Ph.D. in Engineering

Assessing the Policies and Factors Impacting Software Engineering Compliance

February 2023

Graduate School of Seoul National University

Technology Management, Economics, and Policy Program

Mohammed Salem Mubarkoot

Assessing the Policies and Factors Impacting Software Engineering Compliance

지도교수 **Jörn Altmann**

이 논문을 공학박사학위 논문으로 제출함

2023 년 2 월

서울대학교 대학원

협동과정 기술경영경제정책 전공

Mohammed Salem Mubarkoot

모하메드의 공학박사학위 논문을 인준함

2023 년 2 월

위 원 장 황준석 (인)

부위원장 **Jörn Altmann** (인)

위 원 **AI-Athwari Baseem Taher Othman** (인)

위 원 윤현영 (인)

위 원 **Bernhard Egger** (인)

Abstract

Assessing the Policies and Factors Impacting Software Engineering Compliance

Mohammed Salem Mubarkoot
Technology Management, Economics and Policy, ITPP
College of Engineering
Seoul National University

This research is motivated by the growing concerns of insider threats, as they account for 56% of software attacks; in addition to the rising number of organizations which experience software attacks and incidents to 67% with an increase of 7% in the last two years. The research is also motivated by consequences raised by software engineering workarounds, as the phenomena is a serious business problem and relatively unexplored. The short-term gains from workarounds in software engineering can result in technical debt, making it more difficult to pay back as time goes. Additionally, workarounds and temporary fixes can impact future software releases and its overall security and maintainability. As such workarounds account for over 25% of waste in efforts, it is worth investigating what further contributes to

development of workarounds, in order to understand and address their causes. This research presents two main studies as follows:

The first study uses systematic literature review to understand the current research focus, evolving theories and concepts, and potential gaps and directions in software compliance. The study uses an evidence-based thinking to answer the review questions. Based on the review protocol, and the inclusion and exclusion criteria, 84 relevant studies were identified. The results identified 55 factors that impact behavioral compliance at different scopes, and 20 policies along with compliance challenges they address. The review reveal several key findings: (1) End user security is a top discussion followed by legal and privacy issues; (2) Security awareness and automation of compliance are top cited policies; (3) There is an emphasis on the gap between domain and compliance experts on one hand, and software engineers on the other hand; (4) Wile the theory of planned is dominating, the theory of workarounds has emerged in the domain; (5) There are several evolving concepts in the domain: compliance and privacy by design, policy as code, security stress, and home-office users. The study delivers a set of theoretical and practical implications that provide researchers and practitioners with potential research directions and policy guidance.

The second study uses a deductive quantitative method, to examine the factors that impact software engineering workarounds and the extent to which the factors of technostress can trigger workarounds. It also investigates the

role of neutralization strategies, professional autonomy, and perceived behavioral controls as moderators on that impact. The study aims to assess factors leading to software engineering workarounds and provide a new understanding on technostress in the context of workarounds, while emphasize how significant their impact is, on software engineering workarounds. The study positions technostress as a new antecedent of workarounds. It contextualizes workarounds focusing on software engineering since they recognize technical intricacies more than any other stakeholder in software ecosystem. While literature reports that the causes of workarounds come from pressure of meeting deadlines, misfit of work practices, inadequate resources, and complexity of overwhelming technologies, our study posits technostress among causes of workarounds in software engineering. In addition to that, the study also argues that neutralization, degree of professional autonomy given to engineers over technical decisions, and perceived behavioral controls can strengthen that impact. Based on a cross-sectional survey data from 306 software engineers, the study applies covariance-based (CB) and partial least square (PLS) structural equation modeling to evaluate the proposed research model. Detailed analysis and comparison between findings of CB-SEM and PLS-SEM is conducted.

Results report dimensions of Technostress (Overload and Invasion) predicts workarounds indirectly through Strain, while Complexity, Overload, and Invasion report a direct impact only on Strain. Furthermore, the findings

show that technostress (Overload and Insecurity) have a direct impact on the workaround intention. The findings of both CB-SEM and PLS-SEM conclude insignificant moderating impact of neutralization. The CB-SEM analysis report a significant moderating impact of autonomy and perceived behavioral control on the relationship between strain and workarounds intention, while PLS-SEM analysis reports an insignificant result. The study extends the theory of workarounds and provide a new understanding of technostress in the context of software engineering and the moderating role of neutralization, autonomy and behavioral control on engineers' workaround behavior. The findings of this study help practitioners and researchers develop policy response in order to control workarounds; and further gain insights for future research.

Keywords: Software Compliance, Software Policy, Technostress, Neutralization, Autonomy, Workaround, Technical Debt.

Student Number: 2019-39915

Contents

List of Tables

List of Figures

Chapter 1. Overall Introduction

1.1 Background and Motivation

Recent statistics show that 67% of organizations experience 21 to 40 incidents in their software systems per year, with a 7% increase over the last 2 years. Many of these incidents are caused by insiders and take a great deal of time and effort to contain (Proofpoint, 2022). In particular, E-type software systems, which businesses use as part of problem-solving processes, are highly sensitive to real-world changes (Lehman & Ramil, 2002). Ensuring compliance of software systems with regulations, corporate policies, and industry best practices is of paramount importance. Disruptions and downtimes of software systems cause substantial financial and reputational damage to organizations. A study conducted by Ponemon Institute reveals that the mean cost of data center outages is close to $650,000 (Ponemon Institute, 2016). People and organizations increasingly depend on the reliability and security of software systems and services (PricewaterhouseCoopers, 2021). With most software services becoming an integral part of our daily business, any disruption could lead to severe consequences.

Studies have also shown that humans are considered the weakest chain in software compliance (Guhr et al., 2019), accounting for more than 50% of security and data breaches (Balozian & Leidner, 2017). PricewaterhouseCoopers (2018) revealed that recovery from such security breaches takes 19 hours on average; the same report also found more than 28% of businesses do not have an idea of the number of attacks they have experienced. Additionally, the report found that 48% of employees are lacking security awareness and training, whereas around 54% of employees report the

absence of strong incident response processes in their organizations (PricewaterhouseCoopers, 2018).

Based on the laws of software evolution, certain forces push the need for innovation in software and other forces constrain it (Lehman & Ramil, 2002). Among these forces are regulations, corporate policies, and industry best practices. Furthermore, the variation and diversity of compliance sources complicate the management of compliance (Mubarkoot et al., 2022; Tran et al., 2012). As software systems are among the most precious assets of organizations, assuring compliance with regulations, corporate policies, and industry best practices is of paramount importance. Therefore, it is crucial to increase understanding of the existing research foci, evolving issues and topics, and the relevance of potential research directions. Technical countermeasures are insufficient to enhance the overall compliance of software systems; procedural countermeasure are considered critical since it is the human factor that is most challenging (Balozian et al., 2021). This, in turn, requires deep understanding of contemporary factors in order to better develop procedural countermeasure policies accordingly.

One serious and misunderstood issue of organizational insiders is the workaround phenomena and the use of shadow information technologies (IT) (de Vargas Pinto et al., 2022; Silic et al., 2017). While the short-term gains of workarounds can enhance productivity and result in faster delivery, their consequences, in the long run, can be severe (Alter, 2014; R. M. Davison et al., 2021). In particular, the long-term consequences of workarounds implemented by software engineers can become more complicated to deal with as time goes on since they require additional rework and refactoring (Yli-

Huumo et al., 2016). Such consequences are referred to as technical debt (Potdar & Shihab, 2014; Ward Cunningham, 2009).

Technical debts can be a form of counterproductivity and are becoming a serious issue in software engineering, in that around 25% of development efforts are wasted on extra rework and refactoring due to technical debts and workarounds (Ramač et al., 2022). Among the top causes of technical debts are time-to-market deadlines, improper planning, and lack of knowledge (Ramač et al., 2022; Rios et al., 2018). Software engineers are considered among the most stressed workers (Ostberg et al., 2020). Typical causes of the stress they experience are pressure to meet deadlines or technological complexity, and the constant need to stay up-to-date with an overwhelming number of technological advances (Pérez et al., 2021; Ramač et al., 2022; Yli-Huumo et al., 2015, 2016). In this regard, it is critical to investigate what could further cause workarounds, in order to provide a better understanding of the workaround phenomenon and, therefore, develop an effective solution accordingly.

The recent advances in capabilities of information technologies and connectivity have led to an invasion of one's personal space, mixing home and work, causing overload and adding fear of job insecurity. Technostress has raised huge concerns and is a dark side of technology that negatively impacts human behavior (Bondanini et al., 2020). Dimensions of technostress include technology complexity overload, uncertainty, invasion, and insecurity (Ragu-Nathan et al., 2008; Tarafdar et al., 2015). While most technology-related research focuses on what technology does ***for*** people (i.e., the positive impact), it is highly important to recognize what technology can do ***to*** people

as well (Tarafdar et al., 2015) (i.e., the negative impact), in order to understand the negative consequences of technology and provide a foundation to address them. Studies have reported that the impact of technostress on counterproductivity can be severe (Bondanini et al., 2020; Jaekang & Taekyung, 2015; H. Kim et al., 2016). The growing concerns about technostress and its consequences are more likely to add up to the stress people experience, and therefore more likely to trigger workarounds. Thus, the extent to which the dimensions of technostress influence workarounds is worth investigating.

1.2 Problem Description

Software systems and services are considered valuable organizational assets, and guaranteeing adherence to multiple requirements, regulations, industry standards, and best practices is a major challenge. The multi-faceted sets of compliance sources a software system faces complicate compliance management (Tran et al., 2012). Technological approaches are insufficient in ensuring the security of information systems in organizations. Studies reveal that software users do not take appropriately prescribed actions as stated in organizational information security policies (Moody et al., 2018). Similarly, developers and engineers also lack a sense of responsibility to deliver beyond functionalities, for example, taking security and privacy into consideration during the design phase or using software engineering best practices (Bednar et al., 2019).

In order to understand the wider context in which this issue falls, it is crucial to investigate the existing research foci, evolving topics and concepts in the domain, and the relevance of potential research directions. Existing

review works primarily focus on a particular industry or a specific compliance aspect. There is a lack of comprehensive review that investigates the state-of-the-art literature on compliance requirements, and the impacting factors, policies, and challenges they address. Therefore, having such a study helps highlight the relevance of potential gaps and helps position further empirical research accordingly.

The theory of workarounds is an evolving theory in the domain of compliance and causes growing concerns about security vulnerabilities and threats (R. Davison et al., 2019; Song et al., 2020; Wong et al., 2022). The theory states that participants in a work system improvise, adapt, or bypass some of the existing procedures in order to overcome or reduce constraints that prevent them from achieving better efficiency or effectiveness (Alter, 2014). In the field of software engineering, the term is referred to as technical debt as engineers compromise quality to gain short-term benefits. Technical debt has emerged as a serious issue in software engineering (Ramač et al., 2022; Yli-Huumo et al., 2015), and the term is also used interchangeably with workarounds.

While previous studies indicate that engineers lack responsibility to deliver beyond functionality and take appropriate action (Bednar et al., 2019; Moody et al., 2018), there is a lack of research on workarounds, their causes, and related impact (R. Davison et al., 2019; Song et al., 2020; Wong et al., 2022). Several studies are built on the concepts of this theory. However, their focus was primarily on end users. Less research attention is paid to the workarounds that are performed by software engineers.

Software engineering is considered one of the most stressful jobs

(Ostberg et al., 2020). In particular, technostress has recently gained more attention (Bondanini et al., 2020). It has been viewed to have a negative impact on policy compliance (Nasirpouri & Biros, 2020). However, the extent to which technostress stimulates and impacts workarounds has not been investigated. Previous studies focus primarily on time pressure, meeting deadlines, and misfit of work practices as the main causes of workarounds (R. M. Davison et al., 2021; Pérez et al., 2021; Ramač et al., 2022). No prior studies investigated the impact of technostress on workarounds. The importance of this study comes from the need to address the serious consequences of workarounds in software engineering since they equate to approximately 25% of efforts wasted on additional rework and refactoring. Thus, the impact of technostress on engineers' workarounds is worth investigating.

1.3 Research Objective and Research Questions

The rapid progress of technologies along with changing corporate policies and business requirements have shortened the evolution cycle of a software, making the status of the software likely to be in a releasable state most of the time. This in turn poses growing concerns about software policy compliance to many stakeholders. As software systems and services are becoming an integral part of our daily business, it is of high importance to clarify and understand critical aspects that practitioners and researchers should pay attention to. In this regard, this research is conducted to achieve two main objectives aimed at improving decisions related to designing effective software compliance policies based on empirical evidence.

First Objective: Investigating software compliance requirements,

impacting factors, policies, and challenges they address. The purpose is to investigate the current research focus, evolving concepts and issues in the domain, and potential research directions and their relevance to the domain. This helps identify top issues discussed, mature and evolving theories, and deliver key highlights on research gaps and their importance in the field. Accordingly, using a systematic literature review, this study attempts to answer the following three research questions:

RQ1: What are software compliance requirements with respect to different industries and user contexts?

RQ2: What are the factors that impact software compliance and which aspects of compliance are impacted?

RQ3: What are the existing software compliance policies and which compliance challenges do they tend to address?

Second Objective: Assessing the factors that impact software engineering workarounds. Using an empirical research survey instrument, this study aims to provide evidence on how technostress contributes to the development of workarounds and technical debts in the context of software engineering. The study further aims to evaluate the extent to which neutralization, autonomy, and perceived behavioral control contribute to the phenomena of workarounds. This can help understand and address the consequences of technostress and work to mitigate its impact on workarounds. As a result, the study poses the following four research questions:

RQ4: What is the impact of technostress on software engineers' intention to implement workarounds?

RQ5: To what extent does neutralization moderate the relationship

between technostress and engineers' intention to implement workarounds?

RQ6: What is the impact of strain resulting from technostress on engineers' intention to implement workarounds?

RQ7: To what extent does an engineer's level of autonomy and perceived behavioral control moderate the relationship between technostress and their intention to implement workarounds?

1.4 Research Philosophy and Methodology

The philosophical approach of this research is inspired by empiricism and pragmatism. The epistemological assumption of empiricism considers that knowledge comes primarily from sensory experience and emphasizes that empirical evidence is central to generating knowledge (Feigl & Scriven, 1956). Pragmatism, on the other hand, entails that knowledge can be viewed as a means to solve problems and, therefore, is evaluated in terms of its practical impact (Putnam, 1995; Saunders et al., 2019).

Scholars argue that the philosophical approaches of empiricism and pragmatism are complementary, which led to the emergence of the "pragmatic empiricism" paradigm, which is viewed to be more compatible with behavioral studies (Brotherston, 1943; Hempel, 1951; Newman, 1991). This complementarity can guide and add more value to the knowledge generated. In other words, as the new knowledge is supported by empirical evidence, the extent to which such knowledge is valuable depends on how impactful it is within the practical world.

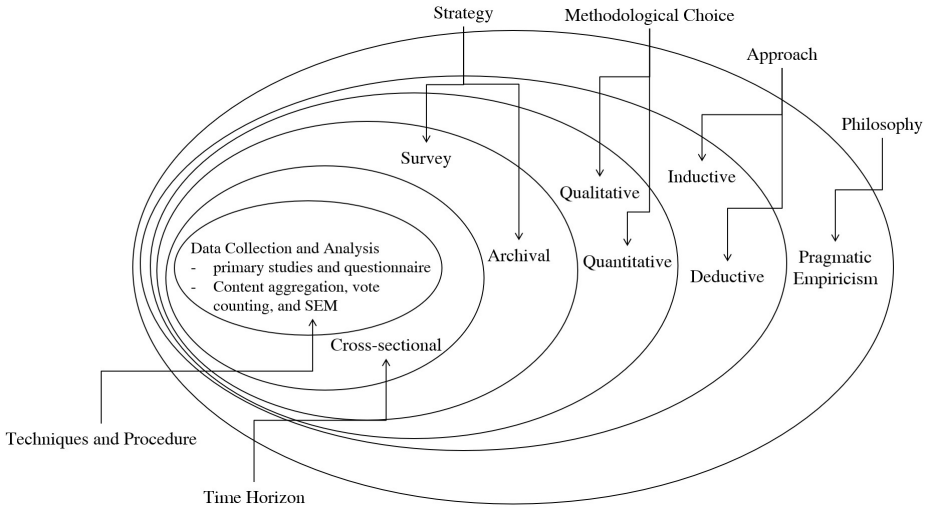

Figure 1. Philosophical Approach and Methodology of the Research

In this regard, this research follows the philosophical approach of pragmatic empiricism in that the first study adopts evidence-based thinking which is based on empiricism (Hjørland, 2011). The study uses an inductive approach to investigate the existing body of knowledge in order to analyze the focus of existing research, evolving concepts, and theories, and develop implications for further research. The study adapts the guideline proposed by Kitchenham, Budgen, and Brereton (2016) for conducting a systematic literature review. A detailed explanation of the methodology is provided in Chapter 2 (Section 2.3).

The second study follows a deductive approach that leverages existing theories along with practical indications to develop the study hypotheses; the relevance of the study is evaluated based on its practical consequences. The data collected for this study is cross-sectional survey-based using a questionnaire as an instrument to collect data from the target respondents. The data is analyzed using covariance-based (CB) and partial least square (PLS)

structural equation modeling (SEM). Accordingly, the results serve as empirical evidence that provides practical insights toward a better understanding of the workaround phenomenon in software engineering. The outcomes of the study are viewed from their practical impact. Fig.1 shows the overall philosophy and methodology adapted throughout this research.

1.5 Research Contribution

The two studies offer detailed analysis and investigation of software policy compliance with a focus on software engineering workarounds. The first study involved a detailed analysis of 84 selected studies identified based on the review protocol. The study adopts evidence-based thinking to investigate requirements, theories, factors, and policies in software compliance. The second study focuses on investigating workarounds in software engineering as one of the findings of the first study highlights the theory of workarounds as an emerging theory. The study uses a deductive quantitative approach and provides an extended explanation of the concepts and arguments in detail. Details are also given on the qualitative part of the proposed theoretical model, empirical data sample and procedure, detailed analysis of two different structural equation models, and the comparison conducted. A discussion is provided that elaborates and connects the key concepts and theories; implications are developed accordingly.

The study contributes to theoretical knowledge through the investigation of the factors that cause workaround behavior in the context of software engineering. It also extends the understanding of technostress, neutralization, professional autonomy, and perceived behavioral controls in the context of software engineering workarounds. In detail, the following are

the main theoretical contributions of this research.

First: This research extends the theory of workarounds with factors of technostress as antecedents to the workaround behavior and contextualizes the understanding of the workaround phenomenon in the field of software engineering. Previous studies argue that time pressure, misfit of work practices, complexity of technology, inadequate IT resources, and misunderstanding between work system stakeholders are the main causes of workarounds. No prior study considered the dimensions of technostress as antecedents that could lead to the development of workarounds, which is the main contribution that this study introduces.

Second: The study also contributes to the literature by providing an understanding of the role of neutralization, autonomy, and perceived behavioral control as moderators on the impact of technostress on workarounds. By evaluating the extent to which these moderators strengthen or weaken the impact of workaround behavior, the study adds to the knowledge base of empirical evidence on the moderating impact of these factors on the workaround phenomenon. This would help researchers consider moderating impact when further studying phenomena in other contexts or perhaps studying similar phenomena.

Third: The contribution of the study to the literature also comes by integrating theories of workarounds and planned behavior with technostress. In other words, the research evaluates the impact of technostress on workarounds from the lens of the theory of planned behavior as an overarching framework. This brings an understanding of the proposed theoretical model from the perspective of planned behavior while calling for

further evaluation from other theoretical lenses.

Fourth: The practical contribution of the study is that it helps practitioners and organizations consider such antecedents of workarounds and the consequences resulting from technostress as inputs in policy formulation and decisions related to software policy compliance. From a policy perspective, the findings of the study provide insights that can guide the setting up of policies which could help further understand the causes of workarounds in order to control their consequences.

Fifth: Additionally, the study provides practitioners with empirical evidence that can guide their decisions on the proper level of professional autonomy that should be given to engineers over technical decisions. A better understanding of the role of autonomy is crucial to develop a balance between responsibility and regulation based on empirical evidence. The study calls for increasing the level of attention given to investigating workarounds in software engineering and analyzing their causes and consequences as the cost of refactoring can be more expensive in the long run.

Sixth: The research further highlights recommendations for future research. These include: (1) The legal concern around end users of E-type software systems; (2) The gaps between compliance and domain experts and software engineers; (3) Compliance of business processes, accessibility, and usability in the context of software developers; (4) Additional exploration on antecedents of workaround phenomenon; (5) Extended application of the theory of planned behavior, namely the reasoned goal pursuit, in software compliance; (6) Distinctions between compliance policies of open source and proprietary software; (7) Supporting mechanisms for enforcing policies and

provide visibility to stakeholders concerned; (8) Compliance related to home-office users; (9) Supporting tools for compliance automation.

1.6 Research Outline

This research is structured as follows. Chapter 2 is the foundation of the research and it presents a systematic literature review on software compliance requirements, factors impacting, policies and challenges they address. Chapter 3 presents the empirical study which assesses the factors that impact software engineering workarounds including dimensions of technostress and the moderating role of neutralization, autonomy and perceived behavioral controls. Chapter 4 presents the discussion, implications, and the contribution and the limitations of the research. The following Figure 2 depicts the overall outline of the dissertation.

Structure	Section	Details
Chapter 1. Overall Introduction	Background	▪ Human is the weakest point in software compliance, accounting for over 50% of security breaches. ▪ Growing concern on technostress and its consequences. ▪ 25% of development efforts wasted due to workarounds.
	Problem Description	▪ Users do not take appropriate actions prescribed in information security policies. ▪ Software systems face a multifaceted set of compliance sources. ▪ Lack of research on workarounds, its causes and related impact.
	Research Objective	▪ Provide an understanding on current research focus, evolving concepts, and potential directions on software compliance requirements, policies, factors and their impact. ▪ Investigate the impact of technostress on software engineering workarounds, and the role of neutralization, professional autonomy and perceived behavioral control.
Chapter 2. a Systematic Literature Review on Software Compliance: Requirements, Policies, Factors and Impact	Research Questions	**RQ1.**What are software compliance requirements with respect to different industries and user contexts? **RQ2.** What are existing policies and which compliance challenges do they tend to address? **RQ3.** What are the factors that impact software compliance and which aspects of compliance are impacted?
	Methodology	Systematic literature review method adapted from Kitchenham et al. (2016): ▪ Analyze existing review works and develop search terms ▪ Establish review protocol (searching, screening, inclusion, analyzing and reporting) ▪ Execute the protocol and address the research questions ▪ Highlight research focus and potential gaps
	Findings	▪ Analysis identified 14 compliance requirements, 20 policies, and 66 factors. ▪ End user security is on top discussion followed by legal and privacy issues. ▪ Security awareness and automation of compliance are top cited policies. ▪ Emphasis on the gap between domain and compliance experts and software engineers. ▪ The theory of workarounds is emerging in the domain of compliance. ▪ Other evolving concepts are: compliance and privacy by design, policy as code, security stress, and home-office users.
Chapter 3. The Impact of Technostress on Software Engineering Workarounds and the Moderating Role of Neutralization, Perceived Behavioral Control, and Autonomy	Research Questions	**RQ4.** What is the impact of technostress on software engineers' workarounds? **RQ5.** To what extent does neutralization moderate the relationship between technostress and engineers' intention to implement workarounds? **RQ6.** What is the impact of strain resulting from technostress on engineers' intention to implement workarounds? **RQ7.** To what extent does engineers' level of autonomy and perceived behavioral control moderate the relationship between technostress and their intention to use workarounds?
	Methodology	Empirical study focusing on software engineers in South Korea ▪ Conduct a literature review on related work and citation network of core theories. ▪ Construct the theoretical development of hypotheses. ▪ Design, validate and translate the measurement instrument. ▪ Data Collection ▪ Use both CB-SEM and PLS-SEM for statistical analysis to test the hypotheses. ▪ Report findings and discuss the implications.
	Findings	▪ Technostress (complexity, overload, invasion) predicts workarounds through strain. ▪ Technology overload and technology invasion directly and indirectly impact workarounds. ▪ CB-SEM report significant moderation of autonomy and perceived behavioral control. ▪ Both CB-SEM and PLS-SEM found no moderating impact of neutralization
Chapter 4. Discussion and Conclusion	Discussion	▪ The human side of compliance is more complicated than the technological one. ▪ While SETA is important, more automation of compliance management is needed. ▪ Evidence indicate that technostress lead to development of workarounds. ▪ Professional autonomy plays a significant role towards development of workarounds.
	Implications	▪ The evolving concepts in the field reflect how important they are in the industry. ▪ There is an emphasis on the gap between domain and compliance experts on one side and software engineers on the other side. ▪ The impact of technostress can be threatful to organizations as it leads to workarounds. ▪ It is crucial to carefully consider the level of autonomy given to engineers.
	Research Contribution	▪ Position technostress as a new antecedent to workarounds. ▪ Extend understanding of technostress in the context of workarounds. ▪ Contribution also comes from evaluating the extent to which neutralization, autonomy and perceived behavioral control play a moderating role in the context of workarounds. ▪ Integrate theories of workarounds, planned behavior with technostress.

Figure 2. Research Outline

Chapter 2. A Systematic Literature Review on Software Compliance: Requirements, Policies, Factors and Impact

2.1 Introduction

Organizations and individuals counting on the reliability and resilience of software systems, as they should be able to have trustworthy technological infrastructures and complex software services (PricewaterhouseCoopers, 2021). The disruption and downtime of software systems can cause a significant loss to organizations. A research conducted by Ponemon Institute reveal that the mean costs of a data center outages are close to $650,000 (Ponemon Institute, 2016), not to mention the reputational damage and other resulting consequences. Insiders' behavior, whether malicious, non-malicious, negligent or compromised ones are regarded as growing risks, in that the cost of credentials' theft have increased to 65% in the last two years taking huge efforts and time to be contained (Proofpoint, 2022). These challenges can be seen from the perspective of technologies and humans. Although the technologies need a continuous checking and maintenance in order to ensure their adherence, the humans are considered the weakest and most vulnerable when it comes to compliance (Guhr et al., 2019). Indeed, prior research consistently report that organizational employees are accountable for more than 50% of security incidents (Balozian & Leidner, 2017). The global security report by PricewaterhouseCoopers (2018) revealed that recovering from security breaches takes around 19 hours on average. The report also shows that over 28% of organizations do not have an idea on the number of attacks they are experiencing. This report concluded that the lack of

awareness accounts for 48%, whereas 54% of respondents reported inexistence of clear incident response processes (PricewaterhouseCoopers, 2018).

IT systems are considered crucial assets to almost every organization, if they are not the core business of an organization. Hence, ensuring adherence with various requirements, industry standards, and security best practices is very challenging. The diversity of compliance sources makes management of software compliance more complicated. This complication can be justified by the highly volatile nature of software technology and the laws governing the evolution of software systems on one hand (Lehman, 1980), and the evolving sources of compliance (i.e., policies, regulations, security requirements and best practices) (Tran et al., 2012) on the other hand. Approaches that focus on technology side are not enough to secure organizational software systems. Studies report that end users of IT systems, mostly, do not take a proper action as prescribed in the security policies (Moody et al., 2018). In similar way, software engineers also lack the responsibility to delivering beyond just functionalities (e.g. implementation of privacy-by-design). As software systems and services can be developed either in house, deployed as a commercial off the shelf, outsourced to a third-party provider, or delivered as cloud services (Hale & Gamble, 2019), shedding the light on the requirements, factors, and policies related to their compliance is highly important. This study aims at providing an understanding on existing academic research foci, evolving issues and directions for potential research on software compliance requirements, determinants impacting, and the policies needed.

Existing review works focus mainly on certain industries or specific

aspects of compliance. No prior review work investigated the literature on compliance requirements, impacting factors, and policies that address different challenges in software compliance in a broader perspective. The importance of this research came from the growing issues on software security and insiders' threat, in addition to the challenges raised due to diverse compliance requirements. In these regards, it is worth investigating and bringing an understanding on existing research focuses and on evolving issues and directions of interest.

In detail, this study poses three research questions, which have been formulated and confirmed through analyzing existing review articles (Ali et al., 2021; Balozian & Leidner, 2017; Cram et al., 2017; D'Arcy & Herath, 2011; Hina & Dominic, 2020; Palanisamy et al., 2019; Trang & Brendel, 2019; Tsohou & Holtkamp, 2018; Zandesh et al., 2019). The results of the analysis, reported in Table 1, shows a lack of research on software compliance requirements, software compliance factors, and policies. The research questions are:

RQ1: What are the software compliance requirements with respect to different industries and user contexts? (Section 2.4.2).

RQ2: What are the factors that impact software compliance and which aspects of compliance are impacted? (Section 2.4.3).

RQ3: What are the existing software compliance policies and which compliance challenges do they tend to address? (Section 2.4.4).

The study adapts the systematic literature review (SLR) methodology of Kitchenham, Budgen and Brereton (2016) to collect evidences in order to answer the aforementioned research questions. The selected SLR method is an

appropriate method to conduct this type of research, because its steps provide explicit and reproduceable way to identify and synthesize existing body of research, while they also minimize biases and information overload. Additionally, this method is considered more suitable for reviewing studies related to software and information systems. In order to achieve that, we systematically searched the scientific databases to retrieve relevant studies. After that, we conducted a first step of eliminating articles, that are irrelevant through an initial screening, which reduces the number of papers to 484. Followed that, a thorough screening is applied with sets inclusion and exclusion criteria, in that 77 studies were identified to be the most relevant for the review. For further inclusion, automated citation analysis is conducted and 7 additional articles included, resulting in 84 research articles selected for the review.

The analysis of these 84 articles revealed 14 compliance requirements. Within the context of end users and software developers, security and legal issues are highly discussed. In addition, twenty policies were identified, and a list of compliance challenges they address was compiled. Since the majority of compliance violations and security breaches are the result of human behavior, security awareness is deemed essential for addressing numerous compliance challenges. Other highly discussed policies include the automation of compliance management, the improvement of organizational climate, and the creation of deterrence instruments. The review also identified 55 factors that influence various aspects of compliance with information systems policies. Individual aspects comprise the majority of these factors, followed by organizational and cultural aspects. In addition to theoretical and

practical implications, the study also suggests potential research directions.

The remaining sections of this paper are organized as follows: Section 2.2 provides a summary of relevant review articles. The methodology and review process are described in Section 2.3. The analysis of results and corresponding research questions are discussed in Section 2.4. In Section 2.5, key highlights of the review are elaborated upon, and implications and future directions are presented.

2.2 Related Work on Software Compliance

Several review studies pertinent to the research objective have been identified. Their research focuses primarily on a particular industry or compliance aspect. Based on what their research was about, we put them into five groups: information security (IS), the theoretical basis of IS, insiders' behavior, factors in a specific industry, and bring your own device (BYOD). The common denominator of these reviews is compliance with security compliance and human subjects. As the human subject is deemed more complex than technological ones, more emphasis is placed on the study of human behavior. Therefore, the level of importance of the aforementioned topics in the domain of software compliance, reflects in the attention that these categories gain in the review literature.

2.2.1 Information Security

Cram et al. (2017) analyzed policies related to organizational information security, and established a framework of five-set relationship. The relationships emphasize on policies design and implementation, the effect of security policies on organizations and their employees, the impact of

organizational and individual factors on policy compliance, the impact of policy compliance on organizational objectives, and changes to the design of policies. Another research of Balozian and Leidner (2017) focus on insider compliance with the policies of information systems. They established four topics as the foundation for indigenous information system security theory. These topics include philosophical management of information security, technical countermeasures, procedural countermeasures, and environmental countermeasures.

2.2.2 Theoretical Foundations of Information Security

Other set of review articles focus, primarily, on theory applications in the realm of information system security compliance. A study by Trang & Brendel (2019) investigate the application of deterrence theory on studies related to information security policy compliance. Their study concludes that, sanctions have influence on deviant behaviors in information security policy, and the deterrence theory predicts deviant behaviors better within malicious contexts, culture with higher degrees of power distance, and cultures with higher degrees of uncertainty avoidance. In a previous study, D'Arcy and Herath (2011) sought to investigate the disparities exist in the literature on information system (IS) deterrence. According to their research, the scientific knowledge on deterrence theory in IS security domain is still lacking. They also demonstrate inconsistencies and, in some other cases, contradicting conclusions of deterrence theory in IS security, resulting in the conclusion that procedures and policies are better directed by faith more by than facts.

2.2.3 Information Security Insider's Behavior

Ali et al. (2021) analyzed information security policy compliance and behaviors in order to determine the behavioral transition from non-compliance to compliance. They found a greater emphasis on the compliance behavior than non-compliance activities. Their research also discovered that value conflict, security stress, and neutralization techniques all contribute to disobedience, whereas internal/external and protective incentives also contribute to compliance behaviors. According to Ali et al. (2021), deterring strategies, management behavior, cultures, and information security knowledge all play important roles in shifting employee's non-compliance to compliance. In similar way, Tsohou and Holtkamp (2018), surveyed the competences linked with user's information security policy compliance behaviors. Their research establishes a set of competences related to information security policy compliance, and delivers evidence that show a lack of focus on information security duties.

2.2.4 Industry-Specific Compliance Factors

Another group of reviews, mainly, focus on a specific context or industry. Zandesh et al. (2019) investigate the determinants that properly shape the legal framework for cloud-based healthcare systems. Their research conceptualizes a framework which can be taken into consideration by the health-care sector before migrating to cloud-based services. The framework comprises 5 main components: compliance, data protections, ownership, identity credentials access management, and quality of services. In similar way, but within a different industry, Hina and Dominic (2020) investigate compliance to information security policy in high education institution. Their

study develops insights from theories and derives the factors, which have significant contribution on information security policy compliance. They conclude that, the awareness on information security policy compliance and the follow up processes should be the first and the most important towards establishing a better information systems security. Their study also revealed that the end users are, typically, not aware of effectiveness of response, and hence, remain vulnerable to attacks most of the time. Hina and Dominic (2020) argue that, workers of high education sector are found the least cautious and aware of the potential risks, that might jeopardize their professional and personal computing environments.

2.2.5 Bring Your Own Device (BYOD)

Lastly, to evaluate the security risks and compliance challenges related to policies of bring your own device (BYOD) to workplaces, Palanisamy et al. (2019) analyze the security risks, issues which are posed by employees' noncompliance to security policies, and strategies that could reduce the risks associated with BYOD. The study revealed and found a lack of emphasis on the social factors that such polices consider within organizations. Furthermore, the surrounding social environment, can also influence employee's compliance related decisions with security policy. Palanisamy et al. (2019) concluded that the existing research on the security policy compliance and efficacy of policies related to BYOD is lacking.

2.2.6 Comparisons

Although related review studies tackle specific area of (non)compliance, for example, the security policy or insider's behaviors; or emphasize, mainly,

on particular contexts or theories (Table.1), there is no previous review study that analyzes existing body of scholarly research on software compliance requirements, factors and their scope of impact on different aspects of compliance, and policies along with their addressed challenges.

Table 1. Summary of related reviews

Study	Focus	Compliance Requirements	Specific Requirement on Information Security	Specific Industry	Theory	Factors	Policies
Cram et al. [8]	Organizational IS		✓				
Balozian & Leidner [6]	Indigenous IS security theory (Insider behavior)				✓	✓	
Trang and Brendel [9]	Deterrence theory		✓		✓		
D'Arcy and Herath [12]	Deterrence theory				✓		
Ali et al. [13]	Behavioral transformation (Insider behavior)		✓			✓	
Tsohou and Holtkamp [14]	Security compliance (Insider behavior)					✓	
Zandesh et al. [15]	Healthcare in the cloud			✓			
Hina and Dominic [16]	Higher education institutions		✓	✓			
Palanisamy et al. [17]	BYOD		✓				
This Study	Requirements, Policies and Impacting Factors	✓				✓	✓

Previous reviews provide very valuable understandings on topics focusing on information security, applications of theories, insider's behavior issues, and context-specific issues; however, we found a lack of studies that

provide a wider analysis of compliance requirements, with regard to their associated industries and concerned stakeholders; policies and their addressed challenges; and the factors impacting as well as their degree and scopes of impacts. Such an inquiry can provide insight into current research priorities, theories in use, emerging notions, and future research areas.

2.3 Methodology

This study adapts Kitchenham et al. (2016) methodology in order to conduct the systematic literature review (SLR). The philosophical underpinning the adapted methodology is empiricism and the use of evidence-based thinking to identify and synthesize the existing body of scholarly research. The reason for selecting this method is that, it provides guidance for the steps that can explicitly executed for identification and synthesis of relevant literature from the existing body of research in a replicable way, while it helps minimize biases and information overloads. Moreover, the selected method is argued to be suitable for reviewing studies which are related with software and information systems (Kitchenham et al., 2016).

The search keywords which are selected for retrieving the relevant primary research articles are: ("*software compliance*") OR ("*compliance of software*") OR (*compliance* AND "*information systems*") OR (*compliance* AND "*distributed systems*") OR (compliance AND "*software systems*") OR (*compliance* AND "*service-oriented systems*"). By selecting the aforementioned keyword combination, we argue that these terminologies cover different alternatives terms related to software compliance. In addition to that, we added the asterisk symbols in at the beginning or ending of some of these terms to include both singular and plural, as well as, in the beginning

of some others to include their opposite words. For instance, (*compliance) and (system*) can retrieve results containing “compliance” and “noncompliance” articles, and the same goes for “system” and “systems”. Having retrieved a total of 8,203 articles based on titles, abstracts and keywords, we can say that these keyword combinations represent the topic sufficiently. On the other hand, a more relaxed query than this would produce huge amount of results and complicate reduction process while at the same time impacting the reproducibility of the review.

One of the challenges in software engineering research is that, the field lacks strong taxonomy compared to other fields (Kitchenham, Budgen, Brereton, et al., 2016). Accordingly, further customization of the search query is developed in accordance with the syntax used by the corresponding scholarly databases. This can, in turn, help retrieving as many articles as possible, while eliminating the chances of missing articles which are relevant for the review. Table.1 in the appendix present the developed search queries to execute along with their correspondent scholarly databases. The databases selected for retrieving primary studies are “Google Scholar”, “ScienceDirect”, “Scopus”, “Web of Science”, “ACM Digital Library”, and “IEEE Xplore”.

Figure 2 depicts the followed steps that this study executes. Step 1 represent the starting point which is setting up the review objective. Based on the review objective, a set of search terms and keywords were formulated in order to construct the search query (Step 2). After that, the search queries were executed in order to retrieve the review studies only (Step 3), that are related to the objective of this review. This would contribute to building the research foundation and linking the related findings. It also helps ensure that

the derived research questions have not been answered by existing research. After the analysis of the retrieved review studies (Step 4), we developed the review research questions and made some improvements on the search keywords accordingly (Step 5).

