Regulatory and Security Standard Compliance Throughout the Software Development Lifecycle

Evelyn Kempe
University of Maryland, Baltimore County
ekempe1@umbc.edu

Aaron K. Massey
University of Maryland, Baltimore County
akmassey@umbc.edu

Abstract

Our systematic literature review aims to survey research on regulatory and security standard requirements as addressed throughout the Software Development Lifecycle. Also, to characterize current research concerns and identify specific remaining challenges to address regulatory and security standard requirements throughout the SDLC. To this end, we conducted a systematic literature review (SLR) of conference proceedings and academic journals motivated by five areas of concern:

1. SDLC & Regulatory Requirement 2. Risk Assessment and Compliance requirements 3. Technical Debt 4. Decision Making Process throughout the SDLC 5. Metric and Measurements of found Software Vulnerability.

The initial search produced 100 papers, and our review process narrowed this total to 20 articles to address our three research questions. Our findings suggest that academic software engineering research directly connecting regulatory and security standard requirements to later stages of the SDLC is rare despite the importance of compliance for ensuring societally acceptable engineering.

1. Introduction

Software Engineers that create and maintain the software in regulated industries must demonstrate regulatory and security standard compliance. Unfortunately, regulations and security standards often contain ambiguities, conflicts, and multiple valid interpretations, making demonstrable compliance challenging. Industry is interested in managing regulatory compliance, as evidenced by the fact that 29% of new cybersecurity funding addresses new or changing regulatory requirements [1]. We, as researchers, need to take a step back and assess where we stand in assisting the software engineering community in becoming more compliant. How much academic research is there in regulatory and security standard compliance?

Our systematic literature review (SLR) aims to survey research on regulatory and security standard requirements as addressed throughout the Software Development Lifecycle. Also, we seek to characterize current research concerns and identify specific remaining challenges to address regulatory and security standard compliance throughout the SDLC. This SLR consists of peer-reviewed articles published over the past 20 years, focusing on five areas of concern: F1: SDLC & Regulatory Requirement, F2: Risk Assessment & Compliance requirements, F3: Technical Debt, F4: Decision-Making Process throughout the SDLC process, and F5: Metric and Measurements of Software Vulnerabilities. These concerns motivate our research questions:

RQ1: What does the academic literature say about Regulatory Compliance (RC) throughout the Software Development Lifecycle (SDLC)?

RQ2: What does the academic literature say about Security Standards compliance (SSC) throughout the Software Development Lifecycle (SDLC)?

RQ3: What does the academic literature say about the cost and benefits of Regulatory and Security Standard compliance?

URI: https://hdl.handle.net/10125/70861
978-0-9981331-4-0

Our initial search produced 100 papers for manual review (See Section 3.2). Through our search and selection process (See Figure 1), we then narrowed our selections to 20 papers that addressed our three research questions.

Our findings suggest that academic software engineering research directly connecting regulatory requirements to later stages of the SDLC is rare despite the importance of regulatory compliance for ensuring societally acceptable engineering. Security standard compliance produced similar findings to regulatory requirements with the later stages. The literature suggests that the SE research community does not consider regulatory or security standards compliance to be a primary, motivating research concern during the latter stages of the SDLC. This differs from industry concerns and spending practices. We describe these findings and limitations in further detail in Section 4 and Section 5, respectively. Section 6 summarizes this work and makes recommendations for future research.

2. Related Work

A few other researchers have studied the SE literature on topics related to regulatory and security standard compliance or propose models to trace regulatory compliance throughout the SDLC. For example, Otto and Antón examine research efforts from 1957 to 2007 "in modeling and using legal text for system development" [2]. The authors' section on "The Nature of Regulation" examines characteristics of regulations, citing the challenges and other applicable factors that we discuss in Section 4. Other secondary studies do speak to our concerns, which we will now discuss.

Meidan *et al.* reviewed papers on software quality measurement methodologies, providing insight into evaluation and testing research [3]. However, this secondary study did not find any research on regulatory compliance and very little on security compliance, focusing overall on Software Development Process improvement.

Romanosky *et al.* conducted empirical research on privacy data breach litigations involving information and communication technologies [4]. The authors investigated "the characteristics of data breach litigation and the outcomes of these cases" [4]. They also provided their coding, which inspires our examination of costs and benefits from case law as applicable to our third research question.

Spanos and Angelis produced another study that we found useful on the "economic consequences of security incidents" [5]. They focused on the aftereffects of decisions, not regulatory and security compliance during software development. Their work motivates researchers seeking to develop software development techniques to address compliance problems.

