2021 IEEE 29th International Requirements Engineering Conference (RE) | 978-1-6654-2856-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/RE51729.2021.00012

Perspectives on Regulatory Compliance in Software Engineering

Evelyn Kempe and Aaron Massey
Department of Information Systems
University of Maryland, Baltimore County
Baltimore, MD, USA
Email: {ekempe1, akmassey}@umbc.edu

***Abstract*—Compliance reviews within a software organization are internal attempts to verify regulatory and security requirements during product development before its release. However, these reviews are not enough to adequately assess and address regulatory and security requirements throughout a software's development lifecycle. We believe requirements engineers can benefit from an improved understanding of how software practitioners treat and perceive compliance requirements. This paper describes an interview study seeking to understand how regulatory and security standard requirements are addressed, how burdensome they may be for businesses, and how our participants perceived them in the software development lifecycle. We interviewed 15 software practitioners from 13 organizations with different roles in the software development process and working in various industry domains, including big tech, healthcare, data analysis, finance, and small businesses. Our findings suggest that, for our participants, the software release process is the ultimate focus for regulatory and security compliance reviews. Also, most participants suggested that having a defined process for addressing compliance requirements was freeing rather than burdensome. Finally, participants generally saw compliance requirements as an investment for both employees and customers. These findings may be unintuitive, and we discuss seven lessons this work may hold for requirements engineering.**

I. Introduction

Society uses laws, regulations, and security standards to embed ethical norms into engineering systems, including software systems. However, this process is not free. Large multinational organizations spend on average about $5.5 million USD annually to comply with regulation at the state, federal, and international levels [1]. Should an organization fail to comply with any applicable regulation, the estimated average cost is about $14.82 million USD [1]. This monetary cost does not include ethical and reputational costs that organizations pay when publicly found to be non-compliant. Despite these consequences, incidents of non-compliance and an overall lack of due diligence regularly make headline news.

Consider, as an example, Zoom's security incidents during the COVID-19 pandemic [2]–[5]. Zoom became massively popular in part because it appeared to have everything a user could want: great quality audio and video with strong privacy protection through end-to-end encryption. The U.S. Department of Health and Human Services even affirmed Zoom for use in telemedicine during the pandemic by relaxing regulatory enforcement [6]. Unfortunately, Zoom's privacy and security practices were not what they appeared to be. Zero-day vulnerabilities [2], broken end-to-end encryption [4], and a myriad of other concerns [3] surfaced in the weeks after the pandemic-related lockdowns became pervasive. These security failures are disturbing because of Zoom's history of failing to protect and secure its platform [7], [8]. In response to the vulnerabilities uncovered during the pandemic, Zoom froze feature development to focus on the security and privacy issues with the platform [5]. Although enforcement actions are pending, they may be nothing more than a "slap on the wrist" with Zoom banking a 317% revenue increase in 2020 [9].

Zoom is not alone. For decades, software organizations have treated regulatory and security standard requirements as tomorrow's problem by operating under the "move fast and break things" (MFBT) motto to release a "minimum viable product" [10]. Developing software this way allows software organizations to keep pace with consumers' expectations for faster product releases, but it exposes the company—and their customers—to greater losses when incidents occur [11]. By then, though, the software organization has disrupted the industry and established a market presence that allows them to hire lawyers and public relations staff to deal with the things they broke when they were moving fast. MFBT is not sustainable long term [11]. Eventually, some incident or event happens, forcing a software organization to fix what they have broken, as Zoom did [12], or risk losing what they have gained in customers and revenue.

To make compliance a first-class concern in the software development process (SDP), requirements engineers need more insight from the software industry. To this end, we present an interview study of 15 software developers, managers, and directors from 13 different organizations. This interview study is meant to answer the question: How software practitioners treat and perceive regulatory and security standard requirements as part of their SDP? This paper has taken an empirical approach to understand the software practitioners perspective on addressing and managing the regulatory and security standard compliance landscape. Little research is currently available that represents the industry perspective on compliance [13]. This gap in the research knowledge motivated us to further and contribute through this interview study.

