Software Compliance in Different Industries: A Systematic Literature Review

Mohammed Mubarkoot[1], Jörn Altmann[1]

[1]Technology Management, Economics and Policy, Seoul National University, Seoul, Korea
mubarkoot@snu.ac.kr, jorn.altmann@acm.org

Abstract. With the emergence of new software development paradigms (e.g., distributed teams and crowd-sourcing), the software supply chain became more complicated than ever. This, in turn, raises concerns in software compliance in many industries, as ensuring adherence beyond functional requirements is very critical. This paper uses a systematic literature review, to investigate the frameworks used for managing compliance of software and software services and their applications across different industries. The review also looked into industry-specific software compliance requirements. A total of 156 primary studies have been collected, of which 63 studies match the criteria indicated in the review protocol. The study develops a classification of these frameworks based on industry-specific needs, business requirements, and the context of compliance. Findings of this research help researchers and practitioners to identify important aspects of software compliance and set directions for future research and development.

Keywords: Software Compliance, Policy, Regulations, Industry Requirements, Systematic Literature Review

1 Introduction

Complex software applications evolve over time and tend to diverge from the intended or documented design models. This deviation makes the system hard to understand, modify, and maintain in the long run [1]. Nevertheless, modifications and updates of software systems are inevitable, in order to respond to changes in business requirements. Nowadays, software development happens globally across geographically distributed and autonomous teams consuming huge amounts of software components drawn from a variety of different sources [14] [75]. Although this helps organizations to deal with technical and economic challenges, it is also increasing unintended risks [2]. These include manageability [1], traceability and auditing [3], adherence to policies and service level agreements (SLAs) [4] [77], service availability [5], security vulnerabilities [2] and use of non-compliant components [3]. Moreover, risks can arise when failing to comply with policies, regulations and industry standards, which is highly critical to not only business continuity [6] but also other consequences that result from non-compliance including cost of litigation and loss of reputation to mention a few. Moreover, typically, whenever the complexity of a software increases, its quality decreases [7] [76].

16th International Conference on Wirtschaftsinformatik,
March 2021, Essen, Germany

Software applications and services should be built in accordance (or compliance) to various policies, best practices, industry-specific needs, and regulations [2]. For most common practices nowadays, ensuring policies adherence to compliance requirements is often held by compliance experts, which is time-consuming and error-prone. What also complicates this process is the gap between compliance experts and domain experts. Eventually, management and monitoring of application behavior become more complicated over time [6]. Typically, requirements are extracted from legal regulations, branch-specific guidelines, internal code of conduct, and other sources. However, challenges arise from the change of these requirements as well as the adaptive environments along with rapid technological changes [9].

Furthermore, in the software supply chain, the philosophy of "assemble more, code less" is becoming very common nowadays, leading to issues in governance, risk management, and compliance (GRC) [2]. Therefore, with modern software applications and services that consist of complex and heterogeneous components, it becomes more challenging to manage their compliance to internal business policies, external regulations, industry standards, infrastructure and security requirements. The task becomes even more complicated, when different deployment technologies are used, in which the alternative manual way of checking and matching compliance requirements tend to be highly risky and mistakes are likely to happen [10]. Moreover, Nick [11] raised an issue with the control problem related to the advances in the capabilities of artificial intelligence (AI) in that self-optimizing AI components can misbehave and go against the boundaries of policies or regulations. All these challenges make the manual way of auditing and checking software compliance useless calling for a more innovative way to check software compliance.

The main objective of this systematic literature review is to survey the existing frameworks used for compliance checking of software and software services, their industry of application and compliance requirements for each industry. The contribution of this research is that it highlights recent progress in the compliance management of software and software services and that it points to future research areas.

Subsequent sections of this paper are organized as follows. Section 2 presents the methodology used, including the research questions formulated and the details on the review protocol used to execute this research. Section 3 presents the analysis and findings of the review. Section 4 discusses the findings and draws directions for future research. Finally, the conclusion section wraps up the key points of the review.

2 Methodology

We based the methodology for conducting a systematic literature review (SLR) on the one of Kitchenham at al. [12], which is one of the more relevant methods in the field of information systems research. We formulated the research questions and, then, developed and validated the review protocol. Afterwards, the collected studies were screened to add those, which are more relevant to our database. After that, we applied a set of criteria for inclusion and quality assessment. Then, after the data is extracted, documented into a database, and analyzed, the results are synthesized. Finally, findings are discussed and mapped against the research questions. The following subsections briefly discuss the research questions and the review protocol.

2.1 Research Questions

There are many aspects to investigate in the area of software compliance. However, we limit our review objective to surveying existing frameworks, their applications in industries, and compliance requirements by each industry. Therefore, we aim at answering the following two research questions:

RQ1. What are the existing frameworks of software compliance management and their applications in industries?

RQ2. What are the compliance requirements and needs of each industry?

2.2 Review Protocol

After setting up the research questions, we developed the review protocol, which includes the strategy applied for searching, selecting, including, and assessing the primary studies. We conducted a manual search using the terms “Software AND Compliance” to retrieve relevant studies. The search process considers the matches of both keywords in the title, abstract, or keywords of scholarly articles.

Selection of Sources: To ensure that the review includes as many relevant studies as possible within the defined search terms, we conducted a manual search in the following sources: *IEEE Xplore, ACM Digital Library, MDPI, Elsevier, HeinOnline, Springer, Web of Science, Scopus and Google Scholar.*

Inclusion Criteria: To keep our review focused on the objectives stated in Section 2.1, we developed a set of inclusion criteria as part of the review protocol. Therefore, the following criteria are applied to include primary studies for the final review:

Criterion 1: Only primary studies published between 2010 and 2020 are included.

