effort) across the software development phases.

4.3.3 Contributions to confident opinions of compliance efforts

SQ3: What factors give practitioners a confident opinions of the compliance process in their organizations? **Finding 3:** An overall culture that promotes compliance and upfront regulatory analysis at the beginning of the software development process.

To get an idea as to why the respondents have certain opinions of their organization's compliance, I asked the survey respondents to rate their agreement to 11 statements organized under two categories of their Organization's Compliance Management and Communication (see Table 4.3) and their Organization's Compliance Strategy (see Table 4.4). These 11 items were not used in the cluster analysis that formed the confident and not confident respondent groups.

In general, the survey respondents who had a 'confident' opinions on compliance management in their organization were much more likely to agree, compared to the respondents in the 'not confident' group, that "compliance was ingrained in the culture of their organization". The coefficient for this survey item in the logistic regression model was 2.36, with a p-value of 0.042 (see Table 4.3). Furthermore, the difference in means between the two categories for this item is 1.98 (that is, the 'confident' (CI) mean 4.58 and the 'not confident' (NCI) mean 2.6). Within the organization's compliance strategy, the statement "My organization spends a lot of time upfront analyzing and understanding the regulatory requirement, which makes the design and implementation of a response straightforward" is statistically significantly more supported in the confident group of respondents, according to the results of the logistic regression model (See Table 4.4). Additionally, the difference in means between the two categories for this item is 2.15 (that

is, confident (CI) mean 4.08 and not confident (NCI) mean 1.93). These findings indicate that confident opinions of compliance efforts are seen in the organization's culture. Furthermore, incorporating compliance is not something done at the end of the development process, but is analyzed and planned from the start of the process and carried out throughout the SDP (Reference Section 4.3.2 and Table 4.1).

4.3.4 Perceived Compliance Difficulties

SQ4: What are the perceived difficulties in achieving regulatory compliance? **Finding 4:** While compliance audits are necessary, enforcement needs better tools and guidance for effectiveness.

Part of understanding regulatory compliance is looking at its governance or enforcement from the software industry point of view (i.e., beyond the organization's practices). Therefore, I asked the respondents to register their agreement with five statements on regulatory compliance governance. I then analyzed these responses in the same way as above, by comparing (via logistic regression and by comparing means) the responses from confident and not confident group. The statement "The best regulations are based on already established industry best practices" was associated with a significant p value in the regression analysis (see Table 4.5), but had a relatively small difference (0.84, where the confident mean 3.77 and the not confident mean 2.93) in means.

A closer look at the responses showed an 80.5% agreement (i.e., 19 respondents 'Somewhat agreed' and 14 'Strongly agreed'); six gave the neutral response of 'Neither agree or disagree' (i.e., three in confident and three in not confident) and two 'Strongly

disagreed' (i.e., one from 'confident' and one 'not confident') to the statement "Compliance audits are necessary but could be better tooled for compliance enforcement within the software industry.".

These reported findings seen in the Results section are consistent with some of the previously reported takeaways from our interview study [84]. In our next section, I further discuss these results and takeaways for researchers, and software practitioners.

4.4 Discussion

In this section, I highlight three key discussion points of this work for software developers. First, the respondents indicated that compliance should not be siloed within any one phase of the software development process nor is it the responsibility of any one group, but a shared responsibility within an organization. The second discussion point is about the importance of accounting for the organizational and cultural environment when trying to understand an organization's compliance and how well they may or may not be doing. Regulation-compliant software organizations require a holistic and ingrained commitment to complying with regulatory and security standards. Third, I expand on the widespread agreement among the respondents with the statement "Compliance audits are necessary but could be better tooled for compliance enforcement within the software industry" and what they could mean for requirements engineers, regulators, and software developers.

4.4.1 Do not silo regulatory compliance in software development.

The results on "who holds responsibility for compliance" and "when compliance activities are applied", indicate that regulatory compliance in practice is not something that can be siloed. A silo mentality means that groups or departments work in isolation. They do not communicate or cooperate with each other or try to understand what other departments or sections are doing within an organization. For regulatory compliance, a silo mentality in which one department or group assumes the responsibility of "regulatory compliance", leaving others free of responsibility might seem like a relief from the compliance burden. However, it is a fundamentally limited approach that does not scale well depending on the organization, the project, and the stakeholders involved. Interpreting and understanding regulations within larger organizations or many stakeholders requires communication amongst groups such as leadership, legal, or the customer (i.e., for whom you are developing the software). As a shared responsibility, regulatory compliance requires everyone to weigh in and do their part to meet or exceed the regulatory requirements. This also means not just addressing regulatory compliance at the end of the software development process within "Testing" or before a software product is released to production, or even just at the beginning as part of requirements analysis. Organizations that want a confident perception of their organization's compliance should address and prioritize it throughout all phases of software development.

This may require a significant investment in compliance resources and everyone accepting their piece of the shared compliance responsibility to "bake-in" compliance within software development. For those who might think baking in compliance will slow

everything down, one of the respondents made this point.

> "We have internal regulatory and security standard compliance teams that incorporate their efforts into our development process from the very beginning, and this allows us to move quickly and abide by regulation"

Software organizations that continue to silo or only address compliance at the end of the software development process put much at risk. In the end, they will probably spend much more time, money, and resources addressing problems that arise from non-compliance with laws and regulations versus taking the time to plan for compliance resources at the beginning of a software development process [79].

4.4.2 An organization's culture of compliance.

Not siloing compliance, but rather "baking it" into the software development process requires more than throwing resources at it. It requires a holistic, cultural, and ingrained commitment toward regulatory and security standard compliance seen within the organization's culture and communicated externally. This result is neither surprising nor counter intuitive. Checklist or going through compliance motions might seem like a way to achieve what is called "bare minimum compliance to standards," but what is lacking is an understanding of the intent behind the law. It is complying with the "letter of law", without regard to why the law exists in the first place. This compliance approach, while technically legal, does not provide much forward movement to improve an organization's compliance process without first some action (i.e., an incident or a contractual obligation) that initiates an update.

Organizations that take a holistic approach to compliance and promote a culture of compliance have strategies to adopt emerging industry standards and when regulatory changes occur within the landscape of a supported domain. These strategies are known throughout the organization and are outwardly communicated to project a confident compliant image of the organization. Consider, one respondent's comment:

> "We stay on top of emerging changes to stay prepared and prevent surprises. Security is the job one and is embedded in all activities. Regulatory changes often lag where we are."

Another respondent's comment on their organization's communication on compliance topics:

> "Our organization has a one-stop portal where we can check the status or any kind of compliance issue. In case of any compliant issues, we would [receive] mail from the concerned teams to act upon it as soon as possible."

The same respondent mentions in a later question that further highlights compliance communications:

> "All the concerned employees in the organization will automatically be notified about the regulatory and security standards. We will be taken through seminars, courses, and sessions to cover and explain."

Some of the implications for software developers are reiterations of some of the lessons learned from the previous interview study [84].

- Incorporating and accounting for compliance throughout all phases of the software development process when planning software systems.
- Accounting for the organizational and cultural environment in addition to personal ethical responsibilities.

A software organization that has a culture of compliance and "bakes" regulatory compliance and security standards throughout their software development process promotes confident organizational images and confidence in the software products produced. Software organizations can leverage that confidence outward to their consumers and the industry as a whole, which could see a return on their investment.

4.4.3 Compliance audits are necessary but could be better supported with improved tooling for enforcement.

[4] This statement has some history behind it. I asked this question within the survey based on comments from the interviewees from the previous interview study [84]:

> "Compliance is necessary but not sufficient"
>
> "[Compliance] is valuable for building trust and making sure that the actual things we do help the people we intend to help.... But maybe if it were written in a little bit more plain English, it could be a little bit more understandable."

I found similar comments within the survey's open responses, such as regulatory compliance being a "field of voodoo" or another that describes their start-up experience

[4]This statement was modified from the survey statement for the context of the discussion.

going through a certification process.

> "Going through the certification process for a large pharma company. We have multiple 300+ row spreadsheets with confusing questions to fill out. We have been slowly slogging through them section by section and then updating our policies (or simply writing policies) to comply. It is a time-consuming, confusing, and frustrating process."

These statements talk about how regulations and regulatory compliance are not straightforward. It can be overwhelming and confusing, especially for those who do not have access to expertise in a particular regulated field or who do not have the resources or strategy to deal with change in regulatory compliance. Regulators and auditors are dealing with similar issues. Regulators are trying to understand the most concerning issues facing a particular industry, but at the same time, they do not want to be restrictive or promote a "one-size fits all" solution toward regulatory or security practices. Auditors or enforcement agencies are having to navigate these uncertain waters trying to figure out who is compliant, who is trying to be compliant, and who is knowingly noncompliant. Researchers can assist both organizations and regulators in having a common understanding and communicating of what a good process toward compliance might look like. Follow-up research can help promote understanding and communication that can translate into better enforcement of regulatory compliance.

4.5 Threats to Validity

This section discusses threats to the validity of my findings.

4.5.1 Threat to Internal Validity

The use of logistic regression to analyze the data can potentially infer a false causality of some survey items when assessed together. It is due to this validity concern that I individually evaluated each survey question used as a predictor or attribute within the logistic regression using the difference of the means for the two groups. Looking at the two groups' (i.e., confident clustered group versus the not confident clustered group) centers and seeing where the differences are at the highest and lowest, I can see where the cluster has overlap versus separation. Where the difference was around two (i.e., 1.67 or more), I was able to support the low p-value and high coefficient impact for a specific question and further discuss the finding in relation to the interview study and as part of the survey analysis. Where the center differences are less than 0.11, I was able to take a closer look at a specific question and the raw answers to see if there was general agreement or disagreement with the statement. I found a question where 80% of the respondents agreed with the statement and presented this finding within the results. Thus, the findings of the logistic regression analysis were triangulated by the differences in means analysis, bolstering the validity of both.

4.5.2 Threat to External Validity

I am limited in generalizing the findings from this study due to the sample size of the responses. 110 people viewed the survey, but only 41 people submitted their survey responses for analysis. Therefore, I was only able to analyze 41 software practitioners' responses to the survey. One of the goals of this survey was to cast a wider net and com-

pare the results from the previous interview study to triangulate and validate the results. It is a larger sample group compared to the interview study [84], but still rather small. Compared to the interview study, I see that some of the lessons learned from the interview study are worthwhile. I was able to add support for the importance of organizational culture as it relates to regulatory compliance and ensuring compliance requirements are part of the more formalized software development process (SDP) from start to finish. A software development process that bakes in compliance and can demonstrate an overall ethical commitment toward compliance outside the organization, will project a compliant image internally to the employees and externally to auditors who check for these things.

4.5.3 Threat to Reliability Validity

This type of survey and analysis can only improve through replication and the collection of more data. Therefore, I have made available survey questions and invite anyone to replicate this study.

4.5.4 Threat to Construct Validity

One of the struggles I had with the analysis was finding and using evaluation techniques to analyze the small data set. To mitigate this, I used techniques to categorize the data into two groups based on the hierarchical scaling of the questions within the survey. Then I analyze the rest of the hierarchical scale questions based on grouping using two statistical techniques (i.e., logistic regression and comparing the difference of the response means) that work for small datasets. Both techniques have validity issues, but by

comparing the results of the techniques and finding where there is agreement, I am able to triangulate findings in agreement from different statistical angles. Repeating the study with more people from a wider selection of backgrounds and industries would increase our sample size and accuracy of our analysis so we could use more robust quantitative and qualitative methods.

4.6 Summary

In this chapter, I examine how software practitioners perceive the regulatory and security standards compliance process within their organization. I grouped the respondents into two categories, confident and not confident opinions. I analyzed and compared logistic regression output of coefficient and p-values and the difference between arithmetic means to determine which survey items most affect the respondents' not confident or confident opinions of their organization's compliance practices. What I found was that software organizations that have a culture of compliance and integrate (i.e., "bake") regulatory compliance and security standards throughout their software development process promote confident organizational images and confidence in the software products produced. Software organizations can leverage that confidence outward to their consumers and the industry as a whole, which could see a return on their investment. However, these compliance efforts must be demonstrated throughout all phases of the software development process. As for who is responsible for compliance, no one group should have the sole responsibility. A silo mentality, that one group has sole responsibility, does not work because interpreting regulations and understanding them requires communication

between many stakeholder groups such as leadership, legal, or the consumer (that is, for whom you are developing the software). Not having a "silo mentality" also means not just addressing regulatory compliance at the end of the software development process within "Testing" or before a software is released to production or at the beginning as part of the requirements analysis. It requires effort and support throughout the entire software development process. Organizations that want a confident opinion of their organization's compliance should address and prioritize compliance throughout all phases of software development. This may require a significant investment in compliance resources and everyone accepting their piece of the shared compliance responsibility for the application of compliance practices. Lastly, 80.5% of the respondents (i.e., 33 out of 41) agree that better tools to facilitate compliance audits would make them more efficient and more effective. With a clearer understanding communicated to the software industry of what adherence to standards, regulations, and other requirements within the law looks like, organizations can implement better strategies toward due diligent regulatory compliance. At the least, enforcement agencies will have better resources available to enforce regulations and determine intent regarding regulatory compliance.

Table 4.2: K-Means Cluster Survey Items

ID	Survey Items	Type
Q33	My organization does everything it can to diligently comply with their regulatory and security requirements	Rating (1 to 5)
Q34	My organization has a process for prioritizing compliance concerns during the software development process.	Rating (1 to 5)
Q35	When resources are tight (i.e. limited staffing, time, or money), the compliance assessment process is the first thing to change.	Rating (5 to 1)*
Q36	I would not change my organization's compliance process.	Rating (1 to 5)
Q37	My organization actively promotes individual employees' professional development and ethics training (i.e., they pay for professional memberships such as ACM and IAPP or encourage conference attendance)	Rating (1 to 5)
Q48	Internal compliance programs provide real benefits for regulatory and security standard compliance.	Rating (1 to 5)
Q49	My organization's compliance program changed my approach to engineering with respect to regulatory and security standard compliance.	Rating (1 to 5)
Q50	I'm more confident in the products my organization produces and maintains because of our internal compliance program(s).	Rating (1 to 5)
Q51	My organization views regulatory and security standard compliance as an investment to ensuring the quality of our software and trust with our customer rather than the cost of doing business.	Rating (1 to 5)

Table 4.3: Logistic Regression & Mean for an Organization's Compliance Communication & Management

ID	Survey Item	Coeff	Std Er	P	Mean (CI)	Mean (NCI)	Diff
	Intercept	-15.11	8.06	0.061			
Q58	My organization's compliance requirements are ingrained into the culture of the organization.	2.36	1.16	0.042	4.58	2.60	1.98
Q56	My organization understands and follows the intent of the law, when it comes to regulatory and security standard compliance.	2.23	1.64	0.173	4.92	3.47	1.46
Q57	My organization has a history of non-compliance.	0.38	0.66	0.561	2.12	2.07	0.049
Q60	My organization communicates our compliance process both to the employees and our customers.	0.26	0.63	0.681	4.38	3.20	1.18
Q61	I wish my organization would be more transparent about our compliance and security processes to our customers.	-0.13	0.74	0.866	3.04	2.80	0.24
Q59	My organization communicates our regulatory and security standards requirements to third party vendors through contracts to ensure compliance to these requirements.	-1.37	1.02	0.18	4.31	3.40	0.91

Table 4.4: Logistic Regression & Mean for an Organization's Compliance Strategy

ID	Survey Item	Coeff	Std Er	P	Mean (CI)	Mean (NCI)	Diff
	Intercept	-5.28	3.33	0.113			
Q72	My organization spends a lot of time upfront analyzing and understanding regulatory requirements, which makes designing and implementing a response straightforward.	1.91	0.84	0.022	4.08	1.93	2.15
Q69	My organization's initial response is to a new regulatory change is to wait and see how new regulatory requirements evolve and are enforced before complying with them.	0.24	0.57	0.674	2.77	3.60	0.83
Q71	Responses to changes (including risks of rushed software changes and non-compliance) are assessed for impact on business.	-0.18	0.54	0.7361	3.61	2.93	0.68
Q68	Responding to regulatory changes consumes a great deal of resources (time, money, effort) in my organization, in comparison to time spent on design and implementation of our products themselves.	-0.10	0.64	0.87	3.81	3.60	0.21
Q70	My organization response to a regulatory change is to form a team of experts to carefully assess its effects and potential responses.	0.03	0.65	0.963	3.96	2.27	1.69

Table 4.5: Logistic Regression & Mean for Perceptions on Compliance Regulation

ID	Survey Item	Coeff	Std Er	P	Mean (CI)	Mean (NCI)	Diff
	Intercept	3.46	2.55	0.174			
Q65	The best regulations are based on already established industry best practices.	1.31	0.5	0.009	3.77	2.93	0.84
Q66	Regulations favor larger companies making it hard for smaller companies to comply and compete.	-0.03	0.39	0.931	3.27	3.53	0.26
Q67	Compliance audits are necessary but could be better tooled for compliance enforcement within the software industry.	-0.48	0.54	0.3691	4.04	2.07	0.028
Q63	Regulators do not understand the best practices of the software industry and cannot draft regulations accordingly.	-0.65	0.45	0.152	3.27	4.07	0.80
Q64	Regulations are too hard to interpret and make my job even harder.	-0.80	0.60	0.181	3.23	3.93	0.70

Chapter 5

Modeling and Communicating Regulatory Ambiguities using GDPR: Multi- Case Study

This study examines how software developers analyze and model ambiguities within a regulation, addressing the following questions:

SQ1: Can software developers analyze and model regulatory ambiguities?

SQ2: What are the difficulties a software developer encounters when analyzing and modeling regulatory ambiguities individually and as a group?

SQ3: Is there value in analyzing and modeling ambiguities during requirements analysis from a software developer's stance?

The Ambiguity Modeling Process allows developers to reason about regulatory ambiguity separately from their system under development and then trace decisions made to resolve regulatory ambiguities to affected requirements specifications. To evaluate this approach, I recruited eleven participants with backgrounds in software design to form groups and model ambiguities in regulation using the Ambiguity Modeling Process and an online tool known as the Ambiguity Heuristic Analysis Builder (AHAB). I wanted to see if they could accomplish the modeling task individually and as a group (i.e., SQ1). I also wanted to identify the difficulties they generally encountered and their effect on the analysis process (i.e., SQ2). Lastly, I wanted to see if the participants saw value in modeling ambiguities

using this process and tool (i.e., SQ3). My results show that software developers can analyze and model regulatory ambiguities. In addition, developers can discuss their rationale with peers and agree on what they find ambiguous within a legal text. Furthermore, the group can present their analysis and models to other parties for further guidance and resolution. This process offers a way to document the mitigation of ambiguity and link software-design decisions in compliance-related regulatory requirements.

This work offers several contributions to regulatory compliance and ambiguity analysis research. First, I expand upon Massey et al.'s previous work on ambiguity identification and classification [94, 96, 97] by operationalizing it within a modeling methodology as a strategy for analysis. Second, I offer a methodology that involves both individual and group regulatory ambiguity analysis, drawing on the strengths of both modes. Lastly, I offer insight into developers' reasoning and heuristics when performing this kind of analysis, which is necessary to effectively offer further support for the process. Overall, my analysis advances the development of the ambiguity analysis methodology and will facilitate further tool and artifact development.

The rest of the chapter discusses the ambiguity modeling process I have refined (Section 5.1), the design of the case study (including pilot study and the first case groups (i.e. Case One) results, protocol changes, and analysis overview) (Section 5.2), the participants' demographics (Section 5.3), findings from my analysis (Section 5.4), discussion points based on the results (Section 5.5), threats to validity (section 5.6), and finally conclusions.

5.1 The Ambiguity Modeling Process

The modeling process aims to analyze and document ambiguities within regulatory text. Through ambiguity modeling, one can examine, brainstorm, storyboard, and organize potentially confusing regulatory and compliance issues within software requirements analysis. Furthermore, modeling is a visualization of rationale. It captures the modeler's perspective on regulatory text and compliance issues. The modeler then can explain their interpretation to a third party using the model as a guide. A version of this process was first presented and analyzed by Massey et al. in 2017 [94]. I simplified the ambiguity modeling process outlined by Massey et al. [94] by removing recursive layering of ambiguity [1]. This section outlines the ambiguity modeling process as executed by the participants. Section 6.1 describes how this process was embedded in the larger group modeling activity in the study.

The Process: I define a regulatory ambiguity as a word or phrase within a regulation having no or multiple meanings. This definition is derived from the IEEE definitions for unambiguous[2]. The ambiguity modeling process applies our ambiguity definition, executed through five high-level steps (See Figure 5.1):

Step One: The first step is reading the regulatory text. The modeler can read the text in its entirety before identifying any ambiguities or they can identify ambiguities as they progress through the text on the first reading.

Step Two: When the modeler comes across a word, phrase, or paragraph that they

[1] Describing the recursive layering of ambiguity is out of the scope of this paper

[2] "unambiguous: 1) Not having two or more possible meanings. 2) Not susceptible to different interpretations. 3) Not obscure, not vague. 4) Clear, definite, certain."

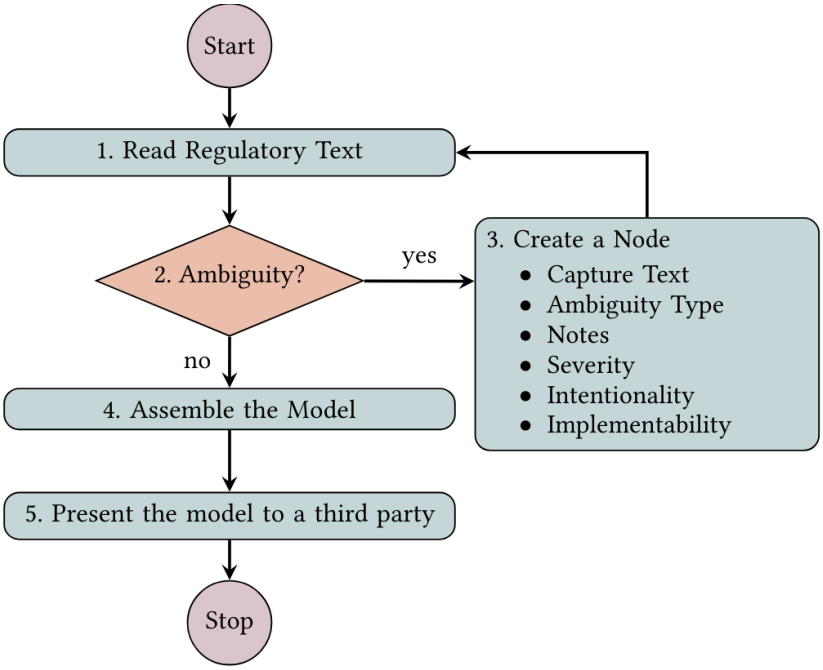

Figure 5.1: The Ambiguity Modeling Process Flowchart [86]

view as ambiguous, they capture the text and begin the process to mark it as ambiguous.

Step Three: To create the ambiguity node, the modeler must expand on their reasoning as to why they view the text as ambiguous by documenting specific prescribed attributes of the ambiguity:

1. **Capture Text**: Identify the ambiguous word or phrase.

2. **Ambiguity Type** Classify the captured text to an ambiguity type. Classification helps clarify the logic as to why the text is ambiguous. Table 1 provides an ambiguity taxonomy to assist the modeler in classifying the ambiguity (See Table 5.1 [96, 97]).

3. **Notes** The modeler further explains the logic behind identifying a regulatory text as ambiguous, beyond the classification within this attribute

4. **Severity** On a scale of 1 to 5, the severity rating indicates the degree to which the resolution of the ambiguity impacts the software design. The severity level increases if the ambiguity challenges the software design process.

5. **Intentionality** Regulators intentionally place ambiguity within regulations or laws, so as the law and its interpretation evolve, so can the regulations, including applicable technology supporting compliance with the law. Therefore, by marking Intentionality as a "Yes," the modeler recognizes that the ambiguity may have been intentional when written.

6. **Implementability** A "Yes" means that the ambiguity exists, but the developer can derive a software requirement specification without further resolution or clarification.

Step Four: Assembling the model involves logically organizing the created ambiguity nodes for presentation to a third party (i.e., Step Five). This organization of the model is the storyboard aspect of modeling, where the modeler highlights potential dependencies, relationships, similarities, and flow of the identified ambiguities.

Step Five: The point of the modeling process is to facilitate communication between stakeholder groups, including people not involved in building the model. This communication solicits further guidance to clarify meaning or intent within a regulation and document further action, interpretation, or decisions made to meet regulatory compliance requirements.

As seen in Figure 5.1, the first three steps (i.e., Reading the Regulatory Text, Identifying the Ambiguity, and Creating the Ambiguity Node) are performed iteratively, until

the modeler identifies all ambiguities, if any, in the regulatory text and creates the associated ambiguity nodes. In step four, the modeler organizes the nodes, thus assembling the model. Then, the modeler proceeds to step five by presenting the model to a third party for further discussion and guidance.

Table 5.1: Case Study Ambiguity Taxonomy [95]

Ambiguity Type	Definition
Lexical	A word or phrase with multiple valid meanings.
Syntactic	A sequence of words with multiple valid grammatical interpretations regardless of context.
Semantic	A sentence with more than one interpretation in its provided context.
Vagueness	A statement that admits borderline cases or relative interpretation.
Incompleteness	A grammatically correct sentence that provides too little detail to convey a specific or needed meaning.
Referential	A grammatically correct sentence with a reference that confuses the reader based on the context.

5.2 Methodology

I conducted a pilot study in November 2021 with two people to test the study design. The primary study with eleven participants was conducted from March 21, 2022, to December 16, 2022 [3]. This section discusses the Case Study design including outcomes

[3]This study was reviewed and approved by the UMBC's Institutional Review Board under Protocol #984 and was partly supported by NSF SaTC Award #1938121.

from the Pilot Study and Case Group One, a description of the AHAB tool, and Data Collection and Analysis.