3. Method and Data Collection

Systematic literature reviews (SLRs) examine the body of literature on a particular subject to understand the state of the subject's research, and identify gaps within a body of research [6]. By identifying gaps through SLRs, researchers can identify and promote needed research in a particular field. We use Kitchenham and Brereton's model [6] for this systemic literature review. In this section, we describe the methods for reviewing articles for this SLR and addressing our research questions.

3.1. Objective and Research Questions

We used a Goal-Question-Metric (GQM) [7] approach to survey research on regulatory and security requirements as addressed throughout the SDLC, characterize current research concerns, and identify remaining challenges from the academic viewpoint. Our three research questions based on this goal are as follows:

RQ1: What does the academic literature say about Regulatory Compliance (RC) throughout the Software Development Lifecycle (SDLC)?

RQ2: What does the academic literature say about Security Standards compliance (SSC) throughout the Software Development Lifecycle (SDLC)?

RQ3: What does the academic literature say about the cost and benefits of Regulatory and Security Standard compliance?

RQ1 seeks practical insight into published academic views on regulatory compliance in the SDLC, including discussion of regulatory compliance, defining the characteristics of regulation, tracing regulatory requirements, and addressing compliance levels. RQ2 extends RQ1 into industry security standards because they are often legally binding through regulatory requirements to meet a baseline standard of care. Security standards are also commonly referenced explicitly as exemplars in regulatory compliance scenarios. RQ3 examines decision models of compliance with regulatory and security standards using a cost and benefits framework. Our approach to RQ3 focused on tradeoffs framed using technical debt or risk management to understand the SE practitioner's management of regulatory and security standard compliance throughout the SLDC.

3.2. Search and Selection Process

Our search and selection process consists of three passes (shown in Figure 1). With each pass, we narrow our selection of articles and map our final selection from our areas of concern to one or more research questions.

Stage 1: Search and Selection – 1st Pass.

Our search queries centered around our five areas of concern, using the following search string:

S1: Software Development AND Regulatory Compliance OR Industry Security Standard
S2: Software Development Lifecycle AND Regulatory Compliance OR Industry Security Standard
S3: Software Development AND Decision Making OR Decision Models
S4: Software Development AND Technical Debt

We used these External Scholarly Search Engines: IEEE Explore, ACM Digital Library, Springer Nature, and Science Direct to search for articles. Our search query applied two auto-search filters: peer-reviewed publications (Conference or Journal) and publication date (1999–2019) incorporating two exclusion criteria (See Figure 1, Process 1.0). We manually excluded papers not written in English and those with less than ten "cited by" references with exceptions made for publications after 2015.

Table 1. Author and Keywords Snowballing Search Results

ID	Focus Area	Author	Keywords
F1:	SLDC	3	3
F2:	RA & Comp. Req.	1	12
F3:	Technical Debt	11	0
F4:	Decision Making	1	5
F5:	Measure and Metrics	0	10

After our initial screening process (Process 1.0), we reviewed the title and metadata of each candidate article against our inclusion criteria (See Process 2.0). We were looking for articles on the following topics: Requirements Engineering and Software Development (21 found), Technical Debt (23 found) & Decision-making within the SDLC (12 found), Risk Assessment or Regulatory Compliance (22 found), or Metrics/Measurements assessments on Software Vulnerabilities (22 found).

Once an article passed the "Filter for Inclusion" Process, we downloaded the full-text article and organized them under our Areas of Concern. Some articles overlapped multiple areas of concern, so we reviewed their abstracts and organized them under their most relevant topic. We also deleted any duplicate articles from our downloaded selection. Prior to the second pass review, the total was 100 articles.

We used reverse snowballing to narrow our selections rather than increase our references (See Figure 1, Process 3.0) [6]. The reverse snowballing process started with a compiled list of authors[1] based on some pre-reviewed articles related to Regulatory Compliance within Software Development. We applied this list to our 100 papers, which narrowed our papers from 100 to 16. Using the 16 articles as a reference, we compiled a list of common tags

[1] For completeness, the list is as follows: T.D. Breaux, M.J. May, C.A. Gunter, I. Lee, P.N. Otto, A.I.Anton, J.C. Maxwell, P. Swire, J. Hayward, S. Ghanavati, R.L. Rutledge, J. Camp, B. Boehm, G. Boella, C. Seaman, Y. Guo, S. Spiekermann, N. Ramasubbu, C.F. Kemerer, F. Shull, E. Shihab, G. Boello, P. Ohm, Z. Zazworka, N. Zeni, L. Mich, and T. Valentien