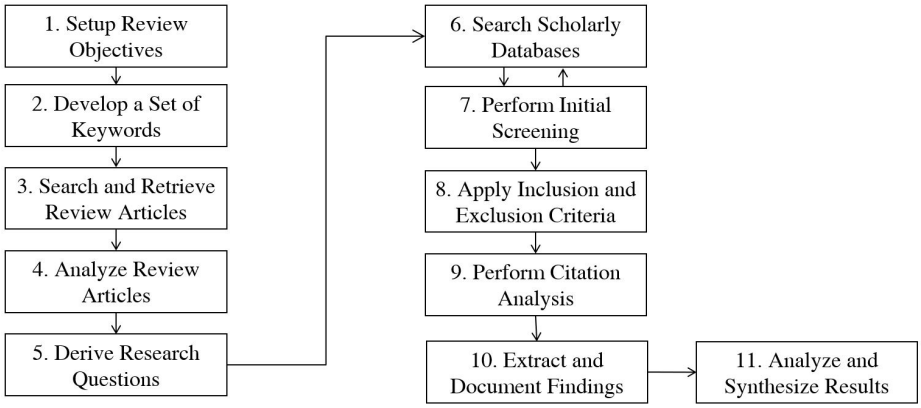

Figure 3. Steps Followed to Conduct the Review

In the following steps (Step 6 of Figure.3; step 1 of Figure. 4), the search queries were executed at the aforementioned scholarly databases, for retrieving the relevant primary studies. An explanation, in detail, about this step is presented in Figure.4 (Steps 1 to 3), including numbers of retrieved articles and the reduction process, based on the specified periods in the criteria. This step is executed parallelly along with the initial screening performed on the retrieved results (Step 7 of Figure 3; Step 4 of Figure 4), in that we checked the title and the abstract of the retrieved studies, in line with their relevancy to the research questions of the review. This filter of initial screening had reduced the number of articles to 484, after removal of duplicated ones (Step 5 of Figure 4).

Once the initial screening of the results has finished, a strict set of criteria for inclusion and exclusion, were applied as a second level of filtering

(Step 8), for reducing number of candidate research articles to high relevant, more focused, as well as, manageable number of studies to consider for the review. Furthermore, an automated search is performed using citation analysis for extra inclusion (step 9 of Figure.3; step 7 of Figure.4) to enhance the review. The criteria for inclusion and exclusion are as follow:

Inclusion criteria:

1. Peer-reviewed studies which are published in international outlets. This criterion ensures the scientific quality of candidate studies, in order to build on a reliable evidence.
2. Full research works are considered, in that contributions of their work are tested and evaluated clearly.
3. Research articles which are published in the period between 2011 and 2021. This period is set due to highly dynamic nature of software technologies, in that, taking into consideration recent studies is important, in order to have an emphasis on the contemporary settings.
4. Hight Relevancy to the review objectives. This means, the study addresses one of the review questions, at least, to become eligible for inclusion.
5. Studies, which are between 2011 and 2016, should have a number of citations more than or equal 30. This is to give more emphasis on the highly influential studies, that are published within that period.

Exclusion criteria:

1. The articles in which “Software Compliance” is marginally discusses, not the main discussion, are excluded. If a study objective is not related to compliance of E-type software.

2. Articles which are written in languages other than English. This is due to limitations on the access and interpretation of research articles, which are written in other languages.
3. Book chapters reports, posters, and presentation materials, are excluded, because they, typically, tend to discuss a wider perspective, and some of these materials are not, usually, reviewed scientifically.
4. Studies, which are published in the period between 2011 and 2016, and have less than 30 citations, are excluded. This is because, the articles, which are published within that period, are a bit old, and in this regard, the citations number serves as another metric that can indicate how influential an article is.
5. Articles that discuss compliance of non-E-Type (i.e. S-Type and P-Type) software are excluded.

The results obtained after applying the criteria of inclusion and exclusion; and citation analysis, are 84 primary research articles. Following that step, is data extraction (Step 10 of Figure 4), with which we use Zotero referencing tool (version 5), to manage, document, and organize the references of the retrieved primary studies. Finally, for the analysis, we use keyword co-occurrence, vote counting techniques and content aggregation. These aforementioned steps rigorously present a systematic way for reproducing the literature review, while sufficiently address the RQs (Step 11 of Figure 4).

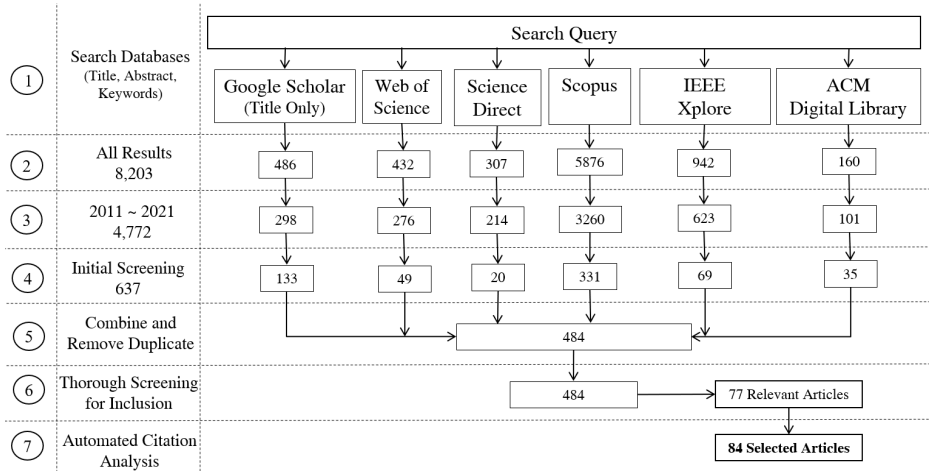

Figure 4. Steps Executed to Reduce and Select Relevant Articles for the Review

For comprehensively enhancing the review, the select studies are further fed into citation analysis in order to conduct an automated search on the studies which cite the selected studies. To achieve this, we use citatioGecko[1] tool for a better visualized search result (Figure 5). This process identified additional seven relevant articles included in the review.

[1] http://citationgecko.com

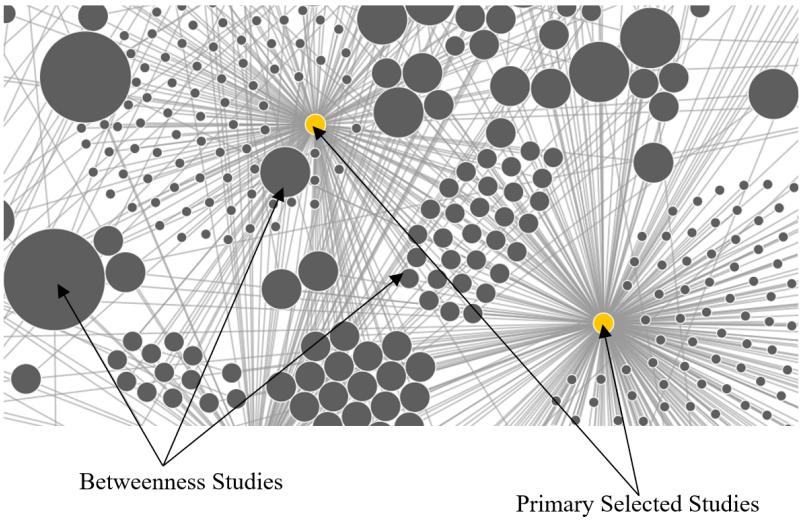

Figure 5. Citation Analysis of Selected Studies for Further Inclusion

2.4 Analysis of Results

2.4.1 Descriptive Analysis

We performed an analysis of co-occurrence of keywords for the 84 selected papers, through VOSViewer (v. 1.6) to assess the keywords and their clusters, which are highly discussed in this domain. We set the thresholds of keyword co-occurrence to twice occurrences. The word co-occurrence indicates the keywords which have been mentioned in the keyword list of one or more articles. The VOSViewer found 54 keywords, which appeare twice, at least, in the articles' keywords lists. This formed 5 clusters, as shown in Figure 5, formulated based on the builtin techniques for clustering, which is introduced by van Eck & Waltman (2018). The topics of discussion of each of these clusters are presented down the legends of the Figure 6.

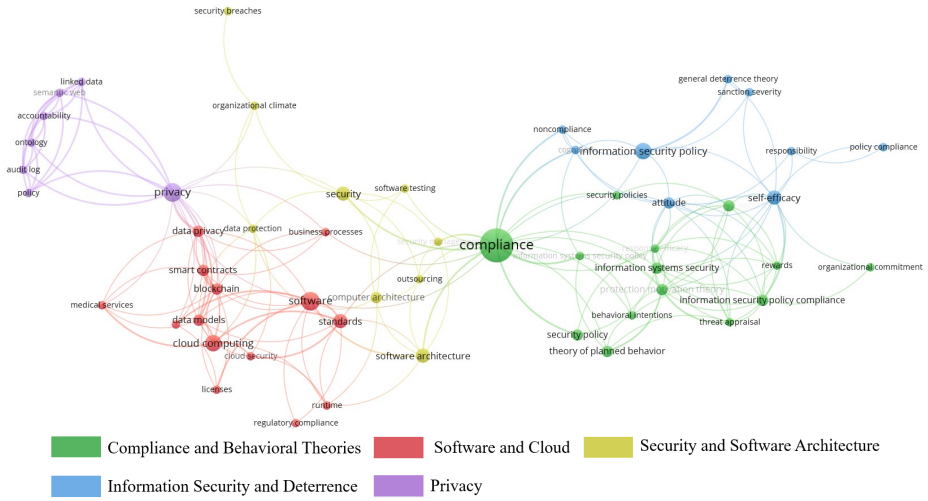

Figure 6. Analysis of Keyword Co-occurrence (Results from VOSViewer)

The analysis of keyword co-occurrence also presents the total strength of links of those keywords. The strength of the link indicates the number of studies, within which a combination of two or more keywords occurred altogether (van Eck & Waltman, 2018). The co-occurred top 10 keywords along with their total strength of links are depicted in Figure 7. These are: “Compliance”, “Software”, “Information Security Policy”, “Self-Efficacy”, “Cloud Computing”, “Privacy”, “Standards”, “Software Architecture”, “Protection Motivation Theory”, and “Attitude”. The highly occurred and strong link of those keywords represent their degree of importance in software compliance.

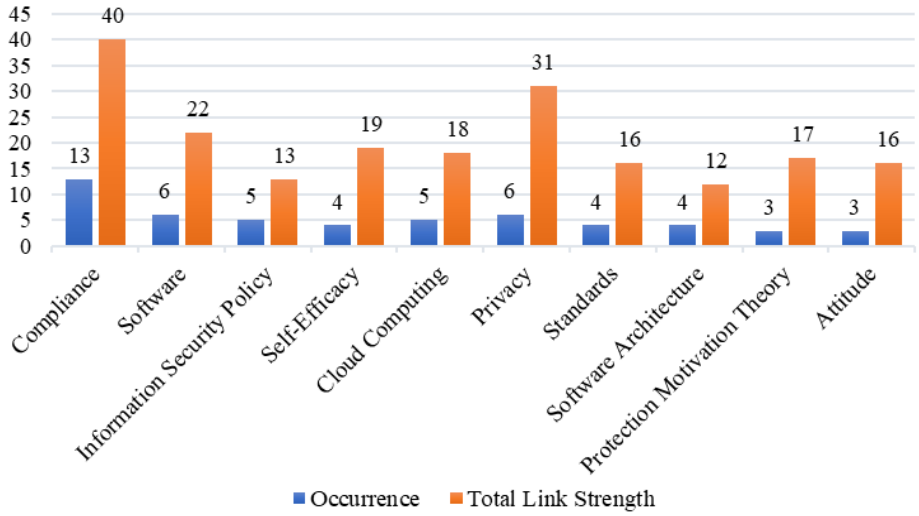

Figure 7. Co-occurred Top 10 Keywords and their Total Link Strength

The following Table.2 presents the publishers and the corresponding numbers of primary studies selected with regard to their type of publication. The overall candidate studies, are 65 journal articles, 16 conference articles, and 3 workshop articles.

Table 2. Publication Database and Selected Studies

Publishers	**Total**	**Journals**	**Conferences**	**Workshops**
ACM Digital Library	3	-	3	-
Association of Information Systems	4	1	3	-
Atlantis Press	1	1	-	-
Elsevier	20	20	-	-
Emerald	8	8	-	-
Hindawi	1	1	-	-
IEEE Xplore	13	5	6	2
MDPI	4	4	-	-
ProQuest	1	-	1	-

Publishers	Total	Journals	Conferences	Workshops
SAGE	3	3	-	-
Springer	11	10	1	-
Taylor & Frances	7	7	-	-
Wiley Online Library	2	2	-	-
World Scientific	1	1	-	-
Other	5	2	2	1
Total	84	65	16	3

The distribution of selected studies based on their country is presented in Figure 8. While, 36 articles have not specified the countries, within which these studies hve been conducted, the 38 remaining articles clearly discuss the country of the study. The legends colored in the map of Figure 8, represent the number of articles with regards to their respective countries. In other words, the legend in dark-blue color, indicates the countries which have 8 articles conducted. The United States is on top of these countries, followed by China, Canada, and Malaysia.

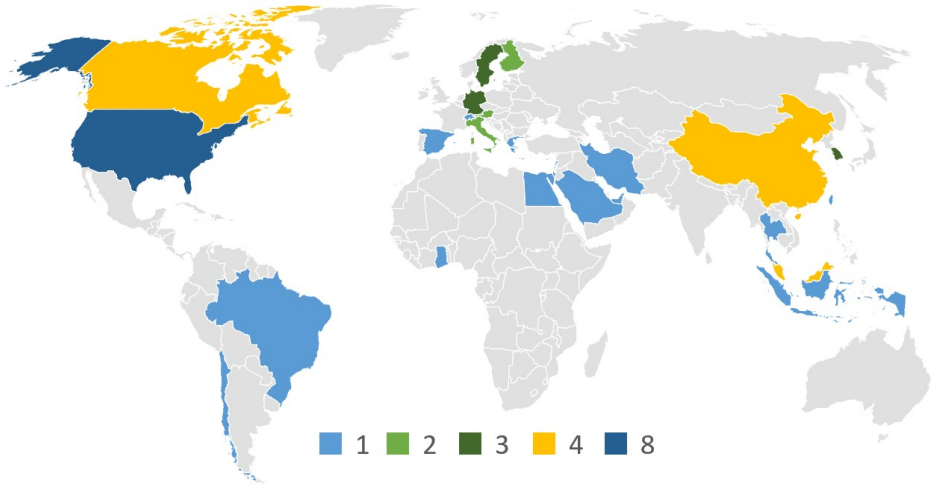

Figure 8. Countries and Number of Studies Conducted

2.4.2 Compliance Requirements and Related Industries and User-Contexts

Regarding the first research question, the analysis of our review extracted the compliance requirements at the category level, and present them in bubble plots graph. This is, in order to visualize the mapping of those requirements along with their applicable industries and user contexts, in a simple way. One of the challenges in analysis is the heterogeneity of primary studies, specifically for the systems hosted in the cloud. For example, some articles discuss healthcare systems in the cloud, or education systems in the cloud. In this case, our analysis classifies that to their main industry which is healthcare of education, not cloud. The reason is that the main discussion focuses on that industry and their compliance challenges or requirements; not cloud related requirements.

The following is a brief description of these requirements:

Accessibility, according to W3C (2022), *Accessibility* means that the system does not have difficulties that exclude a certain group of people, for

example people with disabilities, from using the system.

Software Architecture, of a software refers to its structure, elements and the relationships among them (Czepa et al., 2017).

Software Auditing, the software auditing indicates an external or internal review of its quality, compliance to documented needs, standards and regulations (Julisch et al., 2011).

Business Process, is defined as an activity or collection of activities which accomplish a specific business goals (Alter, 2015).

Continuous Delivery is an approach in which developers produce a software in short cycle in a continuous way ensuring that releases are reliable and well tested (Humble & Farley, 2010)

Software Licensing refers to the legal agreement that restrict use or redistribution of a software component providing rights and terms of use (Sojer et al., 2014).

Privacy is the extent to which a system protects and provides a control over a user's personally identifiable information (Barati et al., 2020)**.**

Quality of Service in general indicates the overall performance of a system with regard to the expected or promised performance (Tran et al., 2012)**.**

Law is a collection of rules established and enforced through certain government institutions in order to regulate behaviors (Willis, 1925).

Software Safety refers to the level through which safety measures, including identification and analysis of risks, are in place and can be controlled (Roland & Moriarty, 1991).

Security is a concept indicates the implementation of mechanisms in

order to make it remain functional and resist against any attack including cyber-attacks (C. Joshi & Singh, 2017).

Transparency and ***Trust*** can be considered from an end user as well as a service provider's perspective, both requirements are associated with values of accountability (Majumdar et al., 2018; Singi, Kaulgud, et al., 2019)

Usability is all about the design of a software product to be effective, efficient, and satisfying. This also include user experience design and other aspects that impact people with diabilities (W3C, 2022).

According to the analysis, the security requirements are on the top of discussion in many industries including health-care (Dong et al., 2021; Humaidi & Balakrishnan, 2017; Karlsson et al., 2017; Kolkowska et al., 2017; Kuo et al., 2021; T. Alanazi et al., 2020), finance (Carmi & Bouhnik, 2020; Y. Chen et al., 2012; Jeon et al., 2020; Merhi & Ahluwalia, 2019; Rongrat & Senivongse, 2018; Westland, 2020), education (Bansal et al., 2020; X. Chen et al., 2018; Hina et al., 2019; Merhi & Ahluwalia, 2019; Wiafe et al., 2020), software (Hale & Gamble, 2019; Thalmann et al., 2014; Truong & Nguyen, 2013; Varela-Vaca et al., 2019), government (Choi & Song, 2018; Jeon et al., 2020; Liu, Wang, Wang, et al., 2020), energy (Ali et al., 2020; S. S. Kim & Kim, 2017), IT (Jeon et al., 2020; Merhi & Ahluwalia, 2019), manufacturing (Jeon et al., 2020), retail (Merhi & Ahluwalia, 2019), and cloud industry (Majumdar et al., 2018).

Legal requirements is the secondly highly discussed topic in health-care (Granlund et al., 2020; Ingolfo et al., 2013; Li et al., 2020; Maxwell et al., 2013; Mohamed et al., 2021), software (Islam et al., 2011), finance (Maxwell et al., 2013) cloud (Joshi et al., 2020) and telecommunication (Usman et al.,

2020). Privacy requirements are discussed in the following industries: health-care (Diamantopoulou & Mouratidis, 2019; Eze et al., 2018; Samavi & Consens, 2018), software (Antignac et al., 2018; Bednar et al., 2019), government (Diamantopoulou & Mouratidis, 2019), and cloud industry (Eze et al., 2018). Licensing requirements are found connected to the software industry (Gangadharan et al., 2012; Moquin & Wakefield, 2016; Sojer et al., 2014); while auditing is discussed in financial (Julisch et al., 2011; Westland, 2020) and health-care (Wickramage et al., 2019). Safety requirements are found related the aviation (Castellanos-Ardila et al., 2021; Marques & da Cunha, 2018) and automobile industries (Antinyan & Sandgren, 2021; Chitnis et al., 2017). Accessibility requirements are discussed in the context of government (Oliveira et al., 2020), health-care (Montazeri et al., 2020) and education sector (Máñez-Carvajal et al., 2021). Figure 9 presents the identified industries along with their associated compliance requirements.

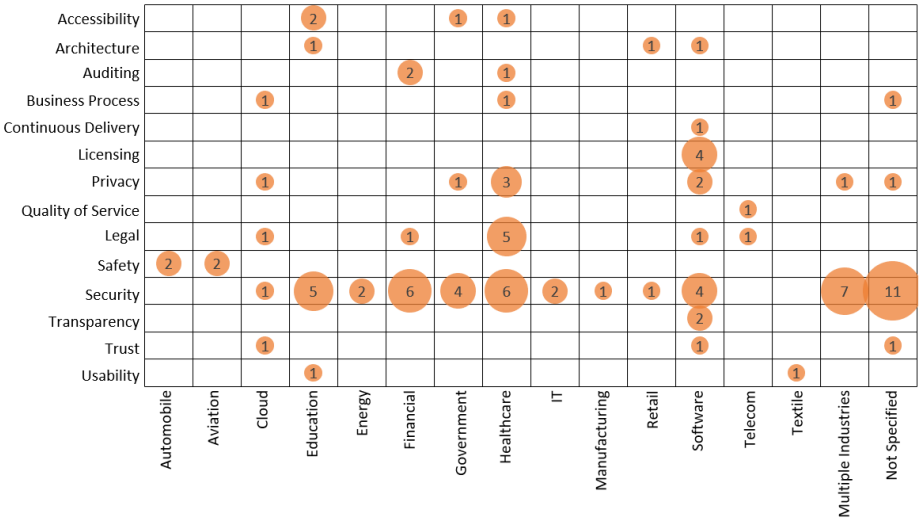

Figure 9. Primary Studies and their Industries and Compliance Requirements

Based on the analysis, different compliance requirements can indicate different levels of standing to their related industries. Nonetheless, the security requirements are critical to most of the industries. There is a less focus on the software compliance requirements including: business process, continuous delivery, trust, quality of service, and transparency. Regarding the least studied industries, among the selected studies are manufacturing, retail and textile.

Regarding compliance requirements, and their respective users' contexts, the majority of the studies discuss issues related to security compliance of software end users (Ali et al., 2020; Balozian et al., 2021; Bansal et al., 2020; Burns et al., 2018; Carmi & Bouhnik, 2020; X. Chen et al., 2018; Y. Chen et al., 2012; Choi & Song, 2018; D'Arcy et al., 2014; Dong et al., 2021; Faizi & Rahman, 2020; Guan & Hsu, 2020; Guhr et al., 2019; Hina et al., 2019; Humaidi & Balakrishnan, 2017; Jeon et al., 2020; Karjalainen et al., 2020; Karlsson et al., 2017; S. S. Kim & Kim, 2017; Kolkowska et al., 2017; Kuo et al., 2021; Liu, Wang, & Liang, 2020; Liu, Wang, Wang, et al., 2020, 2020; Majumdar et al., 2018; Merhi & Ahluwalia, 2019; Ormond et al., 2019; Putri & Hovav, 2014; Stafford et al., 2018; T. Alanazi et al., 2020; Van Slyke & Belanger, 2020; Wiafe et al., 2020). End users are also found related to other requirements, including: privacy (Barati et al., 2020; Diamantopoulou & Mouratidis, 2019; Samavi & Consens, 2018), accessibility (Máñez-Carvajal et al., 2021; Montazeri et al., 2020; Oliveira et al., 2020), usability (Davison et al., 2019) and licensing (Moquin & Wakefield, 2016).

Developers are on the second top discussion, and are found to have more concerns with security (Hale & Gamble, 2019; Rongrat & Senivongse,

2018; Truong & Nguyen, 2013; Varela-Vaca et al., 2019), legal requirements (Islam et al., 2011; Li et al., 2020; Maxwell et al., 2013; Usman et al., 2020), safety requirements (Antinyan & Sandgren, 2021; Castellanos-Ardila et al., 2021; Chitnis et al., 2017; Marques & da Cunha, 2018), licensing (Gangadharan et al., 2012; Sojer et al., 2014), software architecture (Czepa et al., 2017; Silva et al., 2020), privacy (Eze et al., 2018), transparency and trust (Singi et al., 2019).

Auditors, architects, and managers are less discussed in comparison to end users and software developers. Managers are found related to security (Ifinedo, 2012, 2014, 2016) and legal requirements (Usman et al., 2020). Auditors are also discussed in the relation to security (Thalmann et al., 2014; Westland, 2020) and auditing (Julisch et al., 2011; Westland, 2020). Software architects are found related to architectural (Czepa et al., 2017; Silva et al., 2020), auditing (Julisch et al., 2011) and privacy requirements (Antignac et al., 2018). The analysis also represents other stakeholders; however, they are discussed to a less extent in the selected studies, see Figure 10.

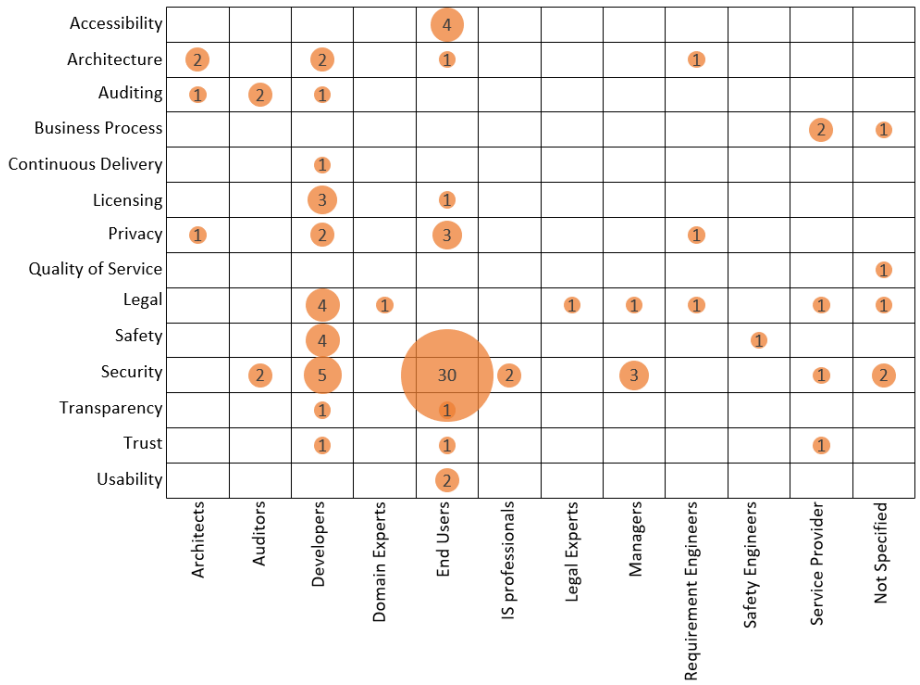

Figure 10. Primary Studies with Type of Users and their Compliance Requirements

Based on the analysis of selected articles in this review, we found higher concerns regarding the security of end users, than other stakeholders. Market studies show that over 50% of security breaches caused by end users. This explains the huge research focus on security issues of end users compared to other compliance requirements. While most requirements address end users of a software, as well as, developers, in which both are considered in the forefront of development or use of software systems, there is less attention paid on other stakeholders including domain experts, legal experts, and safety engineers.

Business managers on the other hand show unexpected results with regard to business processes as they are more concerned with addressing compliance related to business problems. However, studies emphasize that

business managers are mostly outcome oriented and focus more on productivity; moreover they can also involve in noncompliance activities whenever they realize that they obstruct productivity and business outcomes are negatively impacted (R. M. Davison et al., 2021).

2.4.3 Factors Impacting Software Compliance

The base concepts and theories, in which the primary selected studies are using, are presented in Figure 11 where x-axis is the number of publications and the y-axis is the theory/concept. The importance of highlighting these concepts and theories is, because they represent a the underlaying understanding of their corresponding domain. In other words, these concepts and theories are helpful in delivering deep interpretations and explanations of the determinants which have been found in the review. This can, in turn, allow controlling and addressing those factors, in a better way while designing software compliance related policies. The following is a brief explanation of these concepts and theories:

Theory of planned behavior: The theory explain that an individual decision to pursue a certain behavior is defined based on a combined set observations; these include attitude, intentions, subjective norms, and perceived behavioral controls (Ajzen, 1991). The theory is used the most among the surveyed studies; which indicates its usefulness in predicting compliance behavior.

Deterrence theory: It refers to the extent to which practices of using a threat or other force by one party can convince the other one to refrain from initiating a behavior or a course of action (Jervis, 1979). While the theory is predominant in military, it is also applicable in software policy compliance

where an insider considers the cost of violation.

Requirement engineering: is a process through which requirements are defined, documented, and maintained; it is commonly a role in software engineering (Chemuturi, 2012). While activities that are involved in requirement engineering can vary depending on the type of system developed, compliance is seen crucial on all activities of this process (Antinyan & Sandgren, 2021; Granlund et al., 2020; Ingolfo et al., 2013; Islam et al., 2011; Marques & da Cunha, 2018; Maxwell et al., 2013; Steffens et al., 2018; Usman et al., 2020; Wickramage et al., 2019).

Protection motivation theory: an individual response to fear appeals is explained by the protection motivation theory. The theory proposes that people tend to protect themselves based on two factors: threat appraisal, which is one's assessment of severity of a situation, and coping appraisal, which is one's response to that threat (Rogers & Prentice-Dunn, 1997). The theory is third most used among the selected studies; it justifies more on an individual engagement in unhealthy practices in which compliance can be of a great deal.

Privacy-by-design: is an approach that calls for privacy to be considered throughout all the software engineering process. The main objective is to take proactive measure and embed privacy into the design. The concept has been incorporated in the European general data protection regulations (GDPR) (Antignac et al., 2018; Barati et al., 2020; Bednar et al., 2019; Diamantopoulou & Mouratidis, 2019).

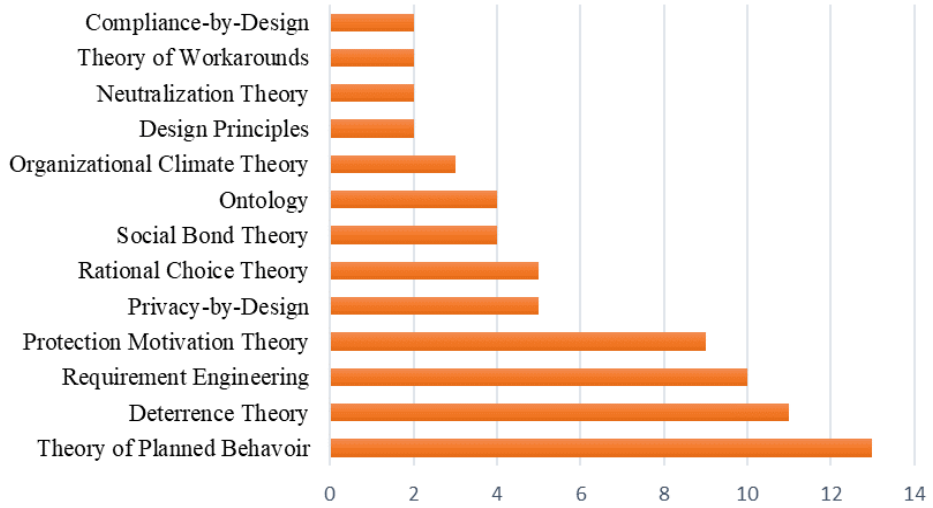

Figure 11. Top Foundational Theories and Concepts Used by the Primary Studies

Rational choice theory: the theory argues that an individual uses a rational calculations based on cost-benefit analysis in order to make rational choices and purse certain outcomes which are in line with their own personal objectives (Boudon, 2003; Carmi & Bouhnik, 2020; Ifinedo, 2016; Stafford et al., 2018; T. Alanazi et al., 2020).

Social bond theory: explaining what prevents people from violating norms and achieve social control is their degree of attachment, commitment, involvement and beliefs (Ali et al., 2020; Choi & Song, 2018; Dong et al., 2021; Ifinedo, 2014).

Ontology: ontologies describe the basic concepts that exist in a certain field and how they are connected and formed; for example the legal ontology can explain legal acts and the connections between them (Hale & Gamble, 2019; K. P. Joshi et al., 2020; Samavi & Consens, 2018).

Design principles: a set of considerations which form the basis guiding a team towards making appropriate decisions and developing a good product

(Máñez-Carvajal et al., 2021; Montazeri et al., 2020).

Neutralization theory: refers to justification of deviant behavior and violation of policy using several strategies including: *denial of a responsibility, denial of an injury, denial of a victim, condemnation of a condemner, appeal to a high loyalty, defense of a necessity*, and *defense of a ubiquity* (Bansal et al., 2020; Coleman, 1987; S. H. Kim et al., 2014; Minor, 1981; Sykes & Matza, 1957).

Organizational climate theory: this theory argues that the environment in an organization has a strong influence on the perception and the behavior of employees (Dong et al., 2021; Ifinedo, 2016).

Theory of workarounds: the misunderstanding of management intentions, designers' intentions and participants' goals, interests and values can lead to development of workarounds, which are adaptations made to overcome work-related obstacles resulted from that misunderstanding or a way to bypass policies to gain short term benefits (Alter, 2015; R. Davison et al., 2019).

Compliance-by-design: a systematic approach applied through embedding and integration of regulatory and policy requirements at the design stage. The approach help control human errors and position compliance at the center (Castellanos-Ardila et al., 2021; Julisch et al., 2011).

For answering the RQ2, the analysis of the reviewed articles, found that the factors identified, primarily impact three behavioral aspects: compliance attitude, compliance intention, and compliance behavior. Having this taxonomy of attitude, intention, and behavior is mainly inspired by the theory of planned behavior of Ajzen (1991). Despite the arguments in literature that

attitudes and intentions are likely good predictors of a certain behavior, they might not lead to the actual compliance behavior, in all circumstances (Ajzen & Kruglanski, 2019). Accordingly, we consider the three aspects of the planned behavior, as an overarching lens that contains the identified factors and their impact as confirmed by the primary studies.

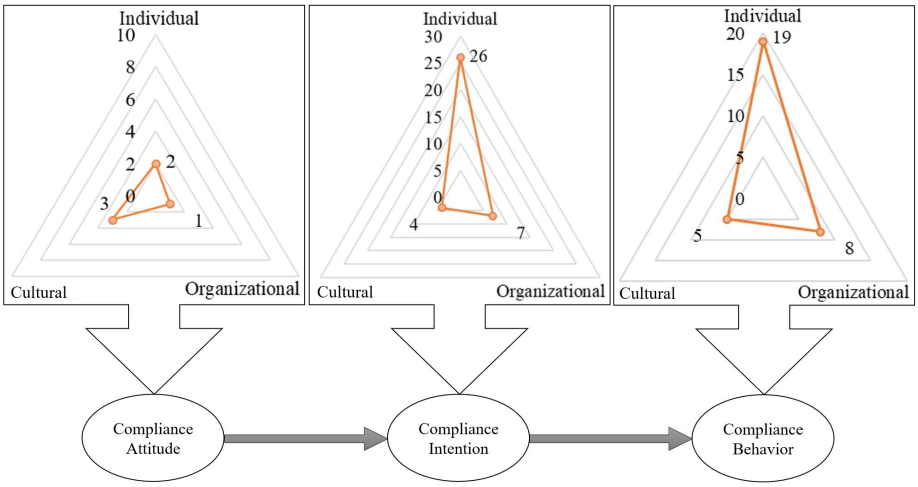

Figure 12. Scope and degrees of impact for the identified factors

In addition to that, we added the scopes of the impact for each of the identified factors (individual, cultural, and organizational). This is to help providing more understanding on the contexts, in which the factors are impacting. For simplifying their visualization, we use vote counting technique, in order to synthesize and present the result (Kitchenham, Budgen, Brereton, et al., 2016).

Figure 12 shows a high-level representation of impact scopes (individual, cultural and organizational), along with their degrees of impacts on the compliance attitudes, intentions, and behaviors). The attitudes toward compliance are more impacted by the factors related to cultures, indicating that more emphasis have to be given to those cultural factors, as they

contribute toward shaping insiders' compliance attitude. In comparison to factors related to cultures factors, the organizational factors show less influence on the attitude towards compliance. The compliance intentions and the compliance behaviors were found more impacted by individual-related factors, while organizational-related factors came right after.

The review analysis identifies 66 factors, which have been empirically tested by the selected primary studies. Among these factors, 11 are found insignificantly impacting the compliance and, hence, we excluded them from our analysis. These are: gender, social pressure, fear, behavioral controls, technical countermeasures, passive/avoidant leaderships, transactional leaderships, differential reinforcement, information security policies, detection probabilities, and the efficacy of measures. The remaining factors, 55, are found confirmed by the selected studies to be impacting, either positively or negatively (Table 3).

Table 3. Factors Influencing Behavioral Compliance and their Scopes of Impact
+ positive impact, - negative impact, Ø insignificant impact., Occurrences indicates number of studies cited.

	Factor	Theory	Scopes of Impact			Impacted Aspects		
			Individual	Cultural	Organizational	Compliance Attitude	Compliance Intention	Compliance Behavior
1	Abusive Supervisions	-			✓		+	
2	Punishments	Deterrence			✓		+ + + +	+
3	Reward	Compliance			✓		+ Ø Ø	
4	Certainty of Controls	Deterrence			✓		+	
5	Security Stress	Technostress	✓				-	
6	Moral Disengagements	Moral Disengagement	✓				-	
7	Top Management Support and Belief	Organizational Climate			✓			+
8	Cost Benefit Analysis	Rational choice	✓					-
9	Sanction	Deterrence			✓	+	Ø	+ +
10	Self-Efficacy	Protection Motivation	✓				+ + + + + + Ø	+ + +
11	Descriptive Norms	Social Norms		✓		+	+	+
12	Differential Association	Social Learning		✓				-

	Factor	Theory	Scopes of Impact			Impacted Aspects		
			Individual	Cultural	Organizational	Compliance Attitude	Compliance Intention	Compliance Behavior
13	Imitations	Social Learning	✓					-
14	Moral Norms	Planned Behavior		✓				+
15	Security Valences	Expectancy	✓				+	
16	Security Instrumentalities	Expectancy	✓				+	
17	Security Expectancy	Expectancy	✓				+	
18	Transformational Leaderships	Full-range Leadership			✓		+	
19	Procedural Countermeasure	Intellectual capital cyber security			✓		+	
20	Socio Cultural Environments	Information Systems Security		✓			+	
21	Neutralization	Neutralization	✓				- -	
22	Attitudes Toward Compliance	Planned Behavior	✓				+ + + + + + + +	+ + +
23	Normative Beliefs	Planned Behavior		✓			+	+
24	Response Efficacy	Protection Motivation	✓				+ + + Ø Ø	+
25	Perception of Compliance Benefit	Rational Choice	✓				+	+
26	Perception of Compliance Cost	Rational Choice	✓				- - Ø	- - Ø
27	Perception of Noncompliance Cost	Rational Choice	✓				+	+
28	Attachment	Social Bond	✓				+	
29	Commitment	Social Bond	✓				+	+
30	Involvement	Social Bond	✓				+	
31	Personal Norms	Planned Behavior	✓			+	+	
32	Perceived Trust	-	✓					+
33	Compliance Behavioral Beliefs	Planned Behavior	✓				+ +	
34	Compliance Knowledge	Social cognitive	✓				+	
35	Subjective Norms	Planned Behavior		✓		+	+ + + Ø	+
36	Religion/Morality	Cognitive moral development	✓					+
37	Personality Traits	Protection Motivation	✓					+
38	General Information Security	-	✓					+
39	Technology Awareness	-			✓			+ Ø
40	Negative Affective flow	Affective flow	✓					- Ø
41	Perceived Severity of Threats	Protection Motivation	✓				+ + Ø	+
42	Perceived Vulnerability	Protection Motivation	✓				+ + +	+

	Factor	Theory	Scopes of Impact			Impacted Aspects		
			Individual	Cultural	Organizational	Compliance Attitude	Compliance Intention	Compliance Behavior
43	Personal Capabilities	Planned Behavior	✓					+
44	Locus of Control	Social Cognitive	✓				+	
45	Social Norms	Social Norms		✓		+		+
46	Information Security Climate	Organizational Climate			✓			+
47	Information Security Training	-			✓			+
48	Compliance Intentions	Planned Behavior	✓					+
49	Perceived Digital Mutualism Justice	Organizational Justice	✓				+	
50	Perceived Freedom Threat	Reactance	✓				-	
51	Perceived Responsibility	-	✓				+	
52	Work Impediment	-	✓				-	
53	Supervisor Subordinate Guanxi	Social exchange			✓			+
54	Perceived Threats	Protection Motivation	✓				+	Ø
55	Ethics	-	✓			+		

2.4.3.1 Factors Impacting Compliance Attitudes

As per the 84 reviewed articles, attitudes toward compliance are found to be impacted by many factors. These include: ethics, personal norms, descriptive norms, social norms, subjective norms, and sanctions. This subsection discusses them in details with regard to their scope of impact.

1. Individual Factors. The individual factors that impact compliance attitude include the personal norms and the ethics. The *Personal Norms*, which refers to an individual's values, is found to enhance one's moral obligation and compliance attitude (Wiafe et al., 2020).