5.2.1 The Multi-Case Study Design

Each case group of three to four participants in the primary study [4] met in three online sessions with two periods of "homework" between the sessions. All participants analyzed the European Union's General Data Protection Regulation (GDPR) Article 17, the "Right to erasure" [5]. The sessions were structured as follows:

Session One was a one-on-one training session on ambiguity modeling with each participant. We provided participants access to training material and the Ambiguity Heuristics Analysis Builder (AHAB) tool(See Section 5.2.2 for details on the tool).The session included an overview of the case study and Ambiguity Taxonomy (See Table 5.1) and a "Hands-On" AHAB demo [6]. During the demo, the facilitator provided the participant with a list of ambiguity modeling tasks using the AHAB tool. The participants discussed their actions as they accomplished the tasks as the facilitator observed. This demo in Session 1 gave each participant some practice with the AHAB tool and the ambiguity modeling technique before building an ambiguity model on their own for Session Two. In addition, this session also allowed the facilitator the ability to provide technical assistance to the participant if necessary. At the end of Session 1, the participant's homework was to examine the assigned section of legal text from the GDPR, mentioned above, and create an ambiguity model before the next session.

[4] Case 1: three participants; Case 2: four participants; Case 3: four participants.

[5] Also known as 'Right to be forgotten' at `https://gdpr-info.eu/art-17-gdpr/`.

[6] Before Session 1, we gave the participant access to AHAB and tutorial material.

Observation sessions were scheduled ”homework” time for participants to build ambiguity models with the facilitator observing the participant's progress. They were optional and allowed the facilitator to note any technical difficulties a participant might have with the online AHAB modeling tool.

Session Two was an online group session where the participants presented their ambiguity models to their group. Presenting the models allowed everyone to see how their peers approached the ambiguity modeling task. At the end of the session, we gave the participants a JSON file with all the group's ambiguity models. This file allowed the participants to compare ambiguity models using the AHAB tool and conduct further analysis before Session 3.

Session Three began with any updates to the models that the participants might have made since Session 2. Then, the session progressed into the group analysis with the participants attempting to achieve consensus to construct a final joint ambiguity model. At the Session's end, if the group reached a consensus, they submitted their joint ambiguity model. If not, they submitted all models in whatever state they ended up in, and the participant's role in the study concluded.

End of Case Survey [7] was an anonymous and optional 10-minute survey hosted through Qualtrics to gain participants' honest feedback on the ambiguity model process and AHAB tool.

The final structure of the Sessions described above was finalized after analyzing data from our pilot study and Case Group One. The four primary changes to our Case Study protocol design were:

[7] The complete survey is found at `https://doi.org/10.6084/m9.figshare.23297717`

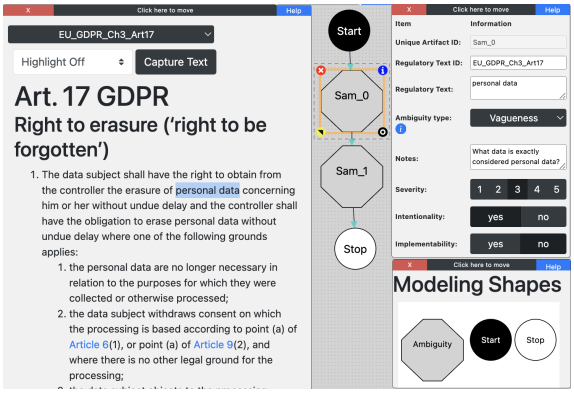

Figure 5.2: Example AHAB version 1 screenshot

1. Expanding the Sessions from two to three online Sessions (From Pilot Study).

2. Incorporating Ambiguity Taxonomy Table (See Table 1) into an AHAB information icon (From Pilot Study).

3. Allowing participants the option to participate in more online "Observation" Sessions (From Case One).

4. Adding the "end of case" survey (From Case One)

Further explanation of these changes are in Section 5.2.3 I made no additional changes to the Case Study's Protocol after Case 1. A more detailed study protocol to include the Session presentation slides are at: `https://doi.org/10.6084/m9.figshare.23297717`.

5.2.2 Ambiguity Heuristics Analysis Builder

To execute the ambiguity modeling process and facilitate an online group analysis and collaboration of regulatory ambiguity, I developed a tool, which I call The Ambiguity

Heuristics Analysis Builder (AHAB). AHAB is designed to allow users to build Regulatory Ambiguity Models [94] online. The tool is written primarily in JavaScript and using a canvas element within the supported web browser as the drawing framework [8]. A user can access the tool through a web browser and build a model by following the process outlined in Section 5.1

Figure 5.2 is a screenshot of AHAB with an example model. The regulatory text is on the left of the picture (i.e., Art. 17 GDPR). An example model is in the middle of Figure 5.2 with two linked ambiguity nodes and a start and stop node. On the top right of Figure 5.2 is the ambiguity node attribute box, outlining all the prescribed ambiguity attributes described in Step three of Section 5.1 and Figure 5.1. The bottom right is the modeling shapes panel, from which a shape (node) can be dragged into the canvas to expand the model further.

AHAB allows for the ambiguity model to be output as a JSON file. The JSON file contains information about every Ambiguity node, including links between nodes. Two other output formats are supported: a tabular format and textual analysis. AHAB also creates a log as a text file that captures every step of making the model, including deleted ambiguity nodes or links.

AHAB has several features that assist with the group analysis of ambiguity models. One feature is the ability to import several models based on the same regulatory text onto the same drawing screen. This feature allows groups to analyze, compare, and combine different ambiguity models built against the same regulatory text without maintaining

[8]The canvas element within HTML5 allows for 2D graphic rendering, such as the octagon ambiguity node within AHAB. `https://en.wikipedia.org/wiki/Canvas_element`

several instances of AHAB

Another feature is the heat mapping of ambiguity nodes. The heat map feature uses coloring to let users see different aspects of the ambiguity nodes within the graphical view of the model. For example, AHAB uses a gradient color ranging from yellow (Severity 1) to orange to a darker red for higher levels of the Severity level attribute. Every attribute of the ambiguity node described in Step Three in Section 5.1 has a heat map instance for selection and viewing.

Overall, AHAB supports the Ambiguity modeling process for individual and group analyses of regulatory text by providing an online, accessible platform and multiple views for documentation and artifact development. To review the AHAB tool in more detail, use the following link:`https://www.sixlines.org/ahab/tutorial/AHAB.html`.

5.2.3 Study Implementation

The previous section outlined the plans for the case study. This section describes the execution of that plan and how it unfolded. I lay out the outcomes of the pilot study (Section 5.2.3.1) and the first case (Section 5.2.3.2). The analysis of those outcomes resulted in changes to the study protocol for Case two and three.

5.2.3.1 Pilot Study

The pilot study goals were to test training tools (i.e., AHAB and the video and written tutorial) and the case study design. I recruited two pilot participants through personal

contacts. I executed a version of Sessions One and Two as group sessions with the pilot participants in November 2021. Session One demonstrated the AHAB tool and ambiguity modeling process and Session Two had the pilot participants present their ambiguity models using AHAB to each other with the facilitator and moderator watching.

The analysis of the pilot study resulted in two changes to the study design and infrastructure. The first was updating the AHAB tool to address two usability issues the pilot participants reported. These changes did not fundamentally alter the functionality of AHAB. The second change occurred because both pilot participants initially had trouble picturing and organizing their ambiguity models. In particular, how to define the relationships between the nodes and how to organize their identified ambiguities. This difficulty was severe enough that one of the participants created an initial ambiguity model independently before Session Two and the other did not. I addressed this difficulty by providing additional guidance and some example models. However, the participant who did not build a model before Session Two quickly made an ambiguity model by closely adopting one of the examples during Session Two. As a result, I decided not to provide examples to the participants in the main case study. This decision was risky because it meant that some participants might not be able to produce a model independently. However, the more considerable risk was providing too prescriptive guidance (in the form of examples) and leading the participants to follow the examples rather than allowing them to reason about the ambiguities freely and express their reasoning in the models they built.

To mitigate the risk and assist participants in initial model construction, I did decide to change the format of Session One. I changed it to an individual session for each participant versus a group session. In addition, I utilized the "Hands-On" exercise with the

AHAB tool by having the participants accomplish a series of tasks and build their models as a facilitator and moderator observed. Having the participants build a model in the first session gave participants some hands-on practice with the AHAB tool and allowed us to provide individual technical assistance if necessary. The intent was to make building the second ambiguity model on their own less intimidating for the case participants.

5.2.3.2 Case One

Case 1 was conducted from March 21 to May 2, 2022 and had three participants (ID 1-3). I initially analyzed Case 1's data from June to August 2022. The analysis of the Case 1's data served as a quality check on the study protocol. In particular, the study team reflected on how I might better facilitate the participants' ambiguity analysis and model building without biasing the results. I concluded that it would be helpful to schedule some dedicated modeling time online between Sessions 1 and 2 for each participant, where the facilitator could observe any difficulties they were having and could assist them accordingly. It also gave the participants access to the facilitator to ask further technical questions on the AHAB tool as required. At this point, I also added the End of Case Survey

The added "Observation sessions" and the survey were the two updates I made to the case study's protocol prior to Case 2 and 3's execution. Cases 2 and 3 overlapped in late 2022 (October 5 to November 22, 2022, and October 26 to December 16. 2022, respectively). Each case had four participants [9] I made no additional changes in the study

[9]Case Three initially had five participants, but ID9 withdrew after Session One because of scheduling conflicts.

protocol.

5.2.4 Data Collection and Analysis

I used data collected from three sources:

1. Online sessions with the case participants recorded through Google Meet and transcribed using Otter.ai [10];

2. The participants' ambiguity models using AHAB;

3. The online close-out survey results [11].

All data was collected and analyzed using NVivo version 12 [12].

I analyzed the data using grounded theory [44] and with-in and cross-case analysis [101]. I generated the initial coding scheme [13] based on the timeline of events, delineating the three sessions in each case, and three parts of each session (i.e., the participant's preparation before the session, what happened during the session, and what happened at the end of the session). I then built content based sub-codes under those initial sequence-based codes. I saw five themes emerge from the initial application of codes. These emerging themes became the final coding scheme (see list below) and helped generate the findings.

1. Common reasoning for identifying an ambiguous legal text

[10] https://otter.ai

[11] We added the survey after Case 1 and hosted it through Qualtrics. The complete survey is found at `https://doi.org/10.6084/m9.figshare.23297717`

[12] `https://lumivero.com/products/nvivo/`

[13] The initial coding scheme is at: `https://doi.org/10.6084/m9.figshare.23297717`

2. Common reasoning for classifying an ambiguous legal text

3. Modeling difficulties

    (a) Understanding the Regulatory Text

    (b) Classifying ambiguities

    (c) Consolidating models

4. Ambiguity analysis and discussion

5. Importance of ambiguity modeling

The next section discusses our findings based on our analysis.

5.3 Recruitment and Participant's Demographics

I recruited participants for the main study from UMBC's[14] Cybersecurity, Data Science, Software Engineering and Information Systems graduate programs. They formed three case groups of 3-4 participants each. All participants were 23-30 years old and had Software Developer (11) or Analyst (2) backgrounds [15]. They reported an average of about three years of work experience in their roles (range of 1-9 years). For Case One, I recruited via class contacts within the Software Engineering and Information Systems departments. For Case Two and Three, I sent recruitment emails to graduate students within the Cybersecurity, Data Science, Software Engineering, and Information Systems

[14]University of Maryland, Baltimore County

[15]Software Developer: A person that builds and maintains software or IT systems. Analyst: A person who gathers and interprets data for requirements.

departments. In total, eleven participated in the primary case study (i.e., Case 1:ID 1-3, Case 2: ID 4-7, and Case 3: ID 8-12) [16]. Table 5.2 is a breakdown of all the participants' demographics including the two pilot study participants [17].

Table 5.2: Case Study Participant's Demographic

Case	ID	Years of Exp	Role
P	PID1	less then 1	Software Developer
P	PID2	1-2	Analyst
1	ID1	2-3	Software Developer
1	ID2	3-4	Analyst
1	ID3	2-3	Software Developer
2	ID4	3-4	Software Developer
2	ID5	2-3	Software Developer
2	ID6	2-4	Software Developer
2	ID7	1-2	Analyst
3	ID8	1-2	Software Developer
3	ID10	8-9	Software Developer
3	ID11	2-3	Software Developer
3	ID12	1-2	Software Developer

[16] One Case Three participant (ID9) withdrew due to scheduling issues after Session 1.

[17] IDP is a pilot study participant; ID is a main study participant.

5.4 Finding

This section highlights the three major findings answering the research questions.

SQ1: Can software developers analyze and model regulatory ambiguities? **Finding 1:** Yes, with a tool and guidance, software developers can perform regulatory ambiguity analysis and modeling individually and as a group.

SQ2: What are the difficulties a software developer encounters when analyzing and modeling regulatory ambiguities individually and as a group? **Finding 2:** Regulatory ambiguity analysis is difficult for software developers. One can expect to see software developers struggle to understand the regulatory text, classify the ambiguities, and consolidate into a group model. However, the difficulties directly lead to identifying ambiguities. Also, discussion of these difficulties is evidence that the analysis is being done through this intermediate documentation.

SQ3: Is there value in analyzing and modeling ambiguities during requirements analysis from a software developer's stance? **Finding 3:** Yes. Engaging software developers in ambiguity analysis can create buy-in and lead developers to value regulatory activities.

5.4.1 Completing the Ambiguity Models - SQ1

Finding 1: With a tool and guidance, software developers can perform regulatory ambiguity analysis and modeling individually and as a group.

The study participants were assigned two tasks. First, they each needed to build a regulatory ambiguity model. Second, they had to discuss and consolidate their models

into one group model.

Everyone accomplished the first task. Two of the three groups accomplished the second task by consolidating their models into a group model by the end of Session Three. Group One was not able to accomplish this in the time allotted [18]. Some participants even reported that ambiguity modeling was easy because of the guidance and the tool, contrary to the results of the pilot study. For example:

> **ID6:**"Yes, same for me, it was really easy. [AHAB] has given various options like...the heat map selection... the text [capture], the ambiguity type, and the severity. [It] was really easy to visualize my whole model. "

Given time, tools, and guidance, software developers can model and communicate concerns about an ambiguous regulatory text. Furthermore, by documenting and sharing these concerns, they can look to third parties (i.e., lawyers) for further guidance to clarify or resolve the ambiguities.

5.4.2 Difficulties with Ambiguity Modeling - SQ2

Finding 2: Regulatory ambiguity analysis is difficult, but the difficulties directly lead to identifying ambiguities. Discussion of these difficulties is evidence that the analysis is being done through this intermediate documentation.

Interpreting regulatory ambiguities can be difficult for software developers with no legal training. This section highlights three modeling process difficulties common among

[18]Screenshots of groups models are in Appendix D and available at: `https://doi.org/10.6084/m9.figshare.23297717`

the participants.

5.4.2.1 Understanding the Regulatory Text

One difficulty in modeling ambiguities was understanding the regulatory text. Most participants found the wording or intent in the legal text difficult to understand. Consider the following comments from Case Group Three's participants:

> **ID8:**"When I read this for the first time, I got confused. I did not know whether they are talking about data subject or the controller. "
>
> **ID10:**"I specifically found understanding the document in the first try [difficult]. I would have to read it a number of times to understand what they're trying to convey. That was one difficulty. "

Seven of the 11 participants used phrasing such as "confused," "unclear," or "complex" when presenting their analysis. I noted this trend as the top reason for identifying ambiguous text amongst the participants. Not all participants expressed difficulty understanding the legal text. Similarly, the survey showed mixed responses to the question about understanding the regulation difficulty [19]. Yet, when such difficulties arose, they led to progress in the analysis.

5.4.2.2 Classifying Ambiguities

Another difficulty pointed out by the participants was classifying ambiguities. Take, for example, ID6's comment from Case Group Two:

[19] Out of the five survey responses, two agreed, one was neutral, and two disagreed.

ID6:"If I read a sentence initially, I [would] think it was one type of ambiguity. If I revisit the model or that text, I [would think] "No, this is something else", interpreting it as [another] ambiguity type. "

The survey results told a slightly different story. Four respondents to the survey disagreed with the statement: "I found it difficult to identify and classify the ambiguities within the regulation.". One of the five respondents agreed, however.

The participants' confidence about the modeling process at the study's end could explain the differences in the data collected. Evidence shows that the taxonomy and the AHAB tool evolved the participants' understanding of regulatory ambiguities. For example:

ID2:"The tool helped me understand what exactly ambiguity is. I didn't know what the word ambiguity meant before this [study]. "

Some participants used the ambiguity taxonomy definitions (See Table 5.1) to explain their ambiguity analysis. Consider the following comments:

ID11:"...the ambiguity type is vagueness, because it is a borderline case "

ID1:"I felt this was an ambiguous statement and [is] vagueness [since] it covers only borderline cases. "

ID4:"This phrase, "legitimate grounds", it may have different interpretations from person to person. So I think that is a semantic ambiguity. "

As the participants' understanding evolved, so did their confidence regarding the modeling process as shown in ID12's comment.

ID12:"I feel like if we had a fourth session, we can [build] another model in one meeting...[it took] three sessions, [for] our [model] because we were new to AHAB...I feel that will be more rapid if we [did] another article. "

Some participants expressed difficulty using the ambiguity classification taxonomy during the sessions; however, some participants felt much more confident in their analysis and the modeling process by the end of the exercise. The ambiguity taxonomy and the AHAB tool were aids to that progression.

5.4.2.3 Consolidating models

The third difficulty with modeling was consolidation. Some participants highlighted that agreeing on ambiguities (identifying or classifying) for model consolidation was challenging. Take, for example, these quotes:

ID2:"I think it's agreeing with others. Trying to get their perspective [versus] your perspective is one thing [that was] difficult. "

ID8:"Choosing an ambiguity type is also a bit difficult, especially in this case study, where we had different opinions. "

Despite the difficulty of consolidation, two groups consolidated and created a group model. Both of these groups quickly developed a systematic approach, involving analyzing the ambiguity nodes one by one, interactively discussing their representations of that node, and coming to a consensus. Group 1 did not complete consolidation, came up with their review approach a little later than the other groups and encountered technical

difficulties with Google Meet. They ran out of time in Session 3 as a result. This process of interactively discussing each node was effective in consolidating the models and as discussed in the next section, also helped change the participants' perspective on the importance of this process.

5.4.3 Valuing Ambiguity Analysis - SQ3

Finding 3: Engaging software developers in ambiguity analysis can create buy-in and lead developers to value regulatory activities.

Some participants started the process with doubts about the value of modeling regulatory ambiguities. However, by the end, opinions changed:

> **ID7:**"When I started working [on the model], I thought "it won't be that important". But then I started to realize that this is an important step in the [requirement analysis] process. "

> **ID4:**"I first thought that it is a simple task, we don't need a model like this. [It] could be done as we progress. I realized that there is a lot more ambiguities than I realized... it will get complex. So, it will make the process easier if we use this model. "

Some participants expressed their thoughts about the modeling process by providing real-world feedback. Others commented on how they might want to use this process to consider other stakeholder perspectives. For example:

> **ID10:**"This is very important, because there are many times in which [I read]

our terms and conditions [contracts and] have a different meaning than what the customer means. ”

ID3:“As a developer with the stakeholder, I would love to have a conversation about this legal text, because I wonder how it can be interpreted. ”

Not all of the participants shared this view. I asked participants the below questions in the survey:

1. “Did you feel there is value in reviewing regulations and building ambiguity models as part of a Software Development Process?”

2. “Would you suggest this modeling process as part of the requirement phase to your software development team?”

Four out of five participants responded with a “Definitely Yes,” or “Probably Yes” to the two questions. One responded with a “Might or might not” to the first question and a “Probably Not” to the second question.

These survey answers indicate that some participants did not see the value in the ambiguity modeling process. Nevertheless, others did. Some participants realized that ambiguity modeling is about the models produced and more. It is also about perspective and understanding the regulatory requirements during the requirements analysis and documentation. The analysis and documentation are evidence of compliance due diligence within software design.

5.5 Discussion

In this section, I distill the findings reported in the previous section into two discussion points that have implications for software practitioners who are concerned with effective regulatory compliance in their software projects.

5.5.1 Certain difficulties aid regulatory analysis

Discussion point 1: Lawyers and other stakeholders seeking to aid software development teams can benefit from hearing developers articulate their difficulties when reviewing a regulation.

Good regulatory analysis requires that everyone be on the same page and that requires communication and engagement between stakeholders. Knowing some of the difficulties a stakeholder might have in understanding a regulatory text should be part of the conversation. The modeling process is a tool to aid in that conversation. Most of the participants said they needed help understanding the regulatory text (i.e., Finding 2). Other studies with legal text have made similar points [96, 111, 97, 40]. A lack of regulatory understanding means developers cannot explain or account for regulatory compliance actions in their software design. Lawyers and other stakeholders wanting to advise their software teams on applicable regulations should note and discuss these struggles. The ambiguity modeling process facilitates and documents the discussion by getting developers to communicate their confusion (i.e., the ambiguities), provide a rationale, and ask meaningful questions about their requirements. Lawyers or other stakeholders can respond with clarifying guidance to assist developers with their understanding and make

better software design choices.

5.5.2 Valuing tools and guidance that support regulatory compliance

Discussion point 2: Well-designed processes and tools are vital to aid and document effective regulatory analysis and build a culture of compliance for software developers.

Some software developers will view ambiguity modeling as an unnecessary hassle. This type of analysis, though, is necessary for regulatory compliance requirements development and documentation. Having proper tools not only reduces the hassle of regulatory analysis and documentation but also, over time, helps build regulatory analysis into the software design process, thus promoting an organizational culture of compliance. Once a software development team has done an initial assessment, they can communicate and discuss their work with other stakeholders, like lawyers, for more guidance or confirmation. More importantly, software developers will internalize and see value in implementing such a process within their requirements development (i.e., Finding 3). Lastly, ambiguity models serve as documentation of due diligence, highlighting how developers addressed risks and trade-offs related to complex compliance concerns like privacy and security and complimenting other software engineering artifacts.

5.6 Threats to Validity

This section covers the limitation of the Multi-Case Study.

5.6.1 Threat to Internal Validity

All the participants built an ambiguity model and explained what they identified as ambiguous. However, within the pilot study, one participant did not complete the model for process and external reasons. Even though I provide resources for any software developer to complete the regulatory ambiguity modeling task, other factors can waylay the process like technical difficulties, competing priorities, and scheduling conflicts. These factors and others might hinder regulatory analysis. Furthermore, I provided incentives for recruitment purposes to conduct the study. Therefore, the participants' motivations to complete the study are different than in the real world.

5.6.2 Threat to External Validity

I am limited in generalizing the findings from this study because the participant sample is small. The study had 11 participants whose ages ranged from 23 to 30 years, had similar cultural backgrounds, and most of the participants' work experience was less than three years [20]. In addition, all the participants identified as software developers, but two had analyst work experience. A repeat of this study would benefit by using a more extensive and diverse selection of participants. Exploring participants and regulations from different domains and jurisdictions may enable other results for comparison.

[20] Four participants had three or more years of experience.

5.6.3 Threats to Reliability Validity

This type of regulatory ambiguity study is novel and does not have a comparison point in the literature. Therefore, I have made available details of the methods and evaluation techniques for others interested in replicating this study [21].

5.6.4 Threat to Construct Validity

This process allows software developers to communicate issues during requirement analysis to other parties to get answers. I did not test the next step by having the groups present their model to an outside expert in this study. The next chapter will explore this next step. Another construct validity threat is that we conducted the study in a lab environment. The study used UMBC graduate students, and they worked on tasks unrelated to their jobs or schoolwork. Therefore, the results may have differed if I had used an established software development team operating in the industry. I tried to mitigate this threat by recruiting participants with real-world experience in the software development industry.

5.7 Summary

I observed software developers interpreting and modeling ambiguities within a regulation. I found that software developers can analyze regulatory ambiguity with a tool and guidance. The analysis can be challenging; however, software developers experiencing

[21] Expanded Case Study Methodology are available at: `https://doi.org/10.6084/m9.figshare.23297717`

and discussing their analysis difficulties is essential to the process. In some ways, the difficulties prove that developers were meaningfully engaging with the regulatory text, and thus actually performing regulatory analysis. The models served as a way to document these due diligence efforts to understand and comply with the law. Overall, engaging developers in these types of activities is vital. It allows them to communicate and document potential issues regarding the understanding of regulatory and compliance requirements. Furthermore, engagement in the regulatory analysis process can change their perspective and create buy-in in the software design analysis and compliance process. The work has limitations, but the next chapters addresses some of these limitations by recruiting auditors and software developers operating in the software industry to provide feedback on the ambiguity modeling process usefulness (i.e., Chapter 6). By assessing the usefulness of the modeling process, researchers can use the data to improve the ambiguity modeling process and AHAB to support regulatory compliance in software design.

Chapter 6

Validating Ambiguity Modeling

Regulatory ambiguities can hinder the development of compliant software. Ambiguity modeling is a proactive measure to communicate and address ambiguity in regulation and develop software that complies with the law's intent. From my multi-case study, I reported that software developers performing ambiguity modeling can identify, document, and communicate unclear, inconsistent regulatory requirements that can put their organization at legal risk. The development team can then use the model to engage internal organizational resources to clarify ambiguous requirements and ensure a clear understanding of their regulatory compliance obligations. Furthermore, documenting the interpretations of ambiguous regulations and decisions demonstrates due diligence efforts to comply with applicable regulations should a compliance inquiry or audit occur.

In this study, we consider the perspective of stakeholders interested in assessing compliance when they were not involved with the project as a developer or engineer. We refer to these stakeholders as auditors. An audit is a process to review and assess an organization's procedures and standards to verify that it operates according to applicable laws or regulations. Internal auditors may be hired by the organization to assess regulatory compliance and potentially compliance with corporate governance. External auditors may be regulatory authorities seeking to assess compliance with laws and regulations. Auditors must be able to assess organizational efforts towards compliance. Assessments may

be conducted as part of a regular internal checkup, as part of a third-party independent audit, as part of a regulatory enforcement action, or a variety of similar scenarios.