This interview study seeks to understand how these requirements are addressed, how burdensome they are for businesses to implement, and how our participants perceived them in the

978-1-6654-2856-9/21/$31.00 ©2021 IEEE
DOI 10.1109/RE51729.2021.00012

software development lifecycle. Our main findings include:

1) The software release process is the aspect of the SDLC where software companies most consistently and regularly examine regulatory and security standard compliance for their products. This is also reflected regardless of the software development process used for products.
2) Software developers view compliance checks and process redundancy related to regulatory compliance and security as freeing rather than burdensome.
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. Compliance is a means of building or establishing trust with customers which yields financial rewards.

We also compiled and discuss seven lessons requirements engineers may take from this study detailed in our Discussion section and summarized in our Summary section.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the methods we used to collect, code, and analyze the data in this interview study. Section IV presents the findings from our study, and Section V discusses the implications of these findings for requirements engineering. Section VI identifies the key limitations of this work. Finally, Section VII summarizes this paper and our plans for future work.

II. Related Work

We first briefly discuss the background and history of research on regulatory and security standard compliance in software systems research. Then we examine the use and design of interview studies to elicit practitioner perspectives on software development.

A. History of Regulatory and Security Standard Compliance Research

Software systems are under regulatory scrutiny with regard to an increasing set of social concerns. For security and privacy, compliance with regulations, standards, and best practices is not entirely new. Domains like healthcare,[1] children's privacy,[2] and finance[3] have been explicitly regulated for over 20 years. Indeed, some regulatory concerns date back to changes that motivated the Bamberger and Mulligan's interview study on privacy nearly a decade ago, which we discuss later on.[4]

Despite this increasing interest in promulgating regulations, enforcement of regulations and security standards has been lax [15]. On one hand, lax enforcement disincentivizes expensive compliance efforts, such as those that would necessitate changes to software development. On the other hand, relatively new regulations, like GDPR,[5] require more case law to set limits and enable stricter enforcement. Companies would respond to increased enforcement by making changes to their software development process [16]. Some domains have a significant history of regulatory compliance requirements, such as healthcare or finance. For software systems, security standards developed long before regulatory compliance standards [17]. In this paper, we are interested in both compliance with non-security regulations as well as compliance with security standards because many security-oriented regulations simply reference compliance with modern standards, like the NIST SP 800-171 standard [18], as sufficient for regulatory compliance. Other domains, like privacy, are still developing regulatory requirements [19]. Scholars are calling for additional regulatory oversight, particularly with respect to security [20]. In short, we may be at an inflexion point.

Within the last 15 years, requirements engineering research has sought to address regulatory compliance as a first-class concern within the software and requirements engineering process. Otto and Antón [21] examined how requirements engineering research managed laws and regulations through their literature review. Other secondary studies focused on regulatory compliance as part of business process compliance [22]–[24] or goal-modeling within requirements engineering [25], [26]. From 2008 to 2018, the International Workshop on Requirements Engineering and Law (RELAW), [6] sought to describe industry perspectives regarding techniques currently used to address regulatory compliance in software systems. Dozens of papers in these years also introduced regulatory compliance approaches for traceability [27], [28], goal modeling [29]–[31], requirements analysis [32], [33], and other areas of requirements engineering as tools for industry to use.

We conducted a systematic literature review finding that almost no work has been done to examine how software engineering practitioners actually respond to and implement changes to address regulatory compliance concerns, motivating us to do this interview study [13]. As an example of work that addresses this area, Usman et al. conducted a case study that investigated the "common practices and challenges with checking and analyzing regulatory compliance" with a product development team at the telecommunications company Ericsson AB [34]. However, they focused on a single company and compliance domain, whereas we conduct herein an interview study broadly focused on perceptions and regulatory compliance challenges across domains. As another example of work addressing this area, Abdullah et al. [35] conducted an interview study in Australia with 11 compliance management experts seeking to characterize challenges experienced by the software industry in 2007, but there has been little follow-on work related to this. Abdullah et al.'s [35] interview study and Usman et al.'s case study [34] were the only studies that we found that were aligned with the goals of our study. However, Abdullah et al. focused exclusively on a single stakeholder group. These examples are the exception. Most academic research on regulatory compliance does not seek to understand how practitioner's

[1] See HIPAA: Pub.L. 104–191, 110 Stat. 1936

[2] See COPPA: Pub.L. 105–277, 112 Stat. 2681–728

[3] See GLBA: Pub.L. 106–102, 113 Stat. 1338

[4] The regulatory history here is a bit complicated, but Hartzog's overview [14] covers it well.