Ethics are also found to provide moral principles and guidance, which can be impactful to a greater extent on the attitudes toward compliance (Moquin & Wakefield, 2016).

2. Cultural Factors. The cultural factors, which impact compliance attitudes, include: descriptive norms, social norms, and subjective norms.

Descriptive Norms motivates compliance attitudes, as a person perceives compliance of other people surrounding (Wiafe et al., 2020). Similarly, for the *Social Norms,* which represent a set of rules that informally guide a particular behavior and, and hence, influence the attitude.

Subjective Norms, which indicates how likely a particular behavior is approved by significant other people (for example: colleagues, friends, and people surrounding), plays important roles in developing the compliance attitudes (Wiafe et al., 2020).

3. Organizational Factors. The only organizational factor, which is found to impact compliance attitude is the *Sanctions*. It refers to consequences that an employee believes to result due to failure to complying with policies. These sanctions could be legal or organizational, are found helpful in directing compliance attitude positively (Moquin & Wakefield, 2016).

The few numbers of research studies and the identified factors that influence the compliance attitude at an individual, cultural or organizational level, indicates lack of studies that influence and shape attitude towards compliance.

2.4.3.2 Factors Impacting Compliance Intentions

The analysis of the review, found that primary studies give more emphasis on exploring the factors that impact compliance intention, than any other behavioral compliance aspects. The factors, which are found to influence compliance intentions are: security stress, moral disengagements, self-efficacy, security valence, security expectancy, security instrumentality, neutralization, attitudes toward compliance, efficacy of response, perceived compliance benefit, perceived compliance costs, perceived noncompliance

costs, commitment, attachment, involvement, personal norms, behavioral beliefs, knowledge on compliance, perception of threat severity, and perception of vulnerabilities, locus of controls, digital mutualism justice, perception of responsibilities, perception of threat, descriptive norms, socio-cultural environments, normative belief, and subjective norms, abusive supervision, rewards, punishment, certainty of control, procedural countermeasures, and transformational leaderships.

1. Individual Factors. *Self-Efficacy* positively impacts one's compliance intention. It refers to the level of a confidence and control capacity that an individual has (X. Chen et al., 2018; Hina et al., 2019; Ifinedo, 2012, p. 201, 2014; Jeon et al., 2020; Siponen et al., 2014). In similar way, the *Response Efficacy* is confirmed by three articles to positively influence one's intention to comply. It indicates the perception of response effectiveness (Ifinedo, 2012; S. H. Kim et al., 2014; Putri & Hovav, 2014). However, two studies by Siponen et al. (2014) and Hina et al. (2019) conclude that the response efficacy has an insignificant influence on the intention to comply. Their study justified that, through the lacking involvement of users in the formulation of security policies.

Security Valences and *Security Instrumentalities* have a positive impact on the compliance intention (Burns et al., 2018). Whereas the security valence is a reflection of the preference of an insider (Feather, 1995), and their perception on policy attractiveness, the security instrumentality refers to one's perceptions the extent to which security of user information help protecting an organization from a potential threat (Burns et al., 2018). In same study, Burns et al. (2018) also found that *Security Expectancy*, has a positive effect on the

compliance intention. The security expectancy refers to one's perception of the efforts needed in order to carry out a protective behavior.

Organizational *Attachment, Commitment, and Involvement* can improve social bonds of an employees, and hence, contribute to influencing compliance intention. They refer to attachment to peers of an organization, commitment to organizational objectives, and involvement in the activities of an organization (Ali et al., 2020).

Compliance Knowledge and its availability is critical to a user's perception of issues and challenges related to compliance (Kim & Kim, 2017), because it, substantially, guides and enhances their compliance intention. Furthermore, investing in compliance knowledge can bring other benefits which can also enhance the overall compliance. These include: *Perceived Compliance Benefits* and *Perceived Noncompliance Costs* (S. H. Kim et al., 2014), *Perceived Threats* (Putri & Hovav, 2014), *Perceived Severity of Threats* (Hina et al., 2019; Siponen et al., 2014), *Perceived Vulnerability* (Hina et al., 2019; Ifinedo, 2012; Siponen et al., 2014), *Perceived Responsibility* (Jeon et al., 2020), and *Perceived Digital Mutualism Justice* (Putri & Hovav, 2014).

Locus of Control can empower an individual become responsible for his/her own actions and, hence, it contributes to enhancing their compliance intention. The locus of control refers to one's abilities in controlling the events which are affecting them (Ifinedo, 2014).

Personal Norms comprise one's views and values, and hence, contribute to his/her compliance intention. The personal norms can guide the beliefs and intentions of an individual, and therefore, they are significant

policy compliance (Ali et al., 2020).

Attitudes Toward Compliance is reported by the selected studies to strongly influence the compliance intentions (Dong et al., 2021; Hina et al., 2019; Ifinedo, 2012, 2014; S. H. Kim et al., 2014; Siponen et al., 2014; Sojer et al., 2014; Wiafe et al., 2020). Despite the argument stated by the theory of planned behavior, that the behavioral beliefs can influence the attitude (Ajzen, 1991), *Compliance Behavioral Beliefs* have a positive influence on compliance intention (S. S. Kim & Kim, 2017; Siponen et al., 2014).

Security Stress is defined as the stress, which is mainly resulting by the demands posed by the growing security requirements (D'Arcy et al., 2014). This concept, security stress, is originally, adapted from the technostress concept of Ragu-Nathan et al. (2008). Security stress can result in a coping behavior, and found to cause *Moral Disengagement* from proper behaviors. All these, in turn, negatively affect the compliance intention (D'Arcy et al., 2014).

Neutralization indicates one's justifications made towards deviant behaviors. This gives the feeling of substituting guilts associated with a particular noncompliant behavior through a use of neutralization techniques. These techniques comprise: denial of an injury, loyalty, condemnation of a condemner, metaphors of ledgers, a necessity, and a defense of a ubiquity (Sykes & Matza, 1957). Rationalizing noncompliant behaviors is further encouraging subsequent policies violation, and therefore, influences compliance intention negatively (Bansal et al., 2020).

The *Perceived Compliance Cost* represents the effort and the time required to achieve compliance with policies. Compliance is viewed as a

burden and counterproductive, in some cases. In this regard, the perception of compliance costs, is found to have a negative impact on one's intention to comply (X. Chen et al., 2018; S. H. Kim et al., 2014). Contrary to these findings, the study of Ifinedo (2012) has rejected that, by justifying that this impact can depend on the effort level required for compliance.

Perception of Freedom Threat, which refers to restricting an individuals' freedom for selecting certain actions, that are related to their own devices. Such a perception is found to have a negative influence on the intention to comply, typically because users are expecting no restrictions or controls over the devices, they own (Putri & Hovav, 2014).

Work Impediments refers to interference or constraints in the ways towards task accomplishment, for example, redundant workflows or excessive security procedures. Such cases can consume time and become cumbersome. This, in turn, can allow alternative workflows and thus influences the compliance intention negatively (Jeon et al., 2020).

2. Cultural Factors: *Descriptive Norms* refers to the perception of a desired behavior by other people. It, typically, represents how people think or act toward certain behaviors (Herath & Rao, 2009). Similarly, to some extent, the *Normative Beliefs* refers to what should (not) be done toward a certain behavior, taking into consideration how the important surrounding people view that behavior (Bulgurcu et al., 2010). Both normative beliefs and descriptive norms contribute positively into shaping the compliance intention (X. Chen et al., 2018; S. H. Kim et al., 2014).

Subjective Norms strongly and positively influence the compliance intentions. They refer to the degree at which a certain majority of people

approve a particular behavior (Ifinedo, 2012, 2014; Sojer et al., 2014). However, a study by Hina et al. (2019) conclude that the impact of subjective norms is insignificant on the compliance intention. Their study justifies this finding by the possibility of a certain desired action is not well established by an organizational culture.

Establishing *Socio-Cultural Environment* helps in the development of compliance culture, through impacting organizational citizenship and habits of people in the long term and, hence, positively influence their intentions to comply with software related polocies (Balozian et al., 2021).

3. Organizational Factors: *Abusive Supervision* creates an attitude of resistance and a negative reaction. Nevertheless, the study of Guan and Hsu (2020) found that, the abusive supervision can be an effective way towards developing an organizational commitment, and, therefore, results in reduction of policy violation. In general, leaderships play a critical role towards guiding and enforcing the compliance. Particularly, the *Transformational Leaderships,* which is found to stimulate employees putting the organizational interests prior to their own ones. This can go beyond just performance achievement, to optimizing overall innovation of individuals, groups and organizations. Consequently, it strongly and positively influences employees' overall compliance (Guhr et al., 2019).

Procedural Countermeasures can act as an extrinsic mean that guides compliance intentions. These include policy development, and conducting regular training, and awareness programs. The technical countermeasures are found not sufficient as human behaviors are, most of the time, difficult to predict (Balozian et al., 2021).

Rewards and Punishment are found to have a positive influence on the overall compliance. There is an interplay effect of both rewards and punishment, on their influence toward employees' compliance (Y. Chen et al., 2012; Choi & Song, 2018; Kuo et al., 2021). Nevertheless, Siponen et al. (2014) found an insignificant effect of the rewards, justifying that, most organizations, typically, do not reward people, just because comply with policies as they obviously have to (Siponen et al., 2014). Additionally, Bansal et al. (2020) concluded that punishment is better deterrence for men and reward is better for women.

Certainty of Control help triggering signals to participants that their activities are under evaluation, monitoring, and amenable for punishment once noncompliance is detected. The certainty of control indicates how likely strategies of policy enforcement are effective (Y. Chen et al., 2012).

The intention of compliance is explored well, from an individual perspective. However, from an organizational and cultural perspectives, further confirmation is needed to make the findings strong, and enhance the validity and generalization.

2.4.3.3 Factors Impacting Compliance Behaviors

The factors that impact the actual compliance behavior are investigated well among the reviewed studies. These factors are: cost benefit analysis, self-efficacy, imitation, attitudes toward compliance, efficacy of response, perceived compliance benefits and costs, perceived noncompliance costs, perception of trust, religion, personality traits, clear general information security policy, negative affective flows, perception of threat severity, perception of vulnerabilities, personal capabilities, compliance intentions,

descriptive norms, differential associations, normative beliefs, moral norms, subjective norms, social norms, punishment, sanctions, top management support, technology awareness, information security climate, information security training, supervisor subordinate guanxi, organizational commitments.

1. Individual Factors: The *Self-Efficacy* has a strong and a positive impact on motivating individual's actual compliance behavior (Humaidi & Balakrishnan, 2017; T. Alanazi et al., 2020; Liu et al., 2020). In similar way, *Personal Capabilities,* can also encourage the compliance behavior. The personal capability refers to the knowledge and competences that an individual has (Carmi & Bouhnik, 2020).

Personality Traits including the level of openness, extraversion, and agreeableness. Such traits are strongly impacting the compliance behaviors, because they are likely to contribute in the spillover of certain values (T. Alanazi et al., 2020).

Response Efficacy is found to be effective in encouraging employees' engagement in a responsible and a compliant behavior (Liu et al., 2020).

Perceived Compliance Benefits and *Noncompliance Costs* are both contributing towards the understanding of values resulting from adherence, as well as the consequences which result from noncompliance. This, in turn, incentivizes the actual compliance behaviors (Carmi & Bouhnik, 2020). On the other hand, the *Perceived Compliance Costs* is found negatively related compliance behaviors (Carmi & Bouhnik, 2020; Liu et al., 2020).

Perception of Trust and the level of confidence in the implementation and enforcement of software policies, positively influences compliance behaviors (Humaidi & Balakrishnan, 2017). In similar way, the employees'

Commitment towards the organizational objectives is found critical to compliance behaviors (Liu et al., 2020).

Religion/Morality were found contributing to development of individual morals and enhancing his/her compliance behaviors (T. Alanazi et al., 2020).

General Information Security refer to the understanding of the policies related to information security, and issues and consequences related associated with them. A proper understanding of these policies can positively influence behavioral compliance of an individual (T. Alanazi et al., 2020).

Perception of Threat Severity and *Vulnerability,* are mainly related to the protection motivation theory (Rogers & Prentice-Dunn, 1997). Both of these factors help enhancing the compliance behaviors (Liu et al., 2020).

Attitudes Toward Compliance is reported strongly influence the compliance behavior (Carmi & Bouhnik, 2020; Moquin & Wakefield, 2016; Ormond et al., 2019). Attitudes should, typically, lead to intentions in order to trigger the compliance behavior, as per the theory of planned behaviors (Ajzen & Kruglanski, 2019). The *Compliance Intention,* in turn, is found by only a single study, which reports a positive influence of the compliance intention on compliance behaviors (Siponen et al., 2014).

Cost Benefit Analysis refers to the individual assessment of costs and benefits resulting from the compliance/noncompliance. The analysis of the selected studies reports a significant effect of cost-benefit analysis on compliance behaviors (Ifinedo, 2016).

Imitating similar behaviors significantly influences compliance behaviors (Lembcke et al., 2019).

Negative Affective Flows, which indicate the immersion of individuals on emotions that are negative. The negative effective flow is found negatively impacting the compliance behavior (Ormond et al., 2019). A study by Ormond et al. (2019) concludes a significant effect of negative effective flow, while studying users experiencing a high degree of frustration. However, they also report an insignificant effect with those experiencing lesser frustration.

2. Cultural Factors. The results report that compliance behavior can be shaped by several cultural factors. These include: descriptive norms, moral norms, and social norms. *Descriptive* and *Moral Norms* are reported to have a positive influence on compliance behaviors (Merhi & Ahluwalia, 2019). The descriptive norm is defined as one's perception about whether other people are also performing a particular behavior (Fishbein & Ajzen, 2009), whereas the moral norm is one's feeling towards moral responsibilities and obligations, that guides an individual on his/her decision on whether to do or obstain from a particular behavior (Ajzen, 1991). In a similar way, the *Social Norms,* can directly influence a compliance behavior. The social norms refer to acceptable behaviors by people or a society (Moquin & Wakefield, 2016).

Furthermore, *Subjective Norms* (T. Alanazi et al., 2020) and *Normative Beliefs* (Carmi & Bouhnik, 2020), are found to have a positive impact on compliance behaviors. Both concepts refer to the extent to which other people, who are important, assess and look at a particular behavior.

Differential Associations, which is the interaction with peers, and the extent that makes someone learns a certain value and attitude. The differential association is found to influence one's behavioral compliance (Lembcke et al., 2019).

3. Organizational Factors: *Punishment* (T. Alanazi et al., 2020), and *Sanctions* (Ifinedo, 2016; Lembcke et al., 2019) are likely powerful deterring instruments, and found positively contributing towards the enhancement of compliance behaviors.

Top Management Support influences compliance significantly (Ifinedo, 2016). In a relatively similar way, the *Supervisor Subordinate Guanxi*, which is the action of exchanging favors and establishment of personal connections, between supervisors and their subordinates, is found critical to compliance. In other words, strong ties of supervisor-subordinate relationships can contribute towards incentivizing the organizational commitments (Liu et al., 2020).

Establishing *General Information Security* guideline along with *Technology Awareness* programs, are found crucial towards enhancing the overall compliance behaviors (T. Alanazi et al., 2020; Carmi & Bouhnik, 2020; Liu et al., 2020). Additionally, having an *Information Security Climate,* within an organization that enables sharing values, and assumptions on information security within all members of an organization, can be strong enabler to compliance behaviors (Liu et al., 2020).

2.4.4 Compliance Policies and their Addressed Challenges

For addressing the third research question, our analysis extracted the policies introduced by the primary studies and grouped them based on the compliance challenges being addressed. In total, 20 policies are found and presented in Table 4. The majority of these policies tend to address behavioral issues of end users, i.e. the human side, whereas few of the identified policies address technology-related challenges.

Table 4. Compliance Policies and their Challenges Addressed

#	Policies	Category	Addressed Challenges	Reference
1	Automating compliance management	Technology / Human	Infrastructure misconfiguration, efforts, checking accessibility, license checking, security attacks, modeling and standards regulations, misinterpretation of requirements, evolution of regulations and standards, neglecting best practices.	(Castellanos-Ardila et al., 2021; Chen et al., 2021; Czepa et al., 2017; Gangadharan et al., 2012; Joshi et al., 2020; Máñez-Carvajal et al., 2021; Rongrat & Senivongse, 2018; Steffens et al., 2018; Usman et al., 2020; Varela-Vaca et al., 2019;)
2	Security education, training and awareness	Human	(non) compliance intentions, (non) compliance behaviors, organizational injustice, affective flows, developers' sense of responsibility, non-malicious insiders, interpretation of compliance requirements, technostress, functional safety.	(Ali et al., 2020; Antinyan & Sandgren, 2021; Balozian et al., 2021; Bednar et al., 2019; Burns et al., 2018; Carmi & Bouhnik, 2020; X. Chen et al., 2018; D'Arcy et al., 2014; Guan & Hsu, 2020; Hina et al., 2019; Humaidi & Balakrishnan, 2017; Ifinedo, 2012, 2014, 2016; S. H. Kim et al., 2014; Kuo et al., 2021; Lembcke et al., 2019; Moquin & Wakefield, 2016; Ormond et al., 2019; Putri & Hovav, 2014; Siponen et al., 2014; Stafford et al., 2018; T. Alanazi et al., 2020; Usman et al., 2020)
3	Establish codes of ethics	Human	Engineers' sense of responsibility.	(Bednar et al., 2019)
4	Rewards and punishment	Human	Insider breach, negligence, neutralization, (non) compliance intentions, (non) compliance behaviors, divergence of preferences, resistance toward information security policy.	(Ali et al., 2020; Bansal et al., 2020; Y. Chen et al., 2012; Ifinedo, 2014; Merhi & Ahluwalia, 2019; Wiafe et al., 2020)

#	Policies	Category	Addressed Challenges	Reference
5	Deterring instruments	Human	(non) compliance intentions, (non) compliance behaviors.	(X. Chen et al., 2018; Ifinedo, 2016; Lembcke et al., 2019; Moquin & Wakefield, 2016)
6	Internal controls and auditing	Human	Information accountability, adherence to security policy, non-malicious insider, lack of transparency in distributed teams, security breaches.	(Kuo et al., 2021; Samavi & Consens, 2018; Singi et al., 2019; Stafford et al., 2018; Westland, 2020)
7	Investigating workarounds	Human	Insider behaviors, shadow IT inadequacy in existing information systems.	(Davison et al., 2019; Van Slyke & Belanger, 2020)
8	Evaluate security related stress	Human	Compliance intentions, technostress.	(D'Arcy et al., 2014)
9	Analyzing rationalities behind noncompliance	Human	Different rationalities, noncompliance motivation, affective flows, organizational injustice.	(Kolkowska et al., 2017; Ormond et al., 2019)
10	Promoting organizational climate and social bonds	Human	Negligence, (non) compliance intentions, (non) noncompliance behaviors.	(Ali et al., 2020; X. Chen et al., 2018; Dong et al., 2021; Hina et al., 2019; Ifinedo, 2014, 2016; Karjalainen et al., 2020; Liu, Wang, & Liang, 2020)
11	Incorporating appropriate responses	Human	Detrimental compliance, compliance intention	(Alter, 2015; Putri & Hovav, 2014)
12	Practice-based discourse analysis	Human	Insiders' threat, workarounds, ambiguity of policies, employee prioritization	(Karlsson et al., 2017)
13	Software certification	Technology	enforcing specific SDLCs and MDD, interpreting regulatory documents, software requirements mismatching of physical and standalone devices, compliance with standards.	(Accorsi et al., 2011 ; Antinyan & Sandgren, 2021; Granlund et al., 2020; Hale & Gamble, 2019; Marques & da Cunha, 2018)
14	Model driven development	Technology	Enforcement of specific SDLCs, diversity of compliance sources, software engineering best practices, familiarity to	(Marques & da Cunha, 2018; Tran et al., 2012)

#	Policies	Category	Addressed Challenges	Reference
			domain experts.	
15	Regulation-oriented architecture	Technology	Regulatory compliance, data interoperability, gaps between technical and legal experts, purpose limitations, data accountability, user rights to erasure, and time-limited retentions, gaps between auditors and designers	(Antignac et al., 2018; Julisch et al., 2011; Li et al., 2020; Mohamed et al., 2021)
16	Use of the most restrictive law	Technology	Ambiguities, contradictions, conflict in requirements, exceptions.	(Maxwell et al., 2013)
17	Outsourcing	Technology	Poor in-house practices	(Thalmann et al., 2014)
18	Run-time security auditing	Technology	Accountability, transparency, trust, infrastructures auditing.	(Majumdar et al., 2018; Samavi & Consens, 2018)
19	Standardizing user accessibility	Technology	Usability, lack of accessibility	(Máñez-Carvajal et al., 2021; Montazeri et al., 2020; Oliveira et al., 2020)
20	Privacy & security by design	Human	Inconsistency between policy makers and developers' mindset	(Arizon-Peretz et al., 2021)

The identified policies are classified into technological and human related policies. The classification of these policies is based on compliance challenges they address as shown in (Figure 13).

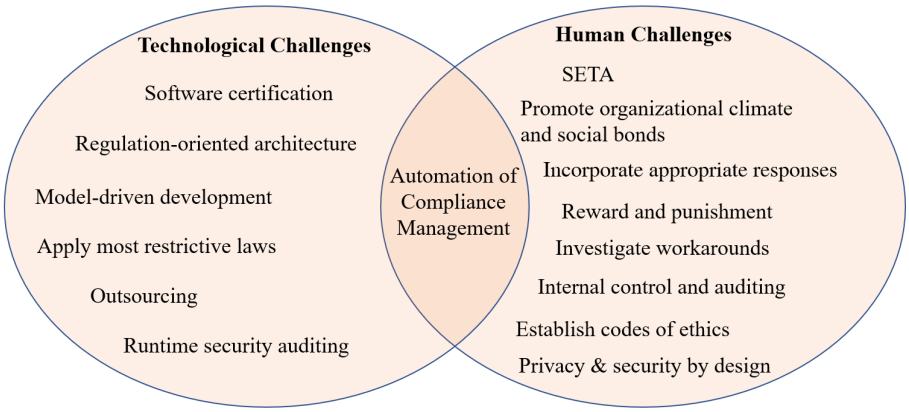

Figure 13. Classifications of Policies based on type of challenges they Tackle

2.4.4.1 Policies Address Human Related Challenges

Several studies found that the human is the weakest when it comes to software compliance, as the human behavior is difficult to predict. In this regard, the majority of the identified policies tend to address issues related to human side (Table 4). These include automation of compliance management, SETA, rewards and punishment, internal controls and auditing, investigation of workarounds, promotion of organizational social bonds, deterring instrument, evaluating security stress, developing code of ethics, analysis of compliance rationalities, and incorporate appropriate responses.

1. *Automating Compliance Management* can help in mitigation of issues related to manual checking. This, in turn, can reduce complexities of analyzing various compliance sources, and help making management of compliance less error-prone (Czepa et al., 2017). Furthermore, creating a system that achieves this purpose can also serve as a reference and source of compliance knowledge (S. S. Kim & Kim, 2017).

2. *Security Educations, Training and Awareness (SETA)* is a highly important policy that organizations should conduct on a regular basis. Noncompliance behaviors happen, mostly, due to negligence and lack of knowledge (S. S. Kim & Kim, 2017), technostress (D'Arcy et al., 2014), organizational injustice (Ormond et al., 2019; Siponen et al., 2014), misinterpretation of requirements (Usman et al., 2020), users' negative emotions (Ormond et al., 2019), and non-malicious insiders, who, unintentionally, perform a noncompliant act (Stafford et al., 2018). Studies found that SETA can help mitigating noncompliance intentions (D'Arcy et al., 2014; Guan & Hsu, 2020) and noncompliance behavior (Hina et al., 2019;

Ifinedo, 2016). Through SETA, organizations can enhance compliance intention (Balozian et al., 2021; Burns et al., 2018; X. Chen et al., 2018; Ifinedo, 2012, 2014; S. H. Kim et al., 2014; Kuo et al., 2021; Putri & Hovav, 2014; Siponen et al., 2014), and behavior (Ali et al., 2020; Carmi & Bouhnik, 2020; Humaidi & Balakrishnan, 2017; Ifinedo, 2012, 2014; Lembcke et al., 2019, 2019; Moquin & Wakefield, 2016; T. Alanazi et al., 2020).

3. *Establish Code of Ethics* can guide engineers and developers, and provide reference and direction, which are helpful to meeting industries best practices. This, in turn, enhances their accountability and responsibility (Bednar et al., 2019).

4. *Rewards and Punishment* are effective in mitigating user negligence (Ali et al., 2020), insider breach (Y. Chen et al., 2012), and user's use of neutralization strategies to rationalize any violation (Bansal et al., 2020; Y. Chen et al., 2012). By default, users do not have a motivation to adhere to certain policies and procedures (Y. Chen et al., 2012). In this regard, having rewards and punishment in place can help minimizing noncompliance intentions (Wiafe et al., 2020) and noncompliance behaviors (Ali et al., 2020; Ifinedo, 2014). Moreover, punishment plays an indirect role in reducing resistance towards information security policy (Merhi & Ahluwalia, 2019).

5. *Deterrence Instrument* can prevent security-related issues and enhance employees' overall compliance intentions (X. Chen et al., 2018) and behaviors (Lembcke et al., 2019; Moquin & Wakefield, 2016). The availability of a deterrence instrument also helps reduce non-compliance behavior and policy violations (Ifinedo, 2016).

6. *Internal Controls and Auditing* can have an effectiveness in

monitoring users security practices of users and improving their conformance information security policies (Kuo et al., 2021) and information accountability (Samavi & Consens, 2018). This, in turn, can assist in the identification of unsafe practices, non-malicious activities on insiders (Stafford et al., 2018), and security breaches (Westland, 2020), while, it also help address transparency issues, especially, of distributed teams (Singi et al., 2019).

7. *Investigation of Workarounds* which are conducted by insiders, are critical to identifying threats and vulnerabilities (Van Slyke & Belanger, 2020). Pressure to meeting deadlines, inadequate software systems, or complexity measure of security, are likely to cause workarounds (Davison et al., 2019). This, in turn, can drive a user to improvise and look for an easy way to accomplish his/her tasks, and as a result, compromise policy compliance.

8. *Evaluate Security Stress.* The stress resulting from complex and ambiguous security requirements, can lead to negative consequences. Evaluating such kind of stress can reduce policy violations and threats (D'Arcy et al., 2014)..

9. *Analyze Rationalities of Noncompliance* can help understanding the motivations behind certain behaviors (Kolkowska et al., 2017), and, in turn, respond to that through development of proper policies. Analyzing rationalities, can reveal causes of noncompliance, including those related to the sense of organizational injustice and user-related emotions (Ormond et al., 2019).

10. *Promote Social Bond and Organizational Climate.* Socializing and regular activities can build a culture of common values and norms, and enhance compliance (X. Chen et al., 2018; Ifinedo, 2014). Although,

developing social values requires an organization investing such activities (Ifinedo, 2014); studies found that, such a policy can positively mitigate noncompliance intention and behavior (Ali et al., 2020; Hina et al., 2019; Ifinedo, 2016). In addition to that, It can promote and normalize certain compliance behaviors (Ifinedo, 2014; Karjalainen et al., 2020; Liu, Wang, Wang, et al., 2020); and, hence, minimizing negligence (Dong et al., 2021).

11. *Incorporate Appropriate Response.* As excessive compliance can be detrimental, contextual responses can be beneficial (Alter, 2015). In this regard, analyzing workarounds of users and incorporating proper responses, can improve efficiency and effectiveness of incident response (Alter, 2015; Putri & Hovav, 2014).

12. *Practices-Based Discourses Analysis.* Ambiguous or incompatible policies can drive employees prioritize and working around them. As a result, such policies can demotivate compliance and encourage workaround behavior. Ensuring that policies are in line with work practices of employees, reduces workarounds and, in turn, less insiders' threat (Karlsson et al., 2017).

13. *Privacy & security by design*: this can bridge the consistency gap between policy makers and developers' mindset and the confusion of interpreting field related vocabularies, allowing policies to be imbedded during design stage (Arizon-Peretz et al., 2021).

2.4.4.2 Policies Address Technology Related Challenges

Eight policies, which address technology-related compliance issues, were identified. These include: automating compliance management, software certifications, regulation driven architectures, standardizing user's accessibility, model driven development, applying the most restrictive law,

run-time auditing of security, and outsourcing (Table 4).

1. *Automating Compliance Management* can help address compliance issues related to humans and technology. Technologically, automation of compliance management can help avoiding errors, and minimize the effort and time required (Castellanos-Ardila et al., 2021). Manually misconfiguring complex IT systems, is likely to result in vulnerabilities (Varela-Vaca et al., 2019). In addition to that, the manual checking can be error prone, for instance, the manual checking of accessibility standards (Máñez-Carvajal et al., 2021) or periodically assessing security risks (Rongrat & Senivongse, 2018). Furthermore, duplication of effort in managing various compliance requirements can be very tedious (Joshi et al., 2020), let alone the likelihood of misinterpreting and misunderstanding by concerned stakeholders of different expertise. In these regards, compliance automation becomes crucial (Czepa et al., 2017). An example of compliance automation is delivering policies-as-code, since software engineers are regarded lacking awareness of compliance requirements (Usman et al., 2020). Policy as code, can overcome issues related to requirements misinterpretation (Steffens et al., 2018).

2. *Software Certifications* minimize risks and vulnerabilities, while simplifying compliance management, through all software life-cycle. This help ensuring proper security measures in place, as well as adherence third party components to required certifications (Hale & Gamble, 2019). Software certifications can also help standardizing compliance for appliances whether physical or virtual ones (Antinyan & Sandgren, 2021; Granlund et al., 2020).

3. *Model Driven Development.* This policy help breaking down a software into modular component, and hence, simplifies alignment of multiple

compliance sources (Tran et al., 2012). With continuous development, using model-driven paradigm can help bridging gaps between domain experts and software engineers. This approach can enhance validation and enforcement of throughout the software life cycle (Marques & da Cunha, 2018). Consequently, making further development less sensitive to changes in business and technologies. The derived approaches from this policy include test-driven and behavior-driven approaches.

4. *Regulation Driven Architectures* help bridge the legal and technical gaps (Mohamed et al., 2021), as the vocabularies and assumptions they tend to use are different (Julisch et al., 2011). Such an approach enables embedding privacy settings in the design phase (Antignac et al., 2018). In turn, this help address the issues related to interoperability and heterogeneity of components (Li et al., 2020).

5. *Use of the Most Restrictive Law* is crucial especially with conflicting, ambiguous, or contradicting requirements. Compliance of software systems can be governed by various regulatory requirements. It is possible that a certain law is more restrictive than others, while a software have to comply with both (Maxwell et al., 2013). Especially, in requirements engineering, following the most restrictive one is crucial.

6. *Outsourcing* is favorable by many organizations as the complexity of software systems increases continuously (Lehman, 1980), while in-house software development can fail to meet best practices (Thalmann et al., 2014).

7. *Run-time Security Auditing* of multi-tenancy environments should be considered, in order to ensure transparency, accountability, and trustworthy services; especially, when consuming cloud-based services (Majumdar et al.,

2018).

8. *Standardize Users' Accessibility* help maximizing the usability of a software systems, and in turn, influence the overall performance and productivity of users (Montazeri et al., 2020). Contrary to that, the poor design, which lacks accessibility, complicates users' acceptance of using the system. Consequently, this can the system less useful, and can even impede their productivity (Máñez-Carvajal et al., 2021; Oliveira et al., 2020).

2.5 Discussion

2.5.1 Summary of Findings

Figure 14 presents main findings of this review. The figure consists of four columns, in which the first two show the categories set up with regard to the topics defined in the research questions. The 3rd and the 4th columns of the figure shows the highly cited topics and evolving ones. The first row of Fig. 14 shows industry requirements and user contexts, in that the security in healthcare and finance is a top issue discussed, and similar for end users and developers (section 2.4.2). The top cited theories are the theory of planned behavior, deterrence theory, and requirements engineering. The evolving theories and concepts are the theory of workarounds, privacy-by-design and compliance-by-design (section 2.4.3). The identified factors are categorized according to their scopes of influence, as presented in 3rd row of Fig. 14. Top cited among the reviewed studies are attitudes toward compliance, punishment, and subjective norms; whereas the security stress has emerged (section 2.4.3). Finally, the identified policies are grouped with regard to the compliance challenges being addressed (human, technology, or both). SETA,

software certifications and regulations driven architectures, and compliance automation, are found the top cited, whereas policy-as-code is emerging (section 2.4.4).

		Highly Cited and Matured	Evolving Concepts
15 Requirements *Section 2.4.2*	15 Industries	Healthcare: Security and Legal	
		Finance: Security and Auditing	
	11 User Context	End User: Security & Accessibility	
		Developers: Security, Safety and Legal	
Theories/Concepts *Section 2.4.3*	37 Theories	Planned Behavior	Workarounds
		Deterrence	
	35 Concepts	Requirement Engineering	Privacy-by-design Compliance by Design
55 Factors *Section 2.4.3*	36 Individual	Attitude Towards Compliance	Security stress
	7 Cultural	Subjective Norms	
	12 Organizational	Punishment	
20 Policies *Section 2.4.4*	11 Human-related	Security Education Training and Awareness	Policy as Code
	8 Technology-related	Certification & Regulation-driven Architecture	
	1 Human & Technology	Automation of Compliance Management	

Figure 14. Key Highlights of Top Cited and Evolving Concepts

The topic of 3rd column of Fig. 13, are investigated well in the software compliance domain. The level of attention that these topics gained, in the literature, indicate their importance. On the other hand, the 4th column, which shows the findings of the evolving concepts, indicates that such topics are either related to growing compliance challenge, new approaches to enhance compliance, or factors gain more emphasis.

2.5.2 Implications

2.5.2.1 Implications on compliance requirements

The review delivers the following implications related to compliance requirements: First, security compliance is on top of discussion in many

industries (Figure 9 of section 2.4.2), and it is mostly related to end users (Figure 10 of section 2.4.2). It is also emphasized by several professional and market research organizations that the vulnerability of end users to non-compliance and security attacks is very high (PricewaterhouseCoopers, 2021). Additionally, the concerns growing on security breaches, have resulted in expectation of the growing research interest in security compliance related to end users. Although, other stakeholders, including developers, managers, domain experts and legal experts, gain less emphasis compared to end users, such result is expected as end users are the ones dealing most frequently with software systems. Moreover, they responsible for over 50% of security and data breaches. Nevertheless, a further investigation is needed around the regulatory concern of end users.

Secondly, the regulatory and privacy requirements in medical industry are on top discussion, while the software industry places security and licensing on top of industry issues. According to the analysis, industry specific needs impose different requirements and priorities on certain requirements over others. For instance, issues related to license compliance come as software industry's second priority; whereas auditing in financial industry is placed as a second top requirement after security. This indicates that each industry prioritizes required policies according to their priorities of their requirements.

Thirdly, from the perspective of software developers, security, legal, safety, license, and software architecture are highly discussed issues (as shown in section 2.4.2). Software engineers are considered the law enforcers, as they are in the forefront, and the code produced is a representation of laws

and policies). In this regard, we found an emphasis on gaps between software developers, compliance and legal experts, which calls for a need to bridging these gaps and address the issues related to misinterpretation, misunderstanding of vocabularies and assumptions. Further research may also consider other key players in the ecosystem, including: software engineers and architects, who are the most highly concerned with the overall implementation of software systems. Lastly, the research on compliance around business process, accessibility, and usability of a software in context of software engineers lacks sufficient exploration in the literature.

2.5.2.2 Implications on factors influencing compliance

Regarding the factors influencing compliance, the findings indicate that a careful analysis of aspects related to individuals and cultures, should be in place, while formulating software related policies (section 2.4.3). As some of the identified factors are found tested in a one or more than one study, this is likely to increase the validity of those factors. However, most the identified factors are found tested by an only one study. Therefore, generalizing such finding to other contexts and cultures, becomes difficult. It is crucial to consider peculiarities related to contexts, in order to develop a proper policy response in accordance to that context. In this regard, decision makers and managers are highly recommended to pay a close attention to their contexts.

The findings report the factors that are critical to the overall compliance, according to their scopes of influence. Firstly, most of those factors emphasize more on individual behaviors from the lens of protection motivations. These include: self-efficacy, attitudes toward compliance, response efficacy, and perceived vulnerabilities and threads. Secondly, business managers need to

consider descriptive, social, and subjective norms, as a confirmation among several primary studies, that they are significantly shape compliance attitudes, and in turn, the behaviors. Thirdly, having deterring instruments in place, whether punishments or sanctions, is effective tool that raises individuals' perceived consequences for noncompliance behavior. In a nut shell, no matter how well-formulated policies, individuals and cultural aspects should not be underestimated. Otherwise, a culture can eat a strategies and policies for a breakfast.

2.5.2.3 Implications for theories

According to the reviewed studies, research on software compliance pay more attention to characteristics of an individual as main factors and rationales behind (non) compliance behaviors. The theoretical analysis of the primary studies found that the planned behavior of Ajzen (1991) is the most dominant theory, followed by deterrence of Jervis (1979), and protection motivation of Rogers and Prentice-Dunn (1997). The review also identified other applied theories including rational choice of Scott (2000), social bond of Hirschi and Stark (1969), neutralization of Sykes and Matza (1957), organizational climate of Schneider et al. (1996) and Alter's (2014) theory of workarounds. The theory of workarounds has emerged in dealing with shadow systems, misfit in work system practices, and technical debt. It argues that workarounds can result from complex compliance measures. Nevertheless, studies on what contributes to development of workarounds for both end users and software engineers is lacking.

As many previous studies argue that human-side of compliance is considered the weakest, the majority of most of the primary studies emphasize

more drivers that motivate/deter individuals' behaviors toward (non) compliance. Furthermore, our review found that previous studies on software compliance miss to identify the ultimate objectives behind the rationale of (non) compliance. In this regard, Ajzen and Kruglanski (2019) introduced the reasoned goal pursuit as enhanced theory of the planned behavior. The reasoned goal pursuit argues that the ultimate gains to obtain out of performing a particular behavior, contributes, to a greater extent, in incentivizing the actual behavior. In this regard, incorporating procurement and active goals in future studies on compliance, can bring deeper understanding on one's compliance behaviors.

2.5.2.4 Implications on policies

Based on the analysis of the reviewed studies, software compliance policies are categorized into: human related and technology related policies, according to issues they tend to tackle (section 2.4.4). The majority of the identified policies emphasize more on addressing compliance issues related to the human side than technological ones. This can give an indication that most compliance challenges are related to human behaviors. The analysis of the top cited policies shows three major policies contributing to solve many of challenges related to software compliance. These are: compliance automation; security education, training, and awareness (SETA); and organizational climate, and social bonds. Compliance automation (sections 2.4.4.1.1 and 2.4.4.2.1) help addressing several human and technology related challenges. In other words, less human involvement results in a less error prone and more effective compliance management. For instance, misconfiguring an infrastructure or misinterpreting requirements by stakeholders of different

expertise, typically happens due to manual compliance management. Automation of compliance help organizations avoid such mistakes.

The second level of policies that organizations should consider, is a regular security educations, training and awareness (SETA) program, because they found to mitigate the threats and compliance issues related insiders, which account for over 50% of software attacks and data breaches (section 2.4.4.1.2). Another highly cited policy is building the organizational climate and social bond (section 2.4.4.1.3). This policy contributes in strengthening the attachment, commitment, and involvement of employees, and enhances their beliefs in the organizational principles. As a result, these can strengthen their sense of belonging; shape compliance attitude and cultures, and, in turn, mitigate insiders' negligence.