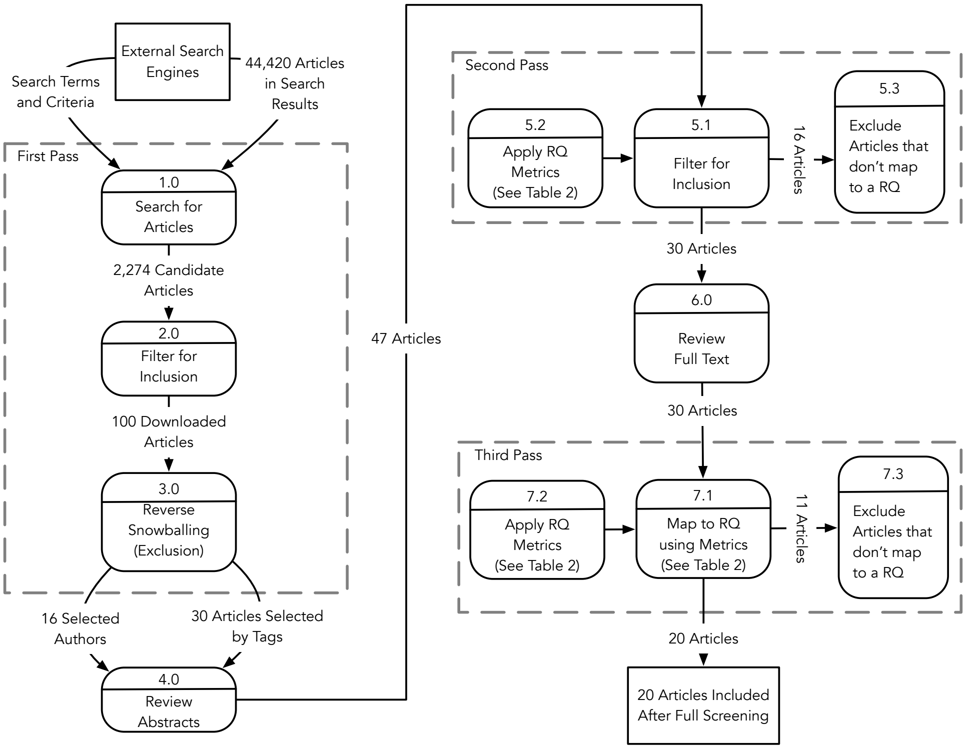
External Search Engines
Search Terms and Criteria
44,420 Articles in Search Results
First Pass
1.0
Search for Articles
2,274 Candidate Articles
2.0
Filter for Inclusion
100 Downloaded Articles
3.0
Reverse Snowballing (Exclusion)
16 Selected Authors
30 Articles Selected by Tags
4.0
Review Abstracts
47 Articles
Second Pass
5.2
Apply RQ Metrics (See Table 2)
5.1
Filter for Inclusion
16 Articles
5.3
Exclude Articles that don't map to a RQ
30 Articles
6.0
Review Full Text
30 Articles
Third Pass
7.2
Apply RQ Metrics (See Table 2)
7.1
Map to RQ using Metrics (See Table 2)
11 Articles
7.3
Exclude Articles that don't map to a RQ
20 Articles
20 Articles Included After Full Screening

Figure 1. Data Flow Diagram for Search Process

(23 total[2]) to review the other 84 articles within our 100 downloaded articles. From the 84 articles, we selected 30 of those articles, for a total of 46 articles for Abstract Review (Process 3.0) and completing our first selection pass. Table 1 maps the number of articles to their respective areas of concern.

Stage 2: Review Abstract – 2nd Pass.

The "Review Abstract" process began with finding articles that addressed our three research questions. First, we want articles discussing a clearly regulated domains. For example, any industry where companies must comply with rigid regulatory requirements by considering fundamental limitations within the Software Development Lifecycle (SDLC) process. Second, we did not want articles that did not reference or discuss regulatory compliance or security standard compliance. Third, we wanted studies that looked at the costs and benefits of compliance or security standard implementation to support our third research questions, focusing on rationale for technical debt accrual or risk management decisions balancing the risk of vulnerability to short term benefits or opportunities. Lastly, we did not want to look at pattern analysis or identification of vulnerabilities for software exploitation because these topics are outside the purpose of this SLR.