[5] The EU's General Data Protection Regulation is available online here: `https://gdpr-info.eu/`

[6] `http://gaius.isri.cmu.edu/relaw/`

respond to regulatory requirements in practice.

B. Interview Studies to Elicit Practitioner Perspectives

Researchers often employ interviews to better understand non-functional aspects of software development, like sustainability [36], or process-related changes to the SDLC, like adoption of continuous software engineering practices [37]. Interviews are now a well-established research technique in software engineering [38]. Indeed, interviews are well-suited to such a task and have been used to that effect in software engineering [37], [39], which is the reason we employ the technique here.

Interviews sometimes identify potential gaps between research and practice. For example, Bamberger and Mulligan interviewed Chief Privacy Officers and other privacy industry leaders to better understand how privacy "on the ground" differs from privacy "on the books" [40]. Their work was conducted thirty years after major changes to privacy law necessitated organizational reform regarding data sharing across several industries. From this work, they were able to make recommendations to both academics and practitioners alike. Ultimately, the approach was so successful, that it was later expanded to a book [19]. Bamberger and Mulligan sought to identify how privacy practices were adopted at regulated organizations, but their study was targeted at organizational compliance rather than the specific impacts regulations may have within the SDLC. Therefore, we decided to conducted our own interview study in software engineering to examine how practitioners perceive and implement compliance efforts.

Haney and her colleagues conducted a series of interview studies that focused on how cybersecurity advocates promoted security practices within their organizations [41], [42]. In particular, they sought to identify what skills successful cybersecurity advocates brought to bear [42] and how these advocates were able to overcome negative perceptions of cybersecurity [41]. We believe this work to be the most similar research to our current study. They examined practitioners through interview studies, but their focus was on organizational adoption of cybersecurity practices. They wanted to know how organizations get their employees to implement a practice that is known to work. In contrast, we want to examine how organizations understand regulations and security standards and how they show that their requirements and implementation are compliant with those regulations or standards. Our goal is to eventually be able to examine how organizations demonstrate compliance to a third party auditor or regulator.

Our work uses qualitative research and analysis methods, described in the Methods, Findings, and Discussions sections of this paper. To start, we did some background research on interview studies within the software engineering field and constructed our interview guide based on that research [39]. Next, we conducted our interview study using a Straussian grounded theory [43] methodology outlined in Corbin and Strauss [44] using a constant comparative approach to analyze, code, and memo the transcripts collected during the interview process [44], [45]. Lastly, we describe and discuss our findings by looking at the human and organizational aspects of software development and regulatory compliance, which is why we used qualitative methods for our empirical study [46]. All these methods shape our research and refine our understanding of regulatory and security requirements management throughout the SDLC.

III. Methods

The interview study consisted of three phases: the pilot study, the main study, and the analysis phase. In this section, we discuss the design of our study III-A, the demographics of the participants III-B, and the methods used to transcribe, code, and analyze the data III-C.

A. The Interview Study design

We 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, we created two interview protocols [7]. The primary difference is one of emphasis technical process of software development and the other content of regulation and policy within the software industry. The goal being to keep all participants, from the different stakeholder groups [8], engaged throughout the interview. Once we incorporated the feedback, we began recruitment for the main interview study in September 2020.

The main interview study was conducted from October 2020-Jan 2021. We began with background questions about the participant and their organization. We then asked the participants about their interest and involvement in regulatory and security standard compliance. Next, we asked questions about their software development process (SDP), how well it worked, and why they use it. Combining the previous two sections (i.e., Background and SDP), we asked participants to reflect on how compliance was integrated in their software process. We wrapped up with open-ended questions regarding their desires or concerns for their compliance in their organization's software development process. Throughout the interview, we encouraged them to give examples to illustrate their decision-making, processes, and concerns. We conducted all of our interviews for the main study virtually and recorded them with video through Google Meets. We then transcribed the interviews using Otter AI and then organized and coded the transcripts using Excel. Our next subsection describes the demographics of our participants and their organizations in further detail.