The other influential policies, which organizations should consider are: deterring instrument, rewards and punishments, and investigation of workarounds. The selection of proper policy mix depends on organizational and industry related requirements should. In other words, the nature of business compliance challenges being faced, are critical to designing and selection of proper policies. Therefore, practitioners and managers should carefully define what they need a policy to address for a more effective decision.

Technology related policies are also considered of equal importance to the aforementioned ones, since the technology nowadays is critical to businesses, if it is not the core business of an organization. Software certifications and regulation-driven architecture are on top discussion of the technology related policies. However, we found that primary studies do not

make distinctions between open source and proprietary software. We believe that making such a distinction is important, as each of these types is likely to raise different compliance challenges. For instance, license compliance has more complications in open-source software, in that software developers can use different components, in which, their underlying licensing scheme might contradict with each other, and hence, likely to result in security and legal consequences and threaten future releases. Furthermore, software piracy can be more problematic to proprietary software than an open-source one.

Another set of technology related policies are concerned with providing mechanisms for enforcement and enhancement of visibility to all the stakeholders concerned, as also argued by Mubarkoot and Altmann (2021). If policies can be modeled in machine-readable code, chances of misinterpreting policies differently by various stakeholders of different expertise can be overcome. Moreover, this can also enable automating and enforcing policies. In this regard, policy-as-code has emerged to help bridging these gaps, however, the research related to policy-as-code is still in its infancy. In addition to that, the research on tools of modeling and supporting software compliance, which consider stakeholders' engagement throughout life cycle of a software is lacking. The challenges can be associated with difficulties, that concerned stakeholders of different expertise have different methods, assumptions, and vocabularies (Julisch et al., 2011).

Lastly, there are growing concerns raised due to multiple devices an individual has, in particular, the threats and consequences resulting from bring yourown device (BYOD) to a workplace. Although, studies on compliance policies of BYOD is lacking (Palanisamy et al., 2020), it is worth

investigating proper policies that address compliance issues of BYOD, since the associated consequences can be severe. In relation to that, and during the COVID-19 pandemic, many companies world-wide responded to safety restrictions and allow their workers to do their jobs home. By granting employees remote access to organizations' resources and software assets, it is likely to raise many vulnerabilities and risks. In this regard, this shift toward home-office working environment might raise several compliance challenges. These include security, privacy, accessibility, and regulatory issues. Hence, future studies can put more emphasis on compliance challenges and policies associated with home-office users.

Table 5. Summary of Further Research Recommendations

Topic	Further Research Recommendation
Requirements	1. Regulatory concern around end users of E-type software systems, requires further investigations. 2. Research effort is needed bridge gaps between compliance experts and domain experts, and software engineers. 3. Further research can investigate compliance of business process, accessibility, and usability, in software developers' context.
Theories	1. The emergence of the theory of workaround, opens a room for exploring what antecedents that can cause workarounds. 2. The theory of reasoned goal pursuit, assist in understanding how individual goals drive (compliance) behavior, and, therefore, deserve investigation.
Policies	1. Policies distancing compliance of open-source and proprietary software requires further research attention. 2. There is a lack of research on enforcement mechanisms and tools that provide visibility to concerned stakeholders. 3. Research efforts need to investigate policies addressing compliance of home office users.

Topic	Further Research Recommendation
	4. As automation help addressing many challenges related to compliance, further research can develop supporting tools for automating compliance management.

2.6 Conclusion

2.6.1 Summery

This study surveys requirements, factors and policies related to software compliance using a systematic literature review. The methodology adapted is of Kitchenham et al. (2016), which uses an evidence-based thinking to systematically collect and analyze existing body of research in a more systematic and replicable way. We made some enhancements in the applied method through adding new step before deriving the review research questions. This include analyzing existing reviews in order to re-assess the objective of the review and formulate the study questions. Accordingly, 84 candidate papers were identified relevant to the review objective.

Based on the analysis of results, the security concerns around end users of software systems are on top discussion. As the end users are considered accountable for more than 50% of software attacks, such findings are expected. Furthermore, the privacy and accessibility concerns are also growing with regard to end users. With regard to software developers, the majority of the selected studies focus more on the security, legal, safety, licensing, architectural compliance. While, typically, software end users and developers are directly dealing with software systems, and more likely to face the aforementioned issues, such results might seem normal to a great extent.

For software developers in particular, there is an emphasis, in the primary studies, gaps between software developers, compliance and legal experts. In this regard, the recently evolving concepts related to privacy by design and compliance by design are expected contribute tremendously towards bridging these gaps, and ultimately optimize compliance management.

The identified factors impacting compliance are categorized according to their scopes of impact: individual factors, cultural factors, and organizational factors. The majority of these factors are mostly associated with individual characteristics. The emergence of the theory of workarounds, in the domain of software compliance is driven primarily, by a deliberate or inadvertent working around prescribed compliance procedures.

Our review also listed a set of compliance policies along with the challenges they tend to tackle. Based on the findings, and as the human factor is found the weakest chain in the compliance, organizations should prioritize security education, training and awareness, in order to mitigate the overall insider threats. The review also found that compliance automation can help overcoming many challenges which are resulting from the manual checking of adherence. The emerging concept “policy as code” can support having machine-readable policies, and in turn, reduce some human involvements in management of compliance. Nevertheless, there is a lack of supporting tools and mechanisms which deliver the enforcement needed, while providing a visibility to the stakeholders concerned. Another effective policy, which enables organizations to build employees’ attachment, involvement, commitment, and beliefs in corporate policies, is the promotion of social bonds and organizational climate. Consequently, this is found to reduce

insider negligence and spread compliance culture.

Surprisingly, this review found no distinction discussed on the compliance policies between open-source and proprietary software. Making such distinctions are crucial because the two types of licensing can pose different specifications, and in turn, likely to result in different challenges associated with licensing, and legal requirements.

This study further delivers a set of implications and potential directions for future research based on the analysis of findings from the selected primary studies. Such implications can guide practitioners develop effective strategies and policies to control compliance and protect the most precious software systems and data. Furthermore, the behavioral factors related to human side is dominating the majority of compliance challenges. This indicates that the human behavior remains most challenging no matter how strong technological aspects. In this regard, we might require less involvement of humans and more automation, in order to tackle such an issue, and enhance the overall compliance.

2.6.2 Limitations and Future Research

Although the process of the systematic literature review and the selection of primary studies were conducted in a rigorous way, there is a possibility of missing some relevant articles that might impact the results and the comprehensiveness of the review.

Additionally, as this review only focuses on the factors that have a direct impact on software compliance, the review did not consider those factors which have an indirect impact. It is also worth mentioning that, the majority of the identified factors and policies are tested in a single context,

therefore, generalizability can be a bit of a challenge. In this regard, additional confirmation could be required in order to enhance the validity of evidences, as well as claim generalizability. Although, having a result of one factor tested in a more than one context, considering peculiarities of a context is critical.

Further research can consider software compliance in relation to business process, accessibility, and usability. There is also an emphasis needed other stakeholders, whom the research has paid less attention; these include managers, and legal and domain experts. In addition to that, the research on compliance around software engineers and architects, who are concerned, in the first place, with designing and developing software systems, deserves more attention. Furthermore, there is a need to contribute in bridging the gaps between domain experts and software engineers, as there are many challenges raised due to misinterpretation of requirements, and usage of different vocabularies and assumptions.

As the theory of workaround been tested in the relation to software compliance of end users, research and applications of the theory in the context of software engineers is still lacking. Additionally, the new extension of the theory of planned behavior, namely reasoned goal pursuits, emphasize incorporating an individual's current active goals as key motivators toward the behavior in question. These motivators can be arguably critical in understanding drivers behind (non)compliance, in which none of the studies selected in this review has tested these extensions. Further research can consider incorporating the procurement and approval goals, to help understand deeper on causes that trigger (non)compliance behaviors.

When it comes to prioritizing which policies to consider over others, as

this likely depends on the challenge being faced and the degree and scope of impact. For example, as the negligence of insider costs organizations over 50% of breaches software related incidents, having regular SETA, and establishing organizational climate, and social bonds could be the remedy that can mitigate this impact in the long run. In general, the issues related to the implementation of such policies depend on needed compliance requirements as well as the contexts of applying these policies. Automating compliance and software certifications, for instance, require huge effort, in the beginning in order to model and align the implementation to fit into that certification or automation scheme. In this regard, in order to help evaluate difficulties related to policies implementation and their expected pay off, further systematic review of case studies that focus on policy applications or even new case studies research can bring valuable understanding on challenges associated with policy implementation.

Chapter 3. The Impact of Technostress on Software Engineering Workarounds and the Moderating Role of Neutralization, Autonomy, and Perceived Behavioral Control

3.1 Introduction

Software systems typically evolve as a response to changing business requirements and policies. This is true for E-Type software systems, which solve and automate real world problems. In fact, the software must continuously grow and change, otherwise it gradually becomes less useful (Lehman, 1980; Lehman & Ramil, 2002). This drives towards shortening software delivery cycles, in order to streamline development and production pipelines; or what is known recently as continuous delivery and integration (Humble & Farley, 2010). In other words, the current trends that demand for more features at a smaller development cycles places a software to be always in a releasable state (Dave Farely, 2021). In general, software engineers are known to be stressed workers. This stress is considered to have detrimental effects on team's morale and motivation, communication and cooperation-dependent work, software quality, maintainability, and requirements management. Therefore, it is crucial to effectively assess, monitor, and reduce stress for software engineers (Ostberg et al., 2020).

Software engineers face management pressure, to meet deadlines and time-to-market schedules. This, in some cases, pushes them to compromise quality and implement workarounds, in order to deliver the required

functionality within deadlines, leaving technical debts behind them. The technical debts can be in the form of requirements, compliance, design, code, architectural, test, dependency, infrastructure, consistency or documentation debts (Yli-Huumo et al., 2014). The pressure of meeting deadlines is considered the most critical cause of software engineering workarounds (Pérez et al., 2021; Ramač et al., 2022). And, in order to respond to this pressure, software engineers have a tendency to assemble more and code less (Singi, R P, et al., 2019), which, in turn, raises many issues in security and compliance. Moreover, studies show an average 25% of development efforts are wasted on issues resulting from workarounds and technical debts, these include delays in delivery, maintainability challenge, and refactoring (Ramač et al., 2022).

In continuous software development, it is of vital importance to keep technical debts as minimum as possible, in order not to compromise the overall quality of the software in the long run (Besker et al., 2022). Encouraging self-reporting of workarounds and technical debts can help to a great extent to analyze and strategize the quality trade-offs made in software engineering (AlOmar et al., 2022). However, developing such a sense of responsibility among engineers is yet very challenging (Bednar et al., 2019).

A wide range of studies in the literature investigated the drivers behind noncompliance and insiders' violation of software related policies. However, there is a lack of research on workarounds, its causes, and its impact (R. Davison et al., 2019; Song et al., 2020; Wong et al., 2022). Specifically, the challenges related to software engineering, which are not investigated in detail, are of paramount importance, since they are highly involved in the underlying

development and operations of software systems. In addition to that, the gaps between software engineers and compliance experts (Gardazi & Ali, 2017) could lead to misunderstanding and misinterpretation of domain-specific vocabularies and assumptions (Julisch et al., 2011). Moreover, with rapid advancements of software development technologies, more components and software libraries were made accessible online. This, in turn, drives developers to do less coding and more assembling of components, which are available off the shelf (Singi, R P, et al., 2019). While this can speed up development in the short run, the consequences and the technical debts, are more likely to be paid in the long run, and the more the delay in acting, the more sever the consequences are. In this regard, it is crucial to shed the light and understand what can cause workarounds in software engineering, so that we can have a better control over such phenomenon.

The literature reports that software engineers are among very stressed workers (Ostberg et al., 2020). While most of the stress that software engineers experience comes from time pressure, tight deadlines, and complexity of technologies (Yli-Huumo et al., 2015, 2016), another source of stress resulting from the use of technology itself has been found; it is known as technostress (Ragu-Nathan et al., 2008). Technostress reflects one's challenges to deal with evolving technologies. It is viewed to have a negative impact on strain and noncompliance (Nasirpouri & Biros, 2020). In other words, technostress is known to be the dark side of technology, and there is a growing research on technostress and its consequences (Bondanini et al., 2020). However, the extent to which technostress plays in stimulating workarounds has not been investigated in the literature. Previous studies focus

on time pressure and meeting deadlines as main causes of workarounds (Pérez et al., 2021; Ramač et al., 2022).

The objective of this study is to investigate the impact of technostress on software engineers' workarounds, and the moderating role of neutralization, autonomy and perceived behavioral controls on that impact. As per the implications of the systematic review in the Chapter 2, the study focus is on investigating antecedents of workarounds the theory has emerged in software compliance. In this regard, technostress is one of the concepts that has a raising concern on human behavior in which our study argues to impact workarounds, and it includes five subfactors. In total, the study considers eight factors that impact the workaround behavior. Based on that, the paper raises the following research questions:

RQ1: What is the impact of technostress on software engineers' workaround intention and behavior?

RQ2: What is the impact of strain resulting from technostress on the intention to implement workarounds?

RQ3: To what extent does neutralization moderate the relationships between technostress, strain and engineers' intention to implement workarounds?

RQ4: To what degree does engineers' level of autonomy and perceived behavioral control moderate the relationship between technostress, strain and their intention to implement workarounds?

The study uses a survey data of 306 respondents who are working in software engineering fields in South Korea. Results of structural equation modeling using both CB-SEM and PLS-SEM show that technostress

(complexity, overload, and invasion) impact strain, which in turn have a significant impact on workaround intention. Findings also report that technostress (overload and insecurity) directly impact workaround intention. Among all dimensions of technostress, technology overload is found to have a direct and indirect impact on workarounds. Furthermore, results report a significant moderating impact of engineers' autonomy and perceived behavioral control towards workaround intention.

The remainder of this article is structured as follows: Section 3.2 provides a detail review the research gap in the related studies on software compliance and technostress, Section 3.3 describes the methodology followed by this study; Section 3.4 presents a detailed literature review on key concepts and underlying theories; Section 3.5 discusses the theoretical development of the proposed research model, Section 3.6 presents detailed description on the empirical data of the study; Section 3.7 shows the analysis of results, Section 3.8 discusses and reports the findings of the study; Section 3.9 reports the implications; and finally Section 3.10 summarizes the concluding remarks and the contribution of the study.

3.2 Research Gap in Related Work on Software Compliance and Technostress

3.2.1 Overview

Existing research on software policy compliance pays more attention to the context of end users than engineers. D'Arcy et al. (2014) investigated the relationship between user stress that is caused by complex and burdensome security requirements, and violation of information security policies. Their

study reported that users, who perceive stress due to complexity, overload and uncertainty of security requirements are more likely to rationalize violation of software related policies. In a later study, D'Arcy & Teh (2019) concluded that security-related stress is positively associated with emotional reactions, which are represented by frustration and fatigue. The latter, in turn, plays an important role in neutralizing behavioral noncompliance to information security policies. Interestingly, D'Arcy & Teh (2019) also find that neutralization is an unstable phenomenon and can vary from time to time within the same individual.

Nasirpouri & Biros (2020) studied the consequences of technostress and the resulting strains on employees' information security policy violation in the context of end users. Their study confirms that technostress leads to strains and makes users neutralize their violation to software related policies. In particular, among all dimensions of technostress, they found that complexity of technology, invasion of technology into one's life, and job insecurity account for a higher impact on noncompliance intentions.

(Silic et al., 2017) studied the role of neutralization on the use of shadow IT (i.e., tools and software services, which are not authorized by an organization's IT department). Their study finds that neutralization positively correlates to employees' intention to use shadow IT. The study also confirmed a positive impact of the intention to use shadow IT and the actual use by tracking respondents' devices on the software components installed. In such a context, the use of shadow ITs is considered a workaround since the user bypasses the prescribed procedure by using an alternative way which is not approved by the organization (Alter, 2015). But, can workarounds be always

considered a violation to policies? Davison et al. (2021) draw from the work systems theory and argue that workarounds are sometimes required if the existing system is inadequate and fails to support necessary work needs. Similarly, Alter (2015) also argues that, in some cases, excessive compliance can be detrimental and noncompliance can be beneficial.

While there are common practices of workarounds conducted by end users and software engineers alike, there are workaround practices which are peculiar to software developers and engineers. Ward Cunningham (2009) referred to workarounds conducted by developers as technical debts which are tradeoffs in quality to gain short term benefits (Kruchten et al., 2012). Yli-Huumo et al. (2015) investigated workarounds of two software companies; and found that workarounds are mostly intentional and forced by managers' decision of time-to-market. Therefore, workarounds performed by engineers and developers can have different consequences from those performed by end users.

The impact of technostress is more studied in relation to productivity on workplace rather than on an individual level (H. Kim et al., 2016). Previous studies of stress indicate that it is associated with the counterproductive work behavior since it results in burnout and low job performance (Do & Lee, 2020; Jaekang & Taekyung, 2015). Therefore, in information systems context, the consequences of technostress can lead to compliance issues as well (Nasirpouri & Biros, 2020).

Given that insiders are recognized as the highest security threat in an organization, it is crucial to analyze how technostress could lead to deviant behavior. No prior study investigated the relationship between 1) technostress

and workaround behavior; 2) neutralization and workaround behavior; and 3) strain and workaround behavior. Moreover, existing research focuses mostly on end users, there is a lack of research investigating the impact of technostress and neutralization on software engineering workarounds. The following Table 6 summarizes the related works along with their key findings.

Table 6. Summary of Related Studies.

Study	Objective	Findings
(Nasirpouri & Biros, 2020)	Understand the impact of technostress on employees' information security compliance	Use of IT imposes high-level perceptions of technostress creators, making users rationalize their ISP violations and engage in on-compliant behaviors.
(Wong et al., 2022)	Examine employees' workaround behavior in the context of inadequate information systems	- Employees experience difficulties in work if the IS are inadequate, leading to creation of workarounds. - Restrictive policies facilitate creation of workarounds.
(D'Arcy & Teh, 2019)	Investigate emotional reactions to security stress, and how they influence rationalization of ISP violations.	- Security stress is positively associated with frustration and fatigue, which in turn impact neutralization of ISP violations. - Frustration and fatigue make employees continue rationalizations of ISP violations.
(Yli-Huumo et al., 2015)	Investigate workarounds in software industry	Workaround decisions to resolve technical issues are often intentional and forced by time-to-market requirements. - Stakeholders are not always familiar with the negative consequences of workarounds: time, costs, and quality.

Study	Objective	Findings
(Davison et al., 2021)	Investigate how employees react to an enterprise system that does not fit with work processes dictated by local realities.	- Employees improvised workarounds to ensure completion of their work. - Workarounds are coordinated and routinised at the team level and documented in standard procedures retained by local managers.
(Silic et al., 2017)	Examine the role of neutralization and deterrence in discouraging use of shadow IT	Neutralization techniques predict intention to use and actual usage of shadow ITs.
(D'Arcy et al., 2014)	Investigate employee stress caused by complex, and ambiguous security requirements and ISP violation	Employees respond to security related stress by disengaging their internal self-sanctions related to ISP violations, which in turn increases their ISP violation intention.

3.2.2 Research gap

The following Table 7 shows the related studies and the foundational theories and concepts used by these studies. The table also shows which study adapts which of the factors from technostress, strain, neutralization, workarounds and behavioral compliance.

Table 7. Theories and Corresponding Factors Used by the Related Studies

Study	Approach	Method	Sample	Technostress					Strain	Neutralization	Autonomy	Perceived behavioral control	Compliance	
				Complexity	Overload	Uncertainty	Invasion	Insecurity					Compliance behavior	Workarounds
Nasirpouri & Biros (2020)	Quantitative	Survey	End users	✓	✓	✓	✓	✓	✓	-	-	-	✓	-
Wong et al. (2022)	Quantitative	Survey	End users	-	-	-	-	-	-	-	-	-	-	✓
D'Arcy & Teh (2019)	Quantitative	Survey	Computer professionals	✓	✓	✓	-	-	-	✓	-	-	✓	-
(Yli-Huumo et al., 2016)	Qualitative	Interviews	Engineers and managers	✓	✓	-	-	-	-	-	-	-	-	✓
R. M. Davison et al. (2021)	Qualitative	Case study	End users	-	-	-	-	-	-	-	-	-	-	✓
Silic et al. (2017)	Quantitative	Survey	End users	-	-	-	-	-	-	✓	-	-	✓	-
D'Arcy et al. (2014)	Quantitative	Survey	End users	✓	✓	✓	-	-	-	-	-	-	✓	-
Alter (2015)	Qualitative	Comparison	-	-	-	-	-	-	-	-	-	-	✓	✓
This Study	Quantitative	Survey	Software engineers	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓

Technostress has been studied quantitatively in relation to noncompliance behavior by (D'Arcy et al., 2014; D'Arcy & Teh, 2019; Nasirpouri & Biros, 2020). Nevertheless, none of them considered studying the workaround behavior. Among the related studies, only a study by Yli-Huumo et al. (2015) attempted to qualitatively investigate causes of workarounds. The study did not quantitatively evaluate the relationship between technostress and workarounds. However, it provided indications that are related to some of the key dimensions of technostress. Similarly, Davison et al. (2021) conducted a qualitative study to investigate the workaround behavior but did not consider technostress as an antecedent to workarounds. Silic et al. (2017) also did not consider the role of technostress in incentivizing the use of shadow IT. What has not been tackled by the aforementioned studies and worth investigating is whether technostress can impact workaround behavior and how neutralization plays in motivating workarounds.

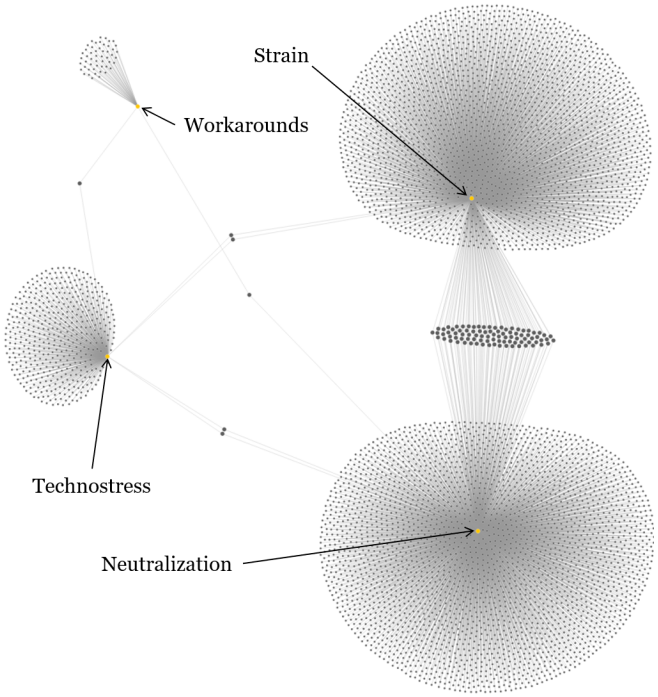

Figure 15. Citation Network of the Underlying Theories and Concepts

To further emphasize the research gap, we conducted a network citation analysis using an online tool named Citation Gecko[2]. The tool is fed with the original articles that grounded these theories to visualize the research articles citing them (Figure.15). The core nodes in the clusters represent these theories namely: Technostress by (Ragu-Nathan et al., 2008), General Strain Theory by (Agnew, 1992) ,the Theory of Workarounds by Alter (2014), and

[2] http://citationgecko.com

Neutralization Theory of Sykes & Matza (1957). Each cluster represents the articles that cite the core theories.

The figure shows the betweenness studies of strain and neutralization. However, there are very few articles that cite technostress and the theory of workarounds. This indicates a lack of research that investigates the relationship between technostress and workarounds.

3.3 Methodology

The study uses a deductive quantitative approach and it is conducted as follows: (1) based on the findings of the systematic literature review, a further investigation of literature is executed along with network citation of the foundational theories in order to elaborate on the research gap; (2) the theoretical development of the hypotheses is conducted with justification and evidence that supports the proposed research model; (3) the measurement instrument of the study is designed based on the literature and then validated with experts and pilot respondents; (4) the survey instrument is translated into Korean language and double checked with experts and the specialized data collection firm[3] in which the data are collected through; (5) the data collection process is then launched through the aforementioned company; (6) data are statistically analyzed and reported using covariance-based structural

[3] https://www.embrain.com/eng/

equation modeling (CB-SEM) with IBM AMOS (version 23), and partial least square (PLS-SEM) with SmartPLS (version 3) to test the hypotheses and compare findings of both; (7) findings are discussed and implications are drawn accordingly. Figure 16 shows the overall flow of the process followed by the study.

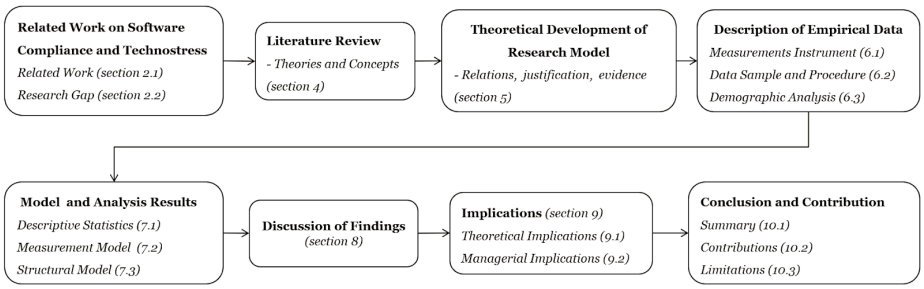

Figure 16. The Overall Steps Followed by the Study

3.4 Literature Review

3.4.1 Technostress:

Advances in information and communications technology (ICT) offer significant benefits for humans and society; however; evidence is growing on the dark negative impact of ICT on organizations and individuals (Bondanini et al., 2020). The increasing use of smart devices and connectivity has become an integral part of daily life for many people. Excessive use of ICT can result in users experiencing a new kind of stress which is referred to as technostress (Tarafdar et al., 2015).

The term Technostress is coined by Brod (1984), a clinical psychologist, who describes technostress as a modern disease caused by one's inability to

cope or deal with ICTs in a healthy manner. Stress in its general definition arises when an individual assesses demands posed by the environment as surpassing his resources and capacity and therefore threatens his wellbeing (Cooper et al., 2001). Most of the recent definitions of technostress associate the term with an organizational context and workplace. Arnetz and Wiholm (1997) define technostress as a state of arousal observed in certain employees who are heavily dependent on computers in their work. Ragu-Nathan et al. (2008) also define it as the stress caused by an individual's attempts to deal with constantly evolving ICTs and the changing physical, social, and cognitive responses demanded by their use. Similarly, Tarafdar et al. (2015) define technostress as the stress that users experience as a result of their use of information systems (IS) in the organizational context.

The constant change is one of the key characteristics of modern information systems; and can cause stress to participants in an organization. Such stress reflects one's challenges to deal with consequences resulting from evolving information technologies. This is referred to as technostress (Ayyagari et al., 2011; Ragu-Nathan et al., 2008). The concept is originally formed in trade literature to describe stressful situations that result due to inability to adapt to new technologies in a healthy way (Tarafdar et al., 2015). Technostress has five different dimensions: overload, complexity, invasion, insecurity, and uncertainty. In this subsection we explain the subconstructs of technostress. These sub constructs are explored in many previous literatures in

the context of end users.

Although, more focus of previous literature on technostress related to end users, the concept can be extended and applies for software developers and engineers as well. From software engineers' perspective, technostress can be viewed similar to end users with variation of impact of its dimensions. In addition to that, software engineers face overwhelming updates in software technologies and best practices which require them to keep up to date, and therefore they are known to be very stressed workers (Ostberg et al., 2020). Stress is caused mostly by time pressure and short time to market, rapidly-changing technological environments, changes in legal requirements (Chilton et al., 2010). While technology lies at the core of this stress, it could lead to negative effects on morale and motivation, cooperation-dependent tasks, communication, software quality, maintenance, and requirements engineering (Ostberg et al., 2020). The following subsections explain in detail the dimensions of technostress.

3.4.1.1 Technology Complexity

Technology complexity can be defined as the degree to which evolving technologies are difficult to understand and deal with (Ragu-Nathan et al., 2008). In other words, the efforts and the time required to figure out and understand various aspects that co-evolve with the technologies (Tarafdar et al., 2015). The complexity can result from the need to upgrade skills and keep up with new technologies. The necessity to keep up with evolving

technologies is also driven by an individual or a business needs in order to stay at a competitive level.

From the perspective of software engineers, complexity arises from the overwhelming and rapidly changing software technologies and environments. This also goes in line with the evolved complexity of software systems according to laws of software evolution (Lehman & Ramil, 2002). The constant need to keep up and develop new skills is frequently required in the software engineering field. Consequently, the time and the efforts needed to go along with new technologies adds up to the stress that software engineers experience.

3.4.1.2 Technology Overload

Technology overload is the degree to which new technologies force users or employees to work much faster and for longer hours. Typically, technology help increase the speed of workflow and forces expectation of more productivity; and hence more workload (Ayyagari et al., 2011). This in turn causes pressure on an individual to meet the expectations of higher workload that technologies bring. The technology overload can also be looked at from a perspective of communications overload (Reinecke et al., 2017). Besides that, the connection and content overload is negatively related to outcomes of using the Internet in everyday life which, in turn, is found significant to perceived stress (LaRose et al., 2014).

As technology speeds up and accelerates business processes, more

workload is generated as a result (Astuti et al., 2018; Devaraj & Kohli, 2003). This in turn allows multitasking with several applications and information processing tasks, as well as time constraining these tasks; leaving insufficient time and attention for accomplishing tasks in creative ways (Tarafdar et al., 2015). The impact of technology overload can be equivalent from the perspective of end users and software engineers, although task estimation in software engineering is more complicated (Wallace, 2015).

3.4.1.3 Technology Uncertainty

Technology uncertainty is defined as an unsettlement of users due to continuing change and upgrade in ICTs. In other words, people need to constantly learn and educate themselves about new technologies (Ragu-Nathan et al., 2008; Tarafdar et al., 2010). While this can be considered context sensitive (Tarafdar et al., 2015), it is obvious that technologies change at a higher rate and the definition of ICT literacy has also become subject to changes. This technology uncertainty is more likely to be accompanied with stress.

Uncertainty in software projects can arise as a result of deficiencies in several areas: contextualizing information, comprehending underlying processes, information on past events and the velocity of changes (Marinho et al., 2015). From a software engineering perspective, uncertainty of technology can be viewed from two different ways: the uncertainty resulting from the evolution of the software project being developed, or the uncertainty due to

new technologies evolving in the industry; and in most cases it presents a threat (Marinho et al., 2018). This two-sided view of uncertainty is expected to worsen the technostress for software engineers.

3.4.1.4 Technology Invasion

Technology invasion refers to a situation where an individual can be reached anytime and anywhere (Tarafdar et al., 2015). With the capabilities provided by smart devices and driven by the need to stay connected, work-related tasks and social interaction keep following an individual wherever he is. Technology invades one's own personal space since, with recent advancement of technology and connectivity, making users are always reachable (Ragu-Nathan et al., 2008). This also creates a need to constantly be connected, which in turn mixes work-related and personal contexts altogether; and consequently, leading to technostress.

Advances in technological capabilities blurs the boundaries between personal life and work; making work-related use of IT during non-working time or vice versa lead to technology invasion (Chen et al., 2022). The context of end users and software engineers can be similar in terms of impact, although the impact on software engineers can be worse compared to end users since they are highly involved in technology. This mixture of work and personal life business which is caused by the technology leads to stress and could be threatful to organizations.

3.4.1.5 Technology Insecurity

Insecurity is the extent to which an individual feel threatened with losing his/her job. This threat is resulting due to the growing automation and advances of information and communications technologies (ICTs), or because of the evolving skills needed in the market (Ragu-Nathan et al., 2008). The fear of being replaced in the job by a more skillful person (Srivastava et al., 2015) or even robots has created a huge amount of stress at workplaces. Although technology has impacted many industries, technology insecurity can vary depending on the industry and the type of profession.

The increase in automation has caused stress to people losing their jobs including IT related jobs (Coupe, 2019). Even though having creative jobs does not change this concern of technology being a threatful to one's job. Studies expect that robots are going to replace many jobs performed by humans, besides that, the advances in artificial intelligence (AI) can be a threat to jobs in the future even for software engineers themselves (Drum, 2017; Ford, 2015).

While most technology related research focuses on what technology does ***for*** people (i.e. the positive impact), it is of highly importance to recognize what technology can do ***to*** people as well (i.e. the negative impact) (Ayyagari et al., 2011), in order to understand the negative consequences of using technology and provide a foundation to address them.

3.4.2 Strain

The general strain theory defines stains as conditions or events which are disliked by an individual or people. Strain could be objective if a condition is disliked by most people in a given group, or subjective if the condition is disliked by a particular person (Agnew & Brezina, 2019). A more specific definition of strain in our context, is the behavioral, psychological and physiological consequences resulting from stress (Kahn & Byosiere, 1992). The transaction theory of stress explains that strain is an individual response to stress in which they are strongly related to each other (Lazarus, 1966). In information systems research, strain is an outcome of stressors that are resulted from different characteristics of technology including its usefulness, complexity, reliability, presenteeism, anonymity and pace of change. These characteristics are found to be related to the aforementioned stressors which are work overload, invasion of privacy, work-home conflict, role ambiguity and job insecurity (Ayyagari et al., 2011).

Psychology studies reported that the physical impact of stress can be burnout, exhaustion, fatigue, restlessness, or irritability of increasing workload because of technology (Arnetz & Wiholm, 1997; Nasirpouri & Biros, 2020). Being an outcome of stress, strain can lead to job dissatisfaction, decrease in performance, lack of creativity and disruptive behavior (Tarafdar et al., 2015). The focus in this study is on strain which is resulted by the use of ICTs.

3.4.3 Workarounds

A workaround is defined as “an improvisation, a goal-driven adaptation, or changes in existing work systems in order to bypass, overcome, reduce the impact of obstacles, anomalies, mishaps or other constraints that prevent participants from achieving a better efficiency or effectiveness” (Alter, 2014). According to the theory of workarounds, the misunderstanding of management intention, designers’ intentions and participants’ goals, interests and values can be the root cause that drives the development of workarounds over time. Although these can be temporary adaptations in the short run, they end up to be routines and transformed into the work system. Eventually, this can lead to consequences on local and broader scopes (Alter, 2014).

There are other definitions of workarounds including three common ones based on a study conducted by Ejnefjäll & Ågerfalk (2019). It could be defined as (1) an action that result from resistance leading to choosing a different path; (2) use of alternative way other than the designed one to accomplish the intended goal; (3) use of alternative path towards the goal when the designed path is blocked (Ejnefjäll & Ågerfalk, 2019). Figure 17 illustrates these definitions of workarounds.

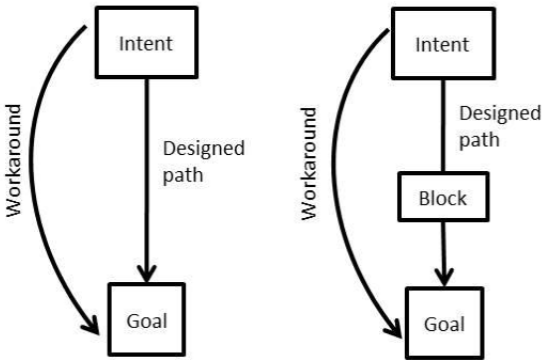

Figure 17. Definitions of Workarounds (Source: Ejnefjäll & Ågerfalk (2019))

In the context of software and information systems, workarounds performed by end users and software can be different to some extent. Typical workarounds performed by end users include manually exporting data from a corporate system and working in a spreadsheet (Aptean, 2018), use of personal removable storage or cloud storage, third-party tools or any other shadow system (Nasirpouri & Biros, 2020). A study by Davison et al. (2021) provides examples of workarounds performed by employees like exporting data from the corporate system, processing it using a spreadsheet, and then sharing it via a removable storage, messaging application or external email system. These kinds of workarounds can jeopardize organizations' most valuable data and could lead to severe consequences.

On the other hand, workarounds performed by software engineers are referred to by Ward Cunningham (2009) as technical debts which are compromises made to gain short term benefits (Kruchten et al., 2012). These compromises are mostly intentional and can be in requirements, compliance,

design, code, architectural, test, dependency, consistency or documentation (Yli-Huumo et al., 2014). Top causes of technical debts are: focus on producing more features at the expense of quality, poor planning and time estimation, lack of qualified professionals, pressure and deadlines, change of requirements, and ineffective project management (C. Becker et al., 2018; Kuutila et al., 2020; Perez et al., 2020). In general, it is argued that engineers have low motivation to deliver beyond functionality, for example privacy design best practices (Bednar et al., 2019). Moreover, business managers care more about meeting deadlines and launching features than the underlying technical details.

There are four quadrants of technical debts which are developed by Martin (2009). These quadrants explain different facets of technical debts performed by engineers. Figure 18 shows that the technical debts can be intentional (deliberate) or accidental; and reckless or prudent. In general, technical debts are more recognizable by engineers compared to other stakeholders who do not see the detailed picture of technical intricacies. While technical debts can be a result of engineering workarounds, both terms were used interchangeably in the literature (Yli-Huumo et al., 2014).

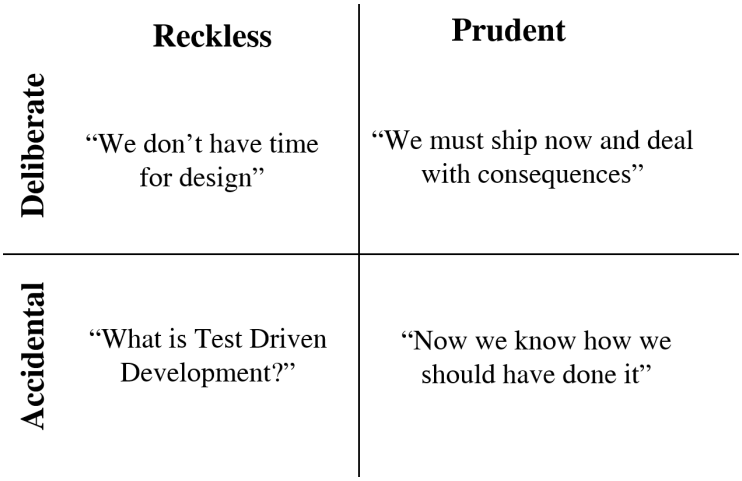

Figure 18. Technical Debt Quadrants[4]

Generally, workarounds are considered noncompliant behaviors since they are not aligned with prescribed policies. Although in some cases workarounds can also be beneficial and excessive compliance can be detrimental (Alter, 2015), it does not indicate that they are always good practices. Yli-Huumo et al., (2015, 2016) argue that those who implement workarounds tend to seek short term benefits and ignore negative consequences in the long run. The negative consequences from software engineering perspective are known as technical debts; which indicate the compromises in quality in exchange for fast delivery of a functionality for instance (Kruchten et al., 2012). Typical consequences of technical debts are

[4] https://praxent.com/blog/brief-history-technical-debt

poor performance, lower maintainability and economic consequences (Buschmann, 2011). Furthermore, workarounds can stimulate more workarounds in future releases if no refactoring decisions are made to avoid that (Yli-Huumo et al., 2015). Davison et al. (2021) argue that workarounds should be seriously tracked and analyzed in order to improve the overall performance and minimize threats at the same time. In this regard, many studies argue that workarounds are deviant and noncompliant behaviors (Beane, 2017; Wong et al., 2022). Therefore, in our context, we consider workarounds as noncompliance behavior.