Regulatory ambiguity modeling is intended to document and communicate compliance efforts, and our study seeks to determine whether and how regulatory ambiguity models are useful for auditors making their assessment.

Auditors are rarely limited to binary assessments that merely report a system as compliant or not. Detailed documentation of the process, including interpretations of regulatory ambiguities and their resolution in software, may provide an auditor with a rationale that mitigates assessments of negligence or malfeasance. Failure to demonstrate due diligence may result in additional penalties beyond those levied for non-compliance. For software organizations who want to operate, or continue to operate, within regulated industries, demonstrating due diligence to comply with applicable laws and regulations to an auditor is essential. To examine the auditor perspective and the usefulness of the ambiguity modeling process, we present this empirical study to answer following sub-research questions on the usefulness of ambiguity modeling within software development:

SQ1: Is ambiguity modeling useful for an auditor assessing a software organization for regulatory compliance?

SQ2: Does ambiguity modeling provide evidence of due diligence of regulatory compliance within a software organization?

SQ3: What can be done to make ambiguity modeling as a process, or the Ambiguity Heuristics Analysis Builder (AHAB) as a tool, more useful?

Six software industry practitioners with auditing experience participated in two fo-

cus groups. We presented the ambiguity modeling process, the AHAB online tool [1], and the findings reported from the Chapter 5's multi-case study. We asked our participants for their feedback on the usefulness of the modeling process from an auditor's perspective. We wanted the auditor's perspective because auditors are an essential part of the audience for this type of research into regulatory compliance in the software development field. They are on the front lines of enforcing these regulations and, as mentioned earlier, are tasked with verifying an organization's compliance with applicable laws and regulations. Our hypothesis is that any tool or documentation that can give an auditor feedback or a visual to help explain why software development teams make certain decisions to mitigate compliance risk is beneficial for the auditor, and by extension, a software organization.

Five out of six participants agreed that the ambiguity modeling tool presented insights and visualizations useful to show an auditor and would benefit a software organization going through an audit. As an internal tool, it can help define an organization's regulatory compliance standards and clarify any legal interpretation issues it may encounter during an audit. Furthermore, when linked to certain artifacts and outcomes, the ambiguity modeling process could help show how an organization interpreted its regulatory obligations and what steps it took to fulfill them. Ambiguity modeling would also be useful as an auditing preparation tool, answering possible questions an auditor might ask about deviations or decisions made during the software development process. Lastly, the groups offered feedback on the AHAB tool and possible updates to improve its usefulness. These are the general findings reported in this study.

[1]The following is a link to the online version of AHAB: `https://www.sixlines.org/ahab/tutorial/AHAB.html`

This chapter reports the research methods used to collect and analyze the data (Section 6.1), the participants' demographics to include background and relevance related to the auditor's perspective (Section 6.2), the findings on the usefulness of the process and tool (Section 6.3), discussion of the implications of those findings (Section 6.4), the threats to validity (Section 6.5) and the end of chapter summary.

6.1 Methodology

Focus Groups are a means of gaining an in-depth understanding of issues or problems based on the participants' experiences and reactions through interactive group discussion [101]. The previous multi-case study finding [86] viewed the value of the ambiguity modeling process from a software developer's perspective. This focus group study is about gaining insight into the usefulness of the Ambiguity Modeling Process from an auditor's perspective. I opted for an auditor's perspective because the previous multi-case study [86] already reported on software developers' perspectives within that study's findings. In additions, auditors assessing an organization for regulatory compliance either for certification (e.g., HIPAA certification) or during a compliance incident investigation, might want to see documentation on how certain design decisions came about, especially when dealing with a regulatory compliance issue that has links to ambiguous language within a law or regulation. Lastly, gaining feedback and insight into the usefulness of the ambiguity modeling process from another perspective could strengthen the multi-case study's claim of value for a software organization to incorporate within a software development process. Therefore, I opted to conduct a Focus Group to gain the auditor's

perspective and contribute to the research area. This section describes the Focus Group study design, including extra tests, pilot study outcomes, and data collection and analysis.

6.1.1 The Focus Group Design

In designing the Focus Group, I started with the goal to gain feedback on the usefulness of an ambiguity model to an auditor. But before asking the participants questions pertaining to that goal, they would need a good deal of background and context, in order to understand the questions. To this end, our study design included presentation of the following background information:

- An overview of the Ambiguity Modeling Process
- A demonstration of someone building a model using the AHAB tool
- Findings from previous research and guidance from the legal researcher
- Discussion questions focused on the auditor's perspective on ambiguity modeling:

FG-Q1: From an auditor's perspective, if presented with an ambiguity model as part of the documentation during an audit, what information could you get from that model that would be useful for your auditing task?

FG-Q2: Would adopting ambiguity modeling help resolve any difficulties you (or auditors) often experience when auditing for regulatory compliance?

FG-Q3: What other artifacts for regulatory compliance could a software development team produce that could help an auditor assess their development process?

I divided the focus group sessions into two sessions, in order for the participants to have some time to reflect on the background information given to them, as well as the discussion questions. Session One covered all of the information listed above, including preview of the discussion questions, but not the actual discussion. Session Two was for the participants to discuss the usefulness of the ambiguity modeling process, based on the questions I posed. I restricted the time between sessions to 48 hours to ensure that our participants could remember the information presented about ambiguity modeling and their thoughts or comments from the first session. Therefore, the sessions were structured as follows:

Session One: This session was an overview of the ambiguity modeling method. This overview included:

- The introduction of the participants and the purpose of the focus group;
- A step-by-step talk through of the ambiguity modeling method;
- A live demonstration of building an ambiguity model using AHAB;
- An initial "Question and Answer" for the Ambiguity Modeling Method;
- The outcomes reported from the multi-case study [86] and the test conducted with Jeffrey Kosseff, Esq. (See 6.1.2);
- A second "Question and Answer" for the Ambiguity Modeling Process;
- The preview of the three discussion questions for Session Two.

Session Two: This session was the discussion session of the focus group. I asked the groups the discussion questions and gave them approximately 15 to 20 minutes for

each question. At the end of the sessions, I thanked everyone for participating and closed the session [2].

6.1.2 Previous work presented to the Focus Group

In Session One, in an effort to present the participants with sufficient background and context to usefully address the discussion questions, we presented prior findings from two sources that shed light on the usefulness of ambiguity modeling. The first the findings from a multi-case study [86] that examined the usefulness of ambiguity modeling for software developers, as follows:

- Despite difficulties, software developers can do this analysis.
- Difficulties aid in the ambiguity analysis
  - Understanding the Regulatory text
  - Classifying ambiguity
  - Consolidating models
- See value through the process and see its importance as a regulatory activity.
  - Promotes communication, discussion, and collaboration.
  - Creates buy-in in the requirements compliance process.
  - Real-world applications for interpreting legal text, not just for the software developers.

[2]The session slides and the protocol outlining sessions at the following DOI: `https://figshare.com/s/7d3752cd8d99631fb8a4`.

In addition to the findings of the previous study, I considered the role guidance documentation can play in an industry. Guidance documentation is " a statement of generally applicable issues by an agency to inform the public of its policies or legal interpretation [1]." Although they are not a requirement set forth by law, guidance documentation can explain more clearly the intent of a regulation or a law and help to establish best practices for an industry or internally within an organization [1]. Guidance documentation is a possible outcome linked to a regulatory ambiguity model that would help clarify ambiguous compliance requirements.

To explore the possibility that an ambiguity model might aid in the process of producing guidance documentation, I worked with a cybersecurity law professor to develop a guidance document for Virginia's Consumer Data Protection Act (VCDPA) Article 59.1-574 titled "Data controller responsibilities; transparency" [3]. The research team first modeled the VCDPA using AHAB and the outlined Ambiguity Modeling Process (Reference Section 5.1 in Chapter 5). I presented our model to our cyber law expert, Jeffery Kosseff, Esq., who then developed a guidance document, and provided additional feedback on the usefulness of the process and tool from their perspective. I used the previous study's findings [86], the VCDPA ambiguity model, and the legal guidance from the cybersecurity law professor as part of our presentation to our focus groups.

6.1.3 Pilot Study

I conducted a pilot study of Session One on August 18, 2023, with two people who fit the demographics of the target participants (i.e., having worked for more than a

[3] https://lis.virginia.gov/cgi-bin/legp604.exe?211+ful+SB1392ES1

year within the software industry within a regulated domain and having gone through a compliance audit). Pilot Participant One was a Chief Technology Officer with a start-up medical device company for over three years. Their duties included Quality Assurance testing and preparing the company for quality audits. Pilot Participant Two is an Army Officer with 20 years of experience in the Department of Defense's Cyber operations.

The session ran for approximately 72 minutes, 12 minutes over our target time. The overrun on time required my demonstrator and me to modify the Session One protocol, changing the demonstration and leaving some additional time for participants to provide feedback during the first session. Overall, though, the feedback from the participants indicated that the presentation and demonstration of the ambiguity modeling process were good, and the questions for discussion would get feedback on the tool's usefulness. Therefore, I proceeded with recruitment for the study.

6.1.4 Data Collection and Analysis

I conducted two focus groups, each comprising the two online sessions as described in Section 6.1.1). The first focus group sessions were on September 5 and 7, 2023, for one hour each within 48 hours of each other. Focus Group Two's Session One lasted about an hour. There was a 10-minute break, and then we conducted Session Two, which lasted 47 minutes. These sessions occurred on December 4, 2023. Other than the scheduling of the sessions and the timing of Focus Group Two's Session Two, which I discuss in Threats to Validity, there was no difference between the execution of the Focus Groups.

All sessions were recorded using Google Meet, comprising the primary data of the

study. The recordings were transcribed using Otter AI. We used NVivo version 12 and shared Google Drive to store and analyze the transcribed sessions.

I used a constructivist grounded theory approach [119] to analyze the data. Before starting the analysis, I began with an initial very general research question regarding the usefulness of ambiguity modeling within software development. I evolved the question, honing in on the auditor's perspective and aligning with the study's goals. To generate the initial coding scheme, we started by creating codes representing the sequence of events in the sessions to help make sense of the collected data (See below).

- Session One:
  - Participant's reported demographics to include current jobs and previous experience.
  - Participant's initial thoughts on the ambiguity modeling process after overview and live demo.
- Session Two: The participant's answers to the discussion questions:
  - From an auditor's perspective, if presented with an ambiguity model as part of the documentation during an audit, what information could you get from that model that would be useful for your auditing task?
  - Would adopting ambiguity modeling help resolve any difficulties you (or auditors) often experience when auditing for regulatory compliance?
  - What other artifacts for regulatory compliance could a software development team produce that could help an auditor assess their development process?

I recruited my peer researcher to review the session transcripts under this initial coding scheme with me. After we had both reviewed all four transcripts, I updated the coding scheme as follows:

1. Participant's Demographics
    (a) Auditing Experience
    (b) Organization
    (c) Previous Experience with the Software Industry
    (d) Current Role in Software Industry
2. Initial thoughts on Ambiguity Modeling
3. Usefulness of Ambiguity Modeling
    (a) Useful
    (b) Not Useful
    (c) Maybe Useful
4. Usability
    (a) For an Audit
    (b) Internal Support Tool
    (c) Clarify Regulatory Requirements
    (d) Documenting discussion on regulatory compliance within Software Development Process

5. Signals an Intent to Comply

6. Suggested Updates to Process and AHAB Tool

7. Other

    (a) Intentional Ambiguity

    (b) Unknown Ambiguity

    (c) Variance of Usability

After the final update and application of the coding scheme to the session transcripts, I re-evaluated and developed the three research questions presented in the introduction. I then created three memos, organizing the categories and codes based on the three research questions. As I developed these memos, essentially performing a fourth pass on the session transcripts, I considered the following:

- Is there data saturation and support from both focus groups?
- Did the data collected answer the research questions?
- Is there some insight or takeaway applicable to the ambiguity modeling process?

After the fourth pass on the session transcripts, we translated the memos to the findings we report in the Results section. The next sections cover the participant's demographics.

6.2 Recruitment and Participant's Demographics

I recruited participants from September 5, 2023, to December 4, 2023, to participate in two Focus Groups, with three participants in each group. I recruited participants using

my peers' professional software industry contacts to recruit candidates who met two or more of the following criteria:

- The participants worked over five years in Software or Information Technology (I.T.) Industry.
- The participants underwent a regulatory audit for a software/I.T. organization.
- The participants performed a regulatory audit either internally organizational or as an external regulatory auditor.
- The participants have advised the software development team on regulatory compliance requirements as part of their contracted duties and responsibilities.

For this study, I targeted candidates with a specific type of experience. I wanted participants working within the software development domain. They also had to have experience as an auditor or had gone through a regulating agency audit for their software or I.T. system. I define an audit as an assessment performed by a third party to determine an organization's compliance with the rules and regulations that pertain to its industry domain. We invited 13 candidates, and six agreed to participate. Table 6.1 summarizes the participants' demographics.

Five out of six participants have more than 20 years of experience within the software industry, with healthcare being the primary regulated domain [4]. The organizational size indicates that the participant works at a large organization with multiple software

[4]Tech indicates that the participant worked at an organization that develops technology for multiple domains. Gov't which is short for Government, indicates the participant operated within a branch of government services like the Department of Defense

Table 6.1: Participant Demographics

ID	Years	Industry	Org size	Auditor Exp.	Audited
ID1	30+	Health	Large	External	Yes
ID2	30+	Health	Small	No	Yes
ID3	20+	Tech/Health	Large	Internal	Yes
ID4	6+	Gov't/Tech	Large	No	Yes
ID5	30+	Health	Large	Internal	Yes
ID6	20+	Gov't/Health	Small	No	Yes

development teams and has access to resources like quality management or software security teams versus a small organization with one development team, where quality assurance/compliance resources are limited. "Auditor experience" indicates whether the participant has performed internal organizational audits or external audits on an organization. Lastly, "Audited" means that the participants have been through an audit by an external third party to ensure compliance with a specific regulation. Overall, we recruited two groups of participants to volunteer two hours to give us feedback based on their background and experience. My next section covers the findings based on the comments and input of our participants.

6.3 Finding

6.3.1 Auditor's perception of the Ambiguity Modeling Process's usefulness – SQ1

Finding 1: The Ambiguity Modeling Process has uses, but it is missing a link to the final output to resolve the ambiguity within the software development process.

Most of the participants saw potential uses with the Ambiguity Modeling Process. Ambiguity modeling documents potential issues and decisions made to resolve those issues. However, it is not standalone documentation. This type of analysis is an intermediate step within requirements analysis. Therefore, it must be linked to an outcome or output to resolve compliance issues.

Between the two focus groups, five out of the six participants saw some usefulness in the process and the AHAB tool. They thought it was useful from an auditor's perspective for the following reasons:

- It documents regulatory ambiguity discussions that are connected to software development artifacts, such as requirements and testing requirements.
- It documents discussions to deviate from an organization's standard practices.
- It documents internal clarifications of regulatory compliance requirements.
- It documents part of the artifact development.

However, they also stated that its usefulness from an auditor's perspective is limited because:

- Ambiguity in regulation is intentional.
- Auditors would want to see a link from the ambiguity models to software artifacts.

The following subsections unpack these points as shared by our participants with cited quotes from both groups.

6.3.1.1 Documenting regulatory discussions and clarifying requirements

Five of the six participants talked about how an auditor might view this tool and how a software organization would use this process to show what steps their company is taking to comply with a regulation. The groups pointed out that the model **documents the discussion of regulatory ambiguities**. These discussions would lead to decisions that would then connect to development artifacts. Consider, for example, ID1's comment:

> **ID1:**" I actually was thinking that this would be a model that would occur between an engineering organization, and either a regulatory or quality organization within the business to establish what those standards are for the business. And then potentially use that as the demonstration if you're talking to an auditor about why it is we picked what we picked. "

Participants also made other points about the tool's usefulness in **documenting reasoning for deviating from an organizational standard practice**.

> **ID3:**" this actually has a really good application, clarifying requirements of the many ways that I try to do with customers; getting them to ask the question in as much of an objective way as possible, which in itself is its own

major effort. But this sort of provides a structured way to say, 'help me, help you' by scoping this down."

An auditor cannot see why certain decisions are made within a software development process when they only examine the outputs or the final compliance reports. Often there is an established standard operating procedure in an organization, linked to a regulatory requirement. Yet, the existing technology or a related system might prevent a software developer from implementing the established standard. In that case, they might have to use a workaround until the development team can address the issue. Documenting such discussions and the resulting decisions is crucial for software maintenance, especially if an organization has a high turnover within its development teams.

There is also the communication piece to consider between the software organization's engineering and business sides. When the organization is small, the two sides are often handled by the same people, so communication may not be an issue. However, as organizations grow and divide their responsibilities, they can have pockets of professionals communicating and interpreting requirements differently within software development. As ID1 pointed out earlier, ambiguity modeling could help internally bridge communication gaps between engineering and other staff when interpreting the regulations.

Other participants, like ID4 from Focus Group Two, also thought that these models could assist internally with the communication between the engineers, business managers, legal, and customers to **clarify regulatory compliance requirements**.

ID4:"this actually has a really good application, clarifying requirements of the many ways that I try to do with customers; getting them to ask the ques-

tion in as much of an objective way as possible, which in itself is its own major effort. But this sort of provides a structured way to say, 'help me, help you' by scoping this down. "

ID2:"'my call at the top of the hour is with [a] company where we're actually trying to figure all [the cybersecurity requirements for medical devices] out. Right now, it's sort of this nightmarish spreadsheet process. And so yes, I do think it could be pretty useful, at least in that context. "

Clarifying regulatory requirements in a structured manner will allow everyone to voice how they view a particular compliance requirement and ensure that everyone is on the same page.

6.3.1.2 Ambiguity modeling is not standalone.

Five out of our six participants supported the usefulness of the modeling process. However, four out of the five participants who thought it was useful also thought it was not a standalone process and would be **part of the documentation for software artifact development**. An output linked to the ambiguity model would be needed, much like the legal guidance shown to them during Session One as a possible outcome. The models may highlight the heuristics behind decisions made internally within the organization. However, the model is only telling part of the story. The auditor must see how the software organization interpreted the regulatory ambiguity and define the organizational standard or policy that defines the interpretation.

Participants were also willing to point out that when going through this process, the

intention should not be to eliminate ambiguity in regulation. **Ambiguity in regulation is intentional**, as two participants pointed out. It is up to organizations or the industry to define their compliance standards internally, not the regulators who are not technical experts.

> **ID6:**"The purpose of intentionally writing regulations with ambiguity is to allow a business to play in the gray space...how each of us interprets the regulation and how we implement that regulation is necessary, necessary for us to gain market "

Therefore, while the model might be a useful visual aid for illustrating regulatory ambiguity that can hinder a software development process, it requires follow-on action. **Auditors want to see the output or link to the ambiguity models**, according to our participants.

> **ID5:**"Auditors do not want to be consultants, actually, most of them cannot be. So even if you presented this information ... It isn't just enough to say [its] not clear. Auditors want to see what are you doing with that and what have you done with it? It's not enough just to say it's unclear "

An example is policy defining the organization's standards to comply with the regulation. Another example is compliance testing linked to an ambiguity model using a developed "use case" from a specific regulation. An ambiguity model itself is less useful without outputs connected to the model showing what the software organization did to comply with an applicable regulation.

6.3.2 Ambiguity modeling is evidence of due diligence - SQ2

Finding 2: Ambiguity modeling signals to an auditor the organization's intention to comply with an applicable regulation. Therefore, it does provide evidence of regulatory compliance due diligence within a software organization's development process.

Four of six participants thought the ambiguity models evidence or could be used as evidence to tell a story of efforts toward compliance. The participants commented on preparing for audits, building documentation packets for review, and showing the heuristics behind certain decisions. Therefore, an ambiguity model would be part of the audit preparation documentation to show that the software organization identified a specific regulatory issue and connected it to some action taken to resolve the issue, as ID5 points out.

> **ID5:**"So where I see this model, sort of fitting in really nicely is the being able to communicate outside of words, that we took good notes, and we have great visuals into the things that we knew we would [encounter] challenges that we knew we would come across. And I think that is a huge signal to auditing bodies, that we're doing the due diligence...if it's not documented, you didn't do it right, you have to always have solid documentation."

Documentation is proof that the regulation and associated guidance were considered and discussed and why certain decisions occurred. Take, for example, ID2's comment.

> **ID2:**"This could be incredibly helpful for a company to prepare for an audit. Yeah, just in terms of something like HIPAA. I would dare guess that in most

organizations, if we don't give them a tool, we will have no documentation on why certain decisions were made, or, any documented proof that we actually thought about this, and that our decision was thoughtful. And, and so I love this for an internal tool. "

One group had a ten-minute conversation about how auditors must assess the severity of a non-compliance violation within an organization and how the models could highlight internal discussions the auditors do not usually see. During the ten-minute conversation, the participants said that these types of discussions and awareness regarding a regulation signal to an auditor an intention within the organization to comply and could positively impact the auditor's assessment of non-compliance investigations.

Although four participants saw the models as evidence of due diligence toward regulatory compliance, two participants did not. One participant did not think that ambiguity modeling was useful or provided any evidence of regulatory compliance that would be useful for an auditor to assess. The other participant thought it was useful but in a narrow capacity as a supplemental tool to other artifacts as seen in the following quote:

ID4:"I think anytime you can bring visibility to pain points in any model, it's great. I think this is a great supplemental to other models out there, because I don't think [in the] SDLC there's no clear space, where you would introduce the concept of something being ambiguous, because there isn't, and this is good. But in terms of a product, [from my] perspective on this, you always want to sort of reduce the friction of folks using this."

6.3.3 Improving the Ambiguity Modeling Process and the AHAB Tool-SQ3

Finding 3: Most of our participants found the Ambiguity Modeling Process and the AHAB tool useful. However, they also had ideas for improving the process and the AHAB tool.

Most of our participants were generally open to the idea of ambiguity modeling playing a role in developing other software artifacts as part of software development. However, they indicated that users of the technique would need a clear understanding of what the resulting model is intended for, i.e. what the link is between an ambiguity model and its contribution to downstream artifacts.

> **ID4:**"**if your output is unclear, people are just not really going to use it, because they won't see any difference, [then] just simply taking notes somewhere, and using whatever status quo exists in their organization. So, whether it's a spreadsheet or visual, I think it's going to be really important to highlight what exactly the expected output is [from this process], and where it [links] into an existing [ambiguity] model.** "

One of the goals of ambiguity modeling is to be able to document confusion in interpreting the regulation so that it can be made explicit and understandable for others and eventually resolved. Therefore, ambiguity modeling could contribute in various ways, depending on the context and the needs of a development project. For example, it might help plan modifications to an existing software product to comply with a new (or newly relevant) regulation. In other cases, ambiguity modeling can identify and mark areas of

the law that must be complied with in a future version of the software. For example, the need to identify areas for future compliance may be a result of the current state of technology, the lack of understanding of the regulation, or the need for further legal interpretation or case law to understand the compliance requirement. Lastly, the situation might call for ambiguity modeling to help develop legal guidance or internal organizational standards for future product development. Whatever the intent behind modeling a regulation, the participants' feedback suggests that the intent behind the ambiguity model must be clear. This ensures that those creating the model are not confused about its scope and goal, thereby enhancing the effectiveness of the modeling process.

Participants also had suggestions for a few more features to add to AHAB. One participant suggested a way to link an ambiguity to an audit finding.

> **ID3:**"what I see would be useful is to kind of map it to the audit findings. What I mean by that is, for some specific ambiguities, which are severe in nature, for example, we could have somebody review these findings by somebody who has expertise in auditing, and flag them in terms of a potentially a major [regulatory] nonconformance [violation] so the team knows. . . that is something that would be useful "

ID1 added to that comment by suggesting that "a risk evaluation of the ambiguity" feature within the AHAB tool would be useful, which ID2 agreed with. ID4 and ID5 thought of linking the models to an anomaly or bug tracker and how the interpretation of the bug may be ambiguous, but the impacts of the bug or ambiguity are not severe enough to stop software release.

Adding new features to AHAB would help connect the model to software design artifacts. These artifacts would be outputs linked to the model, documenting the development team's decisions to address the ambiguity and define their interpretation of regulatory compliance standards within an organization. Related to Finding 1, these linkages are also what auditors will want to see and track when assessing a software organization's regulatory compliance.

I received some excellent feedback on the potential of the ambiguity modeling process from both groups. There were also negative comments, but mostly positive responses overall. Participants saw it as an internal communication tool or method for requirements development. The models can assist an organization in preparing for an audit. More importantly, a majority agreement that this type of documentation highlights these conversations happening internally in the organization signals to an auditor that there is an intent to comply within the software organization. I also got suggestions for improving the tool and process for future work.

Therefore, based on these findings, we discuss some points for consideration.

6.4 Discussion

6.4.1 Intent to comply with regulation matters.

Discussion point 1: Intent to comply with applicable regulations matters. Processes and tools that support and document your intent to comply also matter.

Intent matters in the eyes of the law. It is the difference between willful neglect and an unfortunate error in judgment. For example, the U.S. HIPAA regulations have four

tiers of compliance violation penalties. These tiers assess the violator's level of culpability from Tier 1 (maximum annual penalty of US$25,000), where the violator did not know, to Tier 4 (maximum annual penalty of US$1.5 million), where the violator knew they had violated HIPAA's rules, thus demonstrating intentional non-compliance or reckless indifference and did nothing to correct their actions [17]. Signaling to a compliance violation auditor that your organization took reasonable care to comply with the applicable regulation can save a considerable amount of money based on this penalty structure.

Ambiguity modeling is a way to signal an intention to comply, which a majority of our participants agreed with. Ambiguity modeling is a process that shows and documents that discussions on regulatory requirements are occurring at more than the leadership level. Incorporating ambiguity modeling supports the story that reasonable amounts of due diligent efforts to comply with applicable regulations are occurring. What ambiguity modeling will not give an organization is plausible deniability. In other words, an organization cannot deny they were unaware of their requirement to comply with specific regulations because having the model proves an organization was aware that they might have some compliance requirement linked to the regulation.