B. Participant 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

[7] Interview protocols can be found online here: `https://doi.org/10.6084/m9.figshare.14842242.v1`

[8] Including software developers, project managers, technical directors, security engineers, data or privacy engineers, lawyers, policy managers, and business managers or directors.

Table I
INTERVIEW PARTICIPANT'S DEMOGRAPHIC

ID	Years of Exp	Industry	Org size	Education Level
D/PE1	<10	Technology	Large	PhD
D/PE2	10–20	Technology	Small	Grad
D/PE3	<10	Technology	Large	Grad
D/PE4	10–20	Other	Large	PhD
D/PE5	<10	Technology	Medium	Grad
D/PE6	10–20	Other	Medium	Grad
SD1	<10	Healthcare	Large	Undergrad
SD2	10–20	Technology	Medium	Grad
SD3	20+	Healthcare	Medium	Grad
SD4	<10	Finance	Large	Grad
M/D1	20+	Healthcare	Large	Grad
M/D2	20+	Other	Small	Undergrad
M/D3	20+	Healthcare	Small	PhD
M/D4	10–20	Technology	Large	PhD
M/D5	10–20	Government	Small	PhD

in their current role. We used the technically oriented protocol for 14 of the interviews and the law-and-policy protocol for the remaining 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. Participants came from different industries, focusing on the technology, healthcare, government, and financial sectors. Table I represents the pertinent demographic of varying backgrounds, roles, and company sizes for the main study participants.

We categorized the companies into five industries:

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

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

Government: Agencies in the U.S. Federal government.

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

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

We then categorized the participants into three groups according to their roles and self-reported titles:

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

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

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

Our definitions for company size represent both the structure development teams and resources available to software development.

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

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

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).

Lastly, years of experience refers only to work experience in a related role. Subsection III-C describes how we analyzed the data collected using these qualitative methods using Straussian grounded theory method [43] [44].

C. Analysis Methods

We conducted a Straussian grounded theory study based off the techniques outlined in Corbin and Strauss book [43], [44]. We did not form explicit research questions (RQs) and hypotheses beforehand because we wanted our RQs to be relevant to the requirements engineering field and any finding grounded in the data. We had a general idea and some guidance in conducting or analyzing the study as well as prior background research as a baseline for this study [13]. We just did not want to form any conclusions ahead of time. Thus, we chose Straussian Grounded Theory, because we wanted to reflect the perceptions of the industry software practitioners within the study [43]. To do so, we organized the analysis into 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. We 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. We 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. This was where we applied memoing, or constant comparison of the coding and data, [44]–[46], and it was conducted in February 2021. Once we formulated our findings, we needed to confirm them. Therefore we built and organized our field notes into 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 [46]) 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 nine high-level findings that had some support and three findings that had significant support within the data. Finally, we had three findings that did not have enough weighted support to merit reporting because only one or two participant provided evidence of that finding. These were excluded. Our next section covers our findings in more detail, focusing on our three best-supported high-level findings, and provides examples from the interviews using a narrative format.

IV. Results

Our three high-level findings with the most direct support from our interview data are as follows:

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

Finding 2: Software developers view compliance checks and process redundancy related to regulatory compliance and security as freeing rather than burdensome.

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.

We use these three findings to structure our presentation of both our results and discussion of this interview study. Although support for these findings was presented as responses to many questions from our interview protocols, they can be thought of as respectively addressing the following questions related to compliance within the software development lifecycle:

1) Where are regulatory and security requirements assessed and addressed in the SDLC?
2) How do non-requirements engineers perceive the regulatory and security compliance process? Why do they hold these perceptions?
3) How burdensome are regulatory and security requirements for businesses to implement?

Because our goal was insight, we did not impose a viewpoint during the interview and encouraged elaboration of their organizational and personal processes 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.

A. 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 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."

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. 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. 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.

B. Compliance-oriented Processes are 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 is crucial because the default position must be that regulators assume non-compliance without evidence demonstrating compliance. 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.

Our 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 gatekeeping 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.

C. 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, 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/PE4:** "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 indicated 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.

V. Discussion

Requirements engineers should be encouraged by these findings because software developers are clearly interested in implementing compliance requirements effectively. 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 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 requirements engineering researchers and practitioners structured around our three high-level findings presented in Section IV.