Inclusion criteria for RQ1

IC1: Views on RC throughout the SDLC (Challenges within Industry)
IC2: Defining the characteristics of Regulation
IC3: Traceability of regulatory requirements

[2] For completeness, our tag list is as follows: security, software development, software development standards, security standards, requirements/criteria, requirements engineering, software lifecycle/SDLC, compliance, legal compliance, legal requirements, regulatory compliance, regulation/law, technical debt, technical debt management, system development, auditing, governance, software engineering, risk/costs, risk management, decision making/decision models, cost-benefit/cost-risk analysis, traceability/Software traceability

IC4: Levels of Regulation Compliance

Inclusion criteria for RQ2

IC1: Views on industry security standard compliance (i.e., challenges within industry)

IC2: Defining characteristics of security standards

IC3: Risk management process as a method of tracing security compliance

IC4: Technical Debt (TD) management as a method of tracing security compliance

Inclusion criteria for RQ3

IC1: Quantify the risk of non-compliance to regulatory or security standards

IC2: Decision Models if TD

IC3: Discussing TD Management

IC4: Views stakeholder roles on cost and benefits of technical debt, RC, & security std

Based on our inclusion criteria, we reviewed the remaining 46 articles (See Figure 1, Process 4.0), mapping them to the research question the article addressed (Process 5.1). The second pass excluded 16 articles, leaving 30 articles for the full-text review.

Stage 3: Full-Text Review – 3rd pass.

In this stage, we conduct a full-text review to determine whether an article addresses our research questions using the Inclusion Criteria outlined in the previous stage. For example, Otto and Antón's article provided specific descriptions on the characteristic of regulations *"that make them both useful and difficult to apply to design methodologies"*. They also surveyed efforts to handle legal texts for regulatory compliance and presented academic viewpoints such as regulation ambiguity as an issue for compliance [2]. We included this paper because it realizes three of our four Inclusion Criteria for RQ1.

The article by Izurieta *et al.* is another example included in this survey. This paper quantifies "Technical Debt associated with security issues" using Common Weakness Enumeration (CWE) or Common Weakness Scoring System (CWSS).[3] Quantifying technical debt as it relates to a security vulnerability can prevent foreseeable exploits [8]. We coded this article to RQ2 and RQ3 because CWSS is a methodology for tracing security compliance and Technical Debt and Risk Management.

After completing the full-text review and article mapping, we excluded 11 articles did not address a research question (See Figure 1, Process 7.3), but added a reference (i.e., [9]) based on an additional review of the technical debt sources which was mapped to RQ3. A total of 20 papers addressed our three research questions out of the 100 initially selected papers and the one (i.e., [9]) added in the third pass. Table 2 shows our final selection of included articles and the research questions they address.

Table 2. Included Articles

Citation	RQ1	RQ2	RQ3
Garg and Camp [10]	X	✓	✓
Otto and Antón [2]	✓	X	X
Izurieta *et al.* [8]	X	✓	✓
Ramasubbu *et al.* [11]	X	✓	✓
Laukkarinen *et al.* [12]	✓	✓	X
Parsons *et al.* [13]	✓	✓	✓
Lim *et al.* [14]	X	✓	✓
Maxwell *et al.* [15]	✓	X	✓
Beach *et al.* [16]	✓	X	X
Cristina-Clara *et al.* [17]	✓	✓	X
Velasco *et al.* [18]	✓	X	X
Parent and Reich [19]	X	✓	✓
Maxwell *et al.* [20]	✓	X	✓
Ramasubbu and Kemerer [21]	X	X	✓
Regan *et al.* [22]	✓	✓	X
Trektere *et al.* [23]	✓	✓	X
Falessi *et al.* [24]	X	✓	✓
Maxwell and Antón [25]	✓	X	✓
Seaman *et al.* [9]	X	X	✓
Ramasubbu and Kemerer [26]	X	✓	✓

4. Results

Our "Search and Selection" Process, identified 20 articles as addressing our three research questions on regulatory and security standard compliance (See Table 2). In this section, we will discuss the papers included in the study, provide examples of selected articles, how we map them to a research question, and our overall findings for each research question.

[3] CWE or CWSS is a standard for weakness identification, prioritization, mitigation, and prevention efforts in risk management [8].