Seeing these models, an auditor can reasonably assume:

- The organization read, thought about and discussed applicable regulations.
- The organization identified potential gray areas (i.e., ambiguities) within applicable regulations.
- The organization at multiple levels is aware of a regulatory compliance requirement that should be documented and implemented within the software development pro-

cess.

Making these assumptions, an auditor can ask detailed questions to form the complete compliance picture needed for their investigation and then make the call of intent and due diligence regarding an organization's culpability in a regulatory compliance investigation.

6.4.2 Ambiguity modeling does not reveal everything.

Discussion point 2: Ambiguity models do not tell the whole story in the software development process that an auditor would need to hear to assess regulatory compliance accurately.

Ambiguity modeling is not sufficient by itself, nor should it be. Ambiguity modeling is about documenting and communicating possible issues within a regulation to other stakeholder-holding parties within the software development process. Software developers and design teams will want to communicate these issues so that they may receive further actionable guidance that they can incorporate into the software design and document why they incorporated that design decision at the time. Therefore, more than ambiguity modeling is required. The models would need a link to a relevant software design artifact. An output that an auditor needs to see includes specific steps to meet regulatory compliance requirements.

6.4.3 Software developers have to navigate compliance requirements.

Discussion point 3: Include software developers in the analysis and development of regulatory compliance requirements. They have a professional obligation to understand regulatory compliance requirements within requirements engineering and analysis [29].

Regulatory analysis processes and tools like the Ambiguity Modeling Process and AHAB assist software developers and their organizations in navigating regulatory compliance requirements. Therefore, include software developers in the analysis. They have a professional obligation to understand regulatory compliance as it applies to their organization and software development process [29]. Furthermore, software development teams and the software engineering community are evolving from the traditional silo roles and fostering collaboration with DevOps and DevSecOps. Therefore, software developers can learn a lot about regulations, communicate with other stakeholders, and understand the establishment of specific policies or organizational standards as part of the profession's evolution. They might not immediately know why they are modeling ambiguities or value the process. However, they need to understand the regulatory requirements and the process that goes into developing those standards to comply with regulatory requirements.

After explaining and demonstrating the ambiguity modeling process, both groups initially resisted the concept of the Ambiguity Modeling Process. They thought software developers should not be the sole focus of a use case for this tool and process. Some participants pointed out that software organizations will have infrastructure and people to interpret the software developers' regulations. An example is ID5's comment:

> **ID5:**"Ideally, I don't think a software development team should be contributing directly to this, based on where I sit currently."

Participants in both groups made similar points early in Session One. The participants stated that a software design team or senior system architect in charge of requirement development and interpreting regulations for a software organization would have better uses for the Ambiguity Modeling Process and AHAB tool. These comments were not about the auditor's perspective but the initial feedback from the study's researchers. However, as discussion amongst the group continued, the focus groups started to consider possible uses when communicating with a compliance auditor. This discussion led to our findings to answer SQ1. They thought that not all software organizations are large and have extensive internal resources to dedicate to compliance requirements.

> **ID2:**"Smaller digital health and medical device companies [in the U.S.], they don't have internal audit departments. In fact, they don't even have an internal audit person. And so, things are being handled by... your VP of engineering to figure out why certain decisions were made. And if that person wasn't there, they don't remember, or they don't have something documented that they can then go back and look at, I think it's going to be very hard to show why they made the decisions they did."

Small organizations of less than 100 employees would have limited resources for an internal audit department to interpret regulations, as ID2 pointed out. If these organizations are start-ups, they are lucky to have someone who can take them by hand and go through the rigamarole required by the regulated industry to certify compliance with

a specific regulation such as HIPAA. Also, just because software developers in large organizations have the resources to interpret rules does not mean that they do not have a personal and professional ethical responsibility to know the laws and regulations that pertain to them and their organization. Although some software developers are happy to do what they are told and not ask why, that sentiment is problematic. The ACM's Code of Ethics requires software developers and engineers to know the laws and regulations applicable to their job [29]. Organizations that engage software developers and get them involved with this type of analysis raise their awareness and ethical responsibility [84]. The software developer becomes much more aware of expectations and the right way to address compliance concerns when they come up.

Software organizations need these regulatory analysis processes and tools, whether large, medium, or small. This is especially true for small organizations with limited resources to assist organizations in fulfilling regulatory compliance requirements. These tools engage software developers, helping them understand their regulatory obligations and communicate their concerns. This type of engagement and communication helps foster a compliant organizational culture. A culture that regulatory auditors want and expect to see when performing regulatory compliance audits. It demonstrates that software organizations are diligently working to comply with applicable regulations.

6.5 Threats to Validity

6.5.1 Threat to Internal Validity

While the protocol for both focus groups was the same, execution timelines differed. Focus Group Two was executed in back-to-back sessions versus Focus Group One, which had a 48-hour gap between sessions. I encountered issues with scheduling for Focus Group Two, which presented its own risk of not happening. Overall, the greater risk to the study was not to have two focus groups. Having only one would have severely limited confidence in the findings. However, the inconsistency in the timing between the two groups made it risky to combine the data.

6.5.2 Threat to External Validity

Our sample size limits our findings' generalization ability. All of our participants have relatable auditing experiences. One participant worked at the federal level as an auditor for a governance agency. Most of the study's participants came from a medical device background in the U.S., so most of the conversation centered around software development within the medical domain. Although relevant, our findings are limited to six people, mostly about software devices in one regulated domain. Therefore, a replication of this study recruiting participants operating within a different regulated domain may offer comparable or counter results that could address this threat to validity.

6.5.3 Threat to Reliability Validity

The comparison point in the literature is the previous multi-case study [86]. I did not replicate the study, but the findings and the previous studies are relatable. This type of work is unique, however. Given the sample size of our groups, a threat to reliability is data saturation. Most of the participants agreed on usefulness, but they had many different ideas on how to develop the process and improve the tool. Replicating the previous study and this one could offer more feedback to reach data saturation and improve this tool and process.

6.5.4 Threat to Construct Validity

This study was about gaining a different perspective on the usefulness of the ambiguity modeling process. I wanted to learn whether the ambiguity modeling process offers insights into the heuristics associated with interpreting regulatory ambiguities. Since none of our participants had prior experience building or assessing ambiguity models, it might have been hard for them to provide valid feedback on its usefulness. I mitigate this threat through the demonstration and extensive background information on the Ambiguity Modeling Process provided to the participants in the first session. However, it remains a threat that the participants might have provided different input and feedback if they had had more extensive experience with ambiguity modeling.

6.6 Summary

Overall, both groups gave excellent feedback on the potential of the ambiguity modeling process. Participants saw it as an internal communication tool or process for requirements development. They thought the models can assist an organization in preparing for an audit. More importantly, a majority agreed that this type of documentation highlights these conversations happening internally in the organization and signals to an auditor that there is an intent to comply within the organization. I also got suggestions for improving the tool for future research on the Ambiguity Modeling Process.

Based on these findings, I discuss three points for software engineering to consider. **Discussion point 1:** Intent to comply with applicable regulations matters. These processes and tools support and document your intent to comply. Therefore, organizations should incorporate these processes and tools within the software development process. **Discussion point 2:** Ambiguity modeling is not sufficient by itself, nor should it be. Ambiguity models do not tell the whole story in the software development process that an auditor would need to hear to assess regulatory compliance accurately. Auditors will need to see outputs and their links to ambiguity models to include specific appropriate steps to meet regulatory compliance requirements.

Discussion point 3: Include software developers in the analysis and requirements development of regulatory compliance. Software developers at all levels can learn much about regulations, communicate with other stakeholders, and understand the establishment of policies or organizational standards. Moreover, the transition of software engineering from the conventional isolated roles of software developers to the more inclusive

software design teams, which promote team collaboration through agile software development methods, brings about significant advantages. Regulatory analysis plays a crucial role in this evolution by actively involving software developers, thereby enhancing their understanding and awareness as part of their professional responsibility and growth. This analysis can also help avoid potential technical challenges that software developers and engineers may encounter during software development and foster compliance awareness and culture within a software organization. That is something to show and demonstrate to regulatory auditors assessing a software organization for compliance.

Chapter 7

Study Synthesis

7.1 Dissertation's Goals

My dissertation had three goals. The first goal was to understand the regulatory compliance landscape within the software development industry. I accomplished this goal through the interview and survey studies presented in Part One of this dissertation. The second goal was to explore a method allowing software developers to analyze the 'gray areas' within regulations. Part two of this dissertation tested the Ambiguity Modeling Process, a method of identifying and documenting areas of regulatory ambiguity, through a multi-case study to achieve this goal. Part of the study's findings support that this method is not just a theoretical concept, but a practical tool useful for internal communication within a software development team and external stakeholders involved in the requirements analysis process [95, 94]. These findings are based on participants' responses from the multi-case study. The third goal was to validate the Ambiguity Modeling Process's usefulness for requirements analysis and documentation as evidence of due diligence toward regulatory compliance. I sought the auditor's perspective through a focus group study within Part Three of the dissertation to accomplish goal three and show that it can be a valuable resource for auditors, aiding decision-making and evaluating due diligence toward compliance. Overall, this method serves both organizations wanting to demonstrate and communicate their due diligence toward compliance with an external

party and for third parties reviewing an organization's software development process for regulatory compliance. In this chapter, first, I highlight two studies that underscore regulatory compliance challenges within the software industry. Then, I compare the findings from my studies to these two studies, reflecting on the dissertation's research questions and how each of these studies connects and contributes to answering the questions and the implications of my research.

7.2 Closely related prior research

Abdullah et al. de [20]signed an empirical study that sought to capture the software industry's compliance management issues. Abdullah et al. interviewed 11 Australian compliance management professionals on the challenges the software industry faced in 2007. They summarized their findings into 14 challenges categorized under customer, regulations, and solution challenges. The tables from the study cite the number of interview participants who identified them under Source and the frequency of identification within the study (See Table 7.1).

Another case study by Usman et al. identifies the challenges surrounding compliance requirements [122]. The authors conducted a literature review and found little research on compliance challenges in practice within the software industry. This finding was also confirmed by a literature review that I conducted that concluded about the same time in 2019 [85]. They conducted their case study with an industry partner to characterize the industry's regulatory compliance challenges, which they categorized into three groups: "Requirements specifications," "Process," and "Resource" related chal-

Table 7.1: Compliance Challenges identified by Abdullah et.al. in 2007 [20]

ID	Compliance Challenges/Factors description	Related to	Citation
AB1	"Lack of Compliance Culture "	Customer	[20]
AB2	"High Cost"	Customer	[20]
AB3	"Lack of Efficient Risk Management "	Customer	[20]
AB4	"Difficulties in Creating Evidence of Compliance"	Customer	[20]
AB5	"Lack of Perception of Compliance as a Value-add"	Customer	[20]
AB6	"Lack of Understanding of its Relevance to Business "	Customer	[20]
AB7	"Lack of Communication among Staff "	Customer	[20]
AB8	"Frequent Changes in Regulations"	Regulations	[20]
AB9	"Legislation Weaknesses"	Regulations	[20]
AB10	"Inconsistencies"	Regulations	[20]
AB11	"Overlap in Regulations"	Regulations	[20]
AB12	"Lack of Holistic Practices "	Solutions	[20]
AB13	"Lack of IT Support/Tools"	Solutions	[20]
AB14	"Lack of Compliance Knowledge Base"	Solutions	[20]

lenges shown in Tables 7.2.

Table 7.2: Compliance Challenges related to Compliance Requirements identified by Usman et.al. in 2021 [122]

ID	Compliance Challenges/Factors description	Related to	Citation
US1	"Interpretation of compliance requirements in the context of a specific product"	Requirements Specification	[122]
US2	"Differences in the understanding of the compliance requirements"	Requirements Specification	[122]
US3	"Difficulties in Creating Evidence of Compliance"	Requirements Specification	[122]
US4	"Trade-offs and conflicts between different compliance requirements"	Requirements Specification	[122]
US5	"Missing linkage with the business use cases "	Requirements Specification	[122]
US6	"Linkage between the compliance requirements and design rules"	Requirements Specification	[122]
US7	" Coordination and alignment of the compliance tasks between sub-systems' teams"	Process	[122]
US8	"Compliance requirements not communicated properly"	Process	[122]

US9	" Different compliance requirements (e.g., security) managed differently"	Process	[122]
US10	"Missing dedicated process at the sub-system level"	Process	[122]
US11	" Lack of coordination between verification and development teams"	Process	[122]
US12	"Change management of compliance requirements"	Process	[122]
US13	"Establishing a balance between compliance and business requirements"	Process	[122]
US14	"'Prioritising the right compliance requirements"	Process	[122]
US15	"Lack of automation"	Process	[122]
US16	"Lack of dedicated resources and time to handle compliance requirements"	Resource	[122]
US17	" Lack of awareness among developers about compliance requirements"	Resource	[122]
US18	"Lack of awareness among developers about design rules"	Resource	[122]
US19	"Tools used to manage compliance requirements are not appropriate"	Resource	[122]

I highlight these two studies because many of the findings and lessons learned I

have reported throughout my studies relate to the challenges that these studies identified (See Section 7.3, Section 7.4, and Section 7.5). Second, the implications of my work, specifically on the usefulness of the "Ambiguity Modeling Process," can help resolve the challenges pointed out within these two studies. Therefore, as I reflect on my work, I will refer back to the difficulties outlined in Table 7.1 and Table 7.2 to support the key insights from my studies.

7.3 Reflections from the Interview Study and Survey

RQ1: What are the software industries perceptions regarding Regulatory and Security Standard Compliance?

In their study, Abdullah et al. [20] listed 14 challenges relating to customers, regulations, and solutions to manage compliance. Usman et al. [122] noted similar findings within their research, stating,

> "Common challenges to our study [compared with Abdullah et al. [20]] are the lack of connecting compliance to business objectives, the lack of communicating a common understanding of compliance continuously to employees, inconsistencies in applying regulations, the lack of compliance practices applied throughout an organization, and the lack of tool support for compliance management and monitoring."

The findings I reported from the interview study and the survey that relate to these challenges listed in Table 7.1 and 7.2 are:

- the impact of an organization's culture on compliance management (**Survey- Find-**

ing 3 to AB1)

- the perception of compliance as an investment versus a cost (**Interview Study - Finding 3 to AB5**)
- assessing compliance requirements concerning the business model (**Interview Study - Finding 6-8 to AB6, US5, and US13**)
- communication of the compliance processes and requirements (**Interview Study - Finding 1 to AB7, US8, and US17**)
- having a strategy or plan to respond to regulatory compliance change and demonstrate adherence to regulation (**Interview Study - Finding 9 to AB3, AB8, AB12, AB13, US15, US16, and US19)**
- compliance requirements and process applied throughout the software development process and organization through holistic practices(**Survey -Finding 2 to AB12, US6, US9, and US11**)

My interview study and survey report on compliance practices seen within the industry and how they influence the perception of the participants within the study. These practices as noted in my findings relate to the challenges listed by Abdullah et al. [20](See Table 7.1) and Usman et al. [122](See Table 7.2) and could be seen as ways the software industry is addressing the challenges. For example, suppose a software organization takes the time to invest in ways to interpret regulations, assess the quality and compliance of their product, have the plan to respond to regulatory change and communicate that plan.

In that case, the organization's employees are confident in their organization's compliance practices. Employees' confidence in the software organization's compliance practice starts to decrease when the perception is that regulatory compliance is not a priority and is an impediment within a regulated domain. The organization's culture focuses more on what it must do to comply than on what it should do. Furthermore, costs associated with compliance start to become a risk calculation of liability or legal penalties.

The interview study and survey highlight that some organizations have invested in systems to manage their regulatory landscape, but others have not. Also, going back to the list of challenges, while I highlighted 18 challenges that the software industry is addressing, there are gaps, specifically for compliance challenges related to Regulation (i.e., Table 7.1: AB4, AB9-11) and Requirements Specifications (i.e. Table 7.2: US1-4). For organizations still trying to figure out regulation and how to communicate some of their confusion regarding regulation, I studied the effectiveness of using regulatory ambiguity modeling, which can help resolve these Regulation and Requirement Specification challenges identified by Abdullah et al. [20] (i.e., Table 7.1: AB4, AB9-11) and Usman et al. [122] (i.e. Table 7.2: US1-4). It also provides a useful way to document and demonstrate an organization's methodology to comply with regulations, which is the focus of Parts Two and Three of my dissertation.

7.4 Reflections from the Multi-Case Study

RQ2: How are regulatory ambiguities within legal text interpreted and reasoned by software stakeholders/practitioners individually and as a group?

In the dissertation, I test a concept to document some of the confusion (i.e., regulatory ambiguity) in interpreting a regulation through the ambiguity modeling process. Part two of the dissertation focuses on the ambiguity modeling process and how software developers perceive regulatory ambiguities. With the multi-case study, I considered whether a group of software developers could communicate, discuss regulatory ambiguities, and collectively build an ambiguity model as a team. I gave them guidance and tools to aid in building ambiguity models, which they were able to accomplish. They had some difficulties, but the reported challenges did not stop them from achieving the collective group task. The difficulties aided them in identifying the regulatory ambiguities and facilitated discussion that would help them to seek further guidance from external experts to establish internal organizational standards for compliance. Lastly, some of the participants in the study started to value the ambiguity modeling process when they initially held doubts about the process. The participants had to consider and understand other group members' perspectives to accomplish the ambiguity modeling task. T They realized that the ambiguity modeling process is more than the models themselves. It is also about understanding the regulatory requirements in support of the requirements analysis and documentation. Some participants shared real-world situations they had experienced with other software development stakeholders and provided thoughts on how they would apply this process based on their past experiences. This type of discussion and analysis is vital for developing and documenting regulatory compliance requirements. It facilitates compliance communication and guidance that internalizes and promotes an organizational culture of compliance. Lastly, ambiguity modeling documents that these discussions are happening internally within a software organization. The models would then connect to software en-

gineering artifacts that document the decision that was taken. We hypothesized that this process provides evidence of diligence efforts of regulatory compliance and compliance being "baked" into the software development process that signals an intention toward compliance to external parties like regulatory auditors or assessors.

The challenges from Abdullah and Usman's research that ambiguity modeling addresses are:

- "Difficulties in Creating Evidence of Compliance" [20] (i.e., Table 7.1: AB4)
- "Inconsistencies with regulation" [20] (i.e., Table 7.1: AB10)
- "Differences in the understanding of the compliance requirements" [122] (i.e., Table 7.2:US2)
- "Abstractness of the compliance requirements" [122](i.e., Table 7.2:US3)

A collateral benefit in making software teams go through this process is the change of perception that compliance analysis and development can add value to the software development process and communicate its relevance, which addresses:

- "Lack of Perception of Compliance as a Value-add" [20] (i.e., Table 7.1: AB5)
- "Lack of Communication among Staff" [20] (i.e., Table 7.1: AB7)

Our final two research questions involved validating the usefulness of the Ambiguity Modeling Process from the auditor's perspective. Auditors tasked with assessing software engineering artifacts for compliance would view ambiguity models as evidence of due diligence, thus making them useful for software organizations trying to prove their efforts to comply with applicable regulations.

7.5 Reflections from the Focus Group

RQ3: Is the ambiguity modeling of a regulation useful for a software organization?

RQ4: Does modeling regulatory ambiguities document due diligence toward regulatory compliance from an auditor's perspective?

Within any research area, there are always different perspectives to consider. For regulatory compliance in software development, there is the business perspective, the operational or engineering perspective, and the enforcement perspective. Part one of the dissertation generalizes the business and operational perspective regarding regulatory compliance. Part Two of the dissertation gives an engineering perspective, specifically on the ambiguity modeling process. I had yet to uncover much insight into the enforcement perspective, with only one participant from the interview study who could offer insights from the enforcement perspective and one survey respondent who could provide a legal perspective. Therefore, for my validation study, I created a focus group study to address the enforcement perspective by recruiting individuals with auditing experience to provide feedback on the usefulness of the Ambiguity Modeling Process and AHAB tool.

Five out of six participants from the focus group study saw the Ambiguity Modeling Process's potential. The participants thought it was useful as an internal organizational tool for communication and compliance documentation, which addresses some of the Usman et al. [122] Requirements Specification related compliance challenges (i.e., Table 7.2:US1-3) and one of Abdullah et al. [20] (i.e., Table 7.1:AB 10). In addition, the participants felt that the ambiguity models provide evidence of compliance– addressing one more of Abdullah et al. [20] challenges (i.e., Table 7.1:AB 4); that the models can

also signal an intent to comply and that auditors would like to see that intention. However, the participants also identified a gap in the current process and tools by pointing out that models only tell part of the story. The models must be linked to engineering artifacts that show the technical or administrative steps an organization took to comply with the regulation.

The focus groups commented on my assumption about the Ambiguity Modeling process. The assumption is that the models document some of the thoughts and discussions that occur about a regulation. Documentation on these discussions is not visible within engineering artifacts. Therefore, it is hard to prove that these discussions occur at the software developers' level. Similar to the multi-case studies findings, the focus group commented on how the Ambiguity Modeling Process works as an internal communication support tool, particularly between some of the different stakeholder groups involved in the software development process, validate some of the real-world examples that my multi-case study participants gave regarding validating regulatory compliance requirements with customers. I thought that was an insightful comparison and highlighted the similarity between the two studies. Overall, when I consider some of the comments I received regarding the work and some of the relatable findings between the multi-case research and the focus group, I did validate the Ambiguity modeling process as a proof of concept and found support for its usefulness within the software development process.

7.6 Legal Community's Perspective

A theme throughout all these studies is perspective.

- The software industry's general perspective on regulatory and security standard compliance (i.e., Part 1).
- The software developer's perspective when interpreting regulatory ambiguities (i.e., Part 2).
- The auditor's perspective on ambiguity models and what it may mean to them when assessing a software organization's regulatory compliance practices (i.e., Part 3).

I did not capture the legal perspective. I started to investigate it by working with Jeffrey Kosseff, Esq. on developing legal guidance for the Virginia Consumer Data Protection Act (VCDPA). However, the work is in a nascent stage and needs further development to provide the legal community's perspective on ambiguity modeling. The legal community weighing in on this analysis would have provided additional feedback on the Ambiguity Modeling Process–such feedback could have expanded the uses or improved the Ambiguity Modeling Process and the AHAB tool.

7.7 Contributions

The dissertation examines the Ambiguity Modeling Process as a way to document and communicate regulatory compliance analysis internally within a software development team and externally to other stakeholders interested in the requirements analysis process [95, 94]. The process allows software organizations to visualize the rationale and decisions that occur while interpreting the regulations. The process also communicates confusion concerns the development team may have about a regulation. The process connects the ambiguity models to development artifacts and actions taken within the software

organization to interpret and resolve the issue from the identified regulatory ambiguity. Thus, a software organization wanting to demonstrate to an outside party inspecting (i.e., auditor) their commitment to comply with relevant regulations within their software development process could use ambiguity models as evidence of their due diligence.

This dissertation offers three contributions:

- The first contribution is to report insight into the software industry's perspective regarding regulatory compliance during software development.
- The second contribution is to promote and further develop the Ambiguity Modeling Process, which analyzes regulatory ambiguities and documents decision-making regarding compliance during software development.
- The third contribution investigates whether the ambiguity modeling process is useful for demonstrating due diligence. It involves working with auditors who assess a software organization's technical specifications, policies, and procedures for compliance with applicable regulations.

The empirical studies, which were conducted to test the Ambiguity Modeling Process from the software developer's perspective and validate its usefulness from an auditor's perspective, yielded takeaways for the software engineering community's consideration in the context of regulatory compliance. The summary of the takeaways are as follows:

- A software organization's intention for regulatory compliance matters to an auditor. Software engineering must socialize the importance of compliance, incorporate

processes and tools that support this intent, and document this intent in the software development process.

- Ambiguity modeling, while a valuable tool, is not a standalone solution. It involves documenting and communicating compliance issues that stem from ambiguous phrasing in regulations, and resolving these ambiguities, at least at an organizational level. Therefore, the models, as interpreted by the software development team, would need to be linked to relevant software design artifacts that address regulatory compliance. This integration is a critical step in building a comprehensive and effective compliance strategy.

- The analysis and development of regulatory compliance requirements should not be limited to quality management or regulatory experts. The active involvement of software developers is crucial. Their understanding of the purpose of a specific regulatory specification is as essential as their ability to implement compliance requirements within the software development process. By asking meaningful questions and understanding regulatory compliance requirements, software development professionals can play a significant role in ensuring compliance. Promoting tools and methods that foster collaboration and understanding among all stakeholders is valuable in avoiding unintentional compliance violations and wasteful compliance efforts. Lastly, including software developers in this process creates buy-in for the regulatory analysis and requirements development process.

My dissertation uncovers the potential of software developers in addressing regulatory ambiguities. They can do so by documenting compliance concerns and implementa-

tion rationales with well-developed processes and tools. This is achievable individually and as a group, bridging communication gaps with other stakeholders. This collaboration can lead to thoughtful software design decisions and support other artifact development, such as requirements traceability, resolving and defining regulatory compliance requirements, and building testing suites for compliance verification. This potential, when realized, can significantly enhance the effectiveness and efficiency of the regulatory compliance process, inspiring software developers to take a more proactive role in this aspect of software development.

7.8 Implications

The interview study and survey highlight that investing in sound systems to manage the regulatory landscape and develop a culture of compliance promotes confidence within an organization's regulatory compliance practices. The Multi-Case Study and Focus Group show that Ambiguity Modeling is a potential tool that can help identify and manage regulatory ambiguities, demonstrate an organization's intention toward regulatory compliance, and create buy-in for regulatory compliance analysis and development within software design, thus helping to promote organizational compliance cultures. Based on these findings, my research has implications for software developers, regulatory auditors, other software stakeholders, and the software engineering community.