A. Compliance in the Release Process

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. **Lesson 1:** We, as requirements engineers, should consider targeting compliance requirements for the 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 first-class citizen from the start of any software engineering effort to ensure that it is done correctly.[9] 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 all that.

Four of our participants from larger companies explained that compliance is part of every step of the development process, from requirements, design, implementation, and change management within the maintenance phase. They also have resources to ensure that a secure 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."

However, that approach is not the norm. Security-focused developers as a roving resource are not something most companies have. This is also true for the legal compliance assessment of a software product. Whenever a big change within the regulatory compliance landscape occurs, such as the GDPR, organizations will review all of their products for compliance:

> **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."

The quote highlights our second lesson. **Lesson 2:** Requirements engineers 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. Also, incorporating compliance into the release process is not automatically done when a development company starts. This may happen only after a near miss or when a big mistake points to obvious defects made in the absence or inadequacy of a release process [49]. 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…"

This leads us to **Lesson 3:** We, as requirements engineers, should learn from organizations that have failed to achieve compliance and incorporate those lessons learned into our organizational practices. We might not be as fortunate enough as SD1 to catch the mistake before it has dire consequences.

[9] Many references exist for this claim, possibly going all the way back to the Ware Report [47]. McGraw [48] has a more modern version of the claim, and it is still 15 years old.

Finally, the release process alone may be incapable of verifying compliance in some circumstances. For example, new efforts to define concepts like "fairness" in algorithms [50] 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.

B. Compliance Checks Alleviate Developer Concerns

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, 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 to take this as a personal responsibility. The ACM's Code of Ethics [51] 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 personal 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, as requirements engineers, must account for the organizational and cultural environment in addition to our own personal ethical responsibilities. 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 organizations and the software industry as a whole.

C. Rewarding Compliance in the Marketplace

Our participants identified a clear market incentive for organizations to achieve demonstrable compliance. **Lesson 5:** We, as requirements engineers, should pitch 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 isn’t only beneficial in that it makes compliance more obvious to customers. Recall an earlier comment from an participant claiming that pushback for formal regulation is limited when 30–40% of the industry is already obviously compliant. Furthermore, transparency makes enforcement easier. In the E.U., GDPR’s low enforcement rates have been criticized [52]. National data protection authorities are slow to take action in part because regulators don’t want to accidentally rule so broadly that their ruling would kill Internet commerce.

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 reflects 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 $14.7B in the U.S. alone to settle their “dieselgate” scandal where they directed engineers to build a defeat device to bypass mandated emissions testing [53]. How much easier would it be to build a defeat device in a software system?

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, 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[10] 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?’”

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 [54]. 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 [55]. 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 trade craft [56]. Someone building a tree house in their backyard does not need the same level of regulatory scrutiny as someone building

[10] Common Vulnerabilities and Exposures number. This is a means of uniquely identifying security concerns.

a twelve story apartment complex. Analogous situations in software are not easily differentiated. A 17-line third party library made available on NPM[11] may have been perfectly acceptable for personal projects but it was incorporated into hundreds of professional projects, ultimately "breaking the Internet" when it was later removed [57]. This is to say nothing of actively malicious packages, which are far more problematic [58]. Both could be addressed more effectively if compliance processes required an examination of third party libraries used.

VI. Limitations

All research is limited, and this study is no exception. One limitation is our population size. Fifteen software practitioners, managers, and directors cannot speak authoritatively for the entire industry. Compared to the number of subjects interviewed in similar interview studies from our related work, we are roughly in the middle (i.e., 12, 14, 15—this study, 17, 24). In addition, we scrupulously followed Straussian grounded theory to conclusion. Our coding scheme resulted in clearly applicable codes, and additional data collection was not necessary to develop the lessons learned in this study. Regardless, our work should not be read as validating or defining practices throughout the entire industry (cf., external validity).