Criterion 2: Relevant studies are only included for the review. By this, we mean studies that contribute to addressing our research questions.

Criterion 3: Only studies, which are accessible through Google Scholar and Seoul National University library, considered for the review.

Criterion 4: Only studies written in English are included for the review.

Criterion 5: Studies included for the review are limited to journal publications, conference proceedings, workshop proceedings, and symposium proceedings. Secondary studies, book chapters, presentations, dissertations, and reports are excluded.

Data Extraction: We used Zotero version 5 as a referencing tool to document, manage, and organize the references of the retrieved studies. We also set up a database, to record and extract relevant content. For that purpose, we used Microsoft Excel 2019, to record and manage findings. This helped making the analyses and investigations of findings simpler. It also provides a reference for further investigations in a systematic way.

3 Analysis of Results

3.1 Descriptive Analysis

Initial search on Google Scholar found 253 scholarly articles. We conducted an initial screening to eliminate irrelevant articles. From that, a total of 156 studies have been collected with respect to the search terms indicated in Section 2. Then, after applying the inclusion criteria, which are indicated in the review protocol, and checking the relevance of the primary studies to the research questions, only 63 primary studies are left for the review. Table 1 shows a summary of the studies selected for the review, including the database and types of studies. The table shows that more than half of the primary studies are conference papers. The rest are journal publications or proceedings from symposia and workshops. From well-known scientific databases, including IEEE, Elsevier, HeinOnline, ACM Digital Library, Springer and CiteSeerX, a total of 47 studies were collected. The remaining 16 studies are from sources other than the abovementioned databases, which include universities journals and proceedings.

Table 1. Summary of Selected Papers

Scientific Database	*Total Number of Papers*	*Journals*	*Conferences*	*Symposium*	*Workshops*
IEEE	29	1	24	2	2
Elsevier	5	5	-	-	-
HeinOnline	2	2	-	-	-
ACM Digital Library	5	-	2	2	1
Springer	5	2	3	-	-
CiteSeerX	1	1	-	-	-
Others	16	7	9	-	-
Total	63	18	38	4	3

Figure 1 shows the distribution of the publication years of the primary studies between 2010 and 2020 as indicated in the protocol of this review (Section 2). The trend in Figure 1 indicates that the research in software compliance is still growing, which is an indicator of the growing importance of the field.

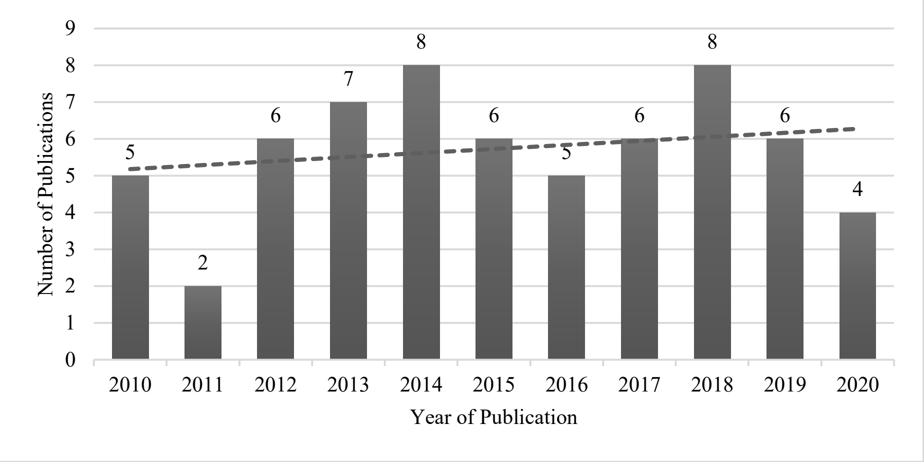

Figure 1. Distribution of Selected Papers by Year of Publication

3.2 Industry Requirements and Compliance Frameworks

Figure 2 summarizes the software compliance applications by industry. The analysis found that certain industries are investigated more than others. In the software industry itself, the review found 36% of the primary studies discuss compliance concerns in the software field. Then, the cloud industry comes with 22% of the studies, followed by the healthcare, in which 13% of the primary studies address issues related to software compliance. The figure also shows that 14% of the studies did not specify the industry of application. The rest of the industries which are discussed by fewer studies are as follows: manufacturing (6%), automobile (2%), financial (3%), aviation (2%), and e-government (2%). Some of the primary studies discuss a certain industry in the context of clouds (e.g., financial software running on clouds). For such scenarios, we classify them to their original industry. In other words, if a study discusses compliance of financial software on clouds, then we consider the focus to be on the financial industry.

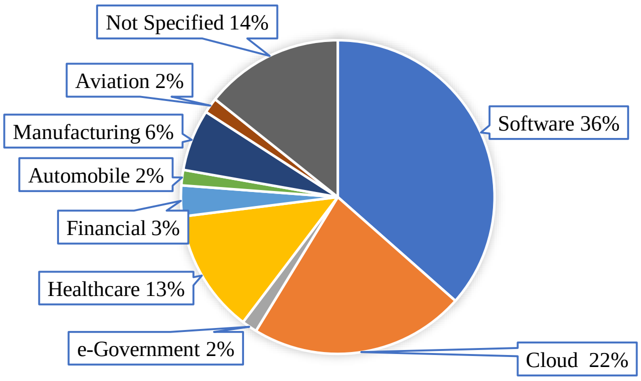

Figure 2. Distribution of primary studies by industry of application

Figure 2 also reflects the amount of challenges that each industry deals with. The majority of primary studies discuss compliance concerns related to software and cloud industries. This indicates that there are a lot of challenges and solutions discussed for these industries. The reason could be that these two industries are highly dynamic and many of their resources are available. Besides that, software and cloud industries are the central discussion in many of the primary studies. Nevertheless, changes in policies, regulations, and requirements are inevitable in every industry. The reviewed studies only discuss industries illustrated in Figure 2. Other industries are not found in primary studies based on our search terms. Perhaps, different terms are used, which do not include any of the search terms specified in our protocol.