For software developers, regulatory compliance is an ethical and professional obligation within the software engineering profession [107]. Understanding these obligations starts with regulatory compliance analysis and requirements development. My studies

and findings demonstrate that software developers can do this type of analysis within the software development and maintenance cycles. Furthermore, software developers can communicate "gray area" for further guidance and resolution, demonstrating a continuous effort within the Software Development Lifecycle to address and maintain regulatory compliance. For auditors assessing organizations for regulatory compliance, ambiguity models demonstrate that software organizations are aware of and discuss their regulatory compliance requirements. This signals an organization's intention to comply and helps relay the organization's due diligent efforts to comply. For other stakeholder groups, such as legal advisors wanting to ensure software organizations meet all relevant regulatory obligations, ambiguity modeling helps identify and communicate regulatory compliance issues. Thus, the models can facilitate meaningful conversations that can result in software design artifacts that "bake in" technical compliance specifications. Models also bridge communication gaps with a software engineering team that can hinder regulatory compliance analysis and development.

To the software engineering community, Abdullah et al. [20] and Usman et al. [122] listed challenges still seen today within the software engineering community regarding regulatory compliance. My work directly addresses some of those challenges (See Section 7.3) while indirectly promoting compliance communication amongst software stakeholders and assisting in developing a compliance culture within software organizations. My research also contributes to the research field by capturing some of the software industry perceptions regarding regulatory compliance, which is underrepresented within academic literature for software engineering [85, 122]. To that end, the findings and lessons learned in my research can serve as a resource for further study in

software engineering.

7.9 Summary

To summarize, software organizations and the industry are trying to understand and demonstrate compliance. It is just challenging when the regulatory language is vague and not clear. Ambiguity modeling is a solution to discuss obscurity and create guidance to resolve it, at least within an internal organization. Furthermore, if linked to engineering artifacts or outputs, auditors will view ambiguity models and documentation as evidence that the regulatory compliance conversation is occurring and that the organization is attempting to comply. This type of analysis signals an intention for regulatory compliance, which is evidence of due diligence—making the Ambiguity Modeling Process useful from an enforcement and operational perspective. Although there is always more to do, I have accomplished my goals regarding this dissertation. My research has limitations, and I have pointed out the threats to validity associated with each of my studies within their chapters and what more can be done to address them. The next chapter is my conclusion, highlighting recommendations for the future of software development and the ambiguity modeling process.

Chapter 8

Conclusion and Future Work

Regulatory ambiguity will always exist within regulation. These ambiguities are intentionally placed in regulation to allow multiple valid interpretations regarding regulatory compliance. This intentional ambiguity provides flexibility within the law and growth within the industry. Despite multiple interpretations to comply with a regulation, a software organization needs to understand the intent of the regulation and diligent efforts must be made to comply with the intent of the regulation.

In this dissertation, I have promoted a process for analyzing regulatory ambiguity that can link to other software design artifacts. Software developers and design teams that struggle to understand the regulatory compliance requirements because of regulatory ambiguity can benefit from the ambiguity modeling process. The ambiguity modeling process provides a means to examine and document the software requirement analysis process. It also fosters critical thinking and encourages consideration of the broader ethical implications of the regulation. This type of documentation signals to third parties reviewing a software organization for regulatory compliance, the organization's intentions to comply. The documentation can further strengthen the organization's position that they are acting in accordance with the intent of a regulation. In addition, the documentation fosters confidence in compliance within their software development process internally in an organization and externally to other interested third-parties.

8.1 Other uses for Ambiguity Modeling

In this dissertation, I focused on Ambiguity Modeling, a method for software organizations to earnestly show due diligence efforts to comply with laws and regulations. The Ambiguity Modeling Process has users identify ambiguous or unclear aspects of regulations and laws, helping organizations interpret and implement regulatory compliance requirements from regulations and laws more effectively. However, Ambiguity Modeling can have other uses such as in the domain of education.

Technology ethics and policy are a crucial part of the educational and professional development journey for the software development community. Ambiguity modeling can be incorporated to be a part of that journey. It can promote ethical compliance within the software development community by emphasizing the importance of understanding the law's intent and regulatory requirements. Ambiguity modeling can educate software developers about regulatory ambiguities and their potential impact on software development and design. By encouraging software developers to articulate their understanding of ambiguity and its potential impact on a proposed software system, the Ambiguity Modeling Process may foster a deeper connection between the regulation's intention and actions to comply within an organization's software development process. The Ambiguity Modeling Process bridges regulatory compliance and ethical considerations, guiding software developers toward more responsible and ethical human rights, fairness, safety, privacy, and security practices. Programs meant to advance cybersecurity education and workforce development, could include techniques like the Ambiguity Modeling Process as a means for regulatory ambiguity and compliance analysis. For example, the National Institute

of Standards and Technology's (NIST) National Initiative for Cybersecurity Education (NICE), can expand their scope to include technology compliance and ethics alongside its cybersecurity initiative [19].

Another use of Ambiguity Modeling is to reveal possible exploitable vulnerabilities within the regulations and laws to regulators and enforcement agencies. An example of exploitation is malicious compliance to regulations and laws. As mentioned earlier in Chapter 1, I define malicious compliance as when organizations follow the "letter of the law," thus they are technically compliant while undermining the law's intention [1]. In the former instance, malicious compliance exposes regulators, enforcement agencies, and organizations to criticism and years of litigation to resolve these regulatory matters. Consider Boeing and their initial assessment of the Maneuver Characteristics Augmentation System (MCAS) risk in the 737-MAX to comply with the U.S. Federal Aviation (FAA) certification for airworthiness. This assessment was malicious compliance. Boeing twisted the "letter" of FAA regulation and evaluated the risk of the MCAS toward the safe operations of the 737-MAX aircraft as "not expected to produce any serious injury and is defined more as something that would increase the cockpit crew's workload" [60]. This assessment was to fit their needs to get a new aircraft into commercial operation and regain market share from Airbus (See Chapter 1). Twisting the FAA's regulations undermined the intent of the self-certification process and the FAA's safety guidance for aircraft

[1] Another definition of malicious compliance as applied to software include Tom DeMarco and Timothy Lister's Peopleware definition [46] that looks at complying with software process inefficiencies based on "written rules" even though they complicate software development processes. Ambiguity modeling may help with this form of malicious compliance but was not studied in this dissertation

design and operations. The two crashes and their follow-up investigation have revealed how vulnerable the FAA's airworthiness certification process is when airline manufacturers do not follow the intent of the process and guidance [72, 8, 10, 13, 60, 90].

Loosely written regulations can allow for legal loopholes contrary to the ethical considerations the regulation is trying to address. In contrast, when regulations are too strict, they become a checklist of requirements where organizations follow the bare minimum to comply without addressing the regulation's intent. This restriction can also hamper problem-solving innovations within the software industry because of a single, dictated standard. Therefore, the organization might technically comply but fail to achieve intended outcomes or improvements to the software or processes for meaningful benefits Malicious compliance undermines a regulation and its enforcement's effectiveness, thus eroding society's trust in regulatory procedures and the agencies that enforce them. The ambiguity modeling process could help show regulators how organizations could subvert the law by exploiting the ambiguities, potentially for malicious compliance.

8.2 Improving Ambiguity Modeling with Future Research

Although we have tested the process and considered its usefulness, we can do more to improve the Ambiguity Modeling Process and the tool used to build models (i.e., AHAB) to further the research. One improvement could be offering an update field or a risk score within the AHAB tool. These fields would link the models to follow-up activities that show the software development team's direction in interpreting and resolving the regulatory ambiguity within the software design process. Another is testing the am-

biguity modeling with regulators and enforcement agency so they might realize some of the loopholes or 'gray areas' within the regulation that might be exploited. In the case of Volkswagen, the "Diesel Gate" scandal was not that Volkswagen and other auto manufacturers used a defeat device, it was the fact they got away with it for six years without being caught [67, 4]. In addition, after the scandal broke, the EPA's response was that they would include road testing in addition to lab testing in their new emission testing. They also decided they would not disclose further information on the road testing procedures to discourage abuse from the automotive industry [67, 4, 77, 66, 48]. Therefore, future research for ambiguity modeling could be to assist governing agency with enforcement.

We have also tested how graduate students with industry software development experience reading a legal text might develop and consolidate an ambiguity model. A recommendation for future work is to test ambiguity modeling through a partnership with an organization designing software that must comply with a regulation using the Ambiguity Modeling Process in a real-world case study. Beyond using professionals to build the models, a researcher could extend the study to see how professionals might link design artifacts to the models that address regulatory ambiguities. The study could further be extended to study how a design team would take the next step and build test suites connected to the model. Lastly, suppose the organization was undergoing a regulatory certification process. In that case, a future research study can show the models to an auditor and get their feedback on the models based on the specific regulations while executing their duties and responsibilities. Such studies could provide more feedback as an iterative evaluation of this work and help address gaps within the process—all to build a core framework of concepts, models, and templates for regulatory compliance.

Chapter A

Interview Study Appendix

A.1 Interview Study's Protocol

We structured two interview protocol divided into five sections. Each section includes the questions asked to the participant to include follow-on questions. Further details on protocol are in the interview guide at the following DOI: `https://doi.org/10.6084/m9.figshare.14842242.v1`.

A.1.1 Project Manager's and Developer's Interview Protocol

1. Participant's Background

    (a) Give a brief history of your professional background with the software system?

    (b) Why are you interested in regulatory and security standards?

2. Participant's Organization and their role

    (a) What is your role and responsibilities in your current organization?

    (b) Follow-on: How long have you been doing this? (If not already stated.)

    (c) What are your organization's mission and goals?

(d) Clarifying questions: Does your organization/the organization you advise, develop, or evaluate software or systems? (If not already stated)

3. Participant's experience with the SDLC

   (a) What is your experience with the software/system development process?

   (b) Follow-on: Why do you/ your organization use that process?

   (c) How involved are you in this process? (Meaning involved in the beginning as a key stakeholder required to provide requirements, brought in during implementation to evaluate the software for security compliance, etc.)

   (d) Follow-on: Can you give some examples?

   (e) Have you ever deviated from your software development process?

   (f) Follow on: A presentation of the rationale: (i.e. what decision factors or influences might have warranted a break from the traditional process?)

4. Participant's experience with Regulatory or Security Standard Compliance

   (a) Segway: Describe why these questions are important or define Regulatory or security compliance RC/SC)

   (b) As a PM/SW Developer in the field of (i.e. medical, safety, automotive, government, financial, etc.), I'm sure there is regulations or security standards you have to comply with. Can you give me brief description of what those standards and regulations are?

   (c) Follow on: Why those standards?

(d) When does regulatory or security compliance fit in your organization software development process? Looking at what phase of the SDLC.

(e) How do you track and manage compliance? (Examples from current or previous projects would be great)

(f) After the release of the software or system, have you had to re-evaluate Regulatory or Security Compliance Standards against the software/system?

(g) . Have you ever had any issues or challenges in complying with a Regulatory or Security Standard?

(h) Follow-on: Would you consider it a form of technical debt?

(i) Follow-on: Did and how did you track such changes (through configuration or knowledge management or another form of documentation tracking)

(j) Follow-on to 4.7: We have asked if you experienced and challenges of issues in complying with Regulation or Security Standards, have there been benefits with having RC/SSC as part of your organization's or your SDP process?

(k) Current Events question:

   i. COVID-19 impacts?

   ii. Change in Administrations impacts?

   iii. Are there any recent changes might have had some regulatory or security standard compliance effects on the software or the organization?

(l) List of questions asked near the end of the interview based on the participants' job role and/or background.

i. What would be something that you would like to see done that, as far as RC/SSC researcher is concerns, that could benefit the Software Development Industry?

ii. . If you had one wish for your organization or the Software Industry regarding RC/SSC, what would it be?

iii. Do you have any closing words or tidbits of wisdom to share about the Software Industry and RC/SSC?

iv. If you could go back and talk to a younger version, what would you say to you?

v. Do you have any thoughts or how to either improve the science or technical expertise on the regulatory side (i.e., the creation of regulation) that might change your perspective on regulatory compliance endeavors?

(m) Is there anything about your views or experiences with applying or evaluating Regulatory or Security Compliance Standards that you would like to add?

5. Summary

(a) Is there anything I should have asked but didn't?

(b) Can you recommend anyone else that would be a source of this topic?

A.1.2 Legal Expert's or Auditor's Interview Protocol

1. Participant's Background
    (a) Give a brief history of your professional background with the software system?
    (b) Why are you interested in regulatory and security standards?
2. Participant's Organization and their role
    (a) What is your role and responsibilities in your current organization?
    (b) Follow-on: How long have you been doing this? (If not already stated.)
    (c) What are your organization's mission and goals?
    (d) Clarifying questions: Does your organization/the organization you advise, develop, or evaluate software or systems? (If not already stated)
3. Participant's experience with the SDLC
    (a) What is your role in their development process?
    (b) Have you ever been asked to weigh in on decisions made during the development of a software package? (Details if they can be provided, please)
4. Participant's experience with Regulatory or Security Standard Compliance (RC/SSC)
    (a) Auditor or Security Advisor
        i. As an Auditor/Security advisor in your software/system development field, you advise on security standards, correct?

ii. Do you think there is a connection between regulation and security standards?

iii. Can you describe some of the challenges with complying with regulation and security standards in your role as an Auditor/Security advisor?

iv. Follow-on: Is this something you help track and manage?

(b) Legal Expert

i. As a Legal expert, you advise on regulation your organization must comply with, correct?

ii. In terms of RC/SSC, what is the most challenging thing about your role in a software development process?

iii. For legal purposes, do you have any requirements to track or manage your organization's compliance?

(c) Follow-on: Have you encountered any challenges in complying with or managing a software/system toward regulatory compliance?

(d) Have you had any experience with enforcement actions that were challenging?

(e) Are there other examples of regulatory or security standard compliance from your past experiences that you would like to add?

5. Summary

(a) Is there anything I should have asked but didn't?

(b) Can you recommend anyone else that would be a source of this topic?

(c) Thank them for their time and participation.

A.2 Interview Study's Coding Scheme

The following is the initial coding scheme used in the interview study analysis and outlines in the Interview Guide available at DOI: `https://doi.org/10.6084/m9.figshare.14842242.v1`.

Code	Subcodes	Definition
Background/Work History		Description of the Participant's background and work history (Note this code is attribute related); **Heuristic: Whenever the participant is describing something not related to their current job, code Background/Work History and look for the buzz words relatable to the subcode.**
	College Educated	Received any four-year degree from a college or university.
	Non-College Educated	On-the-job training, high school diploma, industry certification, associate's degree, or partially completed four year degrees.
	Technical Background	Participants that have previously worked as a Software or Application Developer, Engineer, Architect or Coder, where their focus is technical implementation of software or system.
	Non-Technical Background	Participants that have previously worked as a Manager, Team leader, or Director, where their focus is overall development and management of software or system. Job titles can include Product Manager, Customer Relations rep, or Data Analysis.
	Compliance Background	Participants that have worked within or supported a regulated field, like healthcare, for more than two years, where regulated compliance is emphasized or their job, examples can include Privacy Engineer, Civil Engineer, or Compliance officer, where their focus is compliance.
	Non-Compliance Background	Participants that have worked within the Software Industry but have not held jobs or had much focus on regulatory or security standard compliance as part of their job. Job descriptions within the Technical

		Background code may fall into this category.
	Cybersecurity Background	Participants have job background in security or risk management, or they have performed responsibilities in implementing, assessing, or enforcing technical security features for software or technical system for 2 or more year. Job Titles can include Security Engineer or Developer, Information Assurance Manager, Quality Assurance Manager, Risk Manager.
	Non-Cybersecurity Background	Participants that have worked within the Software Industry but have not held jobs or had much focus on Cybersecurity as part of their Job. Job descriptions within the Technical Background code may fall into this category.
	10+ years of experience	10+ years of experience working within industry and/or extensive research with the Software Industry.
	Less than 10 years of experience	Less than 10 years of experience working within industry and/or extensive research with the Software Industry.
Current Job		Description of the Participant's current Organization, Job, Roles and Responsibilities, and Customer base. Heuristic: When participant is describing their current job, what they focus on, who their customers are, Code CurrentJob
	Compliance Focused	Description that indicates the participant's job is manage or assess compliance within the product
	Cybersecurity focused	Description that indicates the participant's job is to mitigate or manage vulnerabilities within the product
	Risk Focused	Description that indicates the participant's job is to mitigate or manage risk within the product
	Functionality Focused	Description that indicates the participant's job is focused on producing a product or new feature for end-users

	Business Focused	Description that indicates the participant's job is focused on business side (i.e., cost and timeline to produce products, customer requirements)
	Policy Focused	Description that indicates the participant's job is focused on adherence to internal policies and procedures. This focus can overlap with the Compliance or Cybersecurity Focus subcode.
	Research Focused	Description that indicates that the participant's job is to improve how the organization does business through research
	Large Organization	Participant's organization More than 10,000 employees, more than $5 Million in revenue, and abundant human & physical resources.
	Small Organization	Participant's organization Less than 1000 employees, less than $1 Million in revenue, and limited human & physical or contracted resources.
	US only Customers	Participant's Job or organization Customers are limited to only U.S.
	International customers	Participant's Job or organization Customers are more than one country.
	Specific Industry	Participant's Job or organization focused on a specific industry.
	General customer base	Participant's Job or organization has a wide customer base with focuses on multiple types of industry.
	Timeline	How long the participant has worked in their current job and/or with the organization.
DevProcess		Description of the software development process in use at the participant's organization, in general, not necessarily about reg compliance; **Heuristic: When someone is describing Software development in general. Does not have to be a particular development process like Scrum or Waterfall.**
	ReqProcess	Description of how requirements are elicited; **Heuristic: When someone is describing development and highlighting requirement (to include**

		customer feedback and problems), code DevProcess and subcode ReqProcess.
	DesignProcess	Description of how software design is carried out **Heuristic: When someone is describing development and highlighting Design.**
	TestProcess	Description of how testing is done **Heuristic: When someone is describing development and highlighting Testing**
	ImplementProcess	Description of how software is released **Heuristic: When someone is describing development and highlighting release or assessment prior to release of a software or system.**
	MaintProcess	Description of how software is maintained and updated throughout its life **Heuristic: When someone is describing development and highlighting updates or maintaining software after production release.**
	DEVDecision	Description of why the participant and/or their organization uses a particular development process; **Heuristic: When someone references on decision-making or why they decided to do something a certain way, that is a DevDecision code**
	DefProcess	Description of a defined SDP; **Heuristic: When someone describes SDP and puts a name to their or organizations SDP, then DevProcess and DefProcess**
	NoDefProcess	Description is not a defined SDP; **Heuristic: When either someone describe their process as "ad-hoc" or they are not familiar with it to comment on what process is used, then Code NoDefProcess - the latter has a caveat that should be noted as the Development Process is unknown to the Participant**
	DevOther	Catch-all
ReqMGT		Description of how requirements are elicited and managed; **Heuristics: When someone is talking in more detailed about requirements and how they are assessed and managed,**

		where they come from. Then code ReqMGT
	ReqSource	Description of where the requirements come from; **Heuristic: When someone is talking about requirements with reference to understanding or source (including customer feedback or problem), code ReqMGT and subcode ReqSource**
	ReqGathering	Description of how requirements are gathered; **Heuristic: When someone is talking about requirements with reference to understanding or what is driving a requirement**; Examples- Use Case, customers requirement, stakeholders meeting.
	ReqPrioritzation	Description of how requirements are prioritized. **Heuristc: When there is a description that suggest Requirements Priorization is a factor, then code ReqMGT -> ReqPriotization**
	ReqChangeMGT	Description of how requirements evolve and how that change is addressed and managed; **Heuristic: When someone is describing Change, or flexibility to Change, or how requirements are track or how the evolve, code ReqChangeMGT**
	ReqOther	Catch -all; **Heuristic: ReqOther is a catch-all code that is used when something is on the topic of requirements or Req management, not covered by other subcodes in this field**
	ReqStakeholder	Description of requirements perspective and to whom that perspective belongs to
	ReqCommunication	Description of how requirements are communicated; relatable to Compliance Communication and ComplianceReq
ComplianceMGT		Description of how the participant and/or their organization assesses, tracks, and manages regulatory and security standard compliance. **Heuristic: Whenever there is a description on the topic of compliance**

		(regulatory, privacy, or security), how it is managed, assessed, tracked, demonstrated, or about how the participant's organization compliance program is structured or operates, then code ComplianceMGT with the appropriate subcode.
	ComplianceAssessment	Describes how compliance is assessed
	ComplianceTracking	Describes how they track compliance requirements
	ComplianceWhy	Describes why they must adhere to a compliance requirement
	ComplianceTimeline	Describes how long they must comply when a change occurs
	ComplianceWhen	Describes when Compliance is addressed within their SDP
	ComplianceReq	Describes what compliance requirements are required and who is responsible
	ComplianceOther	Catch-all; **Heuristic: ComplianceOther is a catch-all code that is used when something is on the topic of compliance management, not covered by other subcodes in this field**
	ComplianceCommunication	Describes how stakeholders communicate about compliance (Note potentially overlaps with ComplianceReq); **Heuristic: Whenever someone is one the topic of compliance and how it is communicated amongst stakeholders or between organizations**
CompliancePreceptions		Description of how the Participant perceives regulatory and security standard compliance. **Heuristic: Whenever the participant offers their opinion, thoughts, and perceptions on Compliance, use the CompliancePerceptions code.**
	ComplianceChallenges	Description by the participant on what are some challenges to compliance (examples from the text- updating legacy system to comply with current or new regulation, communication or interpretation between stakeholders, resource availability); **Heuristic: When something in the text points to**

		compliance might be a issue or a challenge within implementation of a SDP, the CompliancePerception -> Compliance Challenges
	ComplianceBenefits	Description by the participant of some benefits to compliance (examples from the text- security and developer awareness, trust, and confidence in develop product, sell and compete to more customers). **Heuristic: Similar to the ComplianceChallenges, except if the data or text points to compliance as a benefit to the participant.**
	ComplianceRisks	Descriptions by the participants what are risks to compliance or non-compliance (ex. 1: Proactive compliance companies risk money and false start requirements by trying to stay ahead of compliance 2: Reactive compliance companies risk delays in features and lost manhours to compliance)
	ComplianceSeparation	Descriptions by the participant that separates security and compliance within SDP;
Opinions/Perceptions		Thoughts or perceptions by the participant; **Heuristic: Whenever someone is offering an opinion or perception, (look for key phrases "I don't think" or "I think") code Opinion/Perception then subcode on what the Opinion or Perception is referencing.**
	Stakeholders	Participant's thoughts or perceptions of other stakeholders (ex.1: Developers view lawyers or legal as overly cautious or that some of their requirements to compliance is overkill. 2: Some Developers view Security assessor's expertise as only running the tool while other developers view Security assessors as highly technical and knowledgeable, but often overwhelmed and stretch thin with the amount of products they must assess); **Heuristic: When the participant is stating their opinion/perception in**

		reference to another stakeholder group, then code opinion/perception -> stakeholder
	SDP	Participant's thoughts or perceptions of how they develop software (i.e., Offer opinions on how their SDP is good or could be better in some areas); **Heuristic: Whenever someone is describing why they do a particular development process and list all the benefits to justify their process there Is 1) Opinion/Perception -> SDP because it is their perception specifically about their SW Development Process**
	Organization	Participant's thoughts or perceptions about their organization and how it is managed in relation to RC/SSC (e.g., Some view their organization as great setup wise because it offers resources and resolution process on a particular implementation issue)
	OnCompliance	Participant's thoughts or perceptions on compliance (e.g., "Compliance is necessary but not sufficient"); **Heuristic: Whenever an opinion or perception points to a specific piece on compliance then also code Opinions/Perceptions -> OnCompliance**
	SWDevIndustry	Participant's thoughts or perceptions on the Software Development Industry in relation to RC/SSC
	Other	Catch -all; **Heuristic: Whenever an opinion or perception points to a something specific (i.e., Stakeholder, SDP, Organization, compliance) use assoicated codes. If there is an outlier use Other subcode.**
Technical Debt		Describes TD as defined as the cost of prioritizing one requirement over another and addressing it after release. **Heuristics: Whenever the participant refers to Technical Debt and/or describes something close to definition, use Technical Debt and the appropriate subcode as described.**

		Data could overlap with the RegMGT -> ReqPrioritization code
	WhyTD?	Describes why TD occurs as related to RC/SSC (ex. 1: Change in regulation resulting in production systems requiring refactoring to comply and new updates to a production system get delayed. 2: Resource availability- certain things must be prioritized and tested within a certain timeframe, which means other items get tested and assessed after release 3: Document debt)
	CostTD	What are the cost or impacts to TD as related to RC/SSC (e.g., Larger Maintenance overhead, delays in feature release
	RiskTD	What are the risks to TD as related to RC/SSC (e.g., Exploitation vulnerability, code not working optimally or buggy source code, unsatisfied customers)
	AddressingTD	Describes how they manage TD
Wishes		This is in response to the "End of Interview questions list in the RC/SSC section question L. & M. and Summary Section. Describes things the participant wishes they or their organization did differently. **Heuristic: Any kind of hypothetical to improve the organizations processes or the software industry in general use this code.**
Recent Events		Describes how recent events may have affected your organization or how you conduct business as related to RC/SSC. **Heuristic: Whenever the participant references a "News worthy" topic notable from Jan 2020 to Jan 2021, use the Recent Events code and the appropriate subcode.**
	COVID	Describes how COVID may have affected them
	Presidential election	Describes how Change in Administration may have affected them
	Regulation changes	Describes how recent regulatory decision (i.e., GDPR, EU Court

		decisions) may have impacted your Processes
	Regulatory Infractions	Describe if any regulatory infractions (i.e., Zoom) may have impacted your business
Technical Difficulty		Pause in the transcript
Clarify Questions		Questions is restated or clarified for the Participant
Other		Catch-all code to be applied and expanded on as a new code

Appendix B

Survey Appendix

B.1 Survey Question Listing

The following it the listing of survey questions. The "ID" corresponds the question number as referenced throughout Chapter 4. The "Category" refers to the topic of questions asked regarding regulatory compliance. The "Type" is the type of response, with "Y/N" referring to Yes or No; "IDK" is "I do not know".;"IRNS" is "I'd rather not say";"MC" is Multiple Choice; "Open" is open-ended response. Rating is a numerical rating of 1 to 5 with 1 is Strongly Disagree to 5 is Strongly Agree (Note: Rating* reverses the numerical rating from 5 to 1 with 5 is Strongly Disagree to 1 is Strongly Agree). "Marked if applied" means that the survey responded is the statement applied to their organization. Lastly, "Ordered response" asked to put the seven answers in ranking order. The "Used" and "Notes"columns reference if the the question was used in the analysis and why to correspond with the explanation given in Data Collection and Analysis in Chapter 4, Section 4.1.2. The full dataset and a print out version of the survey is available at the following DOI: `https://doi.org/10.6084/m9.figshare.25078061`.