4.1. RQ1: Regulatory Compliance

We found 11 articles that discussed regulatory compliance against our inclusion criteria for RQ1. We included articles focusing on RC [2], [12], [15]–[18], [20], [25] and articles that highlight RC throughout the SLDC [13], [18], [22], [23]. However, only finding 11 articles points to a concern regarding the RC research body of work: How can practical insights into RC be examined if little published research is out compared to other research areas? Most of the RC articles included are requirements engineering articles (i.e., eight out of 11). Traditional software engineering (SE) techniques (i.e., Waterfall) only focus on regulatory compliance as part of the SDLC's requirements phase. We assumed with other development techniques (DevOps, Agile, Continuous Integration), we would find more RC research within the SDLC's design, testing, deployment, and maintenance phase. It seems that the SE academic community treats regulatory compliance as a concern exclusive to requirements engineering and does not push more for accountability throughout the SDLC process. That said, we found some articles that present RC research addressing the later stages of the SLDC. The next couple of paragraphs discuss examples from the included papers.

Velasco *et al.* presented a web compliance framework for developers to promote their framework for Web Engineering as a form of traceability and verification to regulation (RQ1, IC3) [18]. They aim to support quality assurance and overcome "deficiencies of existing evaluation tools" to ensure the successful implementation of Rich Internet Application that is web compliant. We found their work related because they defined web compliance engineering by describing its challenges at a multinational level (RQ1, IC1), as seen in the excerpt below:

> *"Web Compliance Engineering as the application of quality assurance testing and management processes and principles that ensure conformance of Web applications to standards, policy environments, and other ad-hoc quality criteria... Web compliance is becoming overwhelmingly complex, especially for multinational organizations, which must comply with many local policy environments" [18]*

Regan *et al.* outlined the challenges of traceability requirements within medical device standards, which must comply with FDA regulations and may also require compliance with the Health Insurance Portability and Accountability Act (HIPAA). They highlight the definition and identification of requirements for traceability through each phase of the software development lifecycle (RQ1, IC1–3):

> *"Software traceability is central to medical device software development And essential for regulatory compliance... However, the requirement for traceability through the software development life-cycle is not as obvious in the regulations" [22]*

They also mention ambiguity in the context of regulation (RQ1, IC1), previously seen in other articles (e.g., Otto and Antón [2]). They also propose a model for traceability to show due diligence for regulatory compliance (RQ1, IC3).

We assumed some of the literature would examine processes, tools, and techniques for tracking partial compliance and measuring progress towards complete compliance. However, none of the articles included for a full text review discussed these topics as outlined by our fourth inclusion criteria for RQ1. An organization's ability to account for current levels of regulatory compliance in their software development process speaks to their ability to understand the challenge of regulatory compliance. The old adage from Lord Kelvin is relevant: "If you cannot measure it, then you cannot improve it." Software processes that can account for and measure regulatory compliance, whether in the form of equally-valued items on a checklist or through some risk analysis of costs and benefits, would be more mature than those that cannot. The fact that this SLR did not find academic work relevant to this effort is concerning, particularly given the increasing complexity of regulations that apply to software systems.

Collectively, the articles we found summarize the SE community's academic views on regulatory compliance. First, the SE community relies on regulation to provide general guidance. Then, the SE community develops strategies and policies to track and show compliance as expressed in software requirements within their SDLC process. The SE academic community clearly discusses the challenges of complying with regulation. Foremost among these challenges is that regulatory guidance for engineering is general, ambiguous, and has not addressed gaps in a fast-changing field.

4.2. RQ2: Security Std. Compliance

To address RQ2, we found 12 out of the using the inclusion outlined previously. Five of these 12 overlap with the articles meeting the inclusion criteria for RQ1, indicating that these authors are explicitly addressing both regulatory compliance (RC) and security standard compliance (SSC). This aligns with the view that security standards are often used by regulators and policy makers when crafting regulations and regulatory guidelines. PCI DSS compliance is a good example. HIPAA also explicitly refers to industry standards. Trektere *et al.* points out *"the challenge that MMA [Mobile Medical app] software development companies face when they want to market an app is the adherence to the large number of regulatory requirements specified in various international standards"* [23] However, we separated this research question because security standards do not necessarily address RC. They often exist for reasons other than regulatory compliance. As Laukkarinen *et al.* put it:

> *"Numerous industrial fields – including for instance automotive, space, and medical devices – require reliability, visibility, and traceability of the software project to ensure high quality, safety, and trustworthiness of the software."* [12]

The cost of cyberattacks cost firms on average US$13 million, but the actual costs to any given firm may vary significantly [27]. Firms are unsure how much to spend to mitigate this threat and where to direct it to get the best value for their security budget [27]. We wanted to see how SE academia was addressing this critical need. To that end, our inclusion criteria for RQ2 explicitly focus on security standard compliance separately from regulatory compliance. (Also, RQ3 emphasized management by also looking at Risk and Technical Debt management when it comes to security standard compliance and traceability.) We did this because we wanted a better understanding of decision-making models and management of SSC, exemplified by work from both Garg and Camp and Lim *et al.*

Garg and Camp examine how heuristics and biases can better support decisions in the design of security technologies using a behavioral approach and framing [10]. This paper was included based on IC 3. They point out that *"security investment is a definite expense of time and money (i.e., a loss), while the risk of not investing in security is a probable loss"* [10]. They explored leadership's or management's decision-making process within security compliance by considering their view point of the probable risk of vulnerability exploitation versus lost opportunity gains due to delays to market. It also accounted for quick and dirty heuristics and biases that can affect rational decision making in security design and development.