We are only providing a limited set of perspectives. Most of our participants were either engineers or product managers. Therefore, we were able to get strong data regarding technical measures or organizational processes related to compliance. However, we were not able to interview a comparable number of regulators or lawyers despite our protocol design and recruitment efforts. We know that getting an hour of a lawyers time to talk about their perceptions on their employer's or their clients' efforts towards regulatory compliance is extraordinarily challenging, even for a confidential research study. But their perspective could prove crucial to better understanding the tradeoffs being made at software organizations seeking to build compliance into their software systems and processes. This may also have helped us to better understand the regulatory enforcement process.

More data collection is better, offering more perspectives on this topic while validating and triangulating the data collected herein. One way to address this limitation in the future is through a survey, which would enable us to cast a wider net and achieve broader data saturation with the analysis and findings. This interview study represents a potential starting point for such a survey seeking to get a holistic understanding of the industry.

We avoid concerns related to internal validity, which refers to causal inferences made based on data collected, by not making causal claims in this study. We discuss perceptions, not causality. We addressed concerns related to construct validity by closely following our procedures as outlined in Section III and by consulting with colleagues during the study's design and execution. We address concerns related to reliability by making our complete interview protocols available online: `https://doi.org/10.6084/m9.figshare.14842242.v1`

Finally, space limitations forced us to leave out several example quotes from the participants that supported and expanded our findings. Additional space would allow us to highlight more nuance and provide further context regarding how the industry is actively working towards regulatory and security standard compliance.

VII. Summary

In this paper, we examine how regulatory and security standard requirements are addressed in the SDLC and how these techniques and procedures are perceived by engineers, managers, and directors. 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. Also, 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.

These findings may seem counterintuitive 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, absent 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 that 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 included in the SDLC.

[11] `https://www.npmjs.com/`

Acknowledgment

The authors would like to thank Dr. Carolyn Seaman and Dr. Sreedevi Sampath for their feedback and comments during the design and analysis of this study. This work was supported in part by NSF SaTC Award #1938121.

References

[1] P. I. LLC and Globalscape. The true cost of compliance with data protection regulations. [Online]. Available: https://www.globalscape.com/resources/whitepapers/data-protection-regulations-study

[2] Z. Whittaker. Ex-NSA hacker drops new zero-day doom for Zoom. [Online]. Available: https://techcrunch.com/2020/04/01/zoom-doom/

[3] G. Fleishman. Every zoom security and privacy flaw so far, and what you can do to protect yourself. [Online]. Available: https://tidbits.com/2020/04/03/every-zoom-security-and-privacy-flaw-so-far-and-what-you-can-do-to-protect-yourself/

[4] L. H. Newman. So wait, how encrypted are zoom meetings really? [Online]. Available: https://www.wired.com/story/zoom-security-encryption/

[5] E. S. Yuan. A message to our users. [Online]. Available: https://blog.zoom.us/a-message-to-our-users/

[6] Department of Health and Human Services. Telehealth: Delivering Care Safely During COVID-19. [Online]. Available: https://www.hhs.gov/coronavirus/telehealth/index.html

[7] J. E. Dunn. Apple quietly removes zoom's hidden web server from macs. [Online]. Available: https://nakedsecurity.sophos.com/2019/07/15/apple-quietly-removes-zooms-hidden-web-server-from-macs/

[8] D. Bohn. Serious zoom security flaw could let websites hijack mac cameras. [Online]. Available: https://www.theverge.com/2019/7/8/20687014/zoom-security-flaw-video-conference-websites-hijack-mac-cameras

[9] M. Iqbal. Zoom revenue and usage statistics (2021). [Online]. Available: https://www.businessofapps.com/data/zoom-statistics/

[10] H. Taneja. The era of "move fast and break things" is over. [Online]. Available: https://hbr.org/2019/01/the-era-of-move-fast-and-break-things-is-over

[11] D. Parzych. (2020) The fallacy of move fast and break things. [Online]. Available: https://devops.com/the-fallacy-of-move-fast-and-break-things/

[12] E. S. Yuan. (2020) Ceo report: 90 days done, what's next for zoom. [Online]. Available: https://blog.zoom.us/ceo-report-90-days-done-whats-next-for-zoom/

[13] E. Kempe and A. K. Massey, "Regulatory and security standard compliance throughout the software development lifecycle," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021.

[14] W. Hartzog, "The Inadequate, Invaluable Fair Information Practices," *Maryland Law Review*, vol. 76, no. 952, 2017.