To give a detailed picture on compliance requirements given by each industry, Table 2 shows the applications of compliance frameworks in the different industries along with the requirements needed by each industry. In the software industry, primary studies focus on compliance concerns related to distributed teams, intellectual property, components licensing, copyrights, reliability, security, trust, auditing, user permission, general data protection regulations (GDPR), privacy, software development lifecycle (SDLC), software design, regulatory requirements, process compliance, maintenance, governance risk & compliance (GRC), transparency, design-code compliance, and accountability. In the cloud industry, we found that studies focus on concerns related to security, privacy, compliance to service level agreements (SLA), trust, adaptation, accountability, resilience, application development, application deployment, management, provisioning, and adherence to regulations. Table 2 also shows that there is little attention to software compliance in governments, especially no attention on interoperability concerns of e-government services. Healthcare is an industry, which gained attention by primary studies. According to the primary study, we found that software systems need to comply with the health information technology for economic and clinical health (HITECH), health insurance portability and accountability (HIPAA), personal health information protection (PHIPA), organization for economic co-operation and development (OECD), requirement engineering, safety-critical aspects, quality, and reliability.

Moving to more safety-critical industries like automobile, manufacturing, and aviation, we found that these industries share some common compliance requirements, including reliability and compliance to safety standards. In addition to reliability and safety requirements, primary studies also show that the manufacturing industry focuses also on concerns including security, deployment & provisioning, privacy, GDPR, and industrial automation. Finally, the rest of primary studies did not specify or target a certain industry, however, those studies focus on compliance issues related to software design, service-oriented architecture (SOA), legal contracts, distributed systems, flexibility, auditing, transparency, security, IT service management (ITSM), business process modeling (BPM), outsourcing, and GRC.

Table 2. Compliance framework applications and compliance requirements in different industries

Industry	*Compliance Requirements*	*Reference*
Software	Distributed teams, intellectual property, components licensing, copyrights, reliability, security, trust, auditing, user permission, GDPR, privacy, SDLC, software design, regulatory requirements, process compliance, maintenance, GRC, transparency, design-code compliance, accountability	Singi et al. [14], Yun et al. [13], Singi et al. [2], van der Burg et al. [15], Hemel et al. [31], German and Di Penta [32], Jeff and Alan [33], Koltun [34], Von Willebrand and Patanen [35], Subramaniam and Natarajan [42], R P et al. [49], Gangadharan et al. [53], Hamou-Lhadj [55], Truong and Nguyen [56], Jensen et al. [58], Marques and Cunha [59], Arogundade et al. [62], Engiel et al. [63], Savarimuthu et al. [65], Chakraborty and Chaki [66], Jorshari and Tawil [67], Vytautas and Friedrich [70], Ozbas-Caglayan and Dogru [72],
Cloud	Security, privacy, SLA, trust, adaptation, accountability, resilience, application development, application deployment, management, provisioning, adherence to regulations, distributed services, SOA	McCarthy et al. [16], Suneel and Guruprasad [17], Hashmi et al. [18], Brandic et al. [36], García-Galán et al. [39], Florian et al. [40], Faniyi and Bahsoon [41], Singh and Sidhu [44], Krieger et al. [45], Carrasco et al. [46], Qanbari et al. [47], Breitenbucher et al. [50], Foster et al. [37], Koetter et al. [48]
e-Government	Interoperability	González and Ruggia [19]
Healthcare	HITECH, HIPAA, PHIPA, OECD, requirement engineering, safety-critical systems, quality, reliability	Gardazi and Ali [20], Sartoli et al. [21], Li et al. [22], Ingolfo et al. [51], Khan and Yun Bai [54], Lepmets et al. [64], Zema et al. [68], Maxwell and Antón [74]
Financial	Transparency, accountability, control, response to change	Magnusson and Chou [73], Koetter et al. [28]
Automobile	Functional safety, reliability	Hocking et al. [69]
Manufacturing	Security, deployment and provisioning, safety standards, privacy, GDPR, industrial automation	Zimmermann et al. [23], Castellanos Ardila and Gallina [43], Kittmann et al. [60], Moyon et al. [61]
Aviation	Safety standards, reliability	Jurnečka et al. [71]

Industry	*Compliance Requirements*	*Reference*
Not Specified	Software design, SOA, legal contracts, flexibility, auditing, transparency, security, ITSM, BPM, outsourcing, GRC, reliability	Fischer et al. [24], Tran et al. [25], Sharifi et al. [26], Loreti et al. [27], Groefsema and van Beest [29], Ingle et al. [30], Correia and Brito e Abreu [38], Thalmann et al. [52], Elhasnaoui et al. [57]

4 Discussion

Many industries heavily rely on software and software services, to automate as many of their business processes as possible. Thus, the use of software and software services becomes inevitable in many industries. With that, however, software projects grow and evolve over time as a response to changes in business and industry needs. This, in turn, has a negative impact on software quality according to the theory of software evolution, which was introduced by Lehman [7] in 1980. While most of this is related to functional requirements, there are also non-functional requirements, in which software and information systems need to comply with. These include security, privacy, licensing, reliability, provisioning, interoperability, data sharing, and adherence to regulations. Priorities of such requirements are also different between industries due to the different needs of each industry. The challenges come in fulfilling industry-specific compliance requirements and enable a degree of flexibility to respond to changes as well as checking whether new changes are reflected and enforced at the software level.