Lim *et al.*, we included based in IC 4. They conducted an interview study to characterize technical debt by asking software practitioners to define it and provide contextual examples of it [14]. This example of how practitioners view and manage security gives insight into the balancing of opportunity and risk within software development. For example, consider how participants dealt with other stakeholders in the development process [14]:

> *"Many participants also found that management didn't recognize the value in addressing technical debt unless doing so provided management with a tangible reward or the customer was paying for it. Similarly, customers weren't easily convinced to allow the development team time to repay its technical debt unless they could derive business value from doing so."* [14].

Addressing RQ2 was more difficult than RQ1 and RQ3. The literature on security standard compliance overlaps with the topics of regulatory compliance and risk and technical debt management. We show the overlap in Table 2 (e.g., in the table, five of the 12 RQ2 articles also mapped to RQ1, and eight of the 12 mapped to RQ3). This overlap may be a limitation related to the framing of RQ2. However, it may also be that academics view security standard compliance as a mixture of regulatory compliance and risk or technical debt management.

4.3. RQ3: Costs and Benefits

Thirteen of the 20 articles that received a full text review addressed RQ3 based on our inclusion criteria. We separated this question from the other two to take a closer look at decision-models and management from a cost versus benefit standpoint. Decisions regarding technical implementation of regulatory compliance objectives and security features are necessarily related to the costs of development for the system as a whole. The balancing of opportunities against risks bears a critical relationship to compliance. Understanding this relationship is what motivated us to ask RQ3 separate from RQ1 and RQ2.

As with RQ1 and RQ2, we mapped the related articles to RQ3 based on the inclusion criteria (See Table 2). We looked for quantitative and qualitative factors in decision-making that affect how the SE stakeholders manage technical debt or balance opportunity and risk in software development. Quantitative examples include the cost of non-compliance in the form of regulatory or government fines, court costs, or drop in stock prices and qualitative being articles that considered non-numerical factors such as customer satisfaction and software maintainability. For example, Maxwell and Antón sought to *"present a production rule framework that software engineers can [use] to specific compliance requirements for software"*. They motivated this work in part with the example of the data broker, ChoicePoint, which had a data breach in 2006 that cost the company 25 million US dollars to show how costly non-compliance can be to a software development company and reasoning to adopting a systematic framework [25].

Ramasubbu and Kemerer's 2014 case study serves as a qualitative example [21]. They examined the cost and benefits of 69 customers adoption of added functionality of a commercial enterprise software package over the software package's ten-year lifespan [21]. Early adoption of software features took on more technical debt with higher customer satisfaction ratings but had lower long-term software maintainability. Late adoption had the reverse effects:

> *Similarly, the late adopter behavior of avoiding immature ("beta") features and custom modification of features can be interpreted as the costs of avoiding technical debt (or perhaps as accumulating a "technical credit"). The cost of avoiding technical debt manifests itself in poor customer satisfaction scores before takeoff. However, the strategy of avoiding technical-debt pays off in the longer run in terms of significantly higher software quality of the package throughout its lifespan and further endows a customer with the ability to continue adding features at a faster pace during the later stage of the package's life-cycle."* [21]

The SE communities views on decision factors and design throughout the SDLC have implications for regulatory compliance. Most articles identified in this SLR do not quantify this relationship or directly account for it in their implementation decisions. The articles identified also do not examine the severity of the risk of non-compliance as part of standard development practices or whether SE stakeholders can effectively assess and manage the risk of non-compliance. Some of the articles assessed and used in this SLR describe a framework for managing these tradeoffs. However, these frameworks focus more on tracing previous decisions regarding regulatory compliance requirements than they do in reassessing compliance in a changing domain or evaluating previous decisions (i.e., technical debt) for regulatory implications.