Table B.1: Survey Question Table

ID	Category	Question	Type	Used	Notes
Q1	Consent	Do you agree to participate in this survey?	Y/N	Yes	Cleaning
Q2	Consent	Do you understand the goals of this survey?	Y/N	No	
Q3	Consent	Do you understand that this survey is anonymous and does not record any personally identifiable information?	Y/N	No	
Q4	Demographic	What is your highest educational degree?	MC	No	See Figure 1
Q5	Demographic	Do you have any software related certifications?	MC	No	
Q6	Demographic	Do you have any software related certifications (Other)?	Open	No	
Q7	Demographic	What is your role in your current organization?	MC	No	See Figure 1
Q8	Demographic	What is your role in your current organization (Other)?	Open	No	
Q9	Demographic	How many years have you been at your current role (In Years)?	Number	No	
Q10	Demographic	How many years have you worked in the software industry all together (In Years)?	Number	No	See Figure 1
Q11	Demographic	What other positions have you held?	MC	No	
Q12	Demographic	What other positions have you held (Other)?	Open	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q13	Demographic	What sector does your organization operate in?	MC	No	See Figure 1
Q14	Demographic	What sector does your organization operate in (Other)?	Open	No	See Figure 1
Q15	Demographic	Which of the following best describes your organization?	MC	No	
Q16	Demographic	Which of the following best describes your organization (Other)?	Open	No	
Q17	Demographic	What software or system development methods do you or your organization use?	No		
Q18	Demographic	What software or system development methods do you or your organization use (Other)?	No		
Q19	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Planning**	Rating	Yes	
Q20	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Requirements**	Rating	Yes	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q21	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Design**	Rating	Yes	
Q22	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Implementation**	Rating	Yes	
Q23	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Testing**	Rating	Yes	
Q24	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development: **Deployment/Release (i.e., review of software before release into production)**	Rating	Yes	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q25	SDP	Please rate your organization's efforts in ensuring Regulatory and Security Standard Compliance (RC/SSC) within each of the following aspect of software development:**Maintenance**	Rating	Yes	
Q26	Responsibility	Who do you think is responsible for ensuring adherence to both regulatory and security standard compliance within your organization?	MC	FI	See Figure 2
Q27	Compliance Requirements	Do you have customers that have specific regulatory or security standards requirements?	Y/N/ IRNS	No	
Q28	Compliance Requirements	Does your organization attempt to include their customers' regulatory or security standard requirements as part of your organization's software development process?	Y/N/ IRNS	No	
Q29	Compliance Requirements	What Regulations or Security standards requirements are your customers required to comply with?	MC	No	
Q30	Compliance Requirements	What Regulations or Security standards requirements are your customers required to comply with (Other)?	Open	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q31	Compliance Requirements	What Regulations or Security standards is your organization required to comply with because they directly apply to your organization or for contractual reasons with clients or partner organizations?	MC	No	
Q32	Compliance Requirements	What Regulations or Security standards is your organization required to comply with because they directly apply to your organization or for contractual reasons with clients or partner organizations?	Open	No	
Q33	Organization's Compliance	Please rate your personal agreement with the following statements on your organization's compliance with regulatory or security standards: **My organization does everything it can to diligently comply with their regulatory and security requirements**	Rating	Cluster	
Q34	Organization's Compliance	Please rate your personal agreement with the following statements on your organization's compliance with regulatory or security standards:**My organization has a process for prioritizing compliance concerns during the software development process.**	Rating	Cluster	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q35	Organization's Compliance	Please rate your personal agreement with the following statements on your organization's compliance with regulatory or security standards:**: When resources are tight (i.e. limited staffing, time, or money), the compliance assessment process is the first thing to change.**	Rating*	Cluster	
Q36	Organization's Compliance	Please rate your personal agreement with the following statements on your organization's compliance with regulatory or security standards:**I would not change my organization's compliance process.**	Rating	Cluster	
Q37	Organization's Compliance	Please rate your personal agreement with the following statements on your organization's compliance with regulatory or security standards: **My organization actively promotes individual employees' professional development and ethics training (i.e., they pay for professional memberships such as ACM and IAPP or encourage conference attendance)**	Rating	Cluster	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q38 to 44	Compliance Answers	Please drag and drop from 1 to 7 in order in which you have pursued and gotten a satisfactory answer to a compliance questions: **Ask a Peer or Team Lead; Search the Internet; Search the organization's share site; Ask a Subject Matter Expert within the organization; Search a professional online forum; Seek professional external expertise at cost; Other**	Ordered	No	
Q45	Compliance Answers	Please drag and drop from 1 to 7 in order in which you have pursued and gotten a satisfactory answer to a compliance questions: **Other**	Open	No	
46	Organization's Compliance	Does your organization have a separate team that assesses compliance?	Y/N	No	
Q47	Organization's Compliance	**If Q46 is No:** Given that you do not have a team, do you have organization-wide policy and/or procedures related to regulatory or security standards requirements your organization is required to follow?	Y/N/ IDK	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q48	Organization's Compliance Program	Please rate your personal agreement on the following statements on your organization's internal compliance program: **Internal compliance programs provide real benefits for regulatory and security standard compliance.**	Rating	Cluster	
Q49	Organization's Compliance Program	Please rate your personal agreement on the following statements on your organization's internal compliance program: **My organization's compliance program changed my approach to engineering with respect to regulatory and security standard compliance.**	Rating	Cluster	
Q50	Organization's Compliance Program	Please rate your personal agreement on the following statements on your organization's internal compliance program: **I'm more confident in the products my organization produces and maintains because of our internal compliance program(s).**	Rating	Cluster	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q51	Organization's Compliance Program	Please rate your personal agreement on the following statements on your organization's internal compliance program: **My organization views regulatory and security standard compliance as an investment to ensuring the quality of our software and trust with our customer rather than the cost of doing business.**	Rating	Cluster	
Q52	Comms&MGT	Does your organization communicate their internal compliance process to their customers?	Y/N	No	
Q53	Comms&MGT	Why does your organization not communicate their internal compliance process to their customers?	Open	No	
Q54	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? My organization freely shares their compliance process with their customers.	Mark if applied	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q54	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? **My organization is contractually required to document and share their compliance process with their customers.**	Mark if applied	No	
Q54	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? **My organization is required by regulation to share information regarding their compliance process.**	Mark if applied	No	
Q54	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? **I really cannot say why, just that our compliance process/program is there, and we communicate it to our customers.**	Mark if applied	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q54	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? **My organization reports their compliance process as a condition of a court settlement involving a compliance or security incident per the guidance of a governing regulatory agency like the Federal Trade Commission.**	Mark if applied	No	
Q55	Comms&MGT	Which of the following characterizes your organization's advertisement of their compliance process? **Other**	No		
Q56	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **My organization understands and follows the intent of the law, when it comes to regulatory and security standard compliance**	Rating	Yes	See Table 3

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q57	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **My organization has a history of non-compliance (i.e., my organization has been found in violation or has had an enforcement action against them because of non-compliance).**	Rating*	Yes	See Table 3
Q58	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **My organization's compliance requirements are ingrained into the culture of the organization.**	Rating	Yes	See Table 3

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q59	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **My organization communicates our regulatory and security standards requirements to third party vendors through contracts to ensure compliance to these requirements.**	Rating	Yes	See Table 3
Q60	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **My organization communicates our compliance process both to the employees and our customers.**	Rating	Yes	See Table 3

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q61	Comms&MGT	Please rate your personal agreement on the following statements on your organization's Communication and Management of Regulatory and Security Standard Compliance: **I wish my organization would be more transparent about our compliance and security processes to our customers.**	Rating	Yes	See Table 3
Q62	Comms&MGT	In no more than a paragraph, describe any additional concerns you may have about your organization's communication of compliance?	Open	No	
Q63	Governance Perception	Please rate your personal agreement with the following statements regarding Perception of Compliance: **Regulators do not understand the best practices of the software industry and cannot draft regulations accordingly.**	Rating	Yes	See Table 5
Q64	Governance Perception	Please rate your personal agreement with the following statements regarding Perception of Compliance: **Regulations are too hard to interpret and make my job even harder.**	Rating	Yes	See Table 5

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q65	Governance Perception	Please rate your personal agreement with the following statements regarding Perception of Compliance: **The best regulations are based on already established industry best practices.**	Rating	Yes	See Table 5
Q66	Governance Perception	Please rate your personal agreement with the following statements regarding Perception of Compliance: **Regulations favor larger companies making it hard for smaller companies to comply and compete.**	Rating	Yes	See Table 5
Q67	Governance Perception	Please rate your personal agreement with the following statements regarding Perception of Compliance: **Compliance audits are necessary but could be better tooled for compliance enforcement within the software industry.**	Rating	Yes	See Table 5

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q68	Compliance Strategy	Please rate your personal agreement with the following statements regarding Responses to and Impacts of Regulatory changes: **Responding to regulatory changes consumes a great deal of resources (time, money, effort) in my organization, in comparison to time spent on design and implementation of our products themselves.**	Rating	Yes	See Table 4
Q69	Compliance Strategy	Please rate your personal agreement with the following statements regarding Responses to and Impacts of Regulatory changes: **My organization's initial response is to a new regulatory change is to wait and see how new regulatory requirements evolve and are enforced before complying with them.**	Rating	Yes	See Table 4
Q70	Compliance Strategy	Please rate your personal agreement with the following statements regarding Responses to and Impacts of Regulatory changes: **My organization response to a regulatory change is to form a team of experts to carefully assess its effects and potential responses.**	Rating	Yes	See Table 4

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q71	Compliance Strategy	Please rate your personal agreement with the following statements regarding Responses to and Impacts of Regulatory changes: **Responses to changes (including risks of rushed software changes and non-compliance) are assessed for impact on business.**	Rating	Yes	See Table 4
Q72	Compliance Strategy	Please rate your personal agreement with the following statements regarding Responses to and Impacts of Regulatory changes: **My organization spends a lot of time upfront analyzing and understanding regulatory requirements, which makes designing and implementing a response straightforward.**	Rating	Yes	See Table 4
Q73	Compliance Strategy	In no more than a paragraph, can you describe your organization's approach to regulatory and security standard compliance?	Open	No	
Q74	Close-Out Question	Is there anything your organization's processes regarding regulatory or security standard compliance you would wish were adopted by other software practitioners?	Open	No	

Continued on next page

Table B.1 – continued from previous page

ID	Category	Question	Type	Used	Notes
Q75	Close-Out Question	Is there anything about these definitions or your organization's processes that we missed that should be included in this survey?	Open	No	
Q76	Close-Out Question	Is there anything outside you organization's processes but related to regulatory or security standard compliance you would like to share?	Open	No	

Appendix C

Multi-Case Study Appendix

C.1 Multi-Case Study's Protocol

The following is the Multi-Case Study's Protocol outlining the three session described in Chapter 5, Section 5.2.1. A detail methodology and the session slides are available at the following DOI: `https://doi.org/10.6084/m9.figshare.23297717`.

1. Session One: Kick-off Meeting and Hands-on exercise with AHAB
    (a) Introductions
        i. Facilitator
        ii. Moderator
        iii. Participant
    (b) Background on Regulatory Compliance and Regulatory Ambiguity
        i. Regulatory Compliance in Traditional Engineering Disciplines
        ii. Why model Regulatory Ambiguity?
    (c) Overview of the Case Study
        i. Purpose of the Case Study
        ii. Goals of the Case Group
    (d) Question and Answer on Training Material provided prior to the session.

(e) Hands-on exercise with AHAB

i. Exercise Objective: For the participant to build your first Regulatory Ambiguity model (RAM) while the facilitator and moderators observe.

ii. Exercise Guidance

A. Participant should talk while using AHAB

B. Facilitator/Moderator are observing, not to tell the participant how they should identify or build their models.

C. Facilitator/Moderator will assist if the participants have technical issues with AHAB.

iii. Tasks to complete:

A. Task 1: Arrange the components on the AHAB tool

B. Task 2: Load a regulatory text into AHAB

C. Task 3: Define your User Perspective

D. Task 4: Create at least three ambiguity nodes

E. Task 5: Create your Ambiguity Mode

F. Task 6: Export (Save) your Ambiguity Model into a JSON file

G. Task 7: Submit your JSON file to Google Drive

item End of Hands-on exercise with AHAB Questions

(f) Instructions for Session Two:

i. Homework: Build an Ambiguity on the participant own.

ii. Where to Submit JSON model files

iii. Reference Material and Point of Contact information

2. Observation Session (optional)

   (a) Purpose of the Observation Session is to schedule dedicated time for the Participants to complete their Homework for Session Two.

   (b) Facilitator/Moderator are observing, not to tell the participant how they should identify or build their models.

   (c) Facilitator/Moderator will assist if the participants have technical issues with AHAB.

3. Session Two:

   (a) Goals of Session Two and the Group

   (b) Group Participant's Introduction

   (c) Model Presentation

      i. Each participant will be given about 10 minutes to present their models

      ii. 5 minutes after each presentation for QA

   (d) Session Question and Answer

   (e) Instructions for Session Three:

      i. Homework:Analysis and compare individual models. Participant's can update their model while they compare and contrast models with their own.

      ii. Where to Submit their updated JSON model files

iii. Reference Material and Point of Contact information

4. Observation Session (optional)

   (a) Purpose of the Observation Session is to schedule dedicated time for the Participants to complete their Homework for Session Two.

   (b) Facilitator/Moderator are observing, not to tell the participant how they should identify or build their models.

   (c) Facilitator/Moderator will assist if the participants have technical issues with AHAB.

5. Session Three:

   (a) Goals of Session Three and the Group

   (b) Model Updates and Analysis Presentation

      i. Each participant will be given about 10 minutes to present analysis and any changes to their models

      ii. Question and Answer after presentation

   (c) Consensus Discussion and Model(s) Consolidation to one Group Model

   (d) Close-Out of the Study (Feedback on the Process

      i. What did you think about the process of Modeling Regulatory Ambiguity?

      ii. What did you find the most difficult thing to do within the process?

      iii. Link to Close Out Survey

iv. Thank the participants

(e) Thank everyone for their time and participation.

C.2 Multi-Case Study's Survey

The following is an outline of the End-of-Case Survey, given the Case Group two and three. A print out survey is available at the following DOI: `https://doi.org/10.6084/m9.figshare.23297717`.

1. How easy was it to model Ambiguities within the given regulation? (Possible Answers: Extremely difficult, Somewhat difficult, Neither easy nor difficult, Somewhat easy, Extremely easy)

2. How much time did you spend on your model outside the online Sessions? (Possible Answer: 1-29 min, 30-59 min, 1-2hrs, more then 2 hours, I spent no time on my model outside of the online sessions)

3. In reviewing the regulation and building your model, please rate your agreement to the following statement (Possible Answer: Strongly Disagree, Somewhat Disagree, Neither Agree or Disagree, Somewhat Agree, or Strongly Agree):

   (a) I found it difficult to read and understand the regulations in general.

   (b) I found it difficult to understand the intent of the regulation.

   (c) I found it difficult to identify and classify the ambiguities within the regulation.

   (d) I found it difficult to organize my model.

(e) I found it difficult to present and explain my model to my peers in general.

(f) I found it difficult to understand and follow how my peers created their model in general.

(g) I found the AHAB tool made all the modeling tasks more difficult than they otherwise would be.

(h) I found it difficult to present and explain my model to my peers using the AHAB tool.

(i) I found it difficult to understand and follow how my peers created their model using the AHAB tool.

4. Do you have something to add regarding your answers to Q3? (Open-ended)

5. Did you feel there is value in reviewing regulation and building ambiguity models as part of a Software Development Process? (Possible Answer: Definitely not, Probably not, Might or might not, Probably yes)

6. If on a Software Development Team, would you suggest this modeling process as part of the requirement phase to your team? (Possible Answer: Definitely not, Probably not, Might or might not, Probably yes)

7. Do you have something to add regarding your answers to Q5 and Q6? (Open-ended)

8. Is there anything else you would like to add? (Open-ended)

C.3 Multi-Case Study's Coding Scheme

This is the initial coding scheme used in the case study analysis. The final consolidated coding scheme is described in Chapter 5, Section 5.2.4.

Code	Subcodes	Definitions
Individual Participant QA		At the end of each participant's model presentation, time is given to the group to ask the presenter questions about their model.
Participants Introduction to Case Group		The participants also introduce themselves to the Case Group.
Presentation of Model		The Participants presentation of their Ambiguity models based on Article 17 of GDPR (The right to be forgotten).
	Ambiguities Identified	Identified Ambiguities within the Transcripts and related to the Tabular outputs from the submitted models
	User Perspective	The participants identified perspectives while presenting their model.
Common Ambiguity Type		This type of ambiguity was commonly picked. IOW, within the case, the participants favoured labelling legal text under this ambiguity. common is if two or more participants
Common Legal Text as an Ambiguities		Within the assigned legal text, this legal text was commonly picked as an ambiguity.
Common Reasoning		Within the notes of the AHAB built models and the transcribed Session video, this code is meant to show common reasoning for an ambiguity
	Confusing	Participants used either the word or phrase indicating that they did not understand the legal text and therefore labelled as ambiguous.
	Using the Definition of Ambiguity	Participants during the session, referred back to the ambiguity taxonomy to justify classification.
Participants Homework		The overview of the Homework for Session 2.
Close out of Session 2		Close out of Session 2 by the Facilitator.
Model Turn-in		When the participants turned in their models to Session 2 or 3.
Observation		Participants optional observation, where they went online with the Facilitator observing and worked on their model.
Model Building Time		The time given to a participant to work on their model.
Consensus		The time in session 3 where the Participants consolidated three models into one group ambiguity model.

	Consensus Discussion	The discussion that occurred between the Participants while they tried to consolidate their models into one group ambiguity model.
	Group Consensus Achieved	Did the Participants Achieve Consensus and create one group ambiguity model.
Analysis of Peer Models		The presentation of the participants' analysis of their and their peers' models during Session 3.
	Differences between models	Indicated difference presented in Session 3 between models noted by the participant during individual analysis.
	Similarities between models	Indicated similarities presented in Session 3 between models noted by the participant during individual analysis.
Updates to Ambiguity Models		Indicated Updates the Participants made to their models between Session 2 and 3.
Group Interview		The Case Group Interview to close-out the Case. The groups were asked about talk about their experience in while modeling regulatory ambiguities.
	Thoughts on modeling ambiguitues	During the Case Group Close-Out Interview, participants where asked: "What did you think about the process of Modeling Regulatory Ambiguity?". This code is for their answers.
	Difficulites modeling ambiguities	During the Case Group Close-Out Interview, participants where asked: "What did you find the most difficult thing to do within the process?". This code is for their answers.
AHAB Exercise		Codes applied during the execution of the Session Exercise
	Explanation of the Exercise	Purpose of the Exercise
	Understanding the Exercise	Indicators Participant understanding of the AHAB Exercise
	Exercise Performed	Time code signalling the execution of the AHAB Tool Exercise
	Performing the Task	Observations within the transcripts while the participants performed the AHAB Tool Exercise.
	Task Completion	The participants' completion of tasks and indicators related to task completion. IOW, did something occur during session that affected the participant task completion (i.e. Technical Difficulty, time constraints, not understanding the task, etc.)

Introduction into the Case Study		Background and Overview of Case Study
Session 1 Preparation		Memo describing what the Participants did in preparation for Session1.
	Open the AHAB tool	Indicators that the Participant opened the AHAB tool prior to Session 1.
	Reviewed Session Slides	Indicators that the Participant did review the Session Slides prior to Session 1.
	Reviewed Slides on Ambiguity Model	Indicators that the Participant Reviewed the Ambiguity Model Slides prior to Session 1.
	Reviewed Tutorial Videos	Indicates that the participants reviewed the 12-minute training video on how to use the AHAB tool given to participants prior to Session 1.
Level of Session Preparation		The participants level of Preparation as indicated during the Sessions.
Number of Ambiguities		Number of Ambiguities Identified by participants in a Table Format within any Session.
	Reason for that many ambiguity	The underlying reason as to why they identified that many ambiguities.
Organizing the Model		Indicators during the presentation of how the participant organized their models. This is a Code that can be applied to any of the three sessions.
	Forming Nodal Relationships	The Participants explanation as to why they linked to nodes together,
	Organizing the Nodes	The structure or grouping of nodes within their models and the justification for the grouping. This code is applicable to both nodes linked together and not linked together.
Questioned asked		Questions asked by Session Participants
	From the Facilitator	Questioned asked by the Facilitator to the Participant
	From the Moderator	Questioned asked by the Moderator to the Participant
	Participant to Participant	Questions asked between the participants
	Questioned asked prior to Sessions	Questions asked by participants prior to Sessions to the Facilitator.
	Questions about Ambiguity Modelling	Questions asked by the participants about Ambiguity Modelling.
	Questions about the AHAB tool	Questions about the AHAB Tool asked by the participants.
	To the Facilitator	Questions asked from a participant to the Facilitator

	To the Moderator	Questions asked from a participant to the Moderator
Technical Difficulty		Indicator that the participant had Technical Difficulty using Google Meet or the AHAB Tool
Technical use of AHAB tools		Participants use the technical features within the AHAB tool.
Time Constraints		Time Constraints hinder task completion

Appendix D

Multi-Case Study Models

All artifacts in the Appendix are available at the following DOI:https://doi.org/10.6084/m9.figshare.23297717

D.1 Case Group One's Models

D.1.1 Individual Models

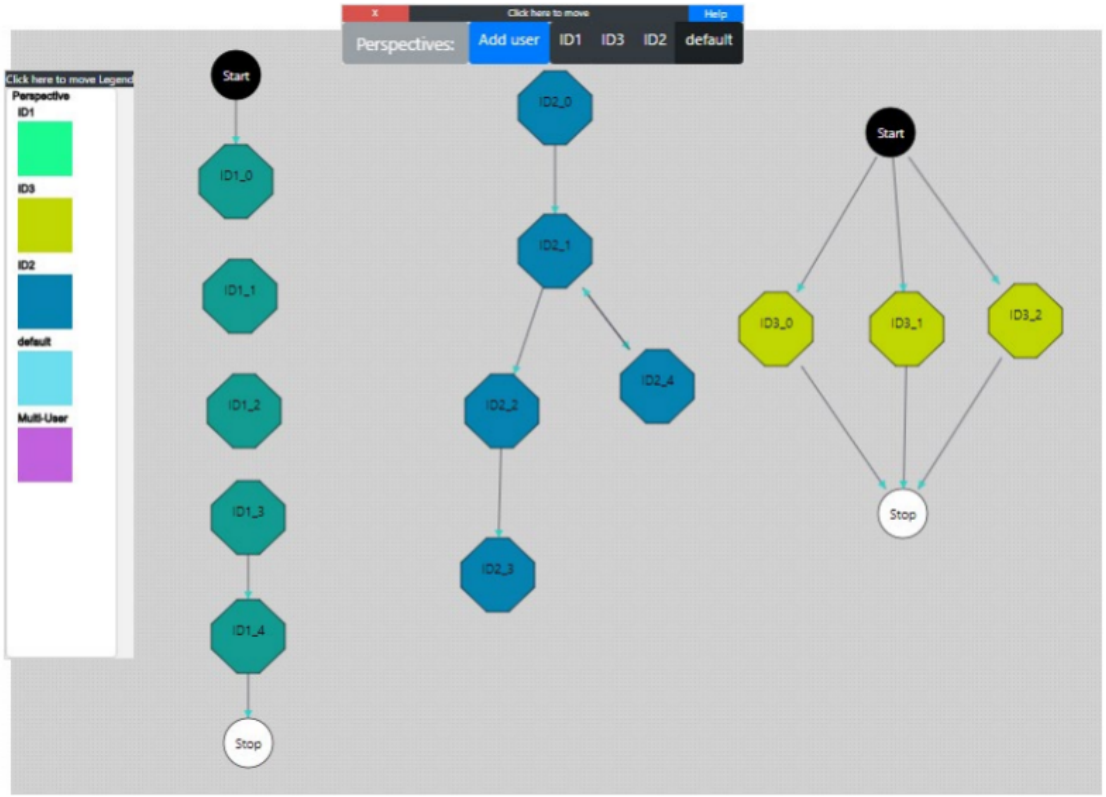

Figure D.1: Case Group One's Session 2 Individual Models

D.1.2 Consensus Model-Not finished

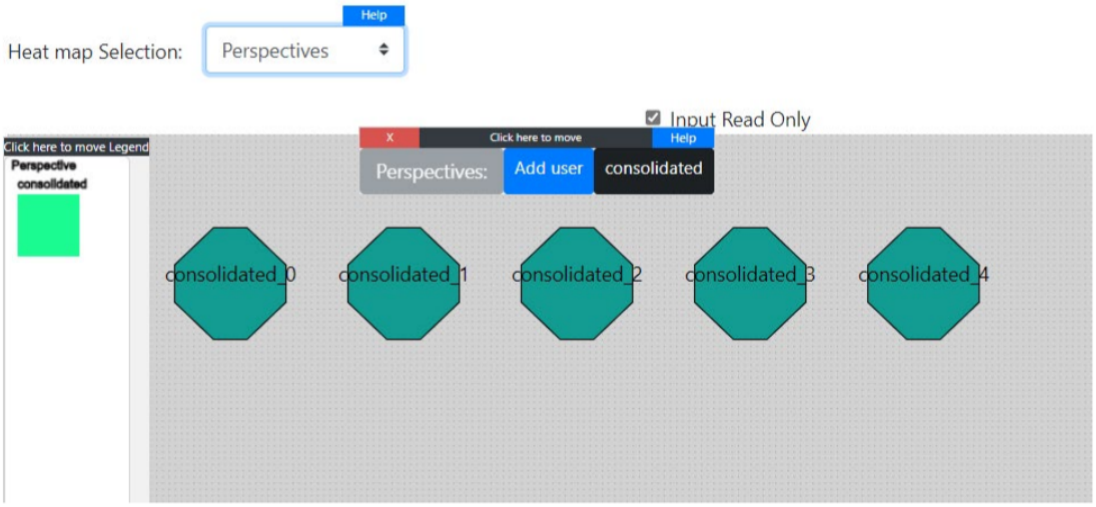

Figure D.2: Case Group One's Session 3 Consolidated Model

D.1.3 Group Consensus Analysis

Element Type	ID	NAME	USER	Ambiguity Type	Severity	Intentionality	Implementability	Regulatory Text	Regulatory Text ID	Notes
Ambiguity Element	consolidated_0	consolidated_0	consolidated	Syntactic	5	y	y	Where the controller has made the personal data public and is obliged pursuant to paragraph 1 to erase the personal data, the controller, taking account of available technology and the cost of implementation, shall take reasonable steps, including technical measures, to inform controllers which are processing the personal data that the data subject has requested the erasure by such controllers of any links to, or copy or replication of, those personal data.	EU_GDPR_Ch3_Art17	too complex sentence. Can not understand the sentence in one readingy notes here...
Ambiguity Element	consolidated_1	consolidated_1	consolidated	Vagueness	3	n	y	Paragraphs 1 and 2 shall not apply to the extent that processing is necessary:	EU_GDPR_Ch3_Art17	not sure of what it really meantotes here...
Ambiguity Element	consolidated_2	consolidated_2	consolidated	Incompleteness	5	n	n	the personal data have been unlawfully processed:	EU_GDPR_Ch3_Art17	need more information on unlawful context
Ambiguity Element	consolidated_3	consolidated_3	consolidated	Semantic	3	n	y	the personal data have to be erased for compliance with a legal obligation in Union or Member State law to which the controller is subject:	EU_GDPR_Ch3_Art17	statement is clear in one context but unclear in another context. Ie, if the data is collected and processed in the same region, the sentence makes sense. if the data is collected from the participant in another country the legal text is unclear.
Ambiguity Element	consolidated_4	consolidated_4	consolidated	Referential	1	y	y	the data subject withdraws consent on which the processing is based according to point (a) of Article 6(1), or point (a) of Article 9(2), and where there is no other legal ground for the processing:	EU_GDPR_Ch3_Art17	the sentence is grammatically sounds correct but refers to two different articles based on the content of the article mentioned the meaning or the meaning or the legal text can change.