The analysis shows that primary studies discussed software compliance frameworks of 8 industries: software, cloud, e-Government, healthcare, financial, automobile, manufacturing, and aviation. Some further studies did not specify the industry, in which their proposed frameworks could be applied. There are some differences among the frameworks proposed by primary studies. These differences are driven by peculiarities of each industry, since each industry has its own business objectives, priorities, compliance requirements, and industry-specific needs. Moreover, the difference between the proposed frameworks is also influenced by the authors' assumptions and the context of compliance that they consider for their framework proposal. Nevertheless, some industries tend to have some compliance needs in common. For example, the manufacturing industry tends to focus on reliability and safety standards, which are also the focus of the automobile and aviation industries. The healthcare industry, however, tends to have different priorities, because they need to meet certain government regulations on healthcare. Furthermore, we found some differences in compliance requirements within the same industry. On top of these, regional-specific compliance requirements add another layer of complexity, especially for globally distributed software services and components.

Referring back to our research questions, there are many frameworks introduced by primary studies according to the analysis. Each has its own peculiarities depending on its application in a certain industry, business requirements, and assumptions considered by authors. In general, there are common issues that the primary studies try to address.

These are the changes in requirements and policies, the gap between IT and laws, the challenge of modeling policies and regulations, and reflecting those changes at a software level. Based on the analysis, compliance requirements, which are discussed most frequently in many industries, are: reliability, safety, security, and privacy, indicating that these requirements are highly critical to most industries.

In the software industry, Singi et al. [14] introduced a framework, in order to help establishing transparency and trust in distributed teams in global software delivery using blockchain. In the same context, other studies also investigated the challenges in crowd sourcing and how the software supply chain is affected in distributed software delivery [2] [25] [27]. Hamou-Lhadj [55] introduced the concept "software compliance engineering", emphasizing that regulatory compliance should be one of the key quality attributes of software products. Jorshari and Tawil [67] also support this argument of including compliance requirement analysis during the software development process, in order to have better governance, risk management and compliance (GRC). Another important aspect of software compliance is software licensing, in which many authors call for checking license compatibility, validation, awareness, dependency check of components, as well as license requirement analysis [15] [31] [32] [33] [35] [53]. The last important compliance issue to emphasize is ensuring design-code compliance. For this matter, Ozbas-Caglayan and Dogru [72] proposed an approach for analyzing software to check the compliance level of design and code using text mining and software repository analysis. To a great extent, the software industry deals with software compliance requirements and concerns from the perspective of software development practices. The aim is to ensure transparency and trust of distributed teams, component licensing, security, privacy, design-code compliance, and process compliance.

The cloud industry has also an increasing concern on compliance issues, especially security and trust between the cloud service providers and service consumers [16] [17] [18]. For this, Suneel and Guruprasad [17] introduced an approach to monitor SLA compliance of a cloud service provider (CSP), which can be implemented at the client end. They assume that a CSP is likely to violate the SLA, spoof the properties of the services, and, then, deliver the services with lower properties. Other studies also try to address the issues of trust, including Florian et al. [40], Singh and Sidhu [44], and Brandic et al. [36]. One of the major challenges in software compliance is modeling policies and legal aspects and enforcing them. For that, Breitenbucher et al. [50] proposed a policy-aware management framework. The framework enables automated provisioning and management of composite cloud applications based on a set of non-functional requirements defined by policies. However, this needs skills of both compliance and domain expertise. To simplify this, Hashmi et al. [18] introduced "security as a service" as a business model. It allows the delivery of managed security services to the user as a cloud service, to provide the end-users with monitoring information on their transaction and, thus, reducing the effect of security concerns. For the same reason, McCarthy et al. [16] introduced "compliance as a service" architecture, which is a cloud brokerage remediation service that checks non-functional security and compliance requirements. They aim at bridging the gap between agility and security, stating that the use of cloud does not guarantee security and legal compliance, which are still the user's obligation. Lastly, when it comes to service provisioning, automated installation of systems, and checking deployment rules, Krieger et al. [45] proposed an approach that enables modeling of reusable deployment compliance rules. Such rules are executed automatically to check declarative deployment models at design time. In the same context but for highly portable and provider-independent cloud applications, Carrasco et al. [46] introduced a model that supports applications, whose components are deployed on different providers. This, in turn, reduces the issues of portability, interoperability, and vendor lock-in. Overall, the software compliance in the cloud industry has similarities with the software industry, however, the cloud takes slightly higher level focusing on compliance concerns related to management and provisioning of software services, (e.g., security, privacy, service level agreement (SLA), adaptation, resilience, application deployment, distributed services).

In the healthcare industry, software projects also encounter many regulatory challenges, in particular, with respect to privacy of personal data. There is a gap between compliance and software architecture [20]. The evolving regulatory requirements affect all phases of the software development life cycle (SDLC), while in most software development practices, ensuring compliance is performed at requirement level. To bridge such a gap, Gardazi and Ali [20] introduced a compliance-driven software architecture based on a set of information security regulations and non-functional requirements. This helps achieving a compliance-aware software architecture. The majority of primary studies focus on security and privacy requirements represented by HITECH, HIPAA, PHIPA, and OECD. In this regard, and with the growing trend of home-based healthcare services, new compliance challenges have been raised in data collection, transferring, and sharing due to the geographical distribution of patients and their care providers. To address this issue, Li et al. [22] introduced the "CareNet" framework that bridges the gap between availability of software-defined infrastructure and compliance with regulatory requirements of a heterogeneous home-edge-core cloud for the home-based healthcare services. Further frameworks also attempt to bridge the gap between compliance and software architecture, by capturing the variability from legal sources and operating environments, real-time response, and modeling legal rules [20] [21] [74]. The growing development of smart healthcare services is a potential area to investigate in software compliance.