[15] D. A. Fou, "Unintended Consequences Of Privacy Regulations? Reduction Of Ad Fraud?" https://www.forbes.com/sites/augustinefou/2020/09/23/unintended-consequences-of-privacy-regulations-reduction-of-ad-fraud/, Sep. 2020.

[16] N. Martin, C. Matt, C. Niebel, and K. Blind, "How Data Protection Regulation Affects Startup Innovation," *Information Systems Frontiers*, vol. 21, no. 6, pp. 1307–1324, Dec. 2019.

[17] W. H. Ware, "Records, Computers and the Rights of Citizens," https://www.rand.org/pubs/papers/P5077.html, 1973.

[18] R. Ross, P. Viscuso, G. Guissanie, K. Dempsey, and M. Riddle, "SP800-171: Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations," *NIST*, 2017.

[19] K. A. Bamberger and D. K. Mulligan, *Privacy on the Ground: Driving Corporate Behavior in the United States and Europe*, 1st ed. Cambridge, Massachusetts: The MIT Press, Oct. 2015.

[20] L. DeNardis, *The Internet in Everything: Freedom and Security in a World with No Off Switch*. New Haven, CT: Yale University Press, Jan. 2020.

[21] P. N. Otto and A. I. Antón, "Addressing Legal Requirements in Requirements Engineering," in *15th IEEE International Requirements Engineering Conference*, 2007, pp. 5–14.

[22] M. Hashmi, G. Governatori, H.-P. Lam, and M. T. Wynn, "Are we done with business process compliance: state of the art and challenges ahead," *Knowledge and Information Systems*, vol. 57, no. 1, pp. 79–133, 2018.

[23] S. Ghanavati, D. Amyot, and L. Peyton, "A systematic review of goal-oriented requirements management frameworks for business process compliance," in *2011 Fourth International Workshop on Requirements Engineering and Law*. IEEE, 2011, pp. 25–34.

[24] S. Sackmann, S. Kuehnel, and T. 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*. Springer, 2018, pp. 409–425.

[25] A. Shamsaei, D. Amyot, and A. Pourshahid, "A systematic review of compliance measurement based on goals and indicators," in *International Conference on Advanced Information Systems Engineering*. Springer, 2011, pp. 228–237.

[26] O. Akhigbe, D. Amyot, and G. Richards, "A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance," *Requirements Engineering*, vol. 24, no. 4, pp. 459–481, 2019.

[27] T. D. Breaux and D. G. Gordon, "Regulatory Requirements Traceability and Analysis Using Semi-formal Specifications," in *Requirements Engineering: Foundation for Software Quality*, J. Doerr and A. L. Opdahl, Eds. Berlin, Heidelberg: Springer, 2013, pp. 141–157.

[28] J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory codes to product specific requirements," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, May 2010, pp. 155–164.

[29] O. Akhigbe, D. Amyot, and G. Richards, "A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance," *Requirements Engineering*, vol. 24, no. 4, pp. 459–481, Dec. 2019.

[30] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *2014 IEEE 22nd international requirements engineering conference (RE)*. IEEE, 2014, pp. 73–82.

[31] S. Ghanavati, D. Amyot, and A. Rifaut, "Legal goal-oriented requirement language (legal grl) for modeling regulations," in *Proceedings of the 6th international workshop on modeling in software engineering*, 2014, pp. 1–6.

[32] Y. Negishi, S. Hayashi, and M. Saeki, "Establishing Regulatory Compliance in Goal-Oriented Requirements Analysis," in *2017 IEEE 19th Conference on Business Informatics (CBI)*, Jul. 2017, pp. 434–443.

[33] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: Supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, vol. 20, no. 1, pp. 1–22, Mar. 2015.

[34] M. Usman, M. Felderer, M. Unterkalmsteiner, E. Klotins, D. Mendez, and E. Alegroth, "Compliance requirements in large-scale software development: An industrial case study," in *International Conference on Product-Focused Software Process Improvement*. Springer, 2020, pp. 385–401.

[35] N. S. Abdullah, S. Sadiq, and M. Indulska, "Emerging challenges in information systems research for regulatory compliance management," in *International Conference on Advanced Information Systems Engineering*. Springer, 2010, pp. 251–265.