The context of this paper considers looking at measuring the workaround from the lens of the theory of planned behavior (Ajzen, 1991), in that we look at the intention to use workarounds and the actual workaround behavior. Hence, we define intention to implement workarounds as an indication of one's readiness to perform a workaround behavior. We also define workaround behavior as an actual observable or habitual use of workarounds. The intention in most of the cases is considered as an immediate antecedent of an actual behavior. However, sometimes a behavior can become a routine as a result of a repeated performance and the intention diminishes. In such a case, the intention can be implicit since the behavior is just performed without a conscious intention (Ajzen & Kruglanski, 2019).

3.4.4 Neutralization

Neutralization refers to the justification of deviant behavior and violation of policy. A person normally rationalizes and gives himself a good excuse using different techniques to justify what he did. According to (Sykes & Matza, 1957) theory of delinquency, there are five techniques of neutralization: 1) *denial of responsibility, 2) denial of injury, 3) denial of the victim, 4) condemnation of condemners, and 5) appeal to higher loyalty*. Other techniques were also introduced later, including 6) *defense of necessity* (Minor, 1981) and 7) *defense of ubiquity* (Coleman, 1987). The neutralization theory explains the reasons that lead to violation of policies and excuses made to rationalize that violation (Y. Chen et al., 2012). For our study, we examine denial of injury, condemnation of condemners, defense of necessity, and defense of ubiquity since they are highly relevant to our context.

Denial of injury: a person looks at the harm or injury resulting, in that the delinquent evaluates his wrongfulness based on whether or not anyone has been hurt by his behavior. For example, a person may violate a prescribed policy thinking that by doing so, no harm is done. And therefore, as long as no harm is done, then it is fine enough to justify that behavior. *Condemnation of condemners:* the delinquent redirects the focus from his own deviant act to those who disapprove his violation. A good example of a policy that prohibits use of personal cloud storage and associated delinquent thoughts with a believe that such a policy is not reasonable.

Defense of necessity is used when an individual believes that he/she is out of choices and therefore does not feel guilty towards a certain violation of policies. An example would be when the corporate policy prohibits use of a USB drive and the delinquent thinks that he has no other alternative to accomplish his work. *Defense of ubiquity* is used when the delinquent justifies his deviant behavior by acknowledging that everyone else is doing it. When an individual observes that his surrounding people are doing a certain act, then he just found a good reason for doing so.

The strategies of neutralization vary from one individual to another, and sometimes, even within the same individual at different times (D'Arcy & Teh, 2019). Although the neutralization theory is very mature in criminology research, there is limited research on examining the role of neutralization strategies in the context of software policy compliance (Bansal et al., 2020).

3.4.5 Autonomy

Autonomy refers to one's level of freedom and control on work related decisions; and often referred to as professional autonomy. People in general value autonomy and recognize it as one of the basic needs. However, in a work-related context, such a freedom has to be regulated and controlled (Coeckelbergh, 2006). Professional autonomy is often thought of as professionals can do what they think is good which is a misleading definition. Instead, it refers to the ability and state of being able to make decisions based on principles with which they identify (Coeckelbergh, 2006; Wall & Palvia,

2013).

Crowdsourcing and distributed teams have emphasized towards more on professional autonomy of software engineers, whether on work schedule, technical related decisions, or methods of accomplishing their tasks (Wu et al., 2022). Engineers' level of autonomy can play an important role in delivering out of the box and innovative solutions; however, it can also lead to delivering a low-quality one or deviating from policies. There are tradeoffs when deciding the level of constraints or responsibility given to engineers. The level of responsibility can be determined more by moral reasoning than some principles. We define autonomy in our context as the extent to which an engineer has freedom and control over technical decisions. While this can give more space towards developing better and innovative solutions, it raises high uncertainty in predicting the outcomes (Jeon et al., 2020). Professional autonomy can be valuable to work in some contexts; however, it may not be applicable to some industries where certain standards and constraints have to be in place.

3.4.6 Perceived Behavioral Control

Perceived behavioral control is one of the key determinants in the theory of planned behavior. It refers to one's perception of his ability to perform a certain behavior (Ajzen, 1991). Behavioral control also reflects how easy it is to pursue the target behavior. For example, the availability of resources and opportunities a person has, can dictate, to some extent, the

likelihood of him doing the intended behavior.

Perceived behavioral control is considered the key difference between the theory of reasoned action and planned behavior; and it found to moderate the relationship between the behavioral intentions and the predictors of the behavior (Ajzen, 1991). In other words, tt strengthens the motivation and intention towards a certain behavior (Ajzen & Kruglanski, 2019). According to Ajzen and Kruglanski (2019), if the target behavior is already experienced by an individual, the perceived behavioral control is more likely to moderate the actual behavior.

3.5 Theoretical Development of Research Model

3.5.1 Technostress and Strain

The aforementioned dimensions of technostress (section 3.3.1) can have psychological and physiological consequences on an individual. They influence emotions of an individual, for instance work exhaustion, drain or burnout. In general, literature concluded a positive relationship between stressors and strain (Keenan & Newton, 1985). Nasirpouri & Biros (2020) studied the impact of technostress on stain and noncompliance intention in the context of employees who have a technology background. Their study concluded a positive effect of technostress on individual emotions and strain. In particular, among all constructs of technostress, they found that complexity, invasion and insecurity are strongly associated with the perceived strain and

noncompliance behavior. D'Arcy & Teh 2019) also investigated the impact of security stress on emotional reactions of employee users in order to assess the extent to which they contribute to information security compliance. They observed that security related stress is positively related to frustration and fatigue. Typical symptoms of strain include job dissatisfaction, poor performance and lack of involvement (Ragu-Nathan et al., 2008; Tarafdar et al., 2015). As technology is an integral part of software engineers' jobs, they are known to be very stressed workers (Ostberg et al., 2020), and thus likely to develop strains. While prior mostly considered testing the impact of technostress on strain in the context of end users, no prior study considered the context of software engineers who are at the heart of technology. With this, the following hypothesis is the derived:

H1. *Technostress positively influences strain.*

3.5.2 Strain and Intention to Implement Workarounds

The general strain theory states that an individual who experiences strain or stressors often becomes upset and sometimes copes with a criminal activity. This can lead an individual to engage noncompliant behavior (Agnew & Brezina, 2019). Previous studies have shown that stain is impactful on an individual's behavior and it could lead to adverse outcomes in many industries. A strainful condition or event is more likely seen as unjust when an individual perceives that it leads to intentional violation of a justice norm (Agnew, 2001).

Nasirpouri & Biros (2020) argue that tiredness and the sense of burnout resulting from human-computer interaction increases the likelihood of one's intention to have more engagement in violation of policies. An individual who experiences fatigue exhibits low performance and is more likely to misbehave; or at least too exhausted to give high priority to strictly following prescribed policies. As a result, this leads an engineer to conduct cost-benefit analysis on policy compliance and likely to impact his intention to violate or use workarounds in order to accomplish work related tasks. A more specific view to our context is that engineers are likely to respond to stains by implementing workarounds in various forms. Examples of workarounds include producing low quality code, skipping security settings or poor documentation. With this, we hypothesize that:

H2. *Strain positively influences intention to implement workarounds.*

3.5.3 Technostress and Intention to Implement Workarounds

As technostress leads to emotional consequences and strain, it can also drive participants in the work system to look for an easier and straightforward way to accomplish their tasks, even if they have to skip some of the prescribed policies and procedures. According to the theory of workarounds, stressful situations lead work system participants to bypass some of the prescribed policies which are followed under normal circumstances in order to achieve an immediate goal (Alter, 2014). In some cases, stressful situations

may not necessarily be a result of technology itself. In fact, time pressure is considered the top reason for engineers to pursue workarounds (Yli-Huumo et al., 2015). Our key focus is on the stress which is caused due to one or more of the aforementioned dimensions of technostress (overload, complexity, invasion, insecurity, and uncertainty). We argue that technostress can trigger an individual to consider thinking of workarounds in order to get his task accomplished. As technostress impacts strain, it also plays a positive role in triggering intention to violate prescribed policies (Nasirpouri & Biros, 2020). When engineers experience technostress, there is a likelihood that they implement workarounds. A study by Bednar et al. (2019) argues that engineers tend to have low motivation and lack responsibility to deliver beyond functionality (e.g. ethic-oriented practices such as Privacy by Design). In some other cases, complexity of technology drives engineers to develop workarounds to get the job done on time (Yli-Huumo et al., 2015). Evidence from literature supports that stressful situations resulting from security requirements can provoke noncompliance behavior (D'Arcy & Teh, 2019). Similarly, the amount of stress resulting from technology increases the likelihood of engineers to bypass policies and use workarounds. With this, we hypothesize that technostress can stimulate intentions to use workarounds.

H3. *Technostress positively influences intention to implement workarounds.*

3.5.4 Intention to Implement Workarounds and Workaround Behavior

It is crucial to draw a distinction between the intention to commit a certain behavior and the actual behavior. Although the intention can have a strong indication of one's motivation to perform a certain act (Silic et al., 2017), it does not always lead to actual behavior (Ajzen, 1991). The relationship between intentions and behavior is theorized in the theory of planned behavior which is derived originally from the theory of reasoned action (Ajzen, 1991). Though, there are discrepancies in the relationship especially when the behavior is directed more by the active goals than the intention to perform that behavior. In such cases, the intention becomes implicit and the behavior is more driven by habits and active goals (Ajzen & Kruglanski, 2019). Nevertheless, there are many studies in the literature that have empirically tested the impact of the intention on the behavior in theories of reasoned actions, planned behaviors, and reasoned goal pursuit. In our context, if a person has an intention to perform a workaround, it makes more sense that he will actually do it, assuming that they have the ability to do so. Hence, we hypothesize:

H4. *Intention to implement workarounds positively influences workaround behavior.*

3.5.5 The Moderating Role of Neutralization

Studies that discuss the relationship between strain and neutralization argue that the strain fosters adoption of beliefs which are favorable to criminal

activity provided that there exists some form of reasoning in one's mind. Such a relationship is not well tested in academic research (Froggio et al., 2009). Agnew & Brezina (2019) argue that there is a relationship between strain, anger and delinquency. However, it all depends on the types of strains and the corresponding neutralization technique may differ. The study of Froggio et al. (2009) found a positive effect of strain and neutralization techniques on deviant behavior. The study also finds that the level of impact is stronger in minor crimes and weaker in major ones. Another study by Lim (2005) found that neutralization has an important moderating impact in the relationship between organizational injustice and the act of cyberloafing. In our context, the fatigue and burnout resulting from technostress can increase the likelihood of using workarounds. And the degree of impact becomes stronger as an engineer uses neutralization to rationalize his/her behavior. As software engineers are among the most stressed workers, the resulting strain can affect their morale and motivation (Ostberg et al., 2020). Besides that, by limiting one's thoughts, strain makes an individual less creative and thinks more of mental justification to cope with strain and drive to rationalize their noncompliant behavior (D'Arcy & Teh, 2019). Therefore, the strength of the relationship between strain and the intention to implement workarounds is moderated by the degree of neutralization.

Ethical theories argue that neutralization allows rationalization of noncompliant behavior in that they provide justification of irresponsibility

using one of the aforementioned strategies explained in the previous section. The neutralization theory is considered as one of the good predictors of noncompliant behavior (Barlow et al., 2013; S. H. Kim et al., 2014). Few studies have examined the relationship between technostress and neutralization theory. The study of D'Arcy & Teh, (2019) uses the coping theory to investigate the stress resulting from security complexity in information systems and how it impacts users' emotions and use of neutralization techniques. Another study by Gwebu et al. (2020) concluded a positive impact of neutralization on noncompliance behavior. The study also points out that neutralization is very common in the context of digital piracy and information security noncompliance. In assessing the extent to which neutralization techniques can play as a predictor of planned behavior, Bauer & Bernroider (2014) confirmed that neutralization is at least of equal importance as other predictors of the theory of planned behavior. Their study found a positive relationship between neutralization and intention towards the desirable information security behavior.

Most engineers advocate much of the responsibility of poor quality to somebody else, therefore neutralize workarounds by refraining from responsibility (Dave Farely, 2021). As we consider workarounds as noncompliant behavior since an engineer who uses workarounds violates prescribed policies anyway, we argue that neutralization moderates the impact of technostress on workarounds. It is also worth mentioning that workarounds

are not necessarily individual improvisations, but they can also be developed collectively and become unofficial local rules (Malaurent & Karanasios, 2020). This collective use of workarounds is easily justifiable by neutralization techniques mentioned before and can eventually become the norm among a group of engineers. While stress in general can have a detrimental impact on one's morale (Ostberg et al., 2020), neutralization can provide a fertile ground for justifying any violation of policies. In this regard, neutralization is likely to strengthen or weaken the relationship between technostress and the intention to implement workarounds.

H5.a. *Neutralization moderates the impact of strain and intention to implement workarounds.*
H5.b. *Neutralization moderates the impact of technostress and intention to implement workarounds.*

3.5.6 The Moderating Role of Autonomy

Autonomy refers to one's level of freedom and control on work related decisions. Engineers' level of autonomy plays an important role in delivering out of the box innovative solutions. According to Coeckelbergh (2006), less control and more trust are expected to improve professional autonomy and enhance quality of services delivered by engineers. On the other hand, regulations and constraints aim at preventing more incidents resulting from the freedom given over technical decisions. While constraining can ensure consistency and predictability of technical solutions; it also indicates that more freedom could result in more uncertainty as well. These two directions

have advantages and disadvantages since issues and consequences related to constraints and autonomy are both complex. For instance, an action of an engineer could be driven by either gains for himself, loyalty to his organization or rules of universal justice. The level of autonomy can be subject to peculiarities and differences of unique situations in that some projects place risk minimization as a top priority and therefore demand more constraints (Coeckelbergh, 2006; Wall & Palvia, 2013).

Previous studies concluded a moderating role of autonomy on the relationship between compliance intention and its antecedents (Jeon et al., 2020). In our context, we argue that the level of autonomy given to engineers can impact their intention to implement workarounds. It can also moderate the relationship between strain and neutralization on one hand, and intention to use workarounds on the other hand. In other words, the impact of strain and neutralization on workarounds is moderated by the level of autonomy that engineers have. When engineers have freedom on technical decisions, they are more likely to have many alternative ways to solve problems; and the decision to select is made up to them. Contrary to that, when engineers are restricted on technical related decisions, they are more likely to be constrained and have fewer alternatives, and therefore, less chances of implementing workarounds. With this, we hypothesize that the level of impact posed by technostress and strains can vary depending on the level of autonomy that engineers have. Hence:

H6.a. *Autonomy moderates the impact of strain and intention to implement workarounds.*
H6.b. *Autonomy moderates the impact of technostress and intention to implement workarounds.*

3.5.7 The Moderating Role of Perceived Behavioral Control

Perceived behavioral control refers to one's perception of his ability to perform a certain behavior. It reflects how easy it is to pursue the intended behavior (Ajzen, 1991). An individual might have an intention to pursue a certain behavior, however, if he does not possess the required abilities and means to do so; then they are more likely to not to perform that behavior. In our context, the perceived behavioral control refers to an engineer's perception of their control and ability to use workarounds. Prior studies have concluded that perceived behavioral control plays a critical role in driving the intention to act towards behavior (Bulgurcu et al., 2010). In this regard, we expect that the perception of engineers' ability to use workarounds can increase their intention to implement and use workarounds. In addition to that, and based on the theory of planned behavior, the perceived behavioral control is expected to strengthen the relationships towards the intention. Therefore, we hypothesize that the impact of technostress and strains on one's intention to implement workarounds vary depending on the behavioral controls that engineers perceive.

H7.a. *Perceived behavioral control moderates the impact of strain on intention to implement workarounds.*

H7.b. *Perceived behavioral control moderates the impact of technostress on intention to implement workarounds.*

Drawing from technostress, general strain theory, neutralization theory, planned behavior, and the theory of workarounds; the research model in Figure 19 shows the factors and the hypothesized relationships. The overall layout of the theoretical model is based on the theory of planned behavior.

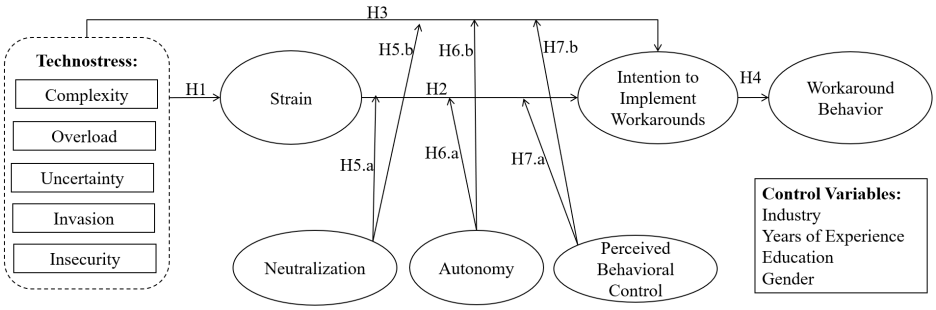

Figure 19. Proposed Research Model

Besides the hypotheses developed, we consider a set of control variables that could be seen to have an impact on the overall analysis of results. These are: industry, years of experience, education and gender. The control variables were selected based on literature (Liu et al., 2020; Merhi & Ahluwalia, 2019). Other control variables such as salary is excluded due to emphasis of various studies which conclude that employees are found to be less motivated by extrinsic variables including salary (Buelens & Van den Broeck, 2007; X. Chen et al., 2018). In addition to that, software engineers are considered among the highly paid professions worldwide (Tekla S. Perry, 2022). Therefore, the salary is excluded from the list of control variable.

3.6 Description of Empirical Data

3.6.1 Measurements Instrument

The measurement of constructs is adapted from the literature. Each of the constructs which are specified in the proposed model contains three or four items quantified using Likert scale of 7 points ranging from (Strongly Disagree) to (Strongly Agree). The detailed measurement instrument along with the translation in Korean is provided in the appendix (3).

Technostress is measured using five constructs (complexity, overload, uncertainty, invasion, and insecurity) adapted from Ragu-Nathan et al. (2008) and Nasirpouri & Biros (2020). Within each of these constructs, three to four questions developed in order to quantify the overall factor of technostress.

Strain is measured using four items adapted from Nasirpouri & Biros (2020), in that respondents were asked to convey their feeling when dealing with different types of new technologies or involved in technology related activities.

Intention to implement workarounds is measured using three items adapted from the theory of planned behavior of Ajzen (1991) and the theory of workarounds (Alter, 2014); in that respondents were asked on their intention to use workarounds before performing the actual workarounds.

Workaround behavior is measured using four items adapted from Alter (2014); Ejnefjäll & Ågerfalk, (2019); Laumer et al., (2017); Wong et al., (2022), in that the derived measurement questions are directly related to the

actual workaround behavior that respondents habitually report at their current time.

Neutralization is measured using five items that correspond to four neutralization techniques namely: denial of injury, condemnation of condemners, defense of necessity, and defense of ubiquity. We adapted the measures introduced by D'Arcy & Teh (2019); Siponen & Vance (2010).

Autonomy is measured using three items adapted from Coeckelbergh (2006), in that respondents were asked to express their level of freedom over technical decisions.

Perceived behavioral control is measured based on three items adapted from the theory of planned behavior of Ajzen (1991).

Demographic information of respondents is also collected with regard to respondents' industry, profession, years of experience, education, and gender. For these items, the options are adopted from a widely known standards of classification (see appendix 3).

3.6.2 Data Sample and Procedure

Data are collected through a data collection firm[5] located in South Korea. The targeted respondents are software engineers who have at least three years of professional work experience in software engineering areas.

[5] https://www.embrain.com/eng/

There are two reasons for selecting Korea as a case for testing the proposed research mode; (1) with high penetration rate of smart devices and network infrastructure index (UN, 2020), the impact of technology on an individual becomes stronger and, in turn, provides a better insights on the dimensions of technostress; (2) Korean software market is very competitive and big attention has been recently paid on the new and evolving software technologies including cloud, big data, artificial intelligence and internet of things (Yoon et al., 2021); therefore, this makes it a good case for conducting this study.

The field of the target respondents are as follows: software engineering, system analysis, consulting, system/network security, system programming, web programming, and application programming. We also consider balancing the number of respondents based on gender, age and years of experience in order to provide a detailed analysis on how such control variables can impact the overall results of the model.

Before proceeding with data collection, the measurement instrument is validated with 9 experts from academic and professional fields. Accordingly, enhancements are made on the survey questions to improve readability and understandability. In the next step, the survey questions were translated into Korean language and cross validated with professional linguistics and field experts. After validating, the instrument is then sent to respondents through a data collection firm located in Korea. A total of 306 valid survey responses has been obtained in professions related to software engineering.

3.6.3 Demographic Analysis of Respondents

The distribution of demographic data based on gender, age, and education is shows in Table 8. With regard to gender, there is a slightly more male (60.8%) respondents than female (39.2%). The majority of respondents age between 30 to 50 years old. The education level of majority of respondents is bachelor degree.

Table 8. Demographic Characteristics of Sample

Gender	Freq.	Age	Freq.	Education	Freq.
Male	186 (60.8%)	20 < 30	47 (15.4%)	High school or equivalent	12 (3.9%)
Female	120 (39.2%)	30 < 40	96 (31.4%)	Junior college graduate	41 (13.4%)
		40 < 50	96 (31.4%)	Bachelor degree	215 (70.3%)
		50 < 60	40 (13.1%)	Master degree	37 (12.1%)
		> 60	27 (8.8%)	Doctoral degree or above	1 (0.3%)

Regarding the professional characteristics of the sample, Table 9 shows the distribution based on the types of profession and years of experience. The majority of respondents have over 13 years of experiences in fields related to software engineering.

Table 9. Professional Characteristics of the Sample

Profession	Frequency	Work Experience	Frequency
Software Engineering, system analysis, consulting	68 (22.2%)	3 - 5 years	49 (16%)

Profession	Frequency	Work Experience	Frequency
System, network, security	73 (23.9%)	5 - 7 years	33 (10.8%)
System programming	19 (6.2%)	7 - 9 years	27 (8.8%)
Web programming	97 (31.7%)	9 - 11 years	39 (12.7%)
Application programming	49 (16%)	11 - 13 years	19 (6.2%)
		More than 13	139 (45.4%)

The distribution of respondents based on industries they are working in, is presented in Figure 20. The sample shows nearly 70% of respondents work in the information technology (IT) industry. While this can drive the analysis and implication a more focus towards the IT industry; it can also provide a better representation of the sample working in software engineering professions.

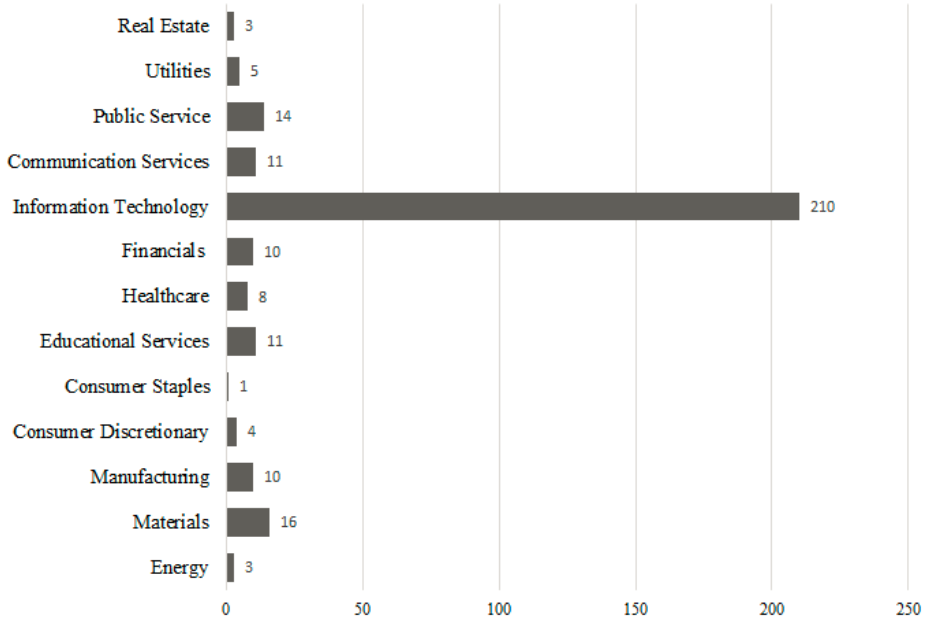

Figure 20. Industries and Corresponding Number of Respondents

Figure 21 shows the distrinution of the firm sizes of the respondents based on the number of employees their firm have. This classification of firm sizes is adapted from OECD (OECD, 2017). The firm sizes of the respondents ranges micro entreprizes to large enterprises, with a majority are from meduim and large entreprizes.

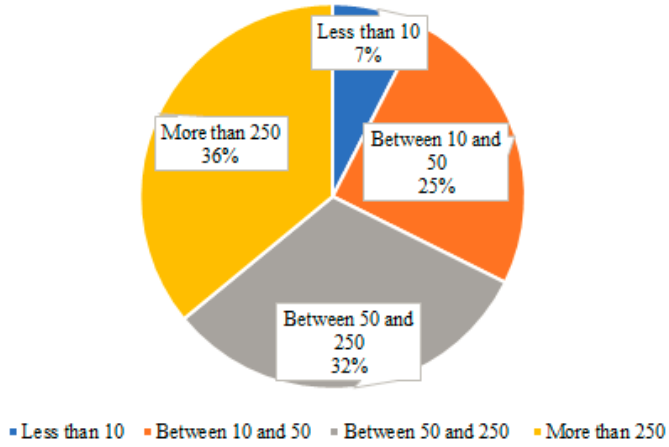

Figure 21. Firm Sizes of Respondents Based on Number of Employees

3.7 Research Model Analysis Results

In order to test the hypotheses, we used structural equation modeling (SEM). SEM is similar to multiple regression in the sense that both techniques test relationships between variables, SEM is able to simultaneously examine multi-level dependence relationships, "where a dependent variable becomes an independent variable in subsequent relationships within the same analysis" in addition to relationships between multiple dependent variables (Shook et al., 2004).

For analyzing the structural model and test the hypotheses, this study applies both covariance-based structural equation modeling (CB-SEM), and partial-least square structural equation modeling (PLS-SEM). For the CB-SEM, IBM SPSS and AMOS are used to analyze the data and assess the measurement and the structural model. To analyze the factors and the

relationships, CB-SEM is preferable for model assessment which is built on well-defined overarching theoretical lens, as well as for analyzing second-order constructs (MacKenzie et al., 2005). In addition to that, we also applied PLS-SEM to further test and confirm the results, in order to build strong conclusions based on both methods.

3.7.1 Descriptive Statistics

The results of the descriptive statistics of the factors and their measurement items are shown in Table 10. The mean and standard deviation (S.D) are considered a basic information that shows the interval and the shape of sample distribution and how far are data from the mean value (Sekaran & Bougie, 2010). Data are normally distributed around their mean indicating that the data describes well majority of the sample. The lowest S.D is with the Perceived Behavioral Control (1.04) and highest S.D is for Strain (1.29). Strain and Technology Invasion show highest S.D, indicating that responses of the survey are slightly sparser from the mean compared to others.

Table 10. Factors and their Descriptive Statistics

Role	Factors	Number of Items	Mean	S.D
Independent	Technology Complexity	3	4.36	1.07
	Technology Overload	4	4.36	1.20
	Technology Uncertainty	3	4.47	1.21
	Technology Invasion	4	3.74	1.25
	Technology Insecurity	3	4.45	1.13

Role	Factors	Number of Items	Mean	S.D
Mediator	Strain	4	4.15	1.29
	Intention to Implement Workarounds	4	4.12	1.06
Dependent	Workaround Behavior	4	4.01	1.05
Moderators	Neutralization	5	3.12	1.18
	Autonomy	3	4.12	1.05
	Perceived Behavioral Control	3	3.88	1.04

3.7.2 Measurement Model

The measurement model is evaluated for testing its reliability through estimation of factor loading, composite reliability (C.R), average variance extracted (AVE), and discriminant validity. Such measures can provide the study with a confidence on how reliable the factors and items are before proceeding with testing the hypotheses.

3.7.2.1 Reliability and Validity

In order to assess the reliability and the validity of the main model, excluding the moderator factors, Table 11 shows the factor loading of the items, the composite reliability (C.R), and the average variance extracted (AVE). The loading indicates the correlation between the items of the factor. The acceptable threshold of loading is greater than or equal 0.5 (Byrne, 2013),

items below that were eliminated. The composite reliability (C.R) represents the internal consistency of the items, and it should be greater than 0.7. In the analysis (Byrne, 2013), The last reliability checking is the AVE, which indicates the level of variance obtained by a factor in relation to the amount of variance due to measurement error. The acceptable threshold for AVE is above 0.5 (Byrne, 2013) in which all the main factors in the study achieved above that threshold. In summary seven factors out of eight passed the reliability checking.

Table 11. Reliability and Validity of the Main Model

Factor	Items	Loading	C.R.	AVE	Cronbach α
Technology Complexity	TCX2	0.536	0.758	0.521	0.748
(TCX)	TCX3	0.893			
	TCX4	0.692			
Technology Overload	TOV1	0.796	0.920	0.741	0.918
(TOV)	TOV2	0.91			
	TOV3	0.873			
	TOV4	0.861			
Technology Uncertainty	TUC1	0.774	0.872	0.695	0.869
(TUC)	TUC2	0.908			
	TUC3	0.813			
Technology Invasion	TNV1	0.751	0.890	0.671	0.889
(TNV)	TNV2	0.81			
	TNV3	0.835			
	TNV4	0.875			
Technology Insecurity	TNS1	0.85	0.790	0.561	0.789
(TNS)	TNS2	0.592			
	TNS3	0.782			

Factor	Items	Loading	C.R.	AVE	Cronbach α
Strain (ST)	ST1	0.872	0.945	0.810	0.944
	ST2	0.916			
	ST3	0.884			
	ST4	0.927			
Intention to Implement	IWA1	0.631	0.740	0.618	0.863
Workaround	IWA2	0.839			
(IWA)	IWA3	0.861			
	IWA4	0.794			
Workaround Behavior	WAB1	0.825	0.740	0.650	0.882
(WAB)	WAB2	0.749			
	WAB3	0.812			
	WAB4	0.837			

Reliable: If CR > 0.70 and AVE > 0.50, *C.R: composite reliability, AVE: the average variance extracted.*

3.7.2.2 Discriminant validity

Discriminant validity refers to the extent that a factor is different from other factors (Henseler et al., 2015). Table 12 shows the correlation of each factor with the rest of other factors. The values represent the square root of AVE which is explained before. The result shows that all factors passed the discriminant validity with a relatively close correlation between intention to implement workarounds (IWA) and the workaround behavior (WAB). While their result is valid, perhaps respondents understand the questions of both factors in a similar way. This challenge is also raised in the theory of planned behavior as there is a difficulty in measuring the intention and the actual behavior simultaneously at the same time (Ajzen & Kruglanski, 2019).

Table 12. Discriminant validity

	TCX	TOV	TUC	TNV	TNS	ST	IWA	WAB
TCX	0.722							
TOV	.466**	0.861						
TUC	.119*	.343**	0.834					
TNV	.231**	.663**	.343**	0.819				
TNS	.481**	.609**	.224**	.485**	0.749			
ST	.453**	.688**	.225**	.557**	.608**	0.900		
IWA	.284**	.468**	0.099	.349**	.479**	.497**	0.786	
WAB	.316**	.438**	0.102	.328**	.414**	.453**	.783**	0.806

If square root of AVE > inter-construct correlations; TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior

3.7.2.3 Model Fit Measures

The following Table 13 describes the reliability information of the whole model and how the dataset fits with the model. If chi-square value (CMIN) divided by the degree of freedom (DF) represented is less than or equal $\leq$ 3; it indicates that the model fit is acceptable fit (Kline, 2015). The goodness of fit index (GFI) represents how well the model fits the data. While the perfect GFI value is 1; the value above 0.8 is considered acceptable (Baumgartner & Homburg, 1996). The comparative fit index (CFI) evaluates the discrepancy between the data and the hypothesized model. When its value is closer to 1, it shows a very good fit while the value of 1 is considered a perfect fit.

Table 13. Model Fit Indices

Fit Statistics	Result	Acceptable Values
CMIN	823.447	-
df	355	-
p	0.000	Insignificance > 0.05 (Jöreskog & Sörbom, 1996)
CMIN/df	2.320	≤ 3 (Kline, 2015)
GFI	0.839	> 0.8 (Baumgartner & Homburg, 1996)
CFI	0.926	0 > 1 (Hu & Bentler, 1998)
RMSEA	0.066	0.05 > 0.08 (Fabrigar et al., 1999)
SRMR	0.0533	≤ 0.08 (Hu & Bentler, 1999)

CMIN := chi-square fit statistics; df := degree of freedom; GFI := goodness-of-fit index; CFI := comparative fix index; RMSEA := root mean square error of approximation; SRMR := standardized root mean square residual.

Root mean square error of approximation (RMSEA) measures the differences between the observed covariance matrix per degree of freedom and the predicted covariance matrix (F. F. Chen, 2007). The values between 0.05 and 0.08 are considered acceptable (Fabrigar et al., 1999). The results of model fit indices indicate that the mode is good enough to test the hypotheses.

3.7.3 Structural Model

In order to test the hypotheses against the data modeled, the analysis applies covariance based structural equation modeling (CB-SEM) as well as partial least square structural equation modeling (PLS-SEM) in order to double check the results of both and develop a comparison on the results obtained. While CB-SEM is preferable when the hypotheses are built based on sufficient evidence and established theoretical foundation (Astrachan et al.,

2014), PLS-SEM is preferable when testing new relations and evolving concepts which have not theoretically matured (Dash & Paul, 2021; Jr et al., 2017). This section presents the results of testing hypotheses using CB-SEM and PLS-SEM; and provides a comparison on the results obtained to solidify the findings and the discussion.

3.7.3.1 Covariance-Based Structural Equation Modeling (CB-SEM) Hypotheses Testing

Using the results of structural equation modeling generated by IBM AMOS (version 23), it is possible to test whether the hypotheses argued in the study are to be supported or not. Table 14 shows the final results of each hypothesis, its path, estimate, standard error (S.E) and the p-value. Based on the significance level represented by the p-value, the hypothesis can either be accepted or rejected. In general, whenever the p-value is less than or equal 0.05; the hypothesis is considered accepted. Details are shown in the legend of Table 14.

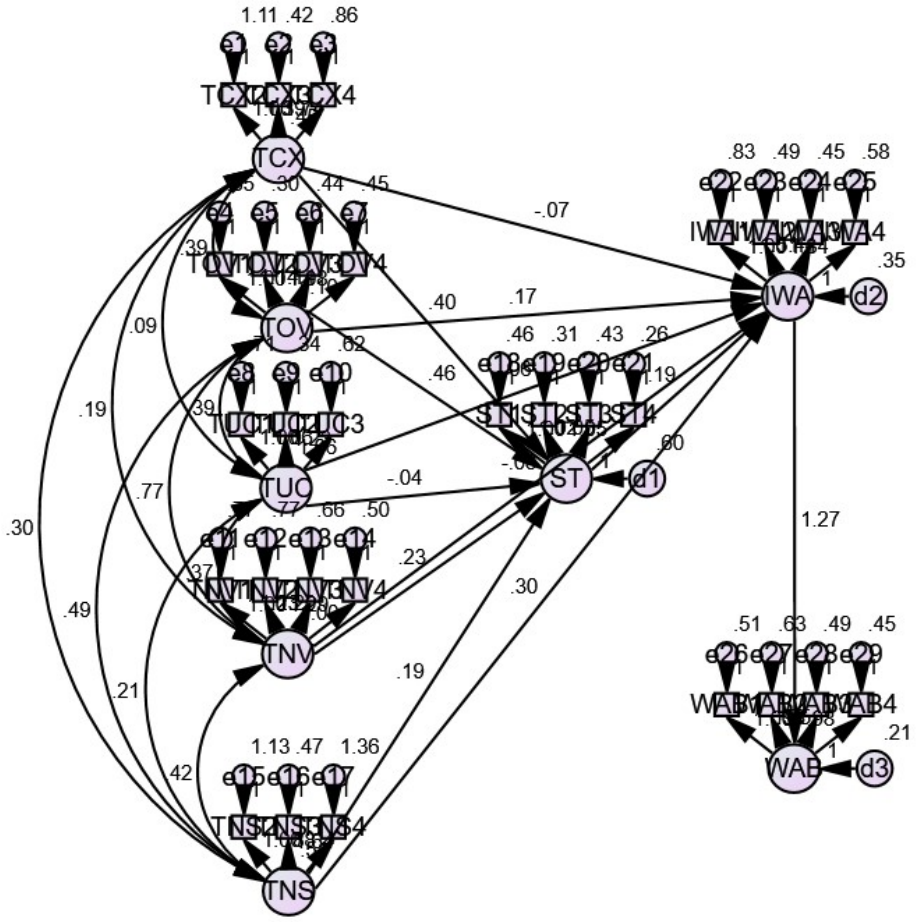

Figure 22. Visualization of Output Results from AMOS

Table 14. Main Hypotheses Testing (Results of CB-SEM)

Hypotheses	Path	Estimate	S.E.	P-value	Result
H1-1	TCX → ST	0.222	0.140	0.004**	Supported
H1-2	TOV → ST	0.399	0.103	0.000***	Supported
H1-3	TUC → ST	-0.036	0.056	0.451	Not Supported
H1-4	TNV → ST	0.192	0.099	0.019*	Supported

Hypotheses	Path	Estimate	S.E.	P-value	Result
H1-5	TNS → ST	0.113	0.145	0.188	Not Supported
H2	ST → IWA	0.302	0.055	0.000***	Supported
H3-1	TCX → IWA	-0.060	0.106	0.530	Not Supported
H3-2	TOV → IWA	0.241	0.081	0.036*	Supported
H3-3	TUC → IWA	-0.089	0.043	0.138	Not Supported
H3-4	TNV → IWA	-0.101	0.077	0.327	Not Supported
H3-5	TNS → IWA	0.289	0.118	0.011*	Supported
H4	IWA → WAB	0.901	0.115	0.000***	Supported

TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.
* := p < .05; ** := p < .01; *** := p < .001.

The result of the main hypotheses indicates that Technology Complexity (TCX), Technology Overload (TOV), and Technology Invasion (TNV) have a significant effect on Strain (ST), while Technology Uncertainty (TUC) and Technology Insecurity (TNS) reported no impact on Strain (ST). TUC is considered context sensitive (Tarafdar et al., 2015) and highly dependent on specific domain or industry, while TNS does not necessarily trigger stain. Strain, in turn, is found to have a significant impact on the Intention to Implement Workarounds (IWA). With regard to the impact of technostress dimensions on the workaround intention, the results show that TOV and TNS have a direct impact on IWA. The result of hypothesis testing also shows that the impact of IWA on WAB is significant.

Indirect Effect

The indirect effect indicates that the impact of a factor on the other is mediated by another factor in between in causality relationships (Alwin & Hauser, 1975). Table 15 shows the analysis results of the indirect effect of technostress dimensions on IWA and WAB. The analysis reports an indirect effect of technology complexity (TCX) and technology overload (TOV) on the intention to implement workarounds (IWA). TOV also shows an indirect impact on the workaround's behavior. Technology insecurity (TNS) also has an indirect impact on the workaround behavior. Lastly, the strain is found to have an indirect impact on the workaround's behavior. The overall results indicate that workarounds can be triggered indirectly by technostress factors (complexity, overload and insecurity).