Figure D.3: Case Group One's Session 3 Consolidated Analysis

D.2 Case Group Two's Models

D.2.1 Individual Models

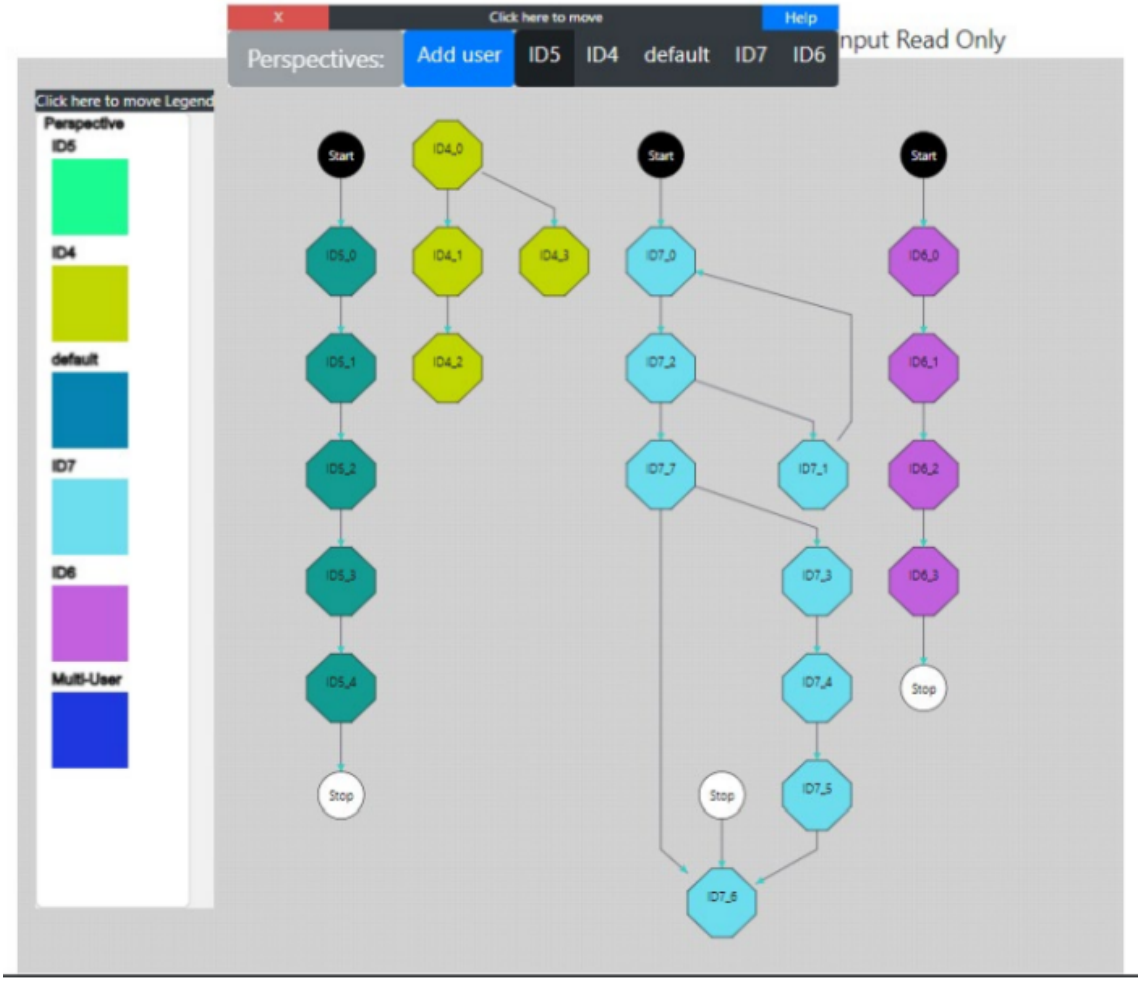

Figure D.4: Case Group Two's Session 2 Individual Models

D.2.2 Consensus Model

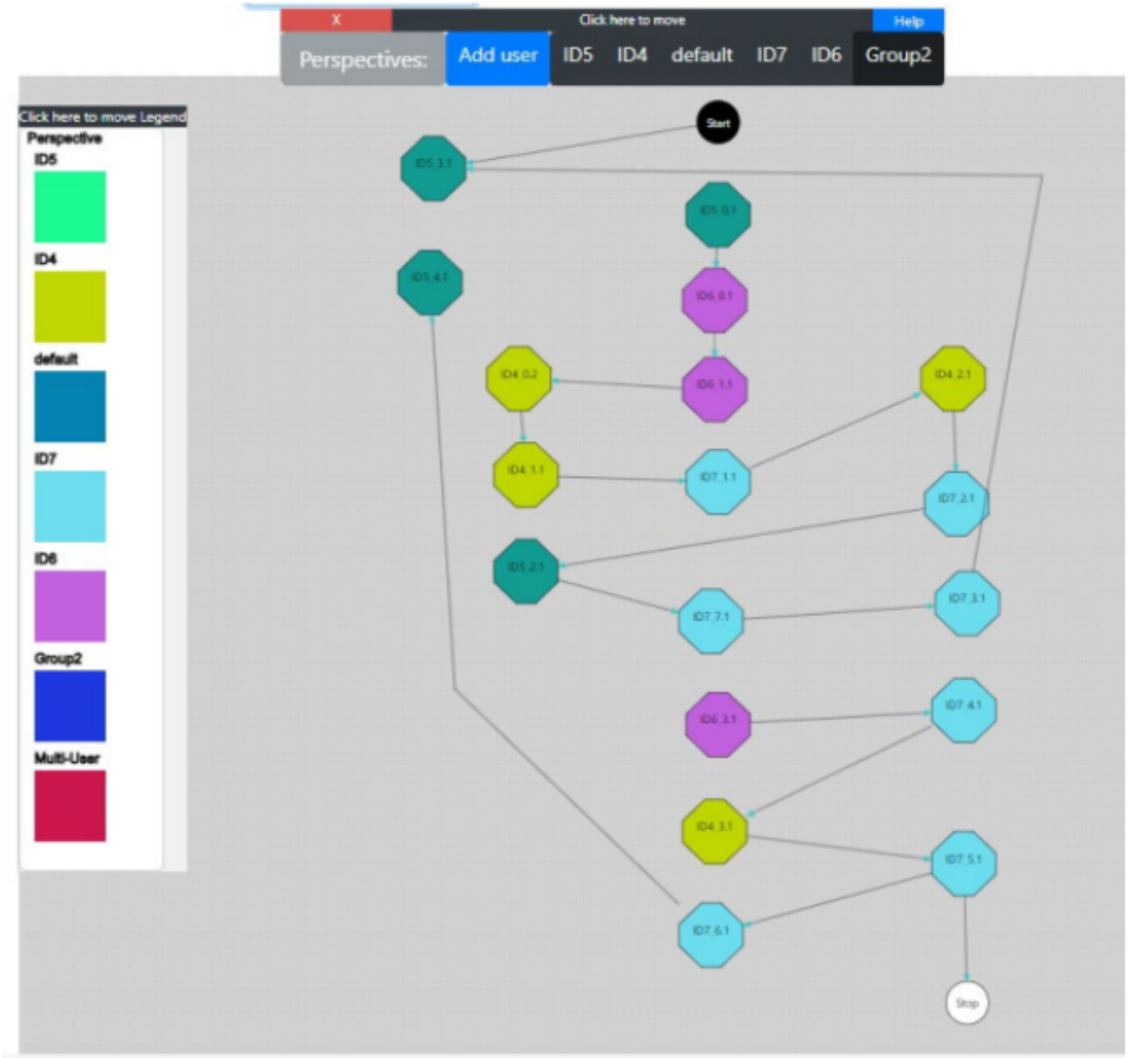

Figure D.5: Case Group Two's Session 3 Consolidated Model

D.2.3 Group Consensus Analysis

Element Type	ID	NAME	USER	Ambiguity Type	Severity	Intentionality	Implementability	Regulatory Text	Regulatory Text ID	Notes
Ambiguity Element	ID5_0.1	ID5_0.1	ID5	Lexical	3y	y		The data subject shall have the ID5 to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:	EU_GDPR_Ch3_Art17	usage of the words multiple times
Ambiguity Element	ID4_0.2	ID4_0.2	ID4	Vagueness	4n	y		personal data	EU_GDPR_Ch3_Art17	The phrase "Personal Data" can have multiple interpretations. For one person, height and weight can be personal data, but for others it might not be that personal. So, we must specify what does it mean by "Personal Data" here.
Ambiguity Element	ID4_1.1	ID4_1.1	ID4	Vagueness	4y	y		purposes for which they were collected	EU_GDPR_Ch3_Art17	The purposes of the study must also be specified in advance. If the purpose of the study is not specified, then there can be problems in future regarding this. Let's say if the data subject wants the controller to erase his data, but the controller says that the reason is not valid because it is one of the purposes of the study. This can be problematic.
Ambiguity Element	ID6_1.1	ID6_1.1	ID6	Referential	3y	y		controller	EU_GDPR_Ch3_Art17	Definition of controller not given. A person might not aware of the term controller.
Ambiguity Element	ID7_1.1	ID7_1.1	ID7	Vagueness	3n	y		the data subject withdraws consent on which the processing is based according to point (a) of Article 6(1), or point (a) of Article 9(2), and where there is no other legal ground for the processing;	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID4_2.1	ID4_2.1	ID4	Semantic	2n	y		legitimate grounds	EU_GDPR_Ch3_Art17	People can have different definitions of what we call legitimate grounds. For one person, a reason can be legitimate, but for another person it might not be legitimate.
Ambiguity Element	ID7_7.1	ID7_7.1	ID7	Syntactic	3n	y		the personal data have been unlawfully processed;	EU_GDPR_Ch3_Art17	Two way relation
Ambiguity Element	ID5_2.1	ID5_2.1	ID5	Incompleteness	4y	y		the personal data have been unlawfully processed;	EU_GDPR_Ch3_Art17	
Ambiguity Element	ID7_3.1	ID7_3.1	ID7	Semantic	2n	y		the personal data have to be erased for compliance with a legal obligation in Union or Member State law to which the controller is subject;	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID5_3.1	ID5_3.1	ID5	Vagueness	5n	y		Where the controller has made the personal data public and is obliged pursuant to paragraph 1 to erase the personal data, the controller, taking account of available technology and the cost of implementation, shall take reasonable steps, including technical measures, to inform controllers which are processing the personal data that the data subject has requested the erasure by such controllers of any links to, or copy or replication of, those personal data.	EU_GDPR_Ch3_Art17	vague , no context
Ambiguity Element	ID6_3.1	ID6_3.1	ID6	Semantic	3n	y		the ID5 of freedom of expression and information;	EU_GDPR_Ch3_Art17	ID5 of only freedom or ID5 of freedom and information is not clear.
Ambiguity Element	ID6_0.1	ID6_0.1	ID6	Referential	3y	y		data subject	EU_GDPR_Ch3_Art17	Defination of data subject not given. A person might not aware of the term 'data subject'.
Ambiguity Element	ID7_2.1	ID7_2.1	ID7	Referential	2n	y		the personal data have been unlawfully processed;	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID5_4.1	ID5_4.1	ID5	Incompleteness	5y	y		for the establishment, exercise or defence of legal claims.	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID4_3.1	ID4_3.1	ID4	Vagueness	3y	y		purposes in the public interest,	EU_GDPR_Ch3_Art17	There must be some clarity over what we consider as the purposes in the public interests. Before signing the consent form, the data subject must review the guidelines on what is being considered in the interest of public.
Ambiguity Element	ID7_6.1	ID7_6.1	ID7	Syntactic	3n	y		for the establishment, exercise or defence of legal claims.	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID7_5.1	ID7_5.1	ID7	Lexical	2n	y		for the establishment, exercise or defence of legal claims.	EU_GDPR_Ch3_Art17	Please add any notes here...
Ambiguity Element	ID7_4.1	ID7_4.1	ID7	Incompleteness	2n	y		for reasons of public interest in the area of public health in accordance with points (h) and (i) of Article 9(2) as well as Article 9(3);	EU_GDPR_Ch3_Art17	Please add any notes here...

Figure D.6: Case Group Two's Session 3 Consolidated Analysis

D.3 Case Group Three's Models

D.3.1 Individual Models

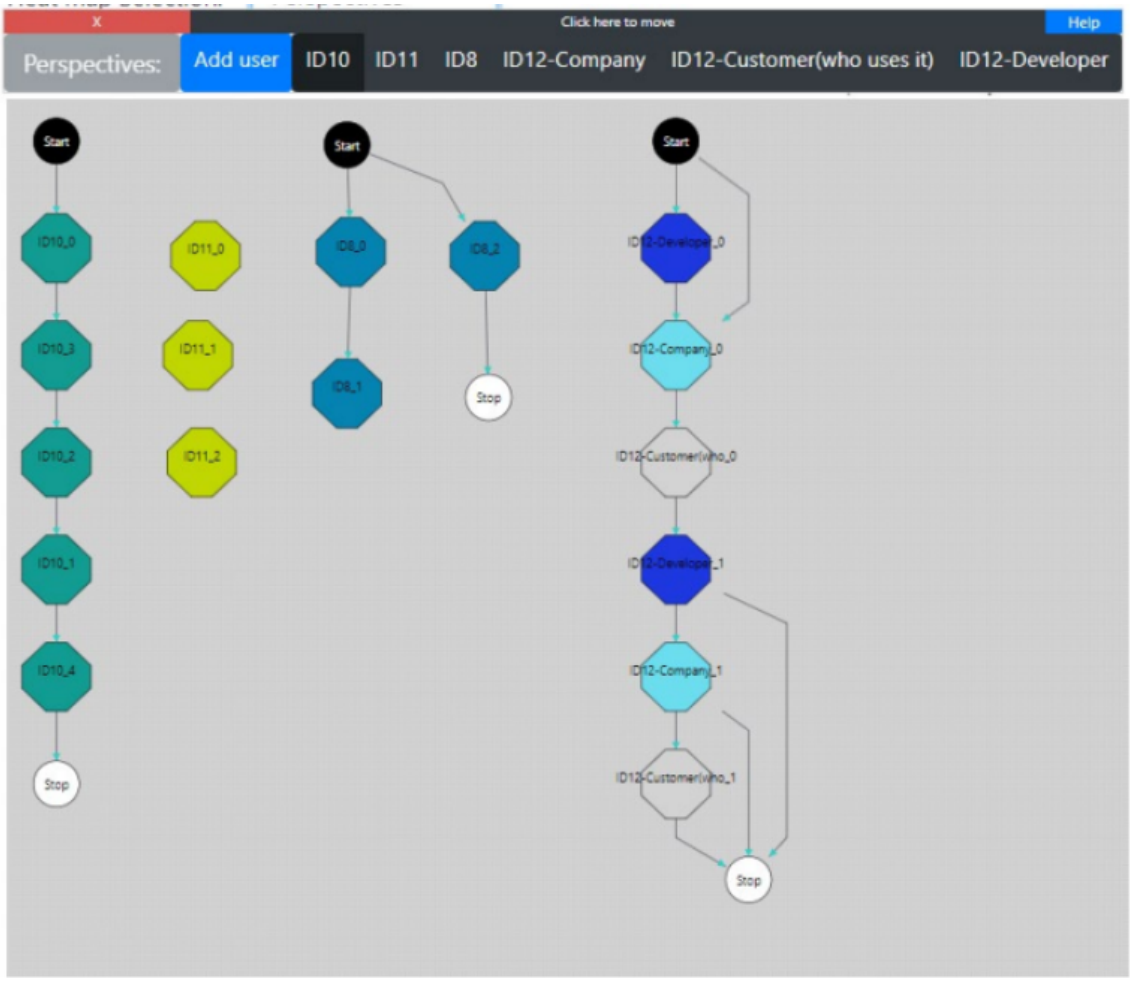

Figure D.7: Case Group Three's Session 2 Individual Models

D.3.2 Consensus Model

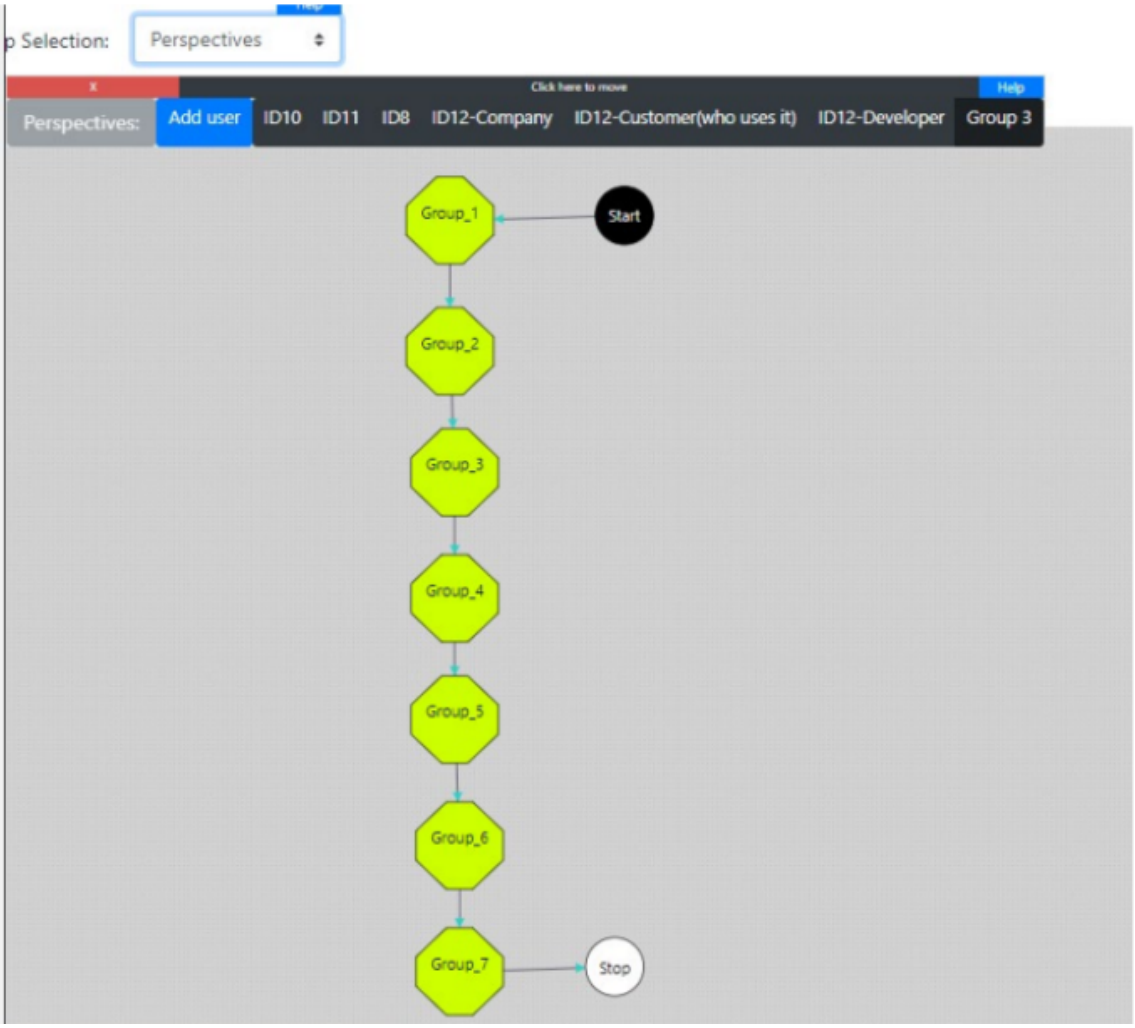

Figure D.8: Case Group Three's Session 3 Consolidated Model

D.3.3 Group Consensus Analysis

Element Type	ID	NAME	USER	Ambiguity Type	Severity	Intentionality	Implementability	Regulatory Text	Regulatory Text ID	Notes
Ambiguity Element	Group_3_1	Group_3_1	Group_3	Referential	3	y	y	erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay	EU_GDPR_Ch3_Art17	Here erasure of personal data concerning user then who is the controller the government or the company. The confusion is whom they are referring to it
Ambiguity Element	Group_3_2	Group_3_2	Group_3	Vagueness	2	y	y	the data subject objects to the processing pursuant to Article 21(1) and there are no overriding legitimate grounds for the processing, or the data subject objects to the processing pursuant to Article 21(2);	EU_GDPR_Ch3_Art17	here the data subject objecting process persuant to article 21(1),(2) at a time
Ambiguity Element	Group_3_3	Group_3_3	Group_3	Incompleteness	3	y	y	the personal data have been unlawfully processed;	EU_GDPR_Ch3_Art17	More detail could have been specified to specify the actions
Ambiguity Element	Group_3_4	Group_3_4	Group_3	Semantic	3	n	y	the personal data have to be erased for compliance with a legal obligation in Union or Member State law to which the controller is subject;	EU_GDPR_Ch3_Art17	how the controller is subject, on what grounds . too many meanings.
Ambiguity Element	Group_3_5	Group_3_5	Group_3	Incompleteness	4	y	y	Where the controller has made the personal data public and is obliged pursuant to paragraph 1 to erase the personal data, the controller, taking account of available technology and the cost of implementation, shall take reasonable steps, including technical measures, to inform controllers which are processing the personal data that the data subject has requested the erasure by such controllers of any links to, or copy or replication of, those personal data.	EU_GDPR_Ch3_Art17	The sentence is incomplete or is delivering less meaning to the reader on the specific activities of the controller.
Ambiguity Element	Group_3_6	Group_3_6	Group_3	Incompleteness	4	y	y	for the establishment, exercise or defence of legal claims.	EU_GDPR_Ch3_Art17	need more information on the highlighted sentence
Ambiguity Element	Group_3_7	Group_3_7	Group_3	Syntactic	2	y	y	for the establishment, exercise or defence of legal claims	EU_GDPR_Ch3_Art17	"For the establishment" part of the sentence can have multiple interpretations as it could mean anything that we have established in the above paragraphs.
Terminal Element	Group_3_Stop	Start								

Figure D.9: Case Group Three's Session 3 Consolidated Analysis

Appendix E

Focus Group Appendix

The following is the Focus Group protocol as described in Chapter 6, Section 6.1.1. Slides and other artifacts are available at the following DOI: `https://figshare.com/s/7d3752cd8d99631fb8a4`

E.1 Focus Group's Protocol

Session 1:

1. Researcher Introduction- The Facilitator and Moderator will introduce themselves and start the session.

2. Introduction of the Focus Group Participant

   (a) Each participant will introduce themselves

   (b) Outline your current job or role in the software industry

   (c) Provide background and industry experience

3. Overview of the Study

   (a) Present the Purpose and the Goal of the Study

   (b) Why we recruited these participants for the study

   (c) Provide Background on Ambiguity Analysis within Software Development

4. Present the process of Regulatory Ambiguity Modeling

    (a) Live demonstration of the Ambiguity Modeling Process

    (b) Question and Answer on Ambiguity Modeling and Live demonstration

5. Present the Data from Previous Work

    (a) Present views of consolidated models and findings

        i. Present Artifacts and Findings from Multi-Case Study

        ii. Present Models and Legal Researcher's Analysis on Ambiguity model from Virginia's Consumer Data Protection Act(VCPDA)

    (b) Question and Answer from Focus Group Participants

6. Close of the Session: (Session 2's discussion questions)

    (a) From an auditor's perspective, if presented with an ambiguity model as part of the documentation during an audit, what information could you get from that model that would be useful for your auditing task?