Similarly, other industries including financial, manufacturing, automobile, aviation, and government look at compliance concerns from an industry-specific perspective. The financial industry focuses on compliance issues related to transparency, accountability, and control. Manufacturing, automobile, and aviation industries have some similarities in compliance concerns, because they share relatively similar industry requirements. Specifically, safety standards and functional reliability are critical requirements for these industries. We also found that software compliance concerns are the least discussed by primary studies in the context of governments. Instead, their main focus is on interoperability aspects of e-Government services. Due to this and the fact that governments are highly complex systems, there is room for research on compliance concerns in governments. In general, all the frameworks discussed by primary studies are industry-dependent and cannot fit into one another. This means that implementing the same software project in two different industries is more likely to experience different compliance issues, which are decided by the industry itself. Therefore, taking into account the industry-specific compliance needs when designing a software architecture is crucial to flexibility and adaptability of the software.

What all these frameworks share in common are the issues of changing requirements and policies, the gap between IT and laws, and the challenge of modeling policies and regulations in a way that can easily be reflected at the software level. However, based on compliance issues and frameworks discussed, we can classify industries into two groups. This classification is based on the level of details that the industries consider for their compliance requirements as well as the aspects that they look into. We classify software and cloud industries as one group, and all other industries as another group. Although there are some overlaps, the justification of this classification is that software and cloud industries tend to look at compliance concerns from the perspective of software development practices and service provisioning, while other industries look at the architectural level and from the industry-specific perspective. In other words, on the one hand, software and cloud industries discuss issues related to distributed teams, component licensing, SLA compliance, reliability, trust, service provisioning, and management. On the other hand, the other industries, including healthcare, manufacturing, finance, aviation and automobile, discuss software compliance at a higher level (i.e., compliance with industry standards, regulations, data sharing policies, and architectural perspective of software). Moreover, the proposed frameworks by primary studies are industry-dependent, emphasizing the importance of considering industry specific compliance requirements when designing a software architecture.

5 Conclusion

We used a systematic literature review, in order to survey existing frameworks and industry requirements regarding software compliance management. The review highlighted that many, different frameworks have been proposed for many industries to manage compliance of software and software services. There is no single solution that fits all scenarios and can be applied across all industries. Each industry has its own peculiarities, compliance requirements, and priorities, which need to be considered when managing software compliance accordingly. Nevertheless, there are common issues emphasized by many primary studies including the gap between compliance and software architecture, modeling policies and regulations, and enforcing those changes at a software level. Based on the analysis, there are two groups of industries that can be distinguished. The group composed of the software and cloud industries views compliance concerns from a component level, while the other group, which is composed of all other industries, looks at it from an architectural level. In other words, software and cloud industries focus on software compliance from a perspective of software development practices and service provisioning, while other industries focus on software compliance from a higher level perspective, which considers industry-specific requirements and regulations. In future work, we will provide an extended study on tools and technologies used to manage and enforce software compliance.

As there is little research on software compliance in some industries (e.g., financial, government, automobile, and aviation), these industries and others are areas for future research. Furthermore, other potential directions for future research are: First, tools and technologies used for management and enforcement of software compliance; Second, technologies used for policy and legal modeling and the extent to which advances in technologies like AI and blockchain can help addressing it; Third, studies of software compliance in the context of government software projects with respect to compliance requirements and challenges.

References

1. Sefika M, Sane A, Campbell RH. Monitoring compliance of a software system with its high-level design models," in Proc of IEEE 18th Intl Conf on Software Engineering, Mar. 1996, pp. 387–396, doi: 10.1109/ICSE.1996.493433.
2. Singi K, RP JCB, Podder S, Burden AP. Trusted Software Supply Chain. In 34th IEEE/ACM Intl Conf on Automated Software Engineering (ASE), Nov. 2019, pp. 1212–1213, doi: 10.1109/ASE.2019.00141.
3. Harutyunyan N, Riehle D. Getting started with open source governance and compliance in companies. 2019, Accessed: Jan. 14, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3306446.3340815.
4. Wurster M, Breitenbücher U, Falkenthal M, Leymann F. Developing, deploying, and operating twelve-factor applications with TOSCA. In Proc of 19th Intl Conf on Information Integration and Web-based Applications & Services, New York, NY, USA, Dec. 2017, pp. 519–525, doi: 10.1145/3151759.3151830.
5. González L. Ruggia R. Controlling Compliance of Collaborative Business Processes through an Integration Platform within an E-government Scenario. Jan. 2020, doi: 10.24251/HICSS.2020.245.
6. Zimmermann M, Breitenbucher U, Krieger C, Leymann F. Deployment Enforcement Rules for TOSCA-based Applications," Proc of Twelfth Intl Conf on Emerging Security Information, Systems and Technologies (SECURWARE 2018), pp. 114–121, 2018.
7. Lehman MM. Programs, life cycles, and laws of software evolution. Proc of IEEE, vol. 68, no. 9, Art. no. 9, Sep. 1980, doi: 10.1109/PROC.1980.11805.
8. Wettinger J, Behrendt M, Binz T, Breitenbücher U, Breiter G, Leymann F, Moser S, Schwertle I, Spatzier T. Integrating Configuration Management with Model-driven Cloud Management based on TOSCA. pp. 437–446, 2013.
9. Koetter F, Kochanowski M, Renner T, Fehling C, Leymann F. Unifying Compliance Management in Adaptive Environments through Variability Descriptors. In IEEE 6th Intl Conf on Service-Oriented Computing and Applications, Dec. 2013, pp. 214–219, doi: 10.1109/SOCA.2013.23.
10. Breitenbucher U, Binz T, Fehling C, Kopp O, Leymann F, Wieland M. Policy-Aware Provisioning and Management of Cloud Applications. International Journal On Advances in Security, vol. 7, p. 23, 2014.
11. Bostrom N. Superintelligence: Paths, Dangers. Strategies, Oxford University Press (2014).
12. Kitchenham BA, et al. Evidence-Based Software Engineering and Systematic Reviews. CRC Press, 2016.