Table 15. Results of Indirect Effect (Results of CB-SEM)

	IWA		**WAB**		**Supported Indirect Impact on**
	Estimate	**p-value**	**Estimate**	**p-value**	
TCX	0.067*	0.023	0.006	0.923	IWA
TOV	0.121**	0.003	0.326*	0.013	IWA, WAB
TUC	-0.011	0.417	-0.09	0.162	-
TNV	0.058	0.058	-0.039	0.783	-
TNS	0.034	0.281	0.291*	0.018	WAB
ST			0.272**	0.004	WAB

TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNV: Technology Invasion, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.
** := p < .05; ** := p < .01; *** := p < .001.*

Moderating Effect

The moderating effect evaluates the extent to which a factor moderates a relationship between an independent factor and a dependent one. It examines how a factor can strengthen or weaken that relationship (Byrne, 2013). The results of the moderating effect presented in hypotheses H5, H6, and H7 are summarized in Table 16. In order to test the moderating impact, the moderation and the interaction impact had to be calculated. This can be done through estimating the impact of the independent and the moderation factors on the dependent one; as well as estimating the impact of the interaction effect. The interaction effect is estimated through multiplication of the independent and the moderator factors, and then measuring the significance level of that result on the dependent factor. For the moderating effect, the most important value that indicates a moderation effect is the interaction effect (Byrne, 2013) as shown in Table 16. If the p-value of the interaction impact is less than or equal 0.05, it indicates there is a moderating impact.

Table 16. Moderating Effect (Results of CB-SEM)

Moderator	H	Path	Moderator Effect (P-value)	Interaction Effect (P-value)	Result
Neutralization	H5a	ST → IWA	0.817**(0.000)	-0.310 (0.148)	Rejected
	H5b	TOV→ IWA	0.70*** (0.000)	-0.065* (0.013)	Supported
		TNS→ IWA	0.798* (0.014)	-0.077 (0.122)	Rejected
Autonomy	H6a	ST → IWA	-0.053 (0.576)	0.496**(0.000)	Supported

Moderator	H	Path	Moderator Effect (P-value)	Interaction Effect (P-value)	Result
	H6b	TOV→ IWA	-0.113 (0.101)	0.062*** (0.000)	Supported
		TNS→ IWA	0.043 (0.576)	0.027* (0.031)	Supported
Perceived	H7a	ST → IWA	0.157 (0.282)	0.420** (0.004)	Supported
Behavioral	H7b	TOV→ IWA	0.165 (0.216)	0.031 (0.115)	Rejected
Control		TNS→ IWA	0.656 (0.139)	-0.050 (0.445)	Rejected

ST: Strain, IWA: Intention to Implement Workarounds; TOV: Technology Overload, TNS: Technology Insecurity.
** := p < .05; ** := p < .01; *** := p < .001.*

Based on the analysis estimation of significance, Neutralization moderates only the impact of technology overload (TOV) on the intention to implement workarounds (IWA). On the other hand, Autonomy is found to significantly moderate the impact of ST, TOV, and TNS on the IWA. This indicates that, the more autonomy given to engineers, the more they are likely to implement workarounds, provided that they feel strained, overloaded, or insecure about their jobs. The Perceived Behavioral Control is also found to have a significant moderating impact on the relationship between ST and IWA. This indicates that the level of control given over technical decisions and engineers ability to use various alternatives could increase the likelihood of them implementing workarounds whenever they perceive a strain resulting from technostress.

Control Effect

The control effect measures the overall impact of a variable on the dependent factor (T. E. Becker, 2005). Based on our analysis of the data,

respondents belong to various working contexts including different firm sizes, different years of experience and different industries. The analysis shows that there is no control effect of industry, years of experience, education, and gender. Although there is a relatively balanced distribution of the dataset regarding gender and years of experience and we expected different perceptions based on these groups of respondents, the control impact of these is found insignificant. Previous studies on information systems security compliance also report an insignificant control effect of gender, age, experience, education, industry, firm size, and job type (X. Chen et al., 2018; Merhi & Ahluwalia, 2019). Similarly, none of the specified control variables is found to have a significant impact on the overall results of the model. In behavioral compliance, extrinsic variables are found less impactful and less motivating towards compliance or noncompliance behavior (Buelens & Van den Broeck, 2007; X. Chen et al., 2018).

3.7.3.2 Partial Least Square Structural Equation Modeling (PLS-SEM)

Hypotheses Testing

Using the results of partial least square structural equation modeling generated by SmartPLS (version 3), it is also possible to test whether the hypotheses posed by the study are to be supported or not (Fig. 23). Table.13 shows the final results of each hypothesis, its path, estimate, standard error (S.E) and the p-value. Based on the significance level represented by the p-

value, the hypothesis can either be accepted or rejected. In general, whenever the p-value is less than or equal 0.05; the hypothesis is considered accepted.

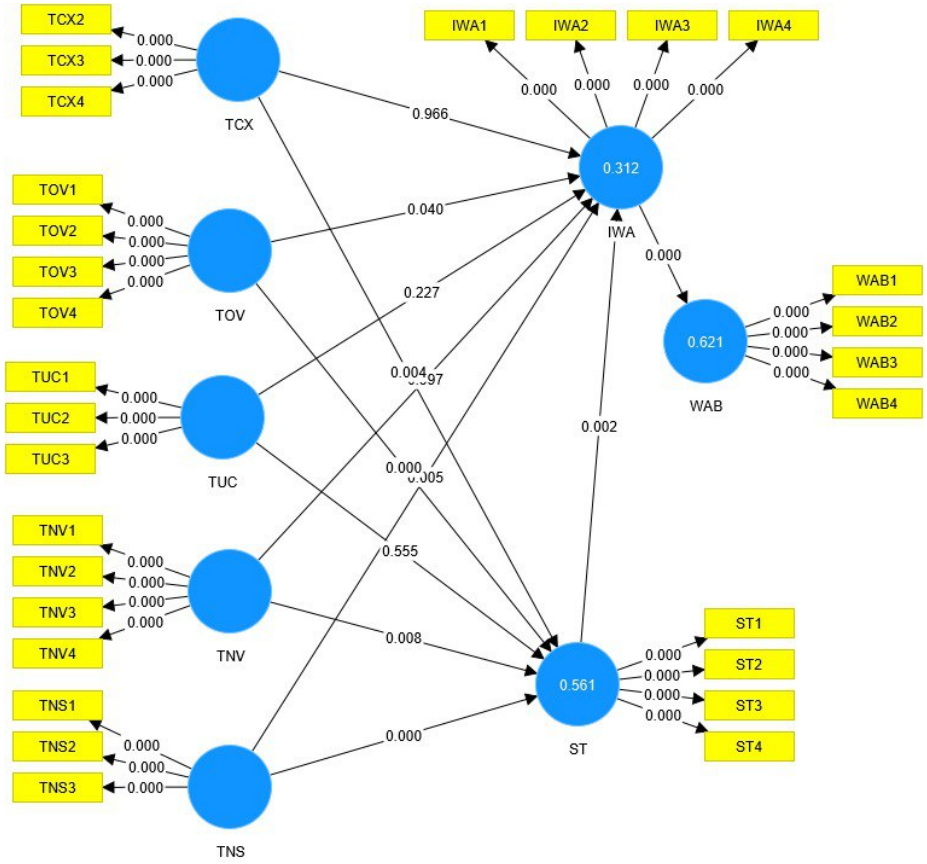

Figure 23. Output Results of SmartPLS Model

The results of PLS-SEM are similar to CB-SEM except for one path in that the hypothesis testing of PLS shows that technology insecurity (TNS) is significant to strain (ST), whereas CB-SEM result show that TNS is insignificant to ST. Table 17 shows the results of hypotheses testing obtained based on PLS estimations.

Table 17. Main Hypotheses Testing (Results of PLS-SEM)

Hypotheses	Path	T Statistics	P-value	Result
H1-1	TCX → ST	2.919	0.004**	Supported
H1-2	TOV → ST	5.173	0.000***	Supported
H1-3	TUC → ST	0.590	0.555	Not Supported
H1-4	TNV → ST	2.670	0.008**	Supported
H1-5	TNS → ST	3.831	0.000***	Supported
H2	ST → IWA	3.039	0.002**	Supported
H3-1	TCX → IWA	0.042	0.966	Not Supported
H3-2	TOV → IWA	2.056	0.040*	Supported
H3-3	TUC → IWA	1.209	0.227	Not Supported
H3-4	TNV → IWA	0.004	0.997	Not Supported
H3-5	TNS → IWA	2.840	0.005**	Supported
H4	IWA → WAB	30.491	0.000***	Supported

TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.
** := p < .05; ** := p < .01; *** := p < .001.*

Indirect Effect

The result of the indirect impact using PLS-SEM (Table 18) also shows similar results to CB-SEM with two differences found, which are insignificant. These are the indirect effects of technology complexity (TCX) and technology insecurity (TNS) on the intention to implement workarounds (IWA).

Table 18. Results of the Indirect Effect (Results of PLS-SEM)

	IWA		WAB		Supported Indirect Impact on
	T Statistics	**p-value**	**T Statistics**	**p-value**	
TCX	1.954	0.051	0.472	0.637	**-**
TOV	2.772**	0.006	2.992**	0.003	**IWA, WAB**
TUC	0.555	0.579	1.313	0.190	-
TNV	2.045*	0.041	0.657	0.511	**IWA**
TNS	2.148*	0.032	3.773***	0.000	**IWA, WAB**
ST			2.334*	0.02	**WAB**

TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNV: Technology Invasion, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.
** := p < .05; ** := p < .01; *** := p < .001.*

Moderating Effect

Results of PLS-SEM shows that moderating effect of neutralization is not supported on the hypothesized relations. While the results of the direct impact of neutralization, autonomy and perceived behavioral control are supported in PLS-SEM, the interaction effect is insignificant. Therefore PLS-SEM did not support the moderating role of autonomy and perceived behavioral control. Only one moderation effect is supported, which is the impact of strain (ST) on the intention to implement workarounds (IWA). Table 19 shows the results of the moderating effect based on PLS.

Table 19. Results of Moderating Effect (Results of PLS-SEM)

Moderator	H	Path	Moderator Effect (P-value)	Interaction Effect (P-value)	Result
Neutralization	H5a	ST → IWA	6.497*** (0.000)	0.825 (0.410)	Rejected
	H5b	TOV → IWA	6.710*** (0.000)	0.123 (0.902)	Rejected
		TNS → IWA	6.710*** (0.000)	0.368 (0.713)	Rejected
Autonomy	H6a	ST → IWA	4.204*** (0.000)	0.763 (0.446)	Rejected
	H6b	TOV → IWA	4.238*** (0.000)	1.192 (0.234)	Rejected
		TNS → IWA	4.238*** (0.000)	0.795 (0.427)	Rejected
Perceived	H7a	ST → IWA	5.991*** (0.000)	1.992* (0.047)	Supported
Behavioral	H7b	TOV → IWA	6.266*** (0.000)	1.010 (0.313)	Rejected
Control		TNS → IWA	6.266*** (0.000)	0.519 (0.604)	Rejected

ST: Strain, IWA: Intention to Implement Workarounds; TOV: Technology Overload, TNS: Technology Insecurity.
** := p < .05; ** := p < .01; *** := p < .001.*

Control Effect

Similar to CB-SEM, PLS-SEM results of the control effect report insignificant control effects of industry, years of experience, education, and gender. As this is also confirmed by previous studies as discussed before in section (3.7.3.2).

3.7.3.3 Comparison of Results: CB-SEM and PLS-SEM

This section provides a comparison between the results generated by the two methods CB-SEM and PLS-SEM. The statistical objective of the two methods is substantially different. According to Hair et al. (2012), the CB-SEM aims at estimating the model parameters that minimize the difference

between the observed sample and the estimated one. In contrast, PLS-SEM aims at maximizing the variance explained in the dependent variables. In addition to that, CB-SEM is a parametric method such that statistical significance is a standard output of that technique, whereas PLS-SEM is a non-parametric method that hinders the immediate determination of inference statistics. (Jr et al., 2017). Therefore, it is typical to see some differences in the estimations and hypothesis testing results.

The overall results are similar, however, there are slight differences in the result of a few factors and moderators that report contradicting results. Based on the comparison of the main mode, the results of both methods are the same except for one path (TNS → ST).

Table 20. Comparison of Hypotheses Testing Results of CB-SEM and PLS-SEM

Hypotheses	**Path**	**CB-SEM**		**PLS-SEM**	
		p-value	**Supported**	**p-value**	**Supported**
H1-1	TCX → ST	0.004**	✓	0.004**	✓
H1-2	TOV → ST	0.000***	✓	0.000***	✓
H1-3	TUC → ST	0.451	x	0.555	x
H1-4	TNV → ST	0.019*	✓	0.008**	✓
H1-5	*TNS → ST*	*0.188*	*x*	*0.000****	✓
H2	ST → IWA	0.000***	✓	0.002**	✓
H3-1	TCX → IWA	0.530	x	0.966	x
H3-2	TOV → IWA	0.036*	✓	0.040**	✓
H3-3	TUC →	0.138	x	0.227	x

Hypotheses	Path		CB-SEM		PLS-SEM	
			p-value	Supported	p-value	Supported
H3-4	IWA TNV	→	0.327	x	0.997	x
H3-5	IWA TNS	→	0.011*	✓	0.005**	✓
H4	IWA IWA WAB	→	0.000***	✓	0.000***	✓

TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNV: Technology Invasion, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.

** := p < .05; ** := p < .01; *** := p < .001.*

The comparison of the indirect effect of technostress factors has resulted in similarity to those of CB-SEM except two paths, in which one is not supported by the CB-SEM and rejected by PLS-SEM, that is technology complexity (TCX); and the technology insecurity (TNS) which is rejected by CB-SEM and supported by PLS-SEM.

Table 21. Comparison of the Indirect Effect

	CB-SEM		PLS-SEM	
	IWA	**WAB**	**IWA**	**WAB**
TCX	✓	x	x	x
TOV	✓	✓	✓	✓
TNV	x	x	x	x
TNS	x	✓	✓	✓
ST		✓		✓

✓ Supported; x rejected, *TCX: Technology Complexity, TOV: Technology Overload, TUC: Technology Uncertainty, TNV: Technology Invasion, TNS: Technology Insecurity, ST: Strain, IWA: Intention to Implement Workarounds, WAB: Workarounds Behavior.*

Comparing the moderating impact of neutralization, autonomy and perceived behavioral control, PLS-SEM reject the moderating role of neutralization, autonomy, perceived behavioral, except the moderating effect of perceived behavioral control on the impact of ST on IWA. On the other hand, CB-SEM shows a moderating effect of neutralization on the impact of technology overload on the workaround intention. Result of CB-SEM also shows a moderation effect of autonomy on: 1) the impact of strain on the workaround intention, and on 2) the impact of technology overload on the workaround intention. Lastly, the perceived behavioral control moderates only the impact of strain on the workaround intention. Table 22, shows a comparison of the moderating effect between CB-SEM and PLS-SEM.

Table 22. Comparison of the Moderating Effect

Moderator	H	Path	CB-SEM	PLS-SEM
Neutralization	H5a	ST → IWA	x	x
	H5b	TOV → IWA	✓	x
		TNS → IWA	x	x
Autonomy	H6a	ST → IWA	✓	x
	H6b	TOV → IWA	✓	x
		TNS → IWA	✓	x
Perceived	H7a	ST → IWA	✓	✓
Behavioral	H7b	TOV → IWA	x	x
Control		TNS → IWA	x	x

ST: Strain, IWA: Intention to Implement Workarounds; TOV: Technology Overload, TNS: Technology Insecurity.
** := p < .05; ** := p < .01; *** := p < .001.*

Although the results of both methods have led to some difference, making a comparison can be somehow misleading. Researchers should not expect similar results from CBSEM and PLS-SEM because, even though they ultimately address the same phenomena, the methods are typically applied in suboptimal situations, where different approaches fall short for different reasons. Within the realm of CB-SEM, for example, maximum likelihood estimation and generalized least squares estimation are asymptotically equivalent when assumptions hold, but equivalence fails when assumptions are violated (Rigdon et al., 2017).

3.8 Discussion

The main objective of this study is to extend the understanding of factors impacting the workaround behavior of software engineers, through examining the impact of technostress dimensions, neutralization, autonomy and behavioral controls. Building on the theoretical lens of planned behavior, the proposed theoretical model draws the main concepts from the theory of workarounds by Alter (2014), technostress by Ragu-Nathan et al. (2008), neutralization by Coleman (1987), Minor (1981), and Sykes & Matza (1957), and autonomy by Coeckelbergh (2006). The results of the hypotheses testing, the results of CB-SEM and PLS-SEM reports a match on most of the findings of the study with slight differences in some results.

3.8.1 Impact of Technostress and Strain on Software Engineers' Workaround Intention and Behavior

Referring back to the research questions, the first RQ1 posed seeks to find the impact of technostress on engineer's workarounds. In a more detailed analysis of both CB-SEM and PLS-SEM, the findings report that factors of technostress (H1): namely technology complexity (H1-1), overload (H1-2), and invasion (H1-4) are confirmed to have a significant impact on strain. According to the general strain theory, strain is a result of stressors that are mainly driven by work overload, invasion of personal space, job insecurity and role ambiguity (Agnew & Brezina, 2019; Ayyagari et al., 2011). This indicates that the factors of technostress stated in hypotheses, namely technology complexity, technology overload and technology invasion, are well explained by the general strain theory. The only difference reported between CB-SEM and PLS-SEM is the impact of technology insecurity (H1-5) on strain; in that CB-SEM report an insignificant impact whereas the PLS-SEM report a significant impact to strain. A further confirmation is needed to ensure more robust conclusion on its significance. Nevertheless, a possible justification if we reject H1-5 would be the high demand for software engineers in the market, which makes the sense of job insecurity is less likely relevant.

On the other hand, the influence of technology uncertainty (H1-3) on strain is rejected based on hypotheses testing of both SEM methods. The

technology uncertainty does not seem to lead to a strain compared to other technostress factors. Perhaps, technology uncertainty is less related to strain or might indirectly contribute to the impact. An alternative justification for this is that, the organizations, to which respondents belong to, experience less frequent changes in software technologies. This finding is confirmed by Nasirpouri & Biros (2020) in that their study rejected the impact of technology uncertainty on strain in the context of end users of a software. In that sense, we conclude no impact of technology uncertainty on strain, although the majority of respondents belong to the IT industry, which is considered highly dynamic. Nevertheless, being rejected by previous studies as well, a further investigation is needed to confirm how well it fits as a factor of technostress.

The results of both SEM estimation methods also show that among the factors of technostress, technology overload (H3-2) and technology insecurity (H3-5) have a direct and significant impact on workarounds intentions. Previous studies also found that technology overload can lead to job dissatisfaction and impact productivity (Fuglseth & Sørebø, 2014). As the technology accelerates business processes, software engineers are required to deliver more functionality within a short time, which, in turn, stimulate their need to find an easy and fast solution to cope up with the overload. In other words, the more technology overload is, the more workarounds are triggered, and consequently accumulated technical debt. The analysis also shows that

technology insecurity, which is the threat of losing one's job because of technology, impacts workaround intention. This can be explained from a perspective that software engineers increase their productivity and throughput in exchange for quality. In most cases, managers care more about fast delivery of services in response to business needs and market pressure, and deal with consequences later (Yli-Huumo et al., 2014). In order to meet these needs, software engineers have to find a way to increase their throughput, and hence, tend to implement workarounds.

Studies on software policy compliance report that technostress in general can lead to violation of policies on cause insider threat (Aggarwal & Dhurkari, 2023; Nasirpouri & Biros, 2020; Shadbad & Biros, 2021). Our study report that technostress can cause workaround behavior. In specific, we found that technology overload and technology insecurity are significant to engineers' intention to implement workarounds. However, our study reports insignificant impact of technology complexity, uncertainty and invasion. This result can be explained by the possibility that software engineers view technology as an integral part of their life, and therefore, do not perceive it as complex, uncertain and invade their personal space. Contrary to previous explanation, they might have viewed technology as complex, uncertain and invade their personal space, however, that does not trigger their intention to implement workarounds unless it resulted in strain.

Regarding the second research question RQ2 (what is the impact of

strain on intention to implement workarounds), the analysis of both SEM methods reports a significant impact of strain on the intention to implement workarounds (H2). This indicates that the more the strain experienced by software engineers, the more it triggers their workaround intention. Prior studies on software policy compliance also confirmed the strain has a significant influence on the intention to violate policies (Agnew & Brezina, 2019; Nasirpouri & Biros, 2020). Considering that strain typically has negative consequences, we can view workaround in this context as a type of noncompliance behavior. These results can also be explained from the perspective of stress and coping theory (Lazarus, 1966), in that the strain resulting from technostress can result in a form of coping behavior. In other words, whenever software engineers experience strain, the workaround behavior can be one of the coping responses to that strain. While workarounds are not always considered violation of policies since policies do not always state what to do and how it should be done (Alter, 2015), it would to the engineers' judgment whether the workaround they implemented is considered noncompliance.

Regarding the indirect impact, both SEM estimation methods confirm that technology overload have an indirect impact on the workaround intention through strain, and on the workaround behavior through the workaround intention. In other words, strain plays a significant mediating impact between the technostress (overload), and the workaround intention. The analysis both

CB-SEM and PLS-SEM show an indirect impact of technology insecurity and strains on the workaround behavior. This impact is mediated by engineers' intention to implement workarounds. This indicates that the more the sense of job insecurity due to technology, the more workaround behaviors performed. This can be explained from the perspective that those who implement workarounds tend to be goal-driven and seek to increase their productivity knowing that managers are outcome-oriented (Davison et al., 2021). On the other hand, both SEM analysis report insignificant mediating role of strain and workaround intention for technology invasion. This means, as technostress (invasion) causes strain, it does not necessarily lead to workarounds behavior. Lastly, the analysis of both CB-SEM and PLS-SEM report contradicting results on the mediating effect of strain between technostress (complexity and insecurity), and the workaround intention.

The findings for the RQ1 and RQ2 conclude that workarounds in software engineering can be directly and indirectly through technostress factors, with stronger impact of technostress (overload) among all other technostress factors. Furthermore, strain plays a strong mediating effect towards the workaround behavior. The analysis of both SEM methods shows consensus in results of most factors, while some need further evaluation.

3.8.2 The Moderating Impact of Neutralization on the Relationships between Technostress, Strain, and Engineers' Intention to Implement Workarounds

To answer the third research question RQ3, which seeks to evaluate the moderating impact of neutralization (H5a, H5b), the results of both CB-SEM and PLS-SEM analysis reject the moderating role of neutralization on the impact of technostress (insecurity) and strain on the workaround intention. This might indicate that respondents do not tend to justify workarounds performed, or neutralization might have been contextualized and understood differently. A Study by D'Arcy & Teh (2019) found that neutralization can be instable phenomena, even within the same individual over different times. However, in order to observe that, it requires a longitudinal study that capture its variability over the time. Another possible justification for rejecting this hypothesis would be the likelihood of looking at workarounds as not necessarily as deviant behavior if policies are not explicit on what, or on how to implement a certain task. Instead, workaround is rather a way for more productivity; as some workarounds can be beneficial according to Alter (2015).

However, the analysis of CB-SEM reports a significant moderating role of neutralization on the relationship between technostress overload and the workaround intention. According to the theory of neutralization, an individual uses one of the neutralization techniques to rationalize his/her deviant

behavior (Sykes & Matza, 1957). This indicates that whenever software engineers experience technostress overload, neutralization techniques can be a good justification that incentivize their intention to implement workarounds. As the study measurement reflects four techniques of neutralization namely: denial of injury, condemnation of condemners, defense of necessity, and defense of ubiquity. In other words, if a software engineer perceives the workaround he tends to implement as nonharmful, the prescribed policy as unreasonable, existing choices are limited, or surrounding people are also doing the same; then he/she is likely to have a good justification that is ready to use whenever technostress overload is very high. Surprisingly, PLS-SEM report an insignificant moderating impact of neutralization on the relationship between technostress overload and the workaround intention, although the direct impact of neutralization on the workaround intention is significant. In this regard, further studies can investigate whether to view neutralization more of a moderator or an independent factor.

3.8.3 Moderating Impact of Autonomy on the Relationships between Technostress, Strain, and Engineers' Intention to Implement Workarounds

Regarding the moderating role of autonomy, the analysis of both CB-SEM and PLS-SEM shows different results. The results of CB-SEM support the moderating impact of autonomy on the relationship between strain and the

workaround intention (H6a), technostress overload and the workaround intention, and technostress insecurity and the workaround intention (H6b). On the other hand, the results of PLS-SEM reject the moderating impact of autonomy for the same hypotheses. Although the analysis of PLS-SEM shows that the direct impact of autonomy on the workaround intention is significant, the results on the interaction impact is insignificant, and therefore no moderating impact.

Based on the results of CB-SEM, the level of control given to engineers over technical decisions plays a significant moderating role towards their intention to implement workarounds. From the self-determination perspective, autonomy is considered a key driver that intrinsically motivates a certain behavior (Jeon et al., 2020). According to Coeckelbergh (2006), less control and more trust are expected to improve professional autonomy and enhance quality of services delivered by engineers. If we accept the results of CB-SEM, we can argue that engineers are more likely to implement workarounds whenever their level of strain, technostress overload, and insecurity are very high, provided that they possess a high autonomy and control over technical decisions. While the level of autonomy plays an important role in delivering out of the box innovative solutions, it can also lead to delivering a low-quality solution or one deviating from policies. In this regard, there are tradeoffs when deciding the level of constraints or the level of responsibility given to engineers, as it can be risky to some industries. Hence, regulations and

constraints might be needed in order to prevent more incidents resulting from the freedom given over technical decisions.

In summary, while the results of CB-SEM show a strong moderating impact of autonomy, the results of PLS-SEM indicate that the impact of strain, overload, insecurity exists on the workaround intention, regardless of the level of autonomy. Furthermore, the results of PLS-SEM confirm a direct impact of autonomy on the workaround intention, indicating that autonomy has a more of a direct impact than a moderation. The findings of this study indicate that the role of autonomy needs a further confirmation in order to build a strong conclusion.

3.8.4 Moderation Impact of Perceived Behavioral Control on the Relationships between Technostress, Strain, and Engineers' Intention to Implement Workarounds

The perceived behavioral control, which represents one's ability to use different alternatives and solutions, is supported to moderate the relationship between the strain and the intention to implement workarounds (H7a) based on the results of both CB-SEM and PLS-SEM. From the perspective of the theory of planned behavior, the perceived behavioral controls play a significant moderating role towards the intended behavior (Ajzen, 1991). This indicates that whenever an individual perceives that he/she has the ability to implement workarounds, the likelihood of him/her doing so increases. The

ability of engineers to accomplish a certain task using various ways is more likely to incentivize them towards developing workarounds whenever they perceive a high strain.

On the other hand, the results of both CB-SEM and PLS-SEM rejected the moderating impact of perceived behavioral control on the impact of technology overload and insecurity on the workaround intention (H7b). This indicates that one's ability to implement workarounds do not incentivize their workaround intention whenever he/she experience high technostress overload and insecurity. In other words, the perceived behavioral control can be a strong moderator, only if an engineer exerts a high degree of strain. In this regard, and from the previous findings, it is critical to address the technostress factors that cause strain, as it plays a strong mediating impact.

In summary, the analysis of both structural models provides the same result on the moderating impact of perceived behavioral control on the impact of technostress and strain on the workaround intention. The only differences that has been remarked are on the moderating role of neutralization and autonomy. Both SEM results confirm a direct impact of neutralization and autonomy on the intention to implement workarounds, as shown in Figure 24. In contrast, the CB-SEM shows a significant moderating impact of autonomy, and rejects its direct impact, whereas, PLS-SEM shows rejects the moderating role of autonomy and confirms its direct impact.

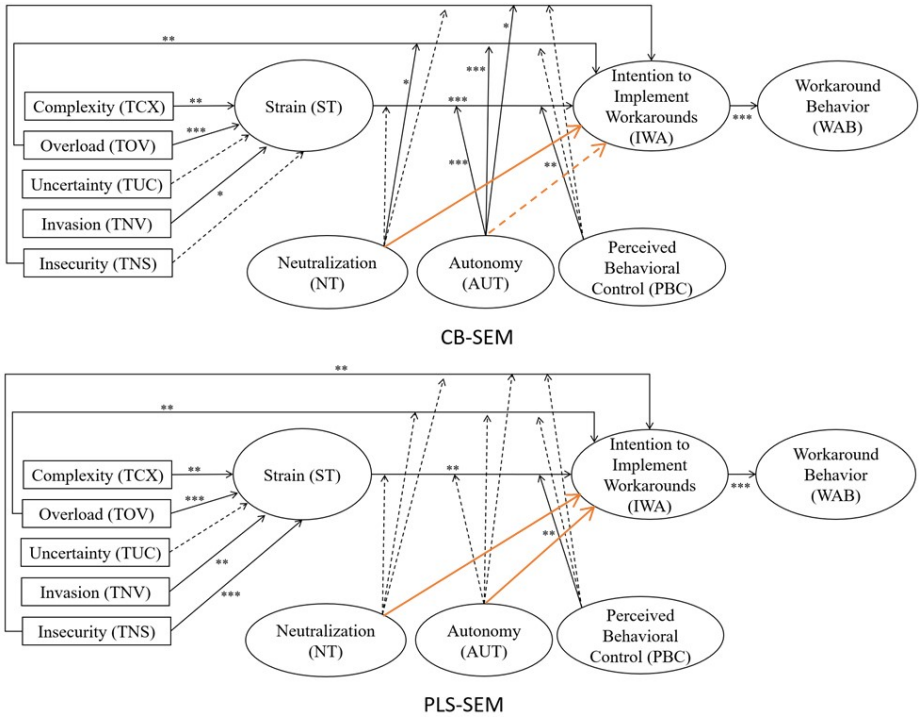

Figure 24. Comparison of Model Results (CB-SEM and PLS-SEM)

Regarding the appropriability of which estimation method the study should consider and build the empirical evidence, scholars introduce criteria to which method is more appropriate for model estimation and hypotheses testing (Astrachan et al., 2014; Jr et al., 2017). Nevertheless, having a combination of two methods and a comparison of results generated by the two methods can enhance the reliability to the findings of this study. Although there are slight differences between the two estimation methods, the overall result shows that factors of technostress can directly and indirectly trigger workarounds through the mediation of strains. In addition to that, the study finds that neutralization

is not significant in the moderation role between strain and workarounds intention. Furthermore, professional autonomy plays a significant moderating role in the impact towards workarounds based on the results of CB-SEM method, whereas PLS-SEM considers autonomy more of a direct impact on the workaround intention. Furthermore, the perceived behavioral control is confirmed in both methods to moderate the impact of strain on the workaround intention. Nevertheless, the overall results of on both CB-SEM and PLS-SEM estimation methods are similar with slight differences on the moderating impact.

3.9 Implications

Based on the empirical evidence obtained through detailed analysis using both CB-SEM and PLS-SEM, the results deliver the following findings: (1) the complexity, overload and invasion of technology are significant to strain; while insecurity directly triggers workarounds; (2) uncertainty of technology is very insignificant to strain and workaround intention; as pervious study by Nasirpouri & Biros (2020) conclude insignificance of technostress (uncertainty) on software policy noncompliance; (3) technology overload and insecurity are both indirectly related to workarounds; (4) the moderating role of neutralization is rejected by both estimation methods.

Surprisingly, the detailed comparison of the results obtained by CB-SEM and PLS-SEM offers the following results: (1) the moderating role of autonomy and perceived behavioral control are significant in CB-SEM but not

in PLS-SEM; (2) slight difference on the significance level is found for insecurity and overload. Accordingly, this section presents the theoretical and practical implications.

3.9.1 Theoretical Implications

From a theoretical perspective, the study extends the software compliance domain by providing an understanding on technostress and its influence on the workaround behavior. In other words, the study adds to the knowledge base through evaluating technostress as a predictor of workarounds. With the evolving research on workarounds, its causes and consequences, the study has the following theoretical implications:

First, the study positions technostress and strain as causes of workarounds. Researchers on software compliance should consider technostress an antecedent, when studying the workarounds phenomenon. While the study looks at the relationship between technostress and workarounds from the lens of the theory of planned behavior (section 3.5.5), further studies can consider viewing such a relationship from the perspective of stress and coping theory in order to investigate and compare whether the workaround can be classified as a coping behavior to stress. This, in turn, helps compare and further assess how well both theories explain the phenomena.

Second, the new theoretical implication also comes from delivering evidence on the moderating role of neutralization in strengthening the impact

on workarounds (section 3.5.6). While no prior study tested neutralization in behavioral workarounds, this adds to the current literature an understanding and empirical evidence on how well neutralization theory fits in the context of workarounds as an explanation of an individual's subjective norms. Although the moderating role of neutralization is rejected in this study, it calls for further investigation on the role of neutralization with respect to its influence on the theory of workarounds.

Third, as the technology uncertainty has been rejected based on both estimation methods, in addition to previous studies on software compliance has also rejected it, research should reconsider whether positioning uncertainty can be a valid dimension of technostress in the context of software policy compliance.

Fourth, with the challenging concerns in agile approaches that call for more autonomous individuals and teams, the study adds to the literature evidence on the extent to which professional autonomy plays a moderator role in the context of workarounds. By understanding this, our study positions autonomy as one of the significant contributors that predicts workaround behavior (section 3.5.7). Although, CB-SEM and PLS-SEM report different results, having such an understanding on the role of autonomy would help improve addressing some of the challenges raised by agile approaches. This entails that future research should not ignore the role of professional autonomy when studying behavioral phenomena of software

compliance. Further research should solidify and confirm this impact of autonomy.

Finally, the perceived behavioral control is derived from the theory of planned behavior (Ajzen, 1991) and tested in the domain of workarounds. Having tested the perceived behavioral control in this study (section 3.7.3.3), this factor has explained well how an individual's perception of his/her ability or difficulty in using different alternatives can strengthen or weaken their behavioral intention towards workarounds. As the theory of planned behavior is well established, accepting the results of CB-SEM can be more appropriate for assessing the moderating role of perceived behavioral control. Therefore, the individual skills can be one of the main determinants that studies should consider in the software compliance behavior and workarounds in particular. This finding also helps viewing workarounds from the lens of the theory of planned behavior.

3.9.2 Managerial Implications

The study offers several managerial implications for practitioners and policy makers helping them mitigate the impact of technostress not only on workarounds but also on the overall software quality and business performance. From managerial perspective, among the policies suggested to identify the technostress experienced by the workers is to investigate practices of insiders including: performing work-related tasks at home, the amount of work overload that technology adds, the rate of changing technologies within

an organization, automation of tasks and replacement of human. This might be very challenging as the perceived technostress and strains are self-observed by an individual which is difficult to grasp. However, organizations can control some the moderators which strengthen such an impact, and incorporate policies so that the consequences become within their tolerable risk appetite. The following are a list of managerial implications the study presents:

First, the overall findings of the study indicate that the overuse of technologies and the factors of technostress can result in not only and strain, but it can also be threatful to an organization from the perspective of security vulnerabilities and technical risks. As the results show that some of the factors of technostress can trigger workarounds (section 3.7.3.1), practitioners should carefully look into such antecedents that lead to workarounds and work on mitigating the impact of technostress or alternatively work on controlling workarounds. In this regard, and according to the findings of this study, the following are some of the policy implications that managers should carefully consider for each of the factors of technostress:

- As the technology complexity and the unpleasant feelings about the multifaceted new software technologies, that require tremendous efforts to understand, is found to cause workarounds directly and indirectly. Therefore, organizations should invest in skill-building and knowledge exchange programs, to help mitigate the pressure raised from technological complexity.

- In order to mitigate the impact of technology overload and control its impact on workarounds, managers and practitioners should properly estimate the time and efforts needed to accomplish tasks, so that they can control the workflow acceleration caused by technology. In addition to that, they need to identify the threshold at which the technology overload started exceeding the pace of engineers to catchup, because after that point, there is a likelihood of workarounds, and quality tradeoffs being made. Addressing this issue of technology overload is very tricky, even with proper task estimation, as there is also a possibility of cyber loafing, i.e. use of organization's time to do personal work, as a result of being connected the whole time.
- To control the impact of technology invasion, organization should pay attention to controlling the tasks that might spill into engineers' private life and endanger their work-life balance. Otherwise, the frequent sense of invasion of one's personal space is likely to lead to strain and, hence, causes more workarounds. Alternatively, an organization might have to be clear about the nature of software engineers' work, especially in case of software incidents which requires engineers to be able to accept tasks anytime and, probably, anywhere.
- The technology insecurity, which is the fear of losing jobs because of technology, causes workarounds according to the results. Since the

number of tasks accomplished is, typically, viewed as a metric for productivity and, in turn, sustaining one's job, organizations need to consider the tradeoffs in throughput and quality of accomplished tasks.

Second, the ongoing advances in technology and connectivity make insiders stay connected the whole time and, as a result, deepen the negative impact of technology on them. This, alongside prior, indicates that the impact of technostress is inevitable and might not be easily controlled at workplaces. Therefore, practitioners and decision makers should develop supporting mechanisms and tools in order to help mitigate and control workarounds. One of the promising policies is the adoption of x-by-design principles; which includes: compliance-by-design, security-by-design, and privacy-by-design. These approaches can provide a level of enforcement using a predefined software blueprint that guides engineers' actions and technical decisions.

Third, although PLS-SEM shows slightly different results, the findings of CB-SEM indicate that engineers' autonomy play a significant role towards contributing to implementation of workarounds (section 3.5.7), whereas the result of PLS-SEM shows a direct impact of autonomy. In this regard, it is vital to carefully consider the level of autonomy given to engineers in that balancing between responsibility and constraints is crucial. The degree of autonomy can be guided, in most cases, by business requirements. In other words, mission-critical businesses are likely to opt for more regulations and constraining rather than autonomy and responsibility compared to less critical

ones. While agile approaches require individuals and teams to be more autonomous, as their role goes beyond developing a software, and they need to involve in other organizational units to understand the problem as a whole. Such autonomy could result in giving more freedom over technical decisions and, hence, more space for implementing workarounds. Furthermore, the notions of crowdsourcing and distributed teams have emphasized towards more on professional autonomy of software engineers, whether on work schedule, technical related decisions, or methods of accomplishing their tasks. As such autonomy can give a space for working arounds the technical decisions, we need to understand how it contributes to development of workarounds, so we can control and mitigate the workarounds. In this regard, organizations should encourage engineers to report any technical debt and workarounds in order to strategize addressing them in future releases before they become more expensive to deal with. Besides that, having clear policies and tools in place that provide a better visibility to all stakeholders on autonomous teams is of paramount importance.

Finally, workarounds performed by software engineers typically result in so-called technical debts and studies report that 25% of efforts are wasted on refactoring. Managers should analyze insiders' practices and identify the antecedents and causes of their workaround behavior. Therefore, based on the empirical evidence from this study, managers need to carefully consider the technostress, neutralization, autonomy, and behavioral controls, and their

consequences on workarounds and technical debt in the long run. Findings of this study can guide practitioners and compliance managers to pay attention to causes of workarounds and provides them with a foundational understanding of the phenomena in order to help mitigating their business impact.

Table 23. Summary of Study Implications

Topic	Implications
Theoretical Implications	1. Provide an understanding of technostress and workarounds from other overarching lenses, including: stress and coping theory; and exit, voice, loyalty, and neglect. 2. Longitudinal investigation on the role of neutralization in the theory of workarounds, is needed, as neutralization is considered unstable phenomena and can vary even within the same individual over the time. 3. As autonomy is crucial to software engineering, while at the same time, significant to workarounds, research efforts need to develop solutions that enhance compliance of autonomous software engineering practices. 4. Behavioral controls, represented by skills and abilities, can guide engineers' response to technostress, and can highly be related to the choice of workarounds to implement. Thus, researchers on workarounds should highly consider the perceived behavioral controls.
Managerial Implications	1. Managers should investigate clues technostress exerted by software engineers. These include: spill of work-related tasks to their private life; (2) the rate of technological changes within an organization; (3) deadline pressure and metrics of productivity (4) drifts noticed between documentation and production. 2. As technology evolves at a high rate, the impact of technostress continues to exist, and, in turn, workarounds. Thus, standardization and automation of software engineering related practices is crucial. 3. Carefully consider the level of autonomy given to engineers, through policies and supporting tools that provide better control and visibility to concerned stakeholders.