5. Limitations

One of the main limitations of this study was the focus on published SE academic views rather than SE industry views. Not including SE industry views has the potential for bias within this study. For example, they may cite academic research that we would find relevant. That said, we chose to focus on academic literature for two reasons. First, academic publications are easier to search and identify given academic publication databases. Second, private industry does not commonly discuss regulatory or security standard compliance because discussing their weaknesses in regulatory compliance may be a competitive disadvantage.

We also limited our survey to a 20 year publication timeframe. We wanted to review the literature holistically, rather than focus on the most recent research trajectory, to understand where the research in regulatory and security standard compliance has been. We felt that anything beyond 20 years would not be relevant to the research field today. However, these assumptions to limit the scope of this work.

A methodological limitation is selecting candidate articles as depicted in Figure 1 described in the Section 3.2. Our first pass systematically incorporated exclusion criteria that cut the candidate articles to less than 10% of the search queries produced. However, we may have excluded articles that would have been deemed relevant had we conducted forward snowballing. On the other hand, our process is more repeatable as conducted. Also, we incorporated snowballing in the latter half of the Stage 1 search and selection process to leverage as many of the benefits of snowballing as possible.

The later stages (i.e., Stage 2 and 3), put more emphasis on content and addressing the three research questions. Our final 20 article selection is the result of wanting articles that directly discuss regulatory and security standard compliance and management models. However, this may mean that our inclusion criteria were too narrow to encompass a complete picture of approaches to regulatory and security standard compliance. Articles that discuss these concerns may exist and may not be included in this study.

Finally, space limitations forced us to leave out several example quotes from the literature that would provide additional context regarding how researchers actively working in this area view regulatory and security standard compliance.

6. Summary and Future Research

The focus of this SLR is to assess the academic views of the current state of regulatory and security standard compliance throughout the SDLC. Our survey searched through 100 candidate articles, finally focusing on 20 articles that addressed our three research questions. We learned that software engineering academics view regulatory and security standard compliance primarily through requirements engineering, especially within regulated industries.

Our findings suggest that academic software engineering research directly connecting regulatory requirements and security standards to later stages of the SDLC is rare despite the industry's focus on regulatory and security standard compliance. The literature suggests that the SE community focuses on other factors during the latter stages of the SDLC. These considerations drive decision models of compliance versus non-compliance. Although the risk of non-compliance is real and motivating, the risk of not capitalizing on opportunities can justify short-term trade-offs over compliance.

None of the articles we found address the challenges of bringing a non-compliant system into compliance with regulatory or security standard requirements. Further, only finding 20 articles directly relevant to our inquiry demonstrates the limited amount of research focusing on the later stages of the SDLC and motivates further work to build methods and tools for tracking and demonstrating regulatory and security standard compliance throughout the SDLC.

Although this work focused exclusively on academic research, it would be interesting to survey industry views on regulatory and security standard compliance. Additional research on framework and decision-models could also help SE professionals manage and demonstrate compliance or progress towards compliance in their systems. Lastly, an investigation of the relationship between regulatory and security standard compliance may prove fruitful. Twenty papers are not enough to determine the extent to which regulatory compliance and security standards compliance are similar or different.

References

[1] K. Lovejoy, "EY Global Information Security Survey," Ernst and Young, Tech. Rep., 2020.

[2] P. N. Otto and A. I. Antón, "Addressing Legal Requirements in Requirements Engineering," in *15th IEEE International Requirements Engineering Conference (RE 2007)*, Oct. 2007, pp. 5–14. DOI: 10.1109/RE.2007.65.

[3] A. Meidan, J. A. Garciá-Garciá, I. Ramos, and M. J. Escalona, "Measuring software process: A systematic mapping study," *ACM Computing Surveys*, vol. 51, no. 3, 58:1–58:32, Jun. 2018. DOI: 10.1145/3186888.

[4] S. Romanosky, D. Hoffman, and A. Acquisti, "Empirical analysis of data breach litigation," *Journal of Empirical Legal Studies*, vol. 11, no. 1, pp. 74–104, 2014.

[5] G. Spanos and L. Angelis, "The impact of information security events to the stock market: A systematic literature review," *Computers & Security*, vol. 58, pp. 216–229, 2016.

[6] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Information and Software Technology*, vol. 55, no. 12, pp. 2049–2075, Dec. 2013. DOI: 10.1016/j.infsof.2013.07.010.

[7] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, "Linking software development and business strategy through measurement," *Computer*, vol. 43, no. 4, pp. 57–65, 2010.

[8] C. Izurieta, D. Rice, K. Kimball, and T. Valentien, "A position study to investigate technical debt associated with security weaknesses," in *Proceedings of the 2018 International Conference on Technical Debt*, ser. TechDebt '18, Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 138–142. DOI: 10.1145/3194164.3194167.