13. Yun HY, Joe YJ, Shin DM. Method of license compliance of open source software governance. In: 8th IEEE Intl Conf on Software Engineering and Service Science (ICSESS). 2017. p. 83–6.
14. Singi K, Kaulgud V, Bose RPJC, Podder S. CAG: Compliance Adherence and Governance in Software Delivery Using Blockchain. In: IEEE/ACM 2nd Intl Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). 2019. p. 32–9.
15. van der Burg S, Dolstra E, McIntosh S, Davies J, German DM, Hemel A. Tracing software build processes to uncover license compliance inconsistencies. In: Proc of 29th ACM/IEEE Intl Conf on Automated software engineering [Internet]. New York, NY, USA: ACM; 2014 [cited 2020 Oct 14]. p. 731–42. (ASE '14). Available from: https://doi.org/10.1145/2642937.2643013
16. McCarthy MA, Herger LM, Khan SM. A Compliance Aware Software Defined Infrastructure. In: IEEE Intl Conf on Services Computing. 2014. p. 560–7.
17. Suneel K, Guruprasad HS. A Novel Approach for SLA Compliance Monitoring in Cloud Computing. International Journal of Innovative Research in Advanced Engineering (IJIRAE). 2015 Jan 1;2(2).
18. Hashmi A, Ranjan A, Anand A. Security and Compliance Management in Cloud Computing. INTERNATIONAL JOURNAL OF ADVANCED STUDIES. 2018;7(1):8.
19. González L, Ruggia R. Controlling Compliance of Collaborative Business Processes through an Integration Platform within an E-government Scenario. In: Proc of 53rd Hawaii Intl Conf on System Sciences | 2020 [Internet]. 2020 [cited 2020 Oct 22]. Available from: http://scholarspace.manoa.hawaii.edu/handle/10125/63986
20. Gardazi SU, Ali A. Compliance-Driven Architecture for Healthcare Industry. International Journal of Advanced Computer Science and Applications. 2017 Jan 1;8.
21. Sartoli S, Ghanavati S, Siami Namin A. Compliance Requirements Checking in Variable Environments. In: IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). 2020. p. 1093–4.
22. Li P, Xu C, Luo Y, Cao Y, Mathew J, Ma Y. CareNet: Building a Secure Software-defined Infrastructure for Home-based Healthcare. In: Proc of ACM Intl Workshop on Security in Software Defined Networks & Network Function Virtualization [Internet]. New York, NY, USA: ACM; 2017 [cited 2020 Oct 17]. p. 69–72. (SDN-NFVSec '17). Available from: https://doi.org/10.1145/3040992.3041007
23. Zimmermann M, Breitenbucher U, Krieger C, Leymann F. Deployment Enforcement Rules for TOSCA-based Applications. Proc of Twelfth Intl Conf on Emerging Security Information, Systems and Technologies (SECURWARE 2018). 2018;114–21.
24. Fischer MP, Breitenbucher U, Kepes K, Leymann F. Towards an Approach for Automatically Checking Compliance Rules in Deployment Models. Eleventh Intl Conf on Emerging Security Information, Systems and Technologies. 2017;5.
25. Tran H, Zdun U, Holmes T, Oberortner E, Mulo E, Dustdar S. Compliance in service-oriented architectures: A model-driven and view-based approach. Information and Software Technology. 2012 Jun 1;54(6):531–52.
26. Sharifi S, Parvizimosaed A, Amyot D, Logrippo L, Mylopoulos J. Symboleo: Towards a Specification Language for Legal Contracts. In: 2020 IEEE 28th Intl Requirements Engineering Conference (RE). 2020. p. 364–9.
27. Loreti D, Chesani F, Ciampolini A, Mello P. A distributed approach to compliance monitoring of business process event streams. Future Generation Computer Systems. 2018 May 1; 82:104–18.
28. Koetter F, Kochanowski M, Weisbecker A, Fehling C, Leymann F. Integrating Compliance Requirements across Business and IT. In: IEEE 18th Intl Enterprise Distributed Object Computing Conference. 2014. p. 218–25.