[36] M. Rosado de Souza, R. Haines, M. Vigo, and C. Jay, "What makes research software sustainable? an interview study with research software engineers," in *12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2019, pp. 135–138.

[37] J. O. Johanssen, A. Kleebaum, B. Paech, and B. Bruegge, "Practitioners' eye on continuous software engineering: An interview study," in *Proceedings of the 2018 International Conference on Software and System Process*. New York, NY, USA: Association for Computing Machinery, 2018, p. 41–50.

[38] 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)*, 2005, pp. 10 pp.–23.

[39] E. Lim, N. Taksande, and C. Seaman, "A Balancing Act: What Software Practitioners Have to Say about Technical Debt," *IEEE Software*, vol. 29, no. 6, pp. 22–27, Nov. 2012.

[40] K. A. Bamberger and D. K. Mulligan, "Privacy on the Books and on the Ground," *Stanford Law Review*, vol. 63, pp. 247–316, 2011.

[41] J. M. Haney and W. 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)*, 2018, pp. 411–425.

[42] J. Haney and W. Lutters, "Skills and Characteristics of Successful Cybersecurity Advocates," in *Thirteenth Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[43] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research: a critical review and guidelines," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 120–131.

[44] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, fourth edition ed. Los Angeles: SAGE Publications, Inc, Dec. 2014.

[45] S. B. Merriam and E. J. Tisdell, *Qualitative Research: A Guide to Design and Implementation*, 4th ed. San Francisco, CA: John Wiley & Sons, Aug. 2015.

[46] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 557–572, Jul. 1999.

[47] Defense Science Board Task Force on Computer Security, "Security Controls for Computer Systems," RAND, Tech. Rep. R-609-PR, 1972.

[48] G. McGraw, *Software Security: Building Security In*, 1st ed. Upper Saddle River, NJ: Addison-Wesley Professional, Jan. 2006.

[49] P. N. Otto, A. I. Antón, and D. L. Baumer, "The ChoicePoint Dilemma: How Data Brokers Should Handle the Privacy of Personal Information," *Security & Privacy Magazine, IEEE*, vol. 5, no. 5, pp. 15–23, Oct. 2007.

[50] S. Mitchell, E. Potash, S. Barocas, A. D'Amour, and K. Lum, "Algorithmic Fairness: Choices, Assumptions, and Definitions," *Annual Review of Statistics and Its Application*, vol. 8, no. 1, p. null, 2021.

[51] Association of Computing Machinery. ACM Code of Ethics and Professional Conduct. [Online]. Available: https://www.acm.org/code-of-ethics

[52] N. Lomas. (2020) GDPR's two-year review flags lack of 'vigorous' enforcement. [Online]. Available: https://techcrunch.com/2020/06/24/gdprs-two-year-review-flags-lack-of-vigorous-enforcement/

[53] Federal Trade Commission. (2016) Volkswagen to spend up to $14.7 billion to settle allegations of cheating emissions tests and deceiving customers on 2.0 liter diesel vehicles. [Online]. Available: https://www.ftc.gov/news-events/press-releases/2016/06/volkswagen-spend-147-billion-settle-allegations-cheating

[54] R. G. Kula, D. M. German, A. Ouni, T. Ishio, and K. Inoue, "Do developers update their library dependencies?" *Empirical Software Engineering*, vol. 23, no. 1, pp. 384–417, Feb. 2018.

[55] M. Zimmermann, C.-A. Staicu, C. Tenny, and M. Pradel, "Small World with High Risks: A Study of Security Threats in the npm Ecosystem," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 995–1010.

[56] W. G. Vincenti, *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*, new edition ed. Baltimore: Johns Hopkins University Press, Feb. 1993.

[57] S. Gallagher. Rage-quit: Coder unpublished 17 lines of javascript and "broke the internet". [Online]. Available: https://arstechnica.com/information-technology/2016/03/rage-quit-coder-unpublished-17-lines-of-javascript-and-broke-the-internet/

[58] T. Claburn. Malicious backdoored npm package masqueraded as twilio library for three days until it was turfed out. [Online]. Available: https://www.theregister.com/2020/11/03/malicious_npm_package_masquerading_as/