[9] C. Seaman and Y. Guo, "Chapter 2 - Measuring and Monitoring Technical Debt," in *Advances in Computers*, M. V. Zelkowitz, Ed., Elsevier, Jan. 2011, pp. 25–46. DOI: 10.1016/B978-0-12-385512-1.00002-5.

[10] V. Garg and J. Camp, "Heuristics and biases: Implications for security design," *IEEE Technology and Society Magazine*, vol. 32, no. 1, pp. 73–79, 2013. DOI: 10.1109/mts.2013.2241294.

[11] N. Ramasubbu, C. F. Kemerer, and C. J. Woodard, "Managing technical debt: Insights from recent empirical evidence," *IEEE Software*, vol. 32, no. 2, pp. 22–25, Mar. 2015. DOI: 10.1109/ms.2015.45.

[12] T. Laukkarinen, K. Kuusinen, and T. Mikkonen, "Devops in regulated software development: Case medical devices," IEEE, May 2017. DOI: 10.1109/icse-nier.2017.20.

[13] D. Parsons, T. Susnjak, and M. Lange, "Influences on regression testing strategies in agile software development environments," *Software Quality Journal*, vol. 22, no. 4, pp. 717–739, Oct. 2013. DOI: 10.1007/s11219-013-9225-z.

[14] 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. DOI: 10.1109/ms.2012.130.

[15] J. C. Maxwell, A. I. Antón, and P. Swire, "A legal cross-references taxonomy for identifying conflicting software requirements," IEEE, Aug. 2011. DOI: 10.1109/re.2011.6051647.

[16] T. Beach, Y. Rezgui, H. Li, and T. Kasim, "A rule-based semantic approach for automated regulatory compliance in the construction sector," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5219–5231, Jul. 2015. DOI: 10.1016/j.eswa.2015.02.029.

[17] A. M. Cristina-Clara, E. D. Canedo, and R. T. de Sousa Júnior, "A synthesis of common guidelines for regulatory compliance verification in the context of ict governance audits," *Information Polity*, vol. 23, no. 2, pp. 221–237, Jun. 2018. DOI: 10.3233/ip-170059.

[18] C. A. Velasco, D. Denev, D. Stegemann, and Y. Mohamad, "A web compliance engineering framework to support the development of accessible rich internet applications," ACM Press, 2008. DOI: 10.1145/1368044.1368054.

[19] M. Parent and B. H. Reich, "Governing information technology risk," *California Management Review*, vol. 51, no. 3, pp. 134–152, Apr. 2009. DOI: 10.2307/41166497.

[20] J. C. Maxwell, A. I. Antón, and P. Swire, "Managing changing compliance requirements by predicting regulatory evolution," IEEE, Sep. 2012. DOI: 10.1109/re.2012.6345793.

[21] N. Ramasubbu and C. F. Kemerer, "Managing technical debt in enterprise software packages," *IEEE Transactions on Software Engineering*, vol. 40, no. 8, pp. 758–772, Aug. 2014. DOI: 10.1109/tse.2014.2327027.

[22] G. Regan, F. McCaffery, K. McDaid, and D. Flood, "Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model," *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 3–9, Nov. 2013. DOI: 10.1016/j.csi.2013.07.012.

[23] K. Trektere, G. Regan, F. McCaffery, D. Flood, M. Lepmets, and G. Berry, "Mobile medical app development with a focus on traceability," *Journal of Software: Evolution and Process*, vol. 29, no. 11, e1861, Mar. 2017. DOI: 10.1002/smr.1861.

[24] D. Falessi, M. A. Shaw, F. Shull, K. Mullen, and M. S. Keymind, "Practical considerations, challenges, and requirements of tool-support for managing technical debt," IEEE, May 2013. DOI: 10.1109/mtd.2013.6608673.

[25] J. C. Maxwell and A. I. Antón, "The Production Rule Framework: Developing a Canonical Set of Software Requirements for Compliance with Law," *Proceedings of the 1st ACM International Health Informatics Symposium*, 2010.

[26] N. Ramasubbu and C. F. Kemerer, "Integrating technical debt management and software quality management processes: A normative framework and field tests," *IEEE Transactions on Software Engineering*, pp. 1–1, 2017. DOI: 10.1109/tse.2017.2774832.

[27] K. Bissell and L. Ponemon. (Mar. 2019). "The Cost of Cybercrime," [Online]. Available: https://www.accenture.com/us-en/insights/security/cost-cybercrime-study.