29. Groefsema H, van Beest N. Design-Time Compliance of Service Compositions in Dynamic Service Environments. In: IEEE 8th Intl Conf on Service-Oriented Computing and Applications (SOCA). 2015. p. 108–15.
30. Ingle C, Samudre A, Bhavsar P, Vidap PS. Audit and Compliance in Service Management using Blockchain. In: 2019 IEEE 16th India Council Intl Conf (INDICON). 2019. p. 1–4.
31. Hemel A, Kalleberg KT, Vermaas R, Dolstra E. Finding software license violations through binary code clone detection. In: Proc of the 8th Working Conf on Mining Software Repositories [Internet]. New York, NY, USA: ACM; 2011 [cited 2020 Oct 14]. p. 63–72. (MSR '11). Available from: https://doi.org/10.1145/1985441.1985453
32. German D, Di Penta M. A Method for Open Source License Compliance of Java Applications. IEEE Software. 2012 May;29(3):58–63.
33. Jeff H, Alan L. A Novel Method for Decentralised Peer-to-peer Software License Validation Using Cryptocurrency Blockchain Technology. In Australian Computer Society (ACS); 2015 [cited 2020 Oct 14]. Available from: https://openrepository.aut.ac.nz/handle/10292/10328
34. Koltun P. Free and Open Source Software Compliance: An Operational Perspective. IFOSS L Rev. 2011;3(1):95–102.
35. Von Willebrand M, Patanen M-P. Package Review as a Part of Free and Open Source Software Compliance. IFOSS L Rev. 2010. 2(1):39–60.
36. Brandic I, Dustdar S, Anstett T, Schumm D, Leymann F, Konrad R. Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. In: IEEE 3rd Intl Conf on Cloud Computing. 2010. p. 244–51.
37. Foster H, Spanoudakis G, Mahbub K. Formal Certification and Compliance for Run-Time Service Environments. In: IEEE Ninth Intl Conf on Services Computing. 2012. p. 17–24.
38. Correia A, Brito e Abreu F. Defining and Observing the Compliance of Service Level Agreements: A Model Driven Approach. In: 2010 Seventh International Conference on the Quality of Information and Communications Technology. 2010. p. 165–70.
39. García-Galán J, Pasquale L, Grispos G, Nuseibeh B. Towards Adaptive Compliance. In: IEEE/ACM 11th Intl Symp on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). 2016. p. 108–14.
40. Florian M, Paudel S, Tauber M. Trustworthy evidence gathering mechanism for multilayer cloud compliance. In: 8th Intl Conf for Internet Technology and Secured Transactions (ICITST-2013). 2013. p. 529–30.
41. Faniyi F, Bahsoon R. Self-managing SLA compliance in cloud architectures: a market-based approach. In: Proc of the 3rd Intl ACM SIGSOFT Symp on Architecting Critical Systems [Internet]. New York, NY, USA: ACM; 2012 [cited 2020 Oct 14]. p. 61–70. (ISARCS '12). Available from: https://doi.org/10.1145/2304656.2304665
42. Subramaniam C, Natarajan K. Software Reliability Compliance Model for Requirements Faults. In: In Recent Trends in Communications and Computers. Proc of 16th WSEAS Intl Conf on Communications. 2012. p. 332–40.
43. Castellanos Ardila JP, Gallina B. Separation of Concerns in Process Compliance Checking: Divide-and-Conquer. In Springer International Publishing; 2020 [cited 2020 Oct 15]. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:mdh:diva-49334
44. Singh S, Sidhu J. Compliance-based Multi-dimensional Trust Evaluation System for determining trustworthiness of Cloud Service Providers. Future Generation Computer Systems. 2017 Feb 1; 67:109–32.
45. Krieger C, Breitenbücher U, Képes K, Leymann F. An Approach to Automatically Check the Compliance of Declarative Deployment Models. In: IBM Research Division. 2018. p. 76–89.
46. Carrasco J, Cubo J, Durán F, Pimentel E. Bidimensional Cross-Cloud Management with TOSCA and Brooklyn. In: IEEE 9th Intl Conf on Cloud Computing. 2016. p. 951–5.

47. Qanbari S, Sebto V, Dustdar S. Cloud Resources-Events-Agents Model: Towards TOSCA-Based Applications. In: Villari M, Zimmermann W, Lau K-K, editors. Service-Oriented and Cloud Computing. Berlin, Heidelberg: Springer; 2014. p. 160–70.
48. Koetter F, Kochanowski M, Renner T, Fehling C, Leymann F. Unifying Compliance Management in Adaptive Environments through Variability Descriptors (Short Paper). In: IEEE 6th Intl Conf on Service-Oriented Computing and Applications. 2013. p. 214–9.
49. R P JCB, Singi K, Kaulgud V, Phokela KK, Podder S. Framework for Trustworthy Software Development. In: 34th IEEE/ACM Intl Conf on Automated Software Engineering Workshop (ASEW). 2019. p. 45–8.
50. Breitenbucher U, Binz T, Fehling C, Kopp O, Leymann F, Wieland M. Policy-Aware Provisioning and Management of Cloud Applications. International Journal on Advances in Security. 2014; 7:23.
51. Ingolfo S, Siena A, Mylopoulos J, Susi A, Perini A. Arguing regulatory compliance of software requirements. Data & Knowledge Engineering. 2013 Sep 1; 87:279–96.
52. Thalmann S, Bachlechner D, Demetz L, Manhart M. Complexity is dead, long live complexity! How software can help service providers manage security and compliance. Computers & Security. 2014 Sep 1; 45:172–85.
53. Gangadharan GR, D’Andrea V, De Paoli S, Weiss M. Managing license compliance in free and open source software development. Inf Syst Front. 2012 Apr 1;14(2):143–54.
54. Khan KM, Yun Bai. Automatic verification of health regulatory compliance in cloud computing. In: IEEE 15th Intl Conf on e-Health Networking, Applications and Services (Healthcom 2013). 2013. p. 719–21.
55. Hamou-Lhadj A. Regulatory compliance and its impact on software development. Software Compliance Research Group, Department of Electrical and Computer Engineering. 2015.
56. Truong N-T, Nguyen V-H. An approach to checking the compliance of user permission policy in software development. Int J Soft Eng Knowl Eng. 2013 Oct 1;23(08):1139–51.
57. Elhasnaoui S, Drissi S, Iguer H, Medromi H. Multi-Agent Architecture of Intelligent and Distributed Platform of Governance, Risk and Compliance of Information Systems. IJACSA [Internet]. 2019 [cited 2020 Dec 17];10(5). Available from: http://thesai.org/Publications/ViewPaper?Volume=10&Issue=5&Code=IJACSA&SerialNo=10
58. Jensen M, Kapila S, Gruschka N. Towards Aligning GDPR Compliance with Software Development: A Research Agenda. 2019. 389 p.
59. Marques J, Cunha AM da. Tailoring Traditional Software Life Cycles to Ensure Compliance of RTCA DO-178C and DO-331 with Model-Driven Design. In: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). 2018. p. 1–8.
60. Kittmann T, Lambrecht J, Horn C. A privacy-aware distributed software architecture for automation services in compliance with GDPR. In: 2018 IEEE 23rd Intl Conf on Emerging Technologies and Factory Automation (ETFA). 2018. p. 1067–70.
61. Moyon F, Beckers K, Klepper S, Lachberger P, Bruegge B. Towards Continuous Security Compliance in Agile Software Development at Scale. In: 2018 IEEE/ACM 4th Intl Workshop on Rapid Continuous Software Engineering (RCoSE). 2018. p. 31–4.
62. Arogundade OT, Abioye TE, Mustapha AM, Adeniji AM, Ikotun AM, Asahiah FO. Specifying and Incorporating Compliance Requirements into Software Development Using UML and OCL. In: Gervasi O, Murgante B, Misra S, Stankova E, Torre CM, Rocha AMAC, ed. Computational Science and Its Applications (ICCSA). Springer. 2018. p. 511–26.
63. Engiel P, Leite JCSDP, Mylopoulos J. A tool-supported compliance process for software systems. In: 11th Intl Conf on Research Challenges in Information Science (RCIS). 2017. p. 66–76.
64. Lepmets M, McBride T, McCaffery F. Towards Safer Medical Device Software Systems: Industry-Wide Learning from Failures and the Use of Safety-Cases to Support Process