    (b) Would adopting Ambiguity Modeling help resolve any difficulties you (or auditors) often experience when auditing for regulatory compliance?

    (c) What other artifacts for regulatory compliance could a software development team produce that could help an auditor assess their development process?

Session 2:

1. Reintroduction and Study Overview

(a) Researcher introduction

(b) Participants introduction

(c) Session Agenda

(d) Study's goal and purpose

2. **Question 1:** From an auditor's perspective, if presented with an ambiguity model as part of the documentation during an audit, what information could you get from that model that would be useful for your auditing task?

3. **Question 2:** Would adopting Ambiguity Modeling help resolve any difficulties you (or auditors) often experience when auditing for regulatory compliance?

4. **Question 3:** What other artifacts for regulatory compliance could a software development team produce that could help an auditor assess their development process?

5. Follow-up or prompts for questions:

   (a) Can you elaborate on your comments?

   (b) Do you think the model show evidence compliant culture?

   (c) Does the ambiguity modeling process align with the Software Developers/Engineers' Professional Responsibilities outlined in the ACM code of ethics?

   (d) Outside regulatory analysis, can the ambiguity modeling process have other applications you can think of?

6. Close-out questions:

(a) Is there something you might include in the ambiguity modeling process to make it more interesting/usable/helpful for you?

(b) What are your closing thoughts?

7. Thank everyone for their time and participation.

E.2 Focus Group's Coding Scheme

Code	Subcodes	Definitions
Demographics		Define: This code highlights our participants' demographics both stated in the interview and verified via other content such as LinkedIn or Bios provided.
	Auditing Experience	Define: Participants stated that they have performed an Audit internally within their organization or as an external auditor.
	Organizations	Define: Participants describe what organizations they come from giving a name, focus, or organizational size.
	Previous Experience with Regulation	Define: Examples participants brought up during discussion on their previous experience in interpreting regulation.
	Regulated Industry	Define: The Regulated domain the participant has worked in.
	Current Role in Software Industry	Define: The Participant's Current and Past Roles within the Software Development Industry
	Time in Software Industry	Define: The Participant's Years in Software Industry combined and in current role in Years.
Initial Thoughts on Ambiguity Modelling		Define: Questions and thoughts asked during or immediately after the demonstration in Session1.
Ambiguity Model on Regulation		Define: The participant's comments on whether they would or would not use ambiguity modelling on a regulation and Why? (Note: I added this because of Aaron comment that them not using on a Regulation is a finding and we should capture their thoughts on the why)
	Maybe Useful	Define: Participant commented that it was " Maybe Useful to model a Regulation" They saw potential BUT they voiced a concern.
	Not Useful	Define: Participant commented that it was "Not Useful to model a Regulation"
	Useful	Define: Participant commented that it was "Useful to model a Regulation"

Usability		Define: Participants comment on how they would use ambiguity modelling within their software development process or what potential they saw in the usability of ambiguity modelling as a practice with software development. (Questions possibly answered: What did our participant say about the usability of ambiguity modelling?)
	Auditing	Define: This code is divided into two subcodes on how our participant responded to the internal and external uses of Ambiguity Modelling.
	External Auditing	Define: The Participant's comment on using an Ambiguity modelling during an external audit by a regulatory agency. (Questions possibly answered: What an External auditor might think, or how would they view the tool? How might it be used for an external audit?)
	Internal Auditing	Define: The Participant's comment on using an Ambiguity modelling during an internal audit within the organization. (Note: This might overlap with the Internal Support Tool subcodes)
	Internal Support Tool	Define: The Participant's comment on Ambiguity modelling as an internal support or supplemental tool within the software development process. This code is divided into five subcodes on how the participant would use the ambiguity model and AHAB tool internally within their software development Process.
	Artifact Development	Define: The Participant's comment on Outputs created based on resulting guidance or clarification linked to the ambiguity model. (Note: This might overlap with the Internal Auditing subcodes)
	Clarify Regulatory Requirements	Define: The Participant's comment on using model to clarify regulatory requirements for a software design
	Deconflicting Requirements	Define: The Participant's comment Ambiguity modelling use to de-conflict requirements for a software design that is linked to a how and why certain decisions or other outputs were created at a moment in time.

	Disambiguating Regulatory requirements	Define: The Participant's comments on how they would use ambiguity modelling to disambiguate a regulation or a regulatory requirement linked to a regulation.
	Documenting discussion on regulatory compliance	Define: The participant's comments on how they would use the tool to document compliance related decisions or outputs within the software development process.
	Signals the Intent to Comply	Define: The participant's comment on additional features or updates for the AHAB tool to increase its usability (i.e., risk score, update field, output linkage)
Other		Define: Points made during the discussion that I do not know how to code.
	Intentional ambiguity	Define: Participants made a point that regulations have intentional ambiguity for reasons like level of available tech, protect vendors from overregulation, etc.
	Unknown Ambiguity	Define: When the perception is that the regulation is clear. What is unknown is how enforcement will interpret it.
	Variance of Usability	Define: The participant's comment on the variance of usability for ambiguity modelling based on Company, a company's size,

Bibliography

[1] Justice manual — 1-19.000 – principles for issuance and use of guidance documents.

[2] Airbus offers new fuel saving engine options for a320 family, Dec 2010.

[3] Airbus with new order record at paris air show 2011, Jun 2011.

[4] What we know: the volkswagen emissions test fraud scandal, Sep 2015.

[5] How an irate developer briefly broke javascript, Mar 2016.

[6] How 'dieselgate' may become a tipping point in the global ev race, Oct 2016.

[7] Volkswagen to spend up to $14.7 billion to settle allegations of cheating emissions tests and deceiving customers on 2.0 liter diesel vehicles, Jun 2016.

[8] Boeing ceo outlines 737 max mcas software fix in congressional hearings, Apr 2019.

[9] Boeing relied on single sensor for 737 max that had been flagged 216 times to faa, Mar 2019.

[10] Managing risk in aircraft certification, Aug 2019.

[11] The many human errors that brought down the boeing 737 max, Mar 2019.

[12] Code section group, Dec 2020.

[13] Airworthiness certification overview, Jun 2022.

[14] Captain "sully" sullenberger testifies to congress about boeing 737 max 8 airplanes and deadly crashes - cbs news, Mar 2024.

[15] Data analytics and ai platform — altair rapidminer, Jan 2024.

[16] Fake security researcher github repositories deliver malicious implant - blog, Apr 2024.

[17] Hipaa violation fines, Feb 2024.

[18] Introduction to k-means clustering, Jan 2024.

[19] Nice, Jun 2024.

[20] Norris Syed Abdullah, Shazia Sadiq, and Marta Indulska. Emerging challenges in information systems research for regulatory compliance management. In *International Conference on Advanced Information Systems Engineering*, pages 251–265. Springer, 2010.

[21] Norris Syed Abdullah, Shazia Sadiq, and Marta Indulska. Information systems research: Aligning to industry challenges in management of regulatory compliance. In *PACIS 2010-14th Pacific Asia Conference on Information Systems*, volume 14, pages 546–557. Pacific Asia Conference on Information Systems, 2010.

[22] Norris Syed Abdullah, Shazia Sadiq, and Marta Indulska. A framework for industry-relevant ontology development. In *ACIS 2011 Proceedings-22nd Australasian Conference on Information Systems*. AIS Electronic Library (AISeL), 2011.

[23] Norris Syed Abdullah, Shazia Sadiq, and Marta Indulska. A study of ontology construction: the case of a compliance management ontology. In *Ontology-Based Applications for Enterprise Systems and Knowledge Management*, pages 276–291. IGI Global, 2013.

[24] Syed Norris Abdullah, Marta Indulska, and Shazia Sadiq. A study of compliance management in information systems research. In *17th European Conference on Information Systems, ECIS 2009*, pages 1–10. European Conference on Information Systems, 2009.

[25] Shams Al-Amin, Nirav Ajmeri, Hongying Du, Emily Z Berglund, and Munindar P Singh. Toward effective adoption of secure software development practices. *Simulation Modelling Practice and Theory*, 85:33–46, 2018.

[26] Orlando Amaral, Sallam Abualhaija, Mehrdad Sabetzadeh, and Lionel Briand. A model-based conceptualization of requirements for compliance checking of data processing against gdpr. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 16–20. IEEE, 2021.

[27] Annie I Antón, Travis D Breaux, Dimitris Karagiannis, and John Mylopoulos. First international workshop on requirements engineering and law (relaw). In *2008 Requirements Engineering and Law*, pages i–iv. IEEE, 2008.

[28] Karthick Arvinth. Vw scandal: Carmaker was warned by bosch about test-rigging software in 2007, Sep 2015.

[29] Association of Computing Machinery. ACM Code of Ethics and Professional Conduct.

[30] Kenneth A. Bamberger and Deirdre K. Mulligan. Privacy on the Books and on the Ground. *Stanford Law Review*, 63:247–316, 2011.

[31] Kenneth A. Bamberger and Deirdre K. Mulligan. *Privacy on the Ground: Driving Corporate Behavior in the United States and Europe*. The MIT Press, Cambridge, Massachusetts, 1st edition edition, October 2015.

[32] Daniel Berry, Erik Kamsties, and Michael Krieger. From contract drafting to software specification: Linguistic sources of ambiguity. Nov 2003.

[33] Daniel M Berry and Erik Kamsties. Ambiguity in requirements specification. In *Perspectives on software requirements*, pages 7–44. Springer, 2004.

[34] Daniel M Berry and Erik Kamsties. The syntactically dangerous all and plural in specifications. *IEEE software*, 22(1):55–57, 2005.

[35] Jaspreet Bhatia and Travis D Breaux. Semantic incompleteness in privacy policy goals. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 159–169. IEEE, 2018.

[36] Jaspreet Bhatia, Travis D Breaux, Joel R Reidenberg, and Thomas B Norton. A theory of vagueness and privacy risk perception. In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 26–35. IEEE, 2016.

[37] Jaspreet Bhatia, Morgan C Evans, and Travis D Breaux. Identifying incompleteness in privacy policy goals using semantic frames. *Requirements Engineering*, 24(3):291–313, 2019.

[38] Stephen Boyd, Didar Zowghi, and Alia Farroukh. Measuring the expressiveness of a constrained natural language: An empirical study. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 339–349. IEEE, 2005.

[39] Travis Breaux and Annie Antón. Analyzing regulatory rules for privacy and security requirements. *IEEE transactions on software engineering*, 34(1):5–20, 2008.

[40] Travis D Breaux, Ana I Anton, and Matthew W Vail. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. Technical report, North Carolina State University. Dept. of Computer Science, 2006.

[41] Travis D. Breaux and Thomas Norton. Legal accountability as software quality: A u.s. data processing perspective. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*, pages 101–113, 2022.

[42] Francis Chantree, Bashar Nuseibeh, Anne De Roeck, and Alistair Willis. Identifying nocuous ambiguities in natural language requirements. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 59–68. IEEE, 2006.

[43] Nicola Clark. Jet order by american is a coup for boeing's rival, Jul 2011.

[44] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, Inc, Los Angeles, fourth edition edition, December 2014.

[45] Sebastian Dännart, Fabiola Moyón Constante, and Kristian Beckers. An assessment model for continuous security compliance in large scale agile environments. In *International Conference on Advanced Information Systems Engineering*, pages 529–544. Springer, 2019.

[46] Tom DeMarco and Tim Lister. *Peopleware: productive projects and teams*. Dorset House Publishing Co., Inc., 353 West 12th Street, New York, NY, 2nd ed edition, 1999.

[47] Christian Denger, Daniel M Berry, and Erik Kamsties. Higher quality requirements specifications through natural language patterns. In *Proceedings 2003 Symposium on Security and Privacy*, pages 80–90. IEEE, 2003.

[48] Jack Ewing. Volkswagen says 11 million cars worldwide are affected in diesel deception. *The New York Times*, Sep 2015.

[49] Jake Ewing. Ex-volkswagen c.e.o. charged with fraud over diesel emissions, May 2018.

[50] Saad Ezzini, Sallam Abualhaija, Chetan Arora, and Mehrdad Sabetzadeh. Automated handling of anaphoric ambiguity in requirements: a multi-solution study. In *Proceedings of the 44th International Conference on Software Engineering*, pages 187–199, 2022.

[51] Saad Ezzini, Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, and Lionel C Briand. Using domain-specific corpora for improved handling of ambiguity in requirements. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1485–1497. IEEE, 2021.

[52] Alessio Ferrari, Beatrice Donati, and Stefania Gnesi. Detecting domain-specific ambiguities: an nlp approach based on wikipedia crawling and word embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 393–399. IEEE, 2017.

[53] Alessio Ferrari and Andrea Esuli. An nlp approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 26(3):559–598, 2019.

[54] Alessio Ferrari, Andrea Esuli, and Stefania Gnesi. Identification of cross-domain ambiguity with language models. In *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 31–38. IEEE, 2018.

[55] Alessio Ferrari, Franco Mazzanti, Davide Basile, and Maurice Ter Beek. Systematic evaluation and usability analysis of formal methods tools for railway signaling system design. *IEEE Transactions on Software Engineering*, 2021.

[56] Alessio Ferrari, Franco Mazzanti, Davide Basile, Maurice H. ter Beek, and Alessandro Fantechi. Comparing formal tools for system design: a judgment study. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 62–74, 2020.

[57] Glenn Fleishman. Every zoom security and privacy flaw so far, and what you can do to protect yourself, Apr 2020.

[58] Vicente Franco, Francisco Posada Sanchez, John German, and Peter Mock. Real-world exhaust emissions from modern diesel cars, Oct 2014.

[59] Sean Gallagher. Rage-quit: Coder unpublished 17 lines of javascript and "broke the internet".

[60] Dominic Gates and Mike Baker. The inside story of mcas: How boeing's 737 max system gained power and lost safeguards. *The Seattle Times*, Jun 2019.

[61] Dominic Gates, Steve Miletich, and Lewis Kamb. Boeing pushed faa to relax 737 max certification requirements for crew alerts, Oct 2019.

[62] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Compliance analysis based on a goal-oriented requirement language evaluation methodology. In *2009 17th IEEE International Requirements Engineering Conference*, pages 133–142. IEEE, 2009.

[63] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. A systematic review of goal-oriented requirements management frameworks for business process compliance. In *2011 Fourth International Workshop on Requirements Engineering and Law*, pages 25–34. IEEE, 2011.

[64] Sepideh Ghanavati, Daniel Amyot, and André Rifaut. Legal goal-oriented requirement language (legal grl) for modeling regulations. In *Proceedings of the 6th international workshop on modeling in software engineering*, pages 1–6, 2014.

[65] Sepideh Ghanavati, André Rifaut, Eric Dubois, and Daniel Amyot. Goal-oriented compliance with multiple regulations. In *2014 IEEE 22nd international requirements engineering conference (RE)*, pages 73–82. IEEE, 2014.

[66] Leah McGrath Goodman. Why volkswagen cheated, Dec 2015.

[67] James Grimmelmann. The vw scandal is just the beginning, Sep 2015.

[68] Julie Haney and Wayne Lutters. Skills and Characteristics of Successful Cybersecurity Advocates. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[69] Julie M. Haney and Wayne G. Lutters. "It's Scary... It's Confusing... It's Dull": How Cybersecurity Advocates Overcome Negative Perceptions of Security. In *Fourteenth Symposium on Usable Privacy and Security ({SOUPS} 2018)*, pages 411–425, 2018.

[70] Mustafa Hashmi, Guido Governatori, Ho-Pun Lam, and Moe Thandar Wynn. Are we done with business process compliance: state of the art and challenges ahead. *Knowledge and Information Systems*, 57(1):79–133, 2018.

[71] Waël Hassan and Luigi Logrippo. Governance requirements extraction model for legal compliance validation. In *2009 Second International Workshop on Requirements Engineering and Law*, pages 7–12. IEEE, 2009.

[72] Joseph Herkert, Jason Borenstein, and Keith Miller. The boeing 737 max: Lessons for engineering ethics. *Science and engineering ethics*, 26:2957–2974, 2020.

[73] S. E. Hove and B. Anda. Experiences from conducting semi-structured interviews in empirical software engineering research. In *11th IEEE International Software Metrics Symposium (METRICS'05)*, pages 10 pp.–23, 2005.

[74] IEEE. Ieee recommended practice for software requirements specifications. *IEEE Std 830-1993*, pages 1–32, 1994.

[75] Chris Isidore. The 737 max crisis costs continues to climb two years after the second fatal crash, Mar 2021.

[76] ITU-T. Z.151: User requirements notation (urn) – language definition, Oct 2018.

[77] Eric Jaffe. The study that brought down volkswagen, Sep 2015.

[78] Jan Ole Johanssen, Anja Kleebaum, Barbara Paech, and Bernd Bruegge. Practitioners' eye on continuous software engineering: An interview study. In *Proceedings of the 2018 International Conference on Software and System Process*, page 41–50, New York, NY, USA, 2018. Association for Computing Machinery.

[79] Russell L Jones and Abhinav Rastogi. Secure coding: building security into the software development life cycle. *Inf. Secur. J. A Glob. Perspect.*, 13(5):29–39, 2004.

[80] Erik Kamsties. Understanding ambiguity in requirements engineering. In *Engineering and Managing Software Requirements*, pages 245–266. Springer, 2005.

[81] Erik Kamsties, Daniel M Berry, and Barbara Paech. Detecting ambiguities in requirements documents using inspections. In *Proceedings of the first workshop on inspection in software engineering (WISE'01)*, volume 13, 2001.

[82] Erik Kamsties and Barbara Peach. Taming ambiguity in natural language requirements. In *Proceedings of the Thirteenth international conference on Software and Systems Engineering and Applications*, 2000.

[83] Erik Kamsties, Antje Von Knethen, Jan Philipps, and Bernhard Schätz. An empirical investigation of the defect detection capabilities of requirements specification languages. In *EMMSAD'01: Proceedings of the Sixth CAiSE/IFIP8. 1 International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design*, 2001.

[84] Evelyn Kempe and Aaron Massey. Perspectives on regulatory compliance in software engineering. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 46–57. IEEE, 2021.

[85] Evelyn Kempe and Aaron K Massey. Regulatory and security standard compliance throughout the software development lifecycle. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021.

[86] Evelyn Kempe, Samin Semsar, Aaron Massey, Sreedevi Sampath, and Carolyn Seaman. Modeling, analyzing, and communicating regulatory ambiguity: An empirical study. In *Workshop on Multi-disciplinary, Open, and RElevant Requirements Engineering (MO2RE 2024)*. ACM, 2024.

[87] Hossein Keramati and Seyed-Hassan Mirian-Hosseinabadi. Integrating software development security activities with agile methodologies. In *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pages 749–754. IEEE, 2008.

[88] Raula Gaikovina Kula, Daniel M. German, Ali Ouni, Takashi Ishio, and Katsuro Inoue. Do developers update their library dependencies? *Empirical Software Engineering*, 23(1):384–417, February 2018.

[89] Ruchika Kumar and Gunter Mussbacher. Textual user requirements notation. In *International Conference on System Analysis and Modeling*, pages 163–182. Springer, 2018.

[90] David Lengyel. Examining risk management failures: The case of the boeing 737 max program. *Enterprise Risk Management*, 8(1):1, 2023.

[91] E. Lim, N. Taksande, and C. Seaman. A Balancing Act: What Software Practitioners Have to Say about Technical Debt. *IEEE Software*, 29(6):22–27, November 2012.

[92] Jonathan Lockhart, Carla Purdy, and Philip Wilsey. Formal methods for safety critical system specification. In *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 201–204. IEEE, 2014.

[93] Luiz Eduardo G Martins and Tony Gorschek. Requirements engineering for safety-critical systems: An interview study with industry practitioners. *IEEE Transactions on Software Engineering*, 46(4):346–361, 2018.

[94] Aaron K Massey, Eric Holtgrefe, and Sepideh Ghanavati. Modeling regulatory ambiguities for requirements analysis. In *International Conference on Conceptual Modeling*, pages 231–238. Springer, 2017.

[95] Aaron K. Massey, Richard L. Rutledge, and Annie I. Antón. Identifying and Classifying Ambiguity for Regulatory Requirements. *22nd IEEE International Requirements Engineering Conference (RE),* Karlskrona, Sweden, pages 83–92, 2014.

[96] Aaron K Massey, Richard L Rutledge, Annie I Antón, Justin D Hemmings, and Peter P Swire. A strategy for addressing ambiguity in regulatory requirements. Technical report, Georgia Institute of Technology, 2015.

[97] Aaron K Massey, Richard L Rutledge, Annie I Antón, and Peter P Swire. Identifying and classifying ambiguity for regulatory requirements. In *2014 IEEE 22nd international requirements engineering conference (RE)*, pages 83–92. IEEE, 2014.

[98] Jeremy C Maxwell and Annie I Antón. Developing production rule models to aid in acquiring requirements from legal texts. In *2009 17th IEEE International requirements engineering conference*, pages 101–110. IEEE, 2009.

[99] Jeremy C Maxwell, Annie I Antón, and Peter Swire. A legal cross-references taxonomy for identifying conflicting software requirements. In *2011 IEEE 19th international requirements engineering conference*, pages 197–206. IEEE, 2011.

[100] Gary McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, 2004.

[101] Sharan B. Merriam and Elizabeth J. Tisdell. *Qualitative Research: A Guide to Design and Implementation*. John Wiley & Sons, San Francisco, CA, 4th edition edition, August 2015.

[102] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic Fairness: Choices, Assumptions, and Definitions. *Annual Review of Statistics and Its Application*, 8(1):null, 2021.

[103] Fabiola Moyón, Christoph Bayr, Daniel Mendez, Sebastian Dännart, and Kristian Beckers. A light-weight tool for the self-assessment of security compliance in software development–an industry case. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 403–416. Springer, 2020.

[104] Fabiola Moyón, Kristian Beckers, Sebastian Klepper, Philipp Lachberger, and Bernd Bruegge. Towards continuous security compliance in agile software development at scale. In *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*, pages 31–34. IEEE, 2018.

[105] Fabiola Moyón, Daniel Méndez, Kristian Beckers, and Sebastian Klepper. Using process models to understand security standards. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 458–471. Springer, 2021.

[106] Lily Hay Newman. So wait, how encrypted are zoom meetings really?, Apr 2020.

[107] Association of Computing Machinery. Acm code of ethics and professional conduct.

[108] U.S. Bureau of Labor Statistics. Software developers, quality assurance analysts, and testers : Occupational outlook handbook.

[109] Jon Ostrower. What is the boeing 737 max maneuvering characteristics augmentation system?, Nov 2018.

[110] P. N. Otto, Annie I. Antón, and D. L. Baumer. The ChoicePoint Dilemma: How Data Brokers Should Handle the Privacy of Personal Information. *Security & Privacy Magazine, IEEE*, 5(5):15–23, October 2007.

[111] Paul N. Otto and Annie I. Antón. Addressing Legal Requirements in Requirements Engineering. In *15th IEEE International Requirements Engineering Conference*, pages 5–14, 2007.

[112] Bruce Potter. Microsoft sdl threat modelling tool. *Network Security*, 2009(1):15–18, 2009.

[113] Joel R Reidenberg, Jaspreet Bhatia, Travis D Breaux, and Thomas B Norton. Ambiguity in privacy policies and the impact of regulation. *The Journal of Legal Studies*, 45(S2):S163–S190, 2016.

[114] Christina Rogers and Mike Spector. Judge slaps vw with $2.8 billion criminal fine in emissions fraud, Apr 2017$.

[115] Stefan Sackmann, Stephan Kuehnel, and Tobias Seyffarth. Using business process compliance approaches for compliance management with regard to digitization: evidence from a systematic literature review. In *International Conference on Business Process Management*, pages 409–425. Springer, 2018.

[116] C. B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, July 1999.

[117] Monika Singh, Ashok Kumar Sharma, Ruhi Saxena, et al. Why formal methods are considered for safety critical systems? *Journal of Software Engineering and Applications*, 8(10):531, 2015.

[118] Stan Stahl. Ccpa and minimum reasonable security procedures and practices: A floor on "defendability" - citadelonsecurity, Jun 2019.

[119] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. Grounded theory in software engineering research: a critical review and guidelines. In *Proceedings of the 38th International conference on software engineering*, pages 120–131, 2016.

[120] Norris Syed Abdullah, Shazia Sadiq, and Marta Indulska. A compliance management ontology: developing shared understanding through models. In *Proceedings of the 24th international conference on Advanced Information Systems Engineering*, pages 429–444, 2012.

[121] Ashfa Umber and Imran Sarwar Bajwa. Minimizing ambiguity in natural language software requirements specification. In *2011 Sixth International Conference on Digital Information Management*, pages 102–107, 2011.

[122] Muhammad Usman, Michael Felderer, Michael Unterkalmsteiner, Eriks Klotins, Daniel Mendez, and Emil Alegroth. Compliance requirements in large-scale software development: An industrial case study. In *International Conference on Product-Focused Software Process Improvement*, pages 385–401. Springer, 2020.

[123] Walter G. Vincenti. *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*. Johns Hopkins University Press, Baltimore, new edition edition, February 1993.

[124] Zack Whittaker. Ex-nsa hacker drops new zero-day doom for zoom, Apr 2020.

[125] Eric S. Yuan. Ceo report: 90 days done, what's next for zoom, 2020.

[126] Eric S Yuan. A message to our users, Apr 2020.

[127] Markus Zimmermann, Cristian-Alexandru Staicu, Cam Tenny, and Michael Pradel. Small World with High Risks: A Study of Security Threats in the npm Ecosystem. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 995–1010, 2019.