Topic	Implications
	4. Since the turnover of software engineers is very high, managers should seriously address the causes that lead to workarounds and technical debts.

3.10 Conclusion and Contribution

3.10.1 Summary

Technostress is viewed as the dark side of technology as it leads to negative consequences in software policy compliance. Besides that, the growing concerns of shadow systems and technical debts led to development of the theory of workarounds in the field of software compliance. This study investigates the impact of technostress on software engineering workarounds, and assesses the moderating role of neutralization, autonomy and perceived behavioral controls on that impact. The study uses structural equation modeling applied to a survey data from a sample of 306 working in software engineering various organizations in South Korea. The analysis of CB-SEM and PLS-SEM methods were applied for a better reliability of results. Findings indicate that technostress and strain can predict workaround behavior. In particular, technology complexity, overload and invasion. This impact is moderated by the degree of professional autonomy and perceived behavioral control, although the PLS analysis does not confirm that. The findings also report an indirect impact of technology complexity and overload on the behavioral intention of workarounds. Another indirect impact found of technology overload, technology insecurity and strain on the workaround

behavior. While some findings confirm strong evidence based on results of both CB-SEM and PLS-SEM, few others need further confirmation to solidify the evidence in order to build implications accordingly. Upon these findings, practitioners can intervene in order to mitigate the impact of technostress and control the consequences resulting from workarounds. The study further introduces theoretical and practical implications.

3.10.2 Contributions

The study contributes to theoretical knowledge through extending the understanding of technostress in the context of software engineering workarounds. The study positions technostress as a new antecedent to workarounds behavior. Previous studies argue that time pressure, misfit of work practices, and complexity of technology are predictors of workarounds. No prior study considered technostress as one of factors that lead to workarounds, which is the key contribution this study introduces. The study takes the lens of planned behavior and extends the theory of workarounds with a technostress as a predictor of workarounds intention and behavior.

While the study provides a strong empirical evidence based on the structural analysis of CB-SEM and PLS-SEM, the detailed comparison of the results obtained both methods offers the following surprising results: (1) the moderating role of autonomy and perceived behavioral control are significant in CB-SEM but not in PLS-SEM; (2) slight difference on the significance level is found for insecurity and overload.

The study also contributes to literature through incorporating neutralization, autonomy, and perceived behavioral control in the theoretical model, and evaluates the extent to which they strengthen or weaken the impact of technostress on workarounds. The strong moderating role of autonomy and perceived behavioral control contributes to more understanding of the workaround phenomena. The study also contributes to the literature by integrating theories of workarounds, planned behavior with technostress.

The practical contribution of the study is that it helps practitioners and organizations consider such antecedents of workarounds and the consequences resulting from technostress. From a policy perspective, the findings of the study provide insights on setting policies that could help mitigate the technostress at workplace. Additionally, the study provides them with evidence to guide them to decide a proper level of professional autonomy to be given to engineers over the technical decisions. Better understanding of the role of autonomy can be great to develop a balance between responsibility and regulation based on empirical evidence. The study calls for paying attention to investigating and analyzing technostress and engineers' workaround behavior, since the consequences of workarounds in the software engineering field can be severe in the long run.

3.10.3 Limitations

The study acknowledges that the findings of this research come with some limitations. Firstly, the impact of control variables has not been

statistically observed by the study using both analysis methods (CB-SEM and PLS-SEM). The same results are also reported by Marchiori et al. (2019). Secondly, while most of the results that are confirmed by both of the aforementioned analysis methods gives strong evidence, some mismatches of the results need further investigation. Thirdly, the findings of the study can be impacted by the culture in Korea which might accordingly confirm or reject some of the hypotheses based on that. Finally, generalizing the findings of the study can be one of the limitations in this research since the theoretical model is tested in Korean context. The peculiarities of cultures could lead to some difference in the results having the study being conducted in a different context.

Chapter 4. Discussion and Conclusion

4.1 Summary

The rapid progress of information and communications technologies, along with changing corporate policies and business requirements, have shortened the evolution cycle of E-Type software systems, making the status of a software is likely to be in a releasable state most of its time. This, in turn, poses growing concerns on maintaining its compliance to policies, and is worsened by the diversity of compliance sources and requirements, that have to be met. Furthermore, as the frequency and cost of insider threats increased over the last two years, it is of high importance to deliver a systematic understanding of the state-of-the-art literature on software policy compliance in order to bring into focus the highly relevant topics and position their impact within the larger ecosystem.

This research is also motivated by the growing concerns of technostress and its unintended consequences on software compliance, particularly from software engineers' perspective, in addition to the wasted efforts resulting from technical debt and workarounds in software engineering. The high turnover rate among software engineers has reached 42% (according to DigitalOcean (2022). This makes workarounds performed in software engineering a serious business problem, as they are likely to impact software stability, security vulnerability and, in turn, business continuity. Furthermore, as technical debt account for 25% of extra efforts wasted due to workarounds,

it is worth investigating what contributes to development of workarounds. Accordingly, this research presents two main studies summarized as follows:

The first study uses an evidence-based systematic literature review on the existing body of research to investigate software policy compliance, in order to provide an understanding on the existing research foci, evolving theories and concepts, and relevance of potential gaps and directions. This, in turn, provides relevance of potential studies and show how important they are from a pragmatic perspective. Based on the review protocol and inclusion criteria (section 2.3), 84 relevant studies identified and analyzed. Results reveal several key findings: (1) End user security is on top discussion followed by legal and privacy issues; (2) Security awareness and automation of compliance are top cited policies; (3) There is an emphasis on the gaps between compliance and domain experts at one hand, and software engineers on the other hand; (4) Wile the theory of planned behavior is dominating, the theory of workarounds has emerged in the domain of compliance; (5) There are several concepts and topics, which are evolving in the domain. These are: privacy and compliance by design, policy-as-code, security related stress, and home-office user environments. The findings of the review can guide practitioners and researchers, and provide them with a foundation for software policy compliance implications and potential research directions.

The second study adapts a deductive approach and uses an empirical quantitative method to examine the antecedents that cause workarounds in

software engineering. In particular, it assesses the extent, to which the factors of technostress (technology complexity, uncertainty, overload, invasion, and insecurity) can trigger workarounds. It also assesses the role of professional autonomy, use of neutralization strategies, and perceived behavioral controls on that impact. The study aims to provide a new understanding of technostress in the context of software engineering, and emphasize how significant these concepts are towards contributing to the development workarounds in software engineering. The study positions the five factors of technostress as new antecedents to workarounds behavior. It contextualizes the definition of workarounds on software engineering and present a distinction from the workarounds performed by end users, since software engineers recognize the technical intricacies more than any other stakeholder in software ecosystem.

While literature reports that the causes of workarounds in software development and operations comes primarily from: pressure of meeting deadlines, misfit of work practices, complexity of overwhelming technologies, inadequate resources, and misunderstanding of intentions between management and software engineers. Our study poses an argument that technostress factors can be among causes of workarounds in software engineering. Furthermore, the study incorporates neutralization, professional autonomy, and behavioral controls as factors that we argue to moderate such an impact. In order to bring a thorough understanding on the impact of technostress and strains on the workarounds, it is critical to further look into

what could strengthen or weaken that impact. In this regard, (1) neutralization techniques can be viewed as means to justify workarounds, and therefore, strengthen/weaken the impact; (2) the level of professional autonomy given to engineers over technical decisions can give engineers more space for deciding various alternative solutions, and therefore, strengthen/weaken that impact; (3) lastly, engineers' perceived abilities and skills to implement different alternatives could also influence their decision to implement workarounds whenever they experience technostress. Using a survey data collected from 306 software engineers in South Korea, the study applies both covariance-based structural equation modeling (CB-SEM) and partial-least square (PLS-SEM) to evaluate the proposed research model and test the hypotheses.

The results of the study report that of *Technology Overload* predicts *Workarounds* indirectly through *Strain*. In other words, the likelihood of engineers to implement workarounds is high whenever the technology overload they experience is also high. The results also report that *Technology Complexity*, *Technology Overload*, and *Technology Invasion* have a direct impact on the *Strain*. Furthermore, *Technology Overload* and *Insecurity* have a direct impact on the intention to implement workarounds. As software projects are considered among highest risk projects, due to evolving technologies and changing business and functional requirements, task estimation in software engineering is inherently difficult, since requirements are subject to change at any time, making it difficult to account for the

unknowns. This gap in estimation is likely to create a fight between management, who push towards shortening the delivery time, and engineers, who perceive the proper estimation for task completion with a consideration of meeting a decent quality for such tasks. This, in turn, puts a pressure on engineers to catchup with the overload while, at the same time, not lose their jobs. And in order to reach a reconciliation, engineers are likely to respond to this overload and insecurity through considering workarounds in order to meet management requirements and cope up with the overload.

The findings of the study also reveal a significant moderating impact of autonomy and perceived behavioral control on the relationship between strain and workarounds intention, according to the analysis of CB-SEM. In other words, the high the degree of professional autonomy is, the stronger is the impact of strain on the workaround behavior. In practice, agile development methodology requires more autonomy given to engineers. This, in turn, gives them more control over technical decisions and the use of various different alternatives to accomplish their tasks. Autonomy is further emphasized by distributed development and crowdsourcing, in which the autonomy evaluated from choice of methods, flexibility in time, and freedom on technical-related decisions. While autonomy is critical to the impact on the behavioral intention of workarounds, possessing the skills and abilities to accomplish tasks using various alternatives, i.e., perceived behavioral controls, also triggers engineers thinking of quality tradeoffs of such alternatives, giving that the strain they

exert is very high. This can also indicate that the highly skilled engineers, who are able to solve the same problem using many different ways, can go with a less costly solution whenever they experience strain.

On the other hand, neutralization shows moderating role only on the impact of technology overload on the workaround intentions, as per the CB-SEM. This indicates that neutralization techniques can serve as a justification mean for implementing workarounds whenever the stress resulting from technology overload is very high, and thus, likely to strengthen the impact towards the workaround behavior. As the detailed analysis of both SEM methods shows that neutralization is directly related to behavioral intentions of workarounds, this might indicate that even with normal circumstances where there is no form of technostress or strain, engineers can implement workarounds by directly relating that with any mean of neutralization (e.g. denial of injury, denial of responsibility, condemnation of condemners, defense of necessity, defense of ubiquity, appeal to higher loyalty). Prior studies have concluded that neutralization is an unstable phenomenon and can change from one individual to another, even with the same individual over time. However, as the data of our study is cross-sectional and collected at a certain point of time, studying the change in the phenomenon within an individual requires longitudinal data. Nevertheless, investigating changes in neutralization over time is out of scope of this study.

The study extends the theory of workarounds and provide a new

understanding of technostress in the context of software engineering. The study also incorporates the use of neutralization strategies, degree of professional autonomy, and the perceived behavioral controls; as moderators on the study of the workaround behavior. The findings of this study help practitioners and researchers revealing more on what further causes workarounds in software engineering. This in turn, assist in developing response policy in order to better control workarounds; and deliver insights for future research. Accordingly, theoretical and practical implications are presented in the following section.

4.2 Implications

4.2.1 Theoretical Implications

From a theoretical perspective, the research provides several implications which are derived based on the findings of the systematic literature review as well as the findings of the empirical study. This section presents the overall theoretical implications that would be valuable for researchers in understanding and extending potential research work.

Based on the findings of the systematic literature review study and the results of analyzing existing research foci, topic evolving, and potential research directions, the following are the theoretical implications derived:

First, while studies on end user security compliance requirements gained more attention, the regulatory concerns around end users of E-type

software systems, the gaps between compliance and domain experts and software engineers, compliance of business process, accessibility and usability, in the context of software engineering are insufficiently explored and remain potential areas for further investigation.

Second, as the theory of workarounds has emerged in the domain of software compliance, potential research should explore more on the antecedents, causes and consequences of software workarounds. In addition to that, the extended theory of planned behavior, namely the reasoned actions of goal pursuits, which incorporates the understanding of an individual's goals that drive a particular behavior and, therefore, more applications of theory are expected to emerge in this domain. The reason for that, in some common cases the intention of doing a certain behavior is no longer seen significant towards performing an actual behavior. Rather, the behavior is driven more by the current active goals and procurement goals which are explained by the theory of reasoned goal purist.

Third, regarding the surveyed policies on software compliance, there is a need to develop distinctions on compliance policies that consider the peculiarities of both open-source software and proprietary software.

Fourth, based on the reviewed studies, there is a lack research efforts on the mechanisms that support the enforcement and provide a better visibility to the concerned stakeholders (e.g., compliance experts, business managers, software architects). A lack of research is also found on policies which are

related to home-office users of organizational software systems, as the remote work and telecommuting has become the norm during pandemics, and more likely to grow.

Fifth, since compliance automation can help addressing various compliance challenges which are caused due to manual human errors and mistakes, further studies and efforts are worth spent on developing the supporting tools for enhancing the automation of compliance management.

The empirical study extends the software compliance domain by providing an understanding on factors influencing the workarounds in software engineering. In other words, the study adds to the knowledge base through evaluating technostress as one of the causes of workarounds, and assesses the moderating role of neutralization, autonomy, and perceived behavioral controls. With the evolving research on workarounds, its causes and consequences, the study has the following theoretical implications:

Sixth, the study positions technostress and strain as causes of the workaround behavior. Researchers on software compliance should consider technostress as an antecedent, when studying the workarounds phenomenon. While the study looks at the relationship between technostress and workarounds from the lens of the theory of planned behavior (section 3.5.5), further studies can also consider viewing such a relationship from the perspective of stress and coping theory, or resistance theories in order to investigate and compare whether the workaround can be classified as a coping

behavior to stress, or a way of resistance. This, in turn, helps compare and further assess how well these theories explain the phenomena from different theoretical points of views.

Seventh, the new theoretical implication also comes from delivering evidence on the moderating role of neutralization in strengthening/weakening the impact of the hypothesized factors on workarounds (section 3.5.6). While no prior study tested neutralization in behavioral workarounds, this adds to the current literature an understanding and empirical evidence on how well neutralization theory fits in the context of workarounds, in which we consider it as an explanation of an individual's subjective norms. Although the moderating role of neutralization is found to only moderate the impact of technology overload, and rejects to moderate others, previous studies argue that the neutralization is an instable phenomenon. It is perceived differently between different people, even differently within the same individual over different times or circumstances. In this regard, to properly evaluate this variation of neutralization and its impact on the workaround behavior, a longitudinal study needs to be conducted. Further studies can also investigate the role of neutralization with respect to its influence on the theory of workarounds, with the aforementioned considerations.

Eighth, with the challenging concerns in agile approaches and the need for autonomous individuals and teams, the study adds to the literature evidence on the extent to which professional autonomy plays a moderator role

in the context of workarounds. Although, the result of CB-SEM shows significant moderating impact of autonomy, the result of PLS-SEM shows a significant direct impact on the workaround intention. By understanding this, our study positions autonomy as one of the significant contributors that predicts the workaround behavior (section 3.5.7). Having such an understanding on the role of autonomy would help improve addressing some of the challenges raised by agile approaches. This entails that future research should not ignore the role of autonomy when studying behavioral phenomena of software compliance.

Finally, the perceived behavioral control is derived from the theory of planned behavior (Ajzen, 1991), and tested in the context of workarounds. Having tested the perceived behavioral control in this study (section 3.7.3.3), this factor well explains how an individual's perception of his/her ability or difficulties in using different alternative solutions, can strengthen or weaken their behavioral intention towards workarounds. Therefore, the individual skills can be one of the main determinants that studies should consider in the software compliance behavior and workarounds in particular. This finding also helps viewing workarounds from the lens of the theory of planned behavior. Reasonably, there seem to be an interplay between the autonomy and the perceived behavioral controls, in which potential research can investigate in the software engineering workarounds.

4.2.2 Practical Implications

The research offers several managerial implications for practitioners and policy makers helping them evaluate software policy compliance, and the antecedents of workarounds in software engineering, for the aim of improving the overall software quality and compliance. Based on the empirical evidences presented, the following are a set of practical and managerial implications that this research delivers:

First: as per the findings of the review, practitioners and decision makers should place a priority to (1) SETA (security education, training and awareness), as it is found effective in mitigating insiders' threat including negligence and non-malicious activities. As software attacks caused by insiders' threat accounts for 56%, it is of paramount importance to address the causes behind such attacks in order to control them. (2) Automation of compliance management can help address most of the manual and error-prone checking for compliance, including misconfiguration or misinterpretation of requirements. (3) Build social bond within an organization enhances the sense of belonging and creates a culture of compliance, and hence, commitment to corporate IT policy.

Second, as the empirical findings of the study reveal that, the factors of technostress cause strain and lead to workarounds behavior, their ultimate consequences can also be threatful to an organization from the perspective of security vulnerabilities and technical risks. This is represented in professional

field by the term “technical debt”, in which extra efforts have to be spent in the future in order to fulfil the quality requirements, which were compromised as a result of accumulated engineering workarounds. The consequences of technical debt can be severe in the long run, if the refactoring decisions are not made on time. What even worsens the problem, is the high rate of turnover of software engineers (42% according to a study by DigitalOcean (2022)), in which alongside with the accumulated workarounds, can put a business at risk. The longer it takes to pay those technical debt, the more expensive it becomes to deal with, as the software itself continues to evolve, in order to meet business needs. As the results of the study show that factors of technostress can trigger workarounds directly and indirectly with a significant moderation of neutralization, autonomy and behavioral controls (section 3.7.3.1), practitioners should carefully work on mitigating the impact of the aforementioned factors and work on controlling workarounds.

Third, the results of the impact of each of the factors of technostress, can be addressed by adapting the policies identified in the second chapter, in order to control the impact of technostress factors. The following are some of the policy prescriptions that managers should consider to mitigate the impact of technostress and control workarounds:

- Software engineers face technological complexities, and unpleasant feelings resulting from the multifaceted new software technologies, requiring constant efforts to understand and use them. This

complexity is found to cause workarounds directly and indirectly, therefore, organizations should invest in skill-building and knowledge exchange programs, to help mitigate the pressure raised from technological complexity. Furthermore, as the turnover rate of software engineers is very high (42%), existing workarounds and technical debt can also add up to this complexity, and as a result, it can trigger more workarounds. Hence, there is a need to constantly evaluate existing technical debt and workarounds on regular basis.

- In order to mitigate the impact of technology overload and control its impact on workarounds, managers and practitioners should properly estimate the time and efforts needed to accomplish tasks, so that they can control the workflow acceleration caused by technology. In addition to that, they need to identify the threshold at which the technology overload started exceeding the pace of engineers to catchup, because after that point, there is a likelihood of workarounds, and quality tradeoffs being made. Addressing this issue of technology overload is very tricky, even with proper estimation of tasks, there is a possibility of cyber loafing, i.e., use of organization's time to do personal work, as a result of being connected the whole time. In this regard, viewing technology overload from the perspective of information overload and connectivity overload, could also result in techno strain, and in turn, more workarounds.

- To control the impact of technology invasion, organization should pay attention to controlling the tasks that might spill into engineers' private life and endanger their work-life balance. Otherwise, the frequent sense of invasion of one's personal space is likely to lead to strain and, hence, causes more workarounds. Alternatively, an organization might have to be clear about the nature of software engineers' work, especially in case of software incidents, which is typical for DevOps engineers, requiring them to be able to accept tasks anytime and, probably, anywhere.
- The technology insecurity, which is the fear of losing jobs because of technology, causes workarounds, according to the results. Since the number of tasks accomplished is, typically, viewed as a metric for productivity and, in turn, sustaining one's job, organizations need to consider the tradeoffs in throughput and quality of accomplished tasks. In such a case, practitioner might need to integrate quality with productivity through enforcement of software best practices frameworks, that require developers to work on predefined software blueprints, and leverage the supporting tools in order to enforce that.

Third, the recent advances in technology and connectivity make insiders stay connected the whole time and, as a result, it deepens the negative impact of technology on the overall productivity as prior studies indicate. While the impact of technostress, in general, seems inevitable and might not

be easily controlled at workplaces, incorporating and enforcing policies, such as software certification and compliance by design, can offer a greater impact on the overall compliance and control, to some extent, the resulting workarounds. Therefore, practitioners and decision makers should develop supporting mechanisms and tools in order to help mitigate and control workarounds.

Fourth, as the findings indicate that engineers' degree of autonomy and control over technical decisions play a significant role towards contributing to implementation of workarounds (section 3.5.7); it is crucial to carefully consider the level of autonomy given to engineers in that balance between responsibility and constraints. While agile approaches require individuals and teams to be more autonomous, as their role goes beyond developing a software, and they need to involve in other organizational units to understand the problem as a whole. Such autonomy could result in giving more freedom over technical decisions and, hence, more space for implementing workarounds; as the findings of this research reveal. Furthermore, the notions of crowdsourcing and distributed teams have emphasized the need towards more professional autonomy of software engineers, whether on work schedule, technical related decisions, or methods of accomplishing their tasks. As this can give a space for working arounds the technical decisions, and in line with prior studies which report that software engineers more likely fail to deliver beyond functional requirements and purse responsible engineering best

practices; the introduction of x-by-design concepts (including compliance-by-design and privacy-by-design), can provide a foundational software blueprint that consider controlling the scope autonomy, while ensuring to some extent compliant software services. Hence, having clear policies and tools in place that empowers autonomy and provide a better visibility to all stakeholders is of paramount importance.

Finally, as mentioned earlier that workarounds performed by software engineers typically result in the so-called technical debt, and studies report that these technical debt cost around 25% of extra efforts wasted on refactoring. Moreover, the accumulation of these technical debt is likely to trigger more workarounds in future releases of a software; as indicated by prior studies. Consequently, the cost of refactoring becomes more expensive in the long run, if not close to the cost rebuilding the software from scratch. Therefore, managers and practitioners need to carefully consider the serious consequences of workarounds and their causes presented by this study and prior studies as well. From an organizational perspective, it is very challenging to evaluate whether engineers experience any dimension of technostress and consider neutralization strategies in order to give themselves a justification for implementing workarounds. This is because such phenomena are self-observed in nature and can only be evaluated by individuals themselves. In this regard, an organization may consider two different ways to control workarounds. One way is to consider restricting the

level of autonomy and shift some of the freedom over technical decisions to upper management; and in this case it might limit their agility and innovativeness. Alternatively, management might need to investigate workarounds and evaluate technical debt in regular way, so that these debts do not accumulate in the long run and become difficult to deal with. Furthermore, organizations should encourage engineers to voluntary report any workarounds implemented, in order strategize proper remedies for addressing them, with respect to organizational priorities and tradeoffs considered.

4.3 Research Contribution

The research consists of two main studies which offer detailed analysis and investigation in software policy compliance with a focus on factors impacting software engineering workarounds. The first study involved detailed analysis of 84 selected studies identified based on the review protocol. The study, adopts evidence-based thinking to investigate requirements, theories, factors and policies in software compliance. The second study focuses on investigating workarounds in software engineering as one of the findings of the first study highlight the theory of workarounds as an emerged theory. The study uses a deductive quantitative approach and provides an extended explanation on the concepts and arguments in detail. Details also given on the qualitative part of proposed theoretical model, empirical data sample and procedure the study conducts. The study presents a detailed analysis of findings of two different structural equation models of CB-SEM

and PLS-SEM and comparison of results from the two models; and discussion is elaborated and connected to the key concepts and theories; and implications are developed accordingly. This research contributes to theoretical and practical knowledge as following:

First: this research extends the theory of workarounds with factors of technostress as antecedents to the workaround behavior, and contextualize the understanding of the workaround phenomenon in the field of software engineering. Previous studies argue that time pressure, misfit of work practices, complexity of technology, inadequate IT resources, and misunderstanding between work system stakeholders; are the main causes of workarounds. No prior study considered the dimensions of technostress as antecedents that could lead to development of workarounds, which is the main contribution that this study introduces.

Second: the study also contributes to literature through incorporating neutralization techniques, professional autonomy and perceived behavioral control as moderators on the impact of technostress and strain on the workaround behavior. By evaluating the extent to which these moderators play in strengthening or weakening the impact towards the workaround behavior, the study adds to knowledge base empirical evidence on the moderating impact of these factors. This would help researchers considers such moderating impact when further studying phenomenon in other contexts or perhaps studying similar phenomena.

Third: the study also contributes to the literature through an integration of the theories of workarounds, planned behavior with technostress. In other words, the research evaluates the impact of technostress on workarounds from the lens of the theory of planned behavior as an overarching theory. This helps bringing an understanding and explanation of the workaround phenomenon from the perspective of planned behavior, while calls for evaluating and explaining the impact of technostress, neutralization, autonomy, and perceived behavioral control on the workaround from other theoretical lenses.

Finally: the practical contribution of the study is that it helps practitioners and organizations consider such antecedents of workarounds and the consequences resulting from technostress. From a policy perspective, the findings of the study provide insights that can guide setting up policies which could help further understands the causes of workarounds in order to control their consequences. Additionally, the study provides practitioners with empirical evidence that can guide them to consider a proper level of professional autonomy to be given to engineers over the technical decisions. Better understanding of the role of autonomy can be crucial to develop a balance between responsibility and regulation. The study calls for paying attention to investigating workarounds in software engineering, and analyzing their causes and consequences of as the cost of refactoring can be more expensive in the long run.

4.4 Research Limitations

This research acknowledges several limitations based on the two studies conducted. Although the review process and selection of articles is conducted rigorously, the likelihood of missing relevant studies, which could have an impact on the findings and comprehensivity of the review, is a limitation of the review study. In addition to that, the review focuses only on factors that directly influence behavioral compliance and does not consider those that have an indirect influence, which is another limitation. It can also be noticed that the results of some policies and factors were not tested in more than a single context and, therefore, might not be generalizable to all contexts. In such cases, additional tests might be needed to obtain more support for generalizability.

The empirical study also acknowledges that the findings of this research come with some limitations. *Firstly*, the impact of control variables has not been statistically observed by the study. The same results are also reported by Marchiori et al. (2019). *Secondly*, while most of the results that are confirmed by both of the CB-SEM and PLS-SEM analysis methods gives strong evidence, some mismatches of the results can, conclusively, limit the evidence and, hence, needs further investigation. *Thirdly*, the findings of the study are likely to be impacted by the culture of respondents, Korea, which might accordingly confirm or reject some of the hypotheses based on that. *Finally*, generalizing the findings of the study can be one of the limitations in

this research since the theoretical model is tested in Korean context. The peculiarities of cultures could lead to some difference in the results having the study being conducted in a different context.

Bibliography

Aggarwal, A., & Dhurkari, R. K. (2023). Association between stress and information security policy non-compliance behavior: A meta-analysis. *Computers & Security*, *124*, 102991. https://doi.org/10.1016/j.cose.2022.102991

Agnew, R. (1992). Foundation for a General Strain Theory of Crime and Delinquency*. *Criminology*, *30*(1), 47–88. https://doi.org/10.1111/j.1745-9125.1992.tb01093.x

Agnew, R. (2001). Building on the Foundation of General Strain Theory: Specifying the Types of Strain Most Likely to Lead to Crime and Delinquency. *Journal of Research in Crime and Delinquency*, *38*(4), 319–361. https://doi.org/10.1177/0022427801038004001

Agnew, R., & Brezina, T. (2019). General Strain Theory. In M. D. Krohn, N. Hendrix, G. Penly Hall, & A. J. Lizotte (Eds.), *Handbook on Crime and Deviance* (pp. 145–160). Springer International Publishing. https://doi.org/10.1007/978-3-030-20779-3_8

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179–211. https://doi.org/10.1016/0749-5978(91)90020-T

Ajzen, I., & Kruglanski, A. W. (2019). Reasoned action in the service of goal pursuit. *Psychological Review*, *126*(5), 774–786. https://doi.org/10.1037/rev0000155

Ali, R. F., Dominic, P. D. D., & Ali, K. (2020). Organizational Governance, Social Bonds and Information Security Policy Compliance: A Perspective towards Oil and Gas Employees. *Sustainability*, *12*(20), Article 20. https://doi.org/10.3390/su12208576

Ali, R. F., Dominic, P. D. D., Ali, S. E. A., Rehman, M., & Sohail, A. (2021). Information Security Behavior and Information Security Policy Compliance: A Systematic Literature Review for Identifying the Transformation Process from Noncompliance to Compliance. *Applied Sciences*, *11*(8), Article 8. https://doi.org/10.3390/app11083383

AlOmar, E. A., Christians, B., Busho, M., AlKhalid, A. H., Ouni, A., Newman, C., & Mkaouer, M. W. (2022). SATDBailiff-mining and tracking self-admitted technical debt. *Science of Computer Programming*, *213*, 102693. https://doi.org/10.1016/j.scico.2021.102693

Alter, S. (2014). Theory of Workarounds. *Business Analytics and Information Systems*. https://repository.usfca.edu/at/40

Alter, S. (2015). *Beneficial noncompliance and detrimental compliance: Expected paths to unintended consequences*. 2015 Americas

Conference on Information Systems, AMCIS 2015. Scopus.

Alwin, D. F., & Hauser, R. M. (1975). The Decomposition of Effects in Path Analysis. *American Sociological Review*, *40*(1), 37–47. https://doi.org/10.2307/2094445

Antignac, T., Scandariato, R., & Schneider, G. (2018). Privacy Compliance Via Model Transformations. *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 120–126. https://doi.org/10.1109/EuroSPW.2018.00024

Antinyan, V., & Sandgren, H. (2021). Software Safety Analysis to Support ISO 26262-6 Compliance in Agile Development. *IEEE Software*, *38*(3), 52–60. https://doi.org/10.1109/MS.2020.3026145

Aptean. (2018). *An Overview of the State of Enterprise Software Workarounds 2018*. Aptean.Com. https://www.aptean.com/en-US/insights/blog/an-overview-of-the-state-of-enterprise-software-workarounds-2018

Arizon-Peretz, R., Hadar, I., Luria, G., & Sherman, S. (2021). Understanding developers' privacy and security mindsets via climate theory. *Empirical Software Engineering*, *26*(6), 123. https://doi.org/10.1007/s10664-021-09995-z

Arnetz, B. B., & Wiholm, C. (1997). Technological stress: Psychophysiological symptoms in modern offices. *Journal of*

Psychosomatic Research, *43*(1), 35–42. https://doi.org/10.1016/S0022-3999(97)00083-4

Astrachan, C. B., Patel, V. K., & Wanzenried, G. (2014). A comparative study of CB-SEM and PLS-SEM for theory development in family firm research. *Journal of Family Business Strategy*, *5*(1), 116–128. https://doi.org/10.1016/j.jfbs.2013.12.002

Ayyagari, R., Grover, V., & Purvis, R. (2011). Technostress: Technological Antecedents and Implications. *MIS Quarterly*, *35*(4), 831–858. https://doi.org/10.2307/41409963

Balozian, P., & Leidner, D. (2017). Review of IS Security Policy Compliance: Toward the Building Blocks of an IS Security Theory. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems*, *48*(3), 11–43. https://doi.org/10.1145/3130515.3130518

Balozian, P., Leidner, D., & Xue, B. (2021). Toward an intellectual capital cyber security theory: Insights from Lebanon. *Journal of Intellectual Capital*, *ahead-of-print*(ahead-of-print). https://doi.org/10.1108/JIC-05-2021-0123

Bansal, G., Muzatko, S., & Shin, S. I. (2020). Information system security policy noncompliance: The role of situation-specific ethical orientation. *Information Technology & People*, *34*(1), 250–296.

https://doi.org/10.1108/ITP-03-2019-0109

Barati, M., Rana, O., Petri, I., & Theodorakopoulos, G. (2020). GDPR Compliance Verification in Internet of Things. *IEEE Access*, *8*, 119697–119709. https://doi.org/10.1109/ACCESS.2020.3005509

Barlow, J. B., Warkentin, M., Ormond, D., & Dennis, A. R. (2013). Don't make excuses! Discouraging neutralization to reduce IT policy violation. *Computers & Security*, *39*, 145–159. https://doi.org/10.1016/j.cose.2013.05.006

Bauer, S., & Bernroider, E. (2014). *An Analysis of the Combined Influences of Neutralization and Planned Behavior on Desirable Information Security Behavior*. 12.

Baumgartner, H., & Homburg, C. (1996). Applications of structural equation modeling in marketing and consumer research: A review. *International Journal of Research in Marketing*, *13*(2), 139–161. https://doi.org/10.1016/0167-8116(95)00038-0

Beane, M. I. (Matthew I. (2017). *Operating in the shadows: The productive deviance needed to make robotic surgery work* [Thesis, Massachusetts Institute of Technology]. https://dspace.mit.edu/handle/1721.1/113956

Becker, C., Chitchyan, R., Betz, S., & McCord, C. (2018). Trade-off decisions across time in technical debt management: A systematic literature

review. *Proceedings of the 2018 International Conference on Technical Debt*, 85–94. https://doi.org/10.1145/3194164.3194171

Becker, T. E. (2005). Potential Problems in the Statistical Control of Variables in Organizational Research: A Qualitative Analysis With Recommendations. *Organizational Research Methods*, *8*(3), 274–289. https://doi.org/10.1177/1094428105278021

Bednar, K., Spiekermann, S., & Langheinrich, M. (2019). Engineering Privacy by Design: Are engineers ready to live up to the challenge? *The Information Society*, *35*(3), 122–142. https://doi.org/10.1080/01972243.2019.1583296

Besker, T., Martini, A., & Bosch, J. (2022). The use of incentives to promote technical debt management. *Information and Software Technology*, *142*, 106740. https://doi.org/10.1016/j.infsof.2021.106740

Bondanini, G., Giorgi, G., Ariza-Montes, A., Vega-Muñoz, A., & Andreucci-Annunziata, P. (2020). Technostress Dark Side of Technology in the Workplace: A Scientometric Analysis. *International Journal of Environmental Research and Public Health*, *17*(21), 8013. https://doi.org/10.3390/ijerph17218013

Boudon, R. (2003). Beyond Rational Choice Theory. *Annual Review of Sociology*, *29*(1), 1–21.

https://doi.org/10.1146/annurev.soc.29.010202.100213

Brotherston, B. W. (1943). The Genius of Pragmatic Empiricism. I. *The Journal of Philosophy*, *40*(1), 14–21. https://doi.org/10.2307/2017530

Buelens, M., & Van den Broeck, H. (2007). An Analysis of Differences in Work Motivation between Public and Private Sector Organizations. *Public Administration Review*, *67*(1), 65–74. https://doi.org/10.1111/j.1540-6210.2006.00697.x

Bulgurcu, B., Cavusoglu, H., & Benbasat, I. (2010). Information Security Policy Compliance: An Empirical Study of Rationality-Based Beliefs and Information Security Awareness. *MIS Quarterly*, *34*(3), 523–548. https://doi.org/10.2307/25750690

Burns, A. J., Roberts, T. L., Posey, C., Bennett, R. J., & Courtney, J. F. (2018). Intentions to Comply Versus Intentions to Protect: A VIE Theory Approach to Understanding the Influence of Insiders' Awareness of Organizational SETA Efforts. *Decision Sciences*, *49*(6), 1187–1228. https://doi.org/10.1111/deci.12304

Buschmann, F. (2011). To Pay or Not to Pay Technical Debt. *IEEE Software*, *28*(6), 29–31. https://doi.org/10.1109/MS.2011.150

Byrne, B. M. (2013). *Structural Equation Modeling With AMOS: Basic Concepts, Applications, and Programming, Second Edition* (2nd ed.).

Routledge. https://doi.org/10.4324/9780203805534

Carmi, G., & Bouhnik, D. (2020). The Effect of Rational Based Beliefs and Awareness on Employee Compliance with Information Security Procedures: A Case Study of a Financial Corporation in Israel. *Interdisciplinary Journal of Information, Knowledge, and Management*, *15*, 109–125.

Castellanos-Ardila, J. P., Gallina, B., & Governatori, G. (2021). Compliance-aware engineering process plans: The case of space software engineering processes. *Artificial Intelligence and Law*. https://doi.org/10.1007/s10506-021-09285-5

Chemuturi, M. (2012). *Requirements Engineering and Management for Software Development Projects*. Springer Science & Business Media.

Chen, X., Wu, D., Chen, L., & Teng, J. K. L. (2018). Sanction severity and employees' information security policy compliance: Investigating mediating, moderating, and control variables. *Information & Management*, *55*(8), 1049–1060. https://doi.org/10.1016/j.im.2018.05.011

Chen, Y., Ramamurthy, K., & Wen, K.-W. (2012). Organizations' Information Security Policy Compliance: Stick or Carrot Approach? *Journal of Management Information Systems*, *29*(3), 157–188.

https://doi.org/10.2753/MIS0742-1222290305

Choi, M., & Song, J. (2018). Social control through deterrence on the compliance with information security policy. *Soft Computing*, *22*(20), 6765–6772. https://doi.org/10.1007/s00500-018-3354-z

Coeckelbergh, M. (2006). Regulation or Responsibility? Autonomy, Moral Imagination, and Engineering. *Science, Technology, & Human Values*, *31*(3), 237–260. https://doi.org/10.1177/0162243905285839

Coleman, J. W. (1987). Toward an Integrated Theory of White-Collar Crime. *American Journal of Sociology*, *93*(2), 406–439. https://doi.org/10.1086/228750

Cram, W. A., Proudfoot, J. G., & D'Arcy, J. (2017). Organizational information security policies: A review and research framework. *European Journal of Information Systems*, *26*(6), 605–641. https://doi.org/10.1057/s41303-017-0059-9

Czepa, C., Tran, H., Zdun, U., Thi Kim, T. T., Weiss, E., & Ruhsam, C. (2017). On the Understandability of Semantic Constraints for Behavioral Software Architecture Compliance: A Controlled Experiment. *2017 IEEE International Conference on Software Architecture (ICSA)*, 155–164. https://doi.org/10.1109/ICSA.2017.10

D'Arcy, J., & Herath, T. (2011). A review and analysis of deterrence theory in

the IS security literature: Making sense of the disparate findings. *European Journal of Information Systems*, *20*(6), Article 6. https://doi.org/10.1057/ejis.2011.23

D'Arcy, J., Herath, T., & Shoss, M. K. (2014). Understanding Employee Responses to Stressful Information Security Requirements: A Coping Perspective. *Journal of Management Information Systems*, *31*(2), 285–318. https://doi.org/10.2753/MIS0742-1222310210

D'Arcy, J., & Teh, P.-L. (2019). Predicting employee information security policy compliance on a daily basis: The interplay of security-related stress, emotions, and neutralization. *Information & Management*, *56*(7), 103151. https://doi.org/10.1016/j.im.2019.02.006

Dave Farely (Director). (2021, June 24). *Where Do The Software Bugs Come From?* https://www.youtube.com/watch?v=fyYIntT1H_Q

Davison, R. M., Wong, L. H. M., Ou, C. X. J., & Alter, S. (2021). The coordination of workarounds: Insights from responses to misfits between local realities and a mandated global enterprise system. *Information & Management*, *58*(8), 103530. https://doi.org/10.1016/j.im.2021.103530

Davison, R., Wong, L. H., Alter, S., & Ou, C. (2019, May 15). Adopted globally but unusable locally: What workarounds reveal about

adoption, resistance, compliance and non-compliance. *Research Papers*. https://aisel.aisnet.org/ecis2019_rp/19

de Vargas Pinto, A., Beerepoot, I., & Maçada, A. C. G. (2022). Encourage autonomy to increase individual work performance: The impact of job characteristics on workaround behavior and shadow IT usage. *Information Technology and Management*. https://doi.org/10.1007/s10799-022-00368-6

Diamantopoulou, V., & Mouratidis, H. (2019). Practical evaluation of a reference architecture for the management of privacy level agreements. *Information & Computer Security*, *27*(5), 711–730. https://doi.org/10.1108/ICS-04-2019-0052

DigitalOcean. (2022). *Currents: A seasonal report on developer and SMB trends in the cloud*. https://www.digitalocean.com/currents/june-2022/

Do, H., & Lee, B. C. (2020). The Effects of Tourism Industry Employee's Technostress on Burnout, Organizational Conflict and Job Performance. *Event & Convention Research*, *16*(4), 135–150. https://doi.org/10.31927/asec.16.4.8

Dong, K., Ali, R. F., Dominic, P. D. D., & Ali, S. E. A. (2021). The Effect of Organizational Information Security Climate on Information Security Policy Compliance: The Mediating Effect of Social Bonding towards

Healthcare Nurses. *Sustainability*, *13*(5), Article 5. https://doi.org/10.3390/su13052800

Ejnefjäll, T., & Ågerfalk, P. J. (2019). Conceptualizing Workarounds: Meanings and Manifestations in Information Systems Research. *Communications of the Association for Information Systems*, *45*, 20. http://dx.doi.org/10.17705/1CAIS.04520

Feather, N. T. (1995). Values, valences, and choice: The influences of values on the perceived attractiveness and choice of alternatives. *Journal of Personality and Social Psychology*, *68*(6), 1135–1151. https://doi.org/10.1037/0022-3514.68.6.1135

Feigl, H., & Scriven, M. (1956). *The Foundations of Science and the Concepts of Psychology and Psychoanalysis*. U of Minnesota Press.

Froggio, G., Zamaro, N., & Lori, M. (2009). Exploring the Relationship between Strain and Some Neutralization Techniques: *European Journal of Criminology*. https://doi.org/10.1177/1477370808098106

Fuglseth, A. M., & Sørebø, Ø. (2014). The effects of technostress within the context of employee use of ICT. *Computers in Human Behavior*, *40*, 161–170. https://doi.org/10.1016/j.chb.2014.07.040

Gardazi, S. U., & Ali, A. (2017). Compliance-Driven Architecture for Healthcare Industry. *International Journal of Advanced Computer*

Science and Applications, *8*. https://doi.org/10.14569/IJACSA.2017.080571

Granlund, T., Mikkonen, T., & Stirbu, V. (2020). On Medical Device Software CE Compliance and Conformity Assessment. *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, 185–191. https://doi.org/10.1109/ICSA-C50368.2020.00040

Guhr, N., Lebek, B., & Breitner, M. H. (2019). The impact of leadership on employees' intended information security behaviour: An examination of the full-range leadership theory. *Information Systems Journal*, *29*(2), 340–362. https://doi.org/10.1111/isj.12202

Gwebu, K. L., Wang, J., & Hu, M. Y. (2020). Information security policy noncompliance: An integrative social influence model. *Information Systems Journal*, *30*(2), 220–269. https://doi.org/10.1111/isj.12257

Hale, M. L., & Gamble, R. F. (2019). Semantic hierarchies for extracting, modeling, and connecting compliance requirements in information security control standards. *Requirements Engineering*, *24*(3), 365–402. https://doi.org/10.1007/s00766-017-0287-5

Hempel, C. G. (1951). Wilfrid Sellars. Language, rules and behavior. John Dewey: Philosopher of science and freedom, a symposium, edited by

Sidney Hook, The Dial Press, New York1950, pp. 289–315. *The Journal of Symbolic Logic*, *16*(3), 209–210. https://doi.org/10.2307/2266396

Henseler, J., Ringle, C. M., & Sarstedt, M. (2015). A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science*, *43*(1), 115–135. https://doi.org/10.1007/s11747-014-0403-8

Hina, S., & Dominic, P. D. D. (2020). Information security policies' compliance: A perspective for higher education institutions. *Journal of Computer Information Systems*, *60*(3), Article 3. https://doi.org/10.1080/08874417.2018.1432996

Hina, S., Panneer Selvam, D. D. D., & Lowry, P. B. (2019). Institutional governance and protection motivation: Theoretical insights into shaping employees' security compliance behavior in higher education institutions in the developing world. *Computers & Security*, *87*, 101594. https://doi.org/10.1016/j.cose.2019.101594

Hjørland, B. (2011). Evidence-based practice: An analysis based on the philosophy of science. *Journal of the American Society for Information Science and Technology*, *62*(7), 1301–1310. https://doi.org/10.1002/asi.21523

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.

Ifinedo, P. (2012). Understanding information systems security policy compliance: An integration of the theory of planned behavior and the protection motivation theory. *Computers & Security*, *31*(1), 83–95. https://doi.org/10.1016/j.cose.2011.10.007

Ifinedo, P. (2014). Information systems security policy compliance: An empirical study of the effects of socialisation, influence, and cognition. *Information & Management*, *51*(1), 69–79. https://doi.org/10.1016/j.im.2013.10.001

Ifinedo, P. (2016). Critical Times for Organizations: What Should Be Done to Curb Workers' Noncompliance With IS Security Policy Guidelines? *Information Systems Management*, *33*(1), 30–41. https://doi.org/10.1080/10580530.2015.1117868

Ingolfo, S., Siena, A., Mylopoulos, J., Susi, A., & Perini, A. (2013). Arguing regulatory compliance of software requirements. *Data & Knowledge Engineering*, *87*, 279–296. https://doi.org/10.1016/j.datak.2012.12.004

Islam, S., Mouratidis, H., & Jürjens, J. (2011). A framework to support

alignment of secure software engineering with legal regulations. *Software & Systems Modeling*, *10*(3), 369–394. https://doi.org/10.1007/s10270-010-0154-z

Jaekang, L., & Taekyung, P. (2015). The Effect of Technostress on Counterproductive Work Behavior. *Journal of Information Technology Services*, *14*(4), 1–14. https://doi.org/10.9716/KITS.2015.14.4.001

Jeon, S., Son, I., & Han, J. (2020). Exploring the role of intrinsic motivation in ISSP compliance: Enterprise digital rights management system case. *Information Technology & People*, *34*(2), 599–616. https://doi.org/10.1108/ITP-05-2018-0256

Jervis, R. (1979). Deterrence Theory Revisited. *World Politics*, *31*(2), 289–324. https://doi.org/10.2307/2009945

Joshi, C., & Singh, U. K. (2017). Information security risks management framework – A step towards mitigating security risks in university network. *Journal of Information Security and Applications*, *35*, 128–137. https://doi.org/10.1016/j.jisa.2017.06.006

Joshi, K. P., Elluri, L., & Nagar, A. (2020). An Integrated Knowledge Graph to Automate Cloud Data Compliance. *IEEE Access*, *8*, 148541–148555. https://doi.org/10.1109/ACCESS.2020.3008964

Jr, J., Matthews, L., Matthews, R., & Sarstedt, M. (2017). PLS-SEM or CB-SEM: Updated guidelines on which method to use. *International Journal of Multivariate Data Analysis*, *1*, 107. https://doi.org/10.1504/IJMDA.2017.10008574

Julisch, K., Suter, C., Woitalla, T., & Zimmermann, O. (2011). Compliance by design – Bridging the chasm between auditors and IT architects. *Computers & Security*, *30*(6), 410–426. https://doi.org/10.1016/j.cose.2011.03.005

Kahn, R. L., & Byosiere, P. (1992). Stress in organizations. In *Handbook of industrial and organizational psychology, Vol. 3, 2nd ed* (pp. 571–650). Consulting Psychologists Press.

Keenan, A., & Newton, T. J. (1985). Stressful Events, Stressors and Psychological Strains in Young Professional Engineers. *Journal of Occupational Behaviour*, *6*(2), 151–156.

Kim, H., Lee, B., & Chung, J. (2016). An Inquiry into the Impact of Technostress on End Users: Research Trends and Future Directions. *Journal of Research in Curriculum Instruction*, *20*(6), 511–520. https://doi.org/10.24231/rici.2016.20.6.511

Kim, S. H., Yang, K. H., & Park, S. (2014). An Integrative Behavioral Model of Information Security Policy Compliance. *The Scientific World*

Journal, *2014*, e463870. https://doi.org/10.1155/2014/463870

Kim, S. S., & Kim, Y. J. (2017). The effect of compliance knowledge and compliance support systems on information security compliance behavior. *Journal of Knowledge Management*, *21*(4), 986–1010. https://doi.org/10.1108/JKM-08-2016-0353

Kitchenham, B. A., Budgen, D., & Brereton, P. (2016). *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press.

Kitchenham, B. A., Budgen, D., Brereton, P., Budgen, D., & Brereton, P. (2016). *Evidence-Based Software Engineering and Systematic Reviews*. Chapman and Hall/CRC. https://doi.org/10.1201/b19467

Kline, R. B. (2015). *Principles and Practice of Structural Equation Modeling, Fourth Edition*. Guilford Publications.

Kruchten, P., Nord, R. L., & Ozkaya, I. (2012). Technical Debt: From Metaphor to Theory and Practice. *IEEE Software*, *29*(6), 18–21. https://doi.org/10.1109/MS.2012.167

Kuutila, M., Mäntylä, M., Farooq, U., & Claes, M. (2020). Time pressure in software engineering: A systematic review. *Information and Software Technology*, *121*, 106257. https://doi.org/10.1016/j.infsof.2020.106257

LaRose, R., Connolly, R., Lee, H., Li, K., & Hales, K. D. (2014). Connection

Overload? A Cross Cultural Study of the Consequences of Social Media Connection. *Information Systems Management*, *31*(1), 59–73. https://doi.org/10.1080/10580530.2014.854097

Laumer, S., Maier, C., & Weitzel, T. (2017). Information quality, user satisfaction, and the manifestation of workarounds: A qualitative and quantitative study of enterprise content management system users. *European Journal of Information Systems*, *26*(4), 333–360. https://doi.org/10.1057/s41303-016-0029-7

Lazarus, R. S. (1966). *Psychological stress and the coping process*. McGraw-Hill.

Lehman, M. M. (1980). Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, *68*(9), Article 9. https://doi.org/10.1109/PROC.1980.11805

Lehman, M. M., & Ramil, J. F. (2002). Software Evolution and Software Evolution Processes. *Annals of Software Engineering*, *14*(1), 275–309. https://doi.org/10.1023/A:1020557525901

Lim, V. (2005). *The Moderating Effect of Neutralization Technique on Organizational Justice and Cyberloafing*. 14.

Liu, C., Wang, N., & Liang, H. (2020). Motivating information security policy compliance: The critical role of supervisor-subordinate guanxi and

organizational commitment. *International Journal of Information Management*, *54*, 102152. https://doi.org/10.1016/j.ijinfomgt.2020.102152

MacKenzie, S. B., Podsakoff, P. M., & Jarvis, C. B. (2005). The Problem of Measurement Model Misspecification in Behavioral and Organizational Research and Some Recommended Solutions. *Journal of Applied Psychology*, *90*(4), 710–730. https://doi.org/10.1037/0021-9010.90.4.710

Majumdar, S., Madi, T., Wang, Y., Jarraya, Y., Pourzandi, M., Wang, L., & Debbabi, M. (2018). User-Level Runtime Security Auditing for the Cloud. *IEEE Transactions on Information Forensics and Security*, *13*(5), 1185–1199. https://doi.org/10.1109/TIFS.2017.2779444

Malaurent, J., & Karanasios, S. (2020). Learning from Workaround Practices: The Challenge of Enterprise System Implementations in Multinational Corporations. *Information Systems Journal*, *30*(4), 639–663. https://doi.org/10.1111/isj.12272

Máñez-Carvajal, C., Cervera-Mérida, J. F., & Fernández-Piqueras, R. (2021). Web accessibility evaluation of top-ranking university Web sites in Spain, Chile and Mexico. *Universal Access in the Information Society*, *20*(1), 179–184. https://doi.org/10.1007/s10209-019-00702-w

Marchiori, D. M., Mainardes, E. W., & Rodrigues, R. G. (2019). Do Individual Characteristics Influence the Types of Technostress Reported by Workers? *International Journal of Human–Computer Interaction*, *35*(3), 218–230. https://doi.org/10.1080/10447318.2018.1449713

Marques, J., & da Cunha, A. M. (2018). Tailoring Traditional Software Life Cycles to Ensure Compliance of RTCA DO-178C and DO-331 with Model-Driven Design. *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 1–8. https://doi.org/10.1109/DASC.2018.8569351

Martin, F. (2009). *TechnicalDebtQuadrant*. Martinfowler.Com. https://martinfowler.com/bliki/TechnicalDebtQuadrant.html

Maxwell, J. C., Antón, A. I., Swire, P., Riaz, M., & McCraw, C. M. (2013). A legal cross-references taxonomy for reasoning about compliance requirements. *Requirements Engineering*, *17*(2), 99–115. https://doi.org/10.1007/s00766-012-0152-5

Merhi, M. I., & Ahluwalia, P. (2019). Examining the impact of deterrence factors and norms on resistance to Information Systems Security. *Computers in Human Behavior*, *92*, 37–46. https://doi.org/10.1016/j.chb.2018.10.031

Minor, W. W. (1981). Techniques of Neutralization: A Reconceptualization and Empirical Examination. *Journal of Research in Crime and Delinquency*, *18*(2), 295–318. https://doi.org/10.1177/002242788101800206

Montazeri, M., Khajouei, R., & Montazeri, M. (2020). Evaluating hospital information system according to ISO 9241 part 12. *DIGITAL HEALTH*, *6*, 205520762097946. https://doi.org/10.1177/2055207620979466

Moody, G. D., Siponen, M., & Pahnila, S. (2018). Toward a Unified Model of Information Security Policy Compliance. *MIS Quarterly*, *42*(1), 285-A22.

Moquin, R., & Wakefield, R. L. (2016). The Roles of Awareness, Sanctions, and Ethics in Software Compliance. *Journal of Computer Information Systems*, *56*(3), 261–270. https://doi.org/10.1080/08874417.2016.1153922

Mubarkoot, M., & Altmann, J. (2021). Towards Software Compliance Specification and Enforcement Using TOSCA. *Economics of Grids, Clouds, Systems, and Services*, 168–177. https://doi.org/10.1007/978-3-030-92916-9_14

Mubarkoot, M., Altmann, J., Rasti-Barzoki, M., Egger, B., & Lee, H. (2022).

Software Compliance Requirements, Factors, and Policies: A Systematic Literature Review. *Computers & Security*.

Nasirpouri, S. F., & Biros, D. (2020). Technostress and its influence on employee information security policy compliance. *Information Technology & People*, *35*(1), 119–141. https://doi.org/10.1108/ITP-09-2020-0610

Newman, B. (1991). Only Empiricism is Compatible with Behavior Analysis: A Response to the Socialism and Behaviorism Debate. *Behavior and Social Issues*, *1*(2), 15–24. https://doi.org/10.5210/bsi.v1i2.164

OECD. (2017). *Entrepreneurship—Enterprises by business size—OECD Data*. TheOECD. http://data.oecd.org/entrepreneur/enterprises-by-business-size.htm

Ostberg, J.-P., Graziotin, D., Wagner, S., & Derntl, B. (2020). A methodology for psycho-biological assessment of stress in software engineering. *PeerJ Computer Science*, *6*, e286. https://doi.org/10.7717/peerj-cs.286

Palanisamy, R., Norman, A. A., & Kiah, M. L. M. (2019). *Bring your own device (BYOD) security policy compliance framework*. Proceedings of the 23rd Pacific Asia Conference on Information Systems: Secure ICT Platform for the 4th Industrial Revolution, PACIS 2019. Scopus.

Palanisamy, R., Norman, A. A., & Mat Kiah, M. L. (2020). BYOD Policy Compliance: Risks and Strategies in Organizations. *Journal of Computer Information Systems*, 1–12. https://doi.org/10.1080/08874417.2019.1703225

Perez, B., Castellanos, C., & Correal, D. (2020). Developing a theory based on the causes of technical debt injection into software projects in Colombia. *Journal of Physics: Conference Series*, *1587*(1), 012022. https://doi.org/10.1088/1742-6596/1587/1/012022

Pérez, B., Castellanos, C., Correal, D., Rios, N., Freire, S., Spínola, R., Seaman, C., & Izurieta, C. (2021). Technical debt payment and prevention through the lenses of software architects. *Information and Software Technology*, *140*, 106692. https://doi.org/10.1016/j.infsof.2021.106692

Ponemon Institute. (2016). *2016 Cost of Data Center Outages*. Ponemon Institute. https://www.ponemon.org/research/ponemon-library/security/2016-cost-of-data-center-outages.html

Potdar, A., & Shihab, E. (2014). An Exploratory Study on Self-Admitted Technical Debt. *2014 IEEE International Conference on Software Maintenance and Evolution*, 91–100. https://doi.org/10.1109/ICSME.2014.31

PricewaterhouseCoopers. (2018). *Global State of Information Security® Survey 2018*. PwC. https://www.pwc.co.uk/issues/cyber-security-services/insights/global-state-of-information-security-survey.html

PricewaterhouseCoopers. (2021). *Securing critical infrastructure: Get ready as voluntary becomes mandatory*. PwC. https://www.pwc.com/us/en/services/consulting/cybersecurity-risk-regulatory/library/biden-memo-signals-private-sector-cyber-performance-goals.html

Proofpoint. (2022). *2022 Ponemon Cost of Insider Threats Global Report*. https://www.proofpoint.com/sites/default/files/threat-reports/pfpt-us-tr-the-cost-of-insider-threats-ponemon-report.pdf

Putnam, H. (1995). Pragmatism. *Proceedings of the Aristotelian Society*, *95*, 291–306.

Putri, F., & Hovav, A. (2014). *Employees' compliance with BYOD security policy: Insights from reactance, organizational justice, and protection motivation theory*. ECIS 2014 Proceedings - 22nd European Conference on Information Systems. Scopus.

Ragu-Nathan, T. S., Tarafdar, M., Ragu-Nathan, B. S., & Tu, Q. (2008). The Consequences of Technostress for End Users in Organizations: Conceptual Development and Empirical Validation. *Information*

Systems Research, *19*(4), 417–433. https://doi.org/10.1287/isre.1070.0165

Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., Lopez, G., Izurieta, C., Seaman, C., & Spinola, R. (2022). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. *Journal of Systems and Software*, *184*, 111114. https://doi.org/10.1016/j.jss.2021.111114

Reinecke, L., Aufenanger, S., Beutel, M. E., Dreier, M., Quiring, O., Stark, B., Wölfling, K., & Müller, K. W. (2017). Digital Stress over the Life Span: The Effects of Communication Load and Internet Multitasking on Perceived Stress and Psychological Health Impairments in a German Probability Sample. *Media Psychology*, *20*(1), 90–115. https://doi.org/10.1080/15213269.2015.1121832

Rios, N., Spínola, R. O., Mendonça, M., & Seaman, C. (2018). The most common causes and effects of technical debt: First results from a global family of industrial surveys. *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–10. https://doi.org/10.1145/3239235.3268917

Rogers, R. W., & Prentice-Dunn, S. (1997). Protection motivation theory. In *Handbook of health behavior research 1: Personal and social determinants* (pp. 113–132). Plenum Press.

Roland, H. E., & Moriarty, B. (1991). *System Safety Engineering and Management*. John Wiley & Sons.

Samavi, R., & Consens, M. P. (2018). Publishing privacy logs to facilitate transparency and accountability. *Journal of Web Semantics*, *50*, 1–20. https://doi.org/10.1016/j.websem.2018.02.001

Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research methods for business students (all UK editions)* (18th ed.). Pearson. https://scholar.google.com/citations?view_op=view_citation&hl=en&user=Myy3jGEAAAAJ&citation_for_view=Myy3jGEAAAAJ:PyEswDtIyv0C

Sekaran, U., & Bougie, R. (2010). *Research Methods for Business: A Skill Building Approach*. John Wiley & Sons.

Shadbad, F. N., & Biros, D. (2021). Does Technostress Trigger Insider Threat? A Conceptual Model and Mitigation Solutions. In Z. W. Y. Lee, T. K. H. Chan, & C. M. K. Cheung (Eds.), *Information Technology in Organisations and Societies: Multidisciplinary Perspectives from AI to Technostress* (pp. 61–83). Emerald Publishing Limited.

https://doi.org/10.1108/978-1-83909-812-320211003

Shook, C. L., Ketchen Jr., D. J., Hult, G. T. M., & Kacmar, K. M. (2004). An assessment of the use of structural equation modeling in strategic management research. *Strategic Management Journal*, *25*(4), 397–404. https://doi.org/10.1002/smj.385

Silic, M., Barlow, J. B., & Back, A. (2017). A new perspective on neutralization and deterrence: Predicting shadow IT usage. *Information and Management*, *54*(8), 1023–1037. Scopus. https://doi.org/10.1016/j.im.2017.02.007

Singi, K., Kaulgud, V., Bose, R. P. J. C., & Podder, S. (2019). CAG: Compliance Adherence and Governance in Software Delivery Using Blockchain. *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 32–39. https://doi.org/10.1109/WETSEB.2019.00011

Singi, K., R P, J. C. B., Podder, S., & Burden, A. P. (2019). Trusted Software Supply Chain. *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 1212–1213. https://doi.org/10.1109/ASE.2019.00141

Siponen, M., Adam Mahmood, M., & Pahnila, S. (2014). Employees' adherence to information security policies: An exploratory field study.

Information & Management, *51*(2), 217–224. https://doi.org/10.1016/j.im.2013.08.006

Siponen, M., & Vance, A. (2010). Neutralization: New Insights into the Problem of Employee Information Systems Security Policy Violations. *MIS Quarterly*, *34*(3), 487–502. https://doi.org/10.2307/25750688

Sojer, M., Alexy, O., Kleinknecht, S., & Henkel, J. (2014). Understanding the Drivers of Unethical Programming Behavior: The Inappropriate Reuse of Internet-Accessible Code. *Journal of Management Information Systems*, *31*(3), 287–325. https://doi.org/10.1080/07421222.2014.995563

Song, D., Zhong, H., & Jia, L. (2020). The Symptom, Cause and Repair of Workaround. *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 1264–1266.

Srivastava, S. C., Chandra, S., & Shirish, A. (2015). Technostress creators and job outcomes: Theorising the moderating influence of personality traits. *Information Systems Journal*, *25*(4), 355–401. https://doi.org/10.1111/isj.12067

Stafford, T., Deitz, G., & Li, Y. (2018). The role of internal audit and user training in information security policy compliance. *Managerial Auditing Journal*, *33*(4), 410–424. https://doi.org/10.1108/MAJ-07-

2017-1596

Steffens, A., Lichter, H., & Moscher, M. (2018). *Towards data-driven continuous compliance testing*. *2066*, 78–84. Scopus.

Sykes, G. M., & Matza, D. (1957). Techniques of Neutralization: A Theory of Delinquency. *American Sociological Review*, *22*(6), 664–670. https://doi.org/10.2307/2089195

T. Alanazi, S., Anbar, M., A. Ebad, S., Karuppayah, S., & Al-Ani, H. A. (2020). Theory-Based Model and Prediction Analysis of Information Security Compliance Behavior in the Saudi Healthcare Sector. *Symmetry*, *12*(9), Article 9. https://doi.org/10.3390/sym12091544

Tarafdar, M., Pullins, E. Bolman., & Ragu-Nathan, T. S. (2015). Technostress: Negative effect on performance and possible mitigations. *Information Systems Journal*, *25*(2), 103–132. https://doi.org/10.1111/isj.12042

Tarafdar, M., Tu, Q., & Ragu-Nathan, T. S. (2010). Impact of Technostress on End-User Satisfaction and Performance. *Journal of Management Information Systems*, *27*(3), 303–334. https://doi.org/10.2753/MIS0742-1222270311

Tekla S. Perry. (2022, April 12). *Today's Software Engineering Salaries, in 5 Charts*. IEEE Spectrum. https://spectrum.ieee.org/software-engineer-salary-2657117801

Tran, H., Zdun, U., Holmes, T., Oberortner, E., Mulo, E., & Dustdar, S. (2012). Compliance in service-oriented architectures: A model-driven and view-based approach. *Information and Software Technology*, *54*(6), 531–552. https://doi.org/10.1016/j.infsof.2012.01.001

Trang, S., & Brendel, B. (2019). A Meta-Analysis of Deterrence Theory in Information Security Policy Compliance Research. *Information Systems Frontiers*, *21*(6), 1265–1284. https://doi.org/10.1007/s10796-019-09956-4

Tsohou, A., & Holtkamp, P. (2018). Are users competent to comply with information security policies? An analysis of professional competence models. *Information Technology & People*, *31*(5), 1047–1068. https://doi.org/10.1108/ITP-02-2017-0052

UN. (2020). *UN E-Government Survey 2020*. https://publicadministration.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2020

Usman, M., Felderer, M., Unterkalmsteiner, M., Klotins, E., Mendez, D., & Alégroth, E. (2020). Compliance Requirements in Large-Scale Software Development: An Industrial Case Study. *Product-Focused Software Process Improvement*, 385–401. https://doi.org/10.1007/978-3-030-64148-1_24

van Eck, N. J., & Waltman, L. (2018). *VOSviewer Manual*. 51.

W3C. (2022). *World Wide Web Consortium (W3C)*. https://www.w3.org/

Wall, J. D., & Palvia, P. (2013). *Control-related motivations and information security policy compliance: The effect of reflective and reactive autonomy*. *2*, 894–902. Scopus.

Ward Cunningham (Director). (2009, February 15). *Debt Metaphor*. https://www.youtube.com/watch?v=pqeJFYwnkjE

Wiafe, I., Koranteng, F. N., Wiafe, A., Obeng, E. N., & Yaokumah, W. (2020). The role of norms in information security policy compliance. *Information & Computer Security*, *28*(5), 743–761. https://doi.org/10.1108/ICS-08-2019-0095

Wickramage, C., Fidge, C., Ouyang, C., & Sahama, T. (2019). Generating Log Requirements for Checking Conformance against Healthcare Standards using Workflow Modelling. *Proceedings of the Australasian Computer Science Week Multiconference*, 1–10. https://doi.org/10.1145/3290688.3290739

Willis, H. E. (1925). Definition of Law, A. *Virginia Law Review*, *12*(3), 203–214.

Wong, L. H. M., Hurbean, L., Davison, R. M., Ou, C. X., & Muntean, M. (2022). Working around inadequate information systems in the

workplace: An empirical study in Romania. *International Journal of Information Management*, *64*, 102471. https://doi.org/10.1016/j.ijinfomgt.2022.102471

Wu, W., Yang, Q., Gong, X., & Davison, R. M. (2022). Understanding sustained participation in crowdsourcing platforms: The role of autonomy, temporal value, and hedonic value. *Information Technology & People*, *ahead-of-print*(ahead-of-print). https://doi.org/10.1108/ITP-09-2019-0502

Yli-Huumo, J., Maglyas, A., & Smolander, K. (2014). The Sources and Approaches to Management of Technical Debt: A Case Study of Two Product Lines in a Middle-Size Finnish Software Company. *Product-Focused Software Process Improvement*, 93–107. https://doi.org/10.1007/978-3-319-13835-0_7

Yli-Huumo, J., Maglyas, A., & Smolander, K. (2015). The Benefits and Consequences of Workarounds in Software Development Projects. In J. M. Fernandes, R. J. Machado, & K. Wnuk (Eds.), *Software Business* (pp. 1–16). Springer International Publishing. https://doi.org/10.1007/978-3-319-19593-3_1

Yli-Huumo, J., Maglyas, A., & Smolander, K. (2016). How do software development teams manage technical debt? – An empirical study.

Journal of Systems and Software, *120*, 195–218. https://doi.org/10.1016/j.jss.2016.05.018

Yoon, J. S., Jang Seob Yoon, & Jang. (2021). *Software industry in South Korea*. Statista. https://www.statista.com/topics/7253/software-industry-in-south-korea/

Zandesh, Z., Ghazisaeedi, M., Devarakonda, M. V., & Haghighi, M. S. (2019). Legal framework for health cloud: A systematic review. *International Journal of Medical Informatics*, *132*, 103953. https://doi.org/10.1016/j.ijmedinf.2019.103953

Appendix

[1] Scholarly databases and corresponding search queries and used for each database to retrieve results.

Scholarly Database	**Search Query**
Google Scholar (Titles Only)	software compliance; compliance "information systems"; compliance "distributed systems"; compliance "software systems"; compliance "service-oriented systems"
Web of Science	("software compliance" OR "compliance of software" OR (compliance AND "information systems") OR (compliance AND "distributed systems") OR (compliance AND "software systems") OR (compliance AND "service-oriented systems"))
ScienceDirect	"software compliance" OR "compliance of software" OR (compliance AND "information systems") OR (compliance AND "distributed systems") OR (compliance AND "software systems") OR (compliance AND "service-oriented systems")
Scopus	("software *compliance" OR "*compliance of software" OR (*compliance AND "information system*") OR (*compliance AND "distributed system*") OR (*compliance AND "software system*") OR (*compliance AND "service-oriented system*"))
ACM Digital Library	("software compliance" OR "compliance of software" OR (compliance AND "information systems") OR (compliance AND "distributed systems") OR (compliance AND "software systems") OR (compliance AND "service-oriented systems"))
IEEE Xplore	"software compliance" OR "compliance of software" OR (compliance AND "information systems") OR (compliance AND "distributed systems") OR (compliance AND "software systems") OR (compliance AND "service-oriented systems")

[2] List of reviewed studies included.

#	Title	Authors
1	Compliance in service-oriented architectures: A model-driven and view-based approach	(Tran et al., 2012)
2	Do Brazilian Federal Agencies Specify Accessibility Requirements for the Development of their Mobile Apps?	(Oliveira et al., 2020)

#	Title	Authors
3	Web accessibility evaluation of top-ranking university Web sites in Spain, Chile and Mexico	(Máñez-Carvajal et al., 2021)
4	Evaluating hospital information system according to ISO 9241 part 12	(Montazeri et al., 2020)
5	On the Understandability of Semantic Constraints for Behavioral Software Architecture Compliance: A Controlled Experiment	(Czepa et al., 2017)
6	Business policy compliance in service-oriented systems	(Weigand et al., 2011)
7	On the verification of mission-related properties in software-intensive systems-of-systems architectural design	(Silva et al., 2020)
8	Generating Log Requirements for Checking Conformance against Healthcare Standards using Workflow Modelling	(Wickramage et al., 2019)
9	Compliance by design − Bridging the chasm between auditors and IT architects	(Julisch et al., 2011)
10	Beneficial noncompliance and detrimental compliance: Expected paths to unintended consequences	(Alter, 2015)
11	Towards data-driven continuous compliance testing	(Steffens et al., 2018)
12	Understanding the Drivers of Unethical Programming Behavior: The Inappropriate Reuse of Internet-Accessible Code	(Sojer et al., 2014)
13	The Roles of Awareness, Sanctions, and Ethics in Software Compliance	(Moquin & Wakefield, 2016)
14	Managing license compliance in free and open source software development	(Gangadharan et al., 2012)
15	GDPR Compliance Verification in Internet of Things	(Barati et al., 2020)
16	Practical evaluation of a reference architecture for the management of privacy level agreements	(Diamantopoulou & Mouratidis, 2019)
17	Publishing privacy logs to facilitate transparency and accountability	(Samavi & Consens, 2018)
18	Engineering Privacy by Design: Are engineers ready to live up to the challenge?	(Bednar et al., 2019)

#	Title	Authors
19	Privacy Compliance Via Model Transformations	(Antignac et al., 2018)
20	Operationalizing Privacy Compliance for Cloud-Hosted Sharing of Healthcare Data	(Eze et al., 2018)
21	Compliance Requirements in Large-Scale Software Development: An Industrial Case Study	(Usman et al., 2020)
22	On Medical Device Software CE Compliance and Conformity Assessment	(Granlund et al., 2020)
23	A legal cross-references taxonomy for reasoning about compliance requirements	Maxwell et al., 2013
24	A framework to support alignment of secure software engineering with legal regulations	(Islam et al., 2011)
25	Arguing regulatory compliance of software requirements	(Ingolfo et al., 2013)
26	ChainSDI: A Software-Defined Infrastructure for Regulation-Compliant Home-Based Healthcare Services Secured by Blockchains	(Li et al., 2020)
27	Law Architecture for Regulatory-Compliant Public Enterprise Model: A Focus on Healthcare Reform in Egypt	(Mohamed et al., 2021)
28	An Integrated Knowledge Graph to Automate Cloud Data Compliance	(Joshi et al., 2020)
29	Tailoring Traditional Software Life Cycles to Ensure Compliance of RTCA DO-178C and DO-331 with Model-Driven Design	(Marques & da Cunha, 2018)
30	Enabling Functional Safety ASIL Compliance for Autonomous Driving Software Systems	(Chitnis et al., 2017)
31	Compliance-aware engineering process plans: the case of space software engineering processes	(Castellanos-Ardila et al., 2021)
32	Software Safety Analysis to Support ISO 26262-6 Compliance in Agile Development	(Antinyan & Sandgren, 2021)
33	Effect of Fear on Behavioral Intention to Comply	(Faizi & Rahman, 2020)
34	Influencing factors of employees' information systems security police compliance: An empirical research in China	(Liu et al., 2020)
35	Employees' compliance with BYOD security policy: Insights from reactance, organizational justice, and protection motivation theory	(Putri & Hovav, 2014)

#	Title	Authors
36	Fostering Information Security Compliance: Comparing the Predictive Power of Social Learning Theory and Deterrence Theory	(Lembcke et al., 2019)
37	CyberSPL: A Framework for the Verification of Cybersecurity Policy Compliance of System Configurations Using Software Product Lines	(Varela-Vaca et al., 2019)
38	Semantic hierarchies for extracting, modeling, and connecting compliance requirements in information security control standards	(Hale & Gamble, 2019)
39	Assessing Risk of Security Non-compliance of Banking Security Requirements Based on Attack Patterns	(Rongrat & Senivongse, 2018)
40	Hospital Staff's Adherence to Information Security Policy: A Quest for the Antecedents of Deterrence Variables	(Kuo et al., 2021)
41	Toward an intellectual capital cyber security theory: insights from Lebanon	(Balozian et al., 2021)
42	The Effect of Organizational Information Security Climate on Information Security Policy Compliance: The Mediating Effect of Social Bonding towards Healthcare Nurses	(Dong et al., 2021)
43	The Effect of Rational Based Beliefs and Awareness on Employee Compliance with Information Security Procedures: A Case Study of a Financial Corporation in Israel	(Carmi & Bouhnik, 2020)
44	Theory-Based Model and Prediction Analysis of Information Security Compliance Behavior in the Saudi Healthcare Sector	(T. Alanazi et al., 2020)
45	Universal and Culture-dependent Employee Compliance of Information Systems Security Procedures	(Karjalainen et al., 2020)
46	The role of norms in information security policy compliance	(Wiafe et al., 2020)
47	Explaining the interactions of humans and artifacts in insider security behaviors: The mangle of practice perspective	(Van Slyke & Belanger, 2020)
48	The role of abusive supervision and organizational commitment on employees' information security policy noncompliance intention	(Guan & Hsu, 2020)
49	Information system security policy noncompliance: the role of situation-specific ethical orientation	(Bansal et al., 2020)
50	Organizational Governance, Social Bonds and Information Security Policy Compliance: A Perspective towards Oil and Gas Employees	(Ali et al., 2020)
51	Motivating information security policy compliance: The critical role of supervisor-subordinate guanxi and organizational commitment	(Liu et al., 2020)

#	Title	Authors
52	Exploring the role of intrinsic motivation in ISSP compliance: enterprise digital rights management system case	(Jeon et al., 2020)
53	Institutional governance and protection motivation: Theoretical insights into shaping employees' security compliance behavior in higher education institutions in the developing world	(Hina et al., 2019)
54	Integrating Cognition with an Affective Lens to Better Understand Information Security Policy Compliance	(Ormond et al., 2019)
55	Examining the impact of deterrence factors and norms on resistance to Information Systems Security	(Merhi & Ahluwalia, 2019)
56	The impact of leadership on employees' intended information security behaviour: An examination of the full-range leadership theory	(Guhr et al., 2019)
57	Social control through deterrence on the compliance with information security policy	(Choi & Song, 2018)
58	Sanction severity and employees' information security policy compliance: Investigating mediating, moderating, and control variables	(Chen et al., 2018)
59	The role of internal audit and user training in information security policy compliance	(Stafford et al., 2018)
60	User-Level Runtime Security Auditing for the Cloud	(Majumdar et al., 2018)
61	Intentions to Comply Versus Intentions to Protect: A VIE Theory Approach to Understanding the Influence of Insiders' Awareness of Organizational SETA Efforts	(Burns et al., 2018)
62	Towards analysing the rationale of information security non-compliance: Devising a Value-Based Compliance analysis method	(Kolkowska et al., 2017)
63	Indirect effect of management support on users' compliance behaviour towards information security policies:	(Humaidi & Balakrishnan, 2017)
64	Practice-based discourse analysis of information security policies	(Karlsson et al., 2017)
65	The effect of compliance knowledge and compliance support systems on information security compliance behavior	(Kim & Kim, 2017)
66	Understanding Employee Responses to Stressful Information Security Requirements: A Coping Perspective	(D'Arcy et al., 2014)
67	Organizations' Information Security Policy Compliance: Stick or Carrot Approach?	(Chen et al., 2012)

#	Title	Authors
68	Critical Times for Organizations: What Should Be Done to Curb Workers' Noncompliance With IS Security Policy Guidelines?	(Ifinedo, 2016)
69	Understanding information systems security policy compliance: An integration of the theory of planned behavior and the protection motivation theory	(Ifinedo, 2012)
70	Information systems security policy compliance: An empirical study of the effects of socialisation, influence, and cognition	(Ifinedo, 2014)
71	An Integrative Behavioral Model of Information Security Policy Compliance	(Kim et al., 2014)
72	Employees' adherence to information security policies: An exploratory field study	(Siponen et al., 2014)
73	Complexity is dead, long live complexity! How software can help service providers manage security and compliance	(Thalmann et al., 2014)
74	An approach to checking the compliance of user permission policy in software development	(Truong & Nguyen, 2013)
75	The information content of Sarbanes-Oxley in predicting security breaches	(Westland, 2020)
76	CAG: Compliance Adherence and Governance in Software Delivery Using Blockchain	(Singi et al., 2019)
77	Adopted globally but unusable locally: what workarounds reveal about adoption, resistance, compliance and non-compliance	(Davison et al., 2019)
78	Publishing privacy logs to facilitate transparency and accountability	(Samavi & Consens, 2018)
79	Trustable outsourcing of business processes to cloud computing environments	(Alsouri et al., 2011)
80	When and why developers adopt and change software licenses	(Vendome et al., 2015)
81	Automated Certification for Compliant Cloud-based Business Processes	(Accorsi et al., 2011)
82	Trust calibration of automated security IT artifacts: A multi-domain study of phishing-website detection tools	(Chen et al., 2021)
83	Accessibility, usability, and security evaluation of universities' prospective student web pages: a comparative study of Europe, North America, and Oceania	(Macakoğlu et al., 2022)
84	Understanding developers' privacy and security mindsets via climate theory	(Arizon-Peretz et al., 2021)

[3] Measurement