Compliance. In: 10th Intl Conf on the Quality of Information and Communications Technology (QUATIC). 2016. p. 193–8.

65. Savarimuthu T, Dam H, Licorish S, Keertipati S, Avery D, Ghose A. Process Compliance in Open Source Software Development – A Study of Python Enhancement Proposals (PEPS). Research Papers [Internet]. 2016 Jun 15; Available from: https://aisel.aisnet.org/ecis2016_rp/48
66. Chakraborty M, Chaki N. A New Framework for Configuration Management and Compliance Checking for Component-Based Software Development. In: Chaki R, Cortesi A, Saeed K, Chaki N, editors. Advanced Computing and Systems for Security: Vol 2 [Internet]. New Delhi: Springer India; 2016 [cited 2020 Dec 18]. p. 173–88. (Advances in Intelligent Systems and Computing). Available from: https://doi.org/10.1007/978-81-322-2653-6_12
67. Jorshari FZ, Tawil RH. A High-Level Scheme for an Ontology-Based Compliance Framework in Software Development. In: IEEE 17th Intl Conf on High Performance Computing and Communications, IEEE 7th Intl Symp on Cyberspace Safety and Security, and IEEE 12th Intl Conf on Embedded Software and Systems. 2015. p. 1479–87.
68. Zema M, Rosati S, Gioia V, Knaflitz M, Balestra G. Developing medical device software in compliance with regulations. In: 2015 37th Annual Intl Conf of the IEEE Engineering in Medicine and Biology Society (EMBC). 2015. p. 1331–4.
69. Hocking AB, Knight J, Aiello MA, Shiraishi S. Arguing Software Compliance with ISO 26262. In: IEEE Intl Symp on Software Reliability Engineering Workshops. 2014. p. 226–31.
70. Vytautas Č, Friedrich L. Compliance and Software Transparency for the Design of Legal Machines. In 2014.
71. Jurnečka P, Hanáček P, Barabas M, Henzl M, Kačic M. A method for parallel software refactoring for safety standards compliance. In: 8th IET Intl System Safety Conference incorporating the Cyber Security Conference 2013. 2013. p. 1–6.
72. Ozbas-Caglayan K, Dogru AH. Software Repository Analysis for Investigating Design-Code Compliance. In: Joint Conf. of 23rd Intl Workshop on Software Measurement and 8th Intl Conf on Software Process and Product Measurement. 2013. p. 231–4.
73. Magnusson C, Chou S. Risk and Compliance Management Framework for Outsourced Global Software Development. In: 5th IEEE Intl Conf on Global Software Engineering. 2010. p. 228–33.
74. Maxwell JC, Antón AI. The production rule framework: developing a canonical set of software requirements for compliance with law. In: Proc of 1st ACM Intl Health Informatics Symp [Internet]. New York, NY, USA: ACM; 2010 [cited 2020 Dec 18]. p. 629–636. (IHI '10). Available from: https://doi.org/10.1145/1882992.1883092
75. Kim K, Altmann J. Platform Provider Roles in Innovation in Software Service Ecosystems. IEEE Transactions on Engineering Management, https://doi.org/10.1109/TEM.2019.2949023, 2020.
76. Haile N, Altmann J. Evaluating Investments in Portability and Interoperability between Software Service Platforms. Future Generation Computer Systems 78(1): 224-241, https://doi.org/10.1016/j.future.2017.04.040, Elsevier, January 2018.
77. Breskovic I, Altmann J, Brandic I. Creating Standardized Products for Electronic Markets," Future Generation Computer Systems, Elsevier, 29(4): 1000-1011, June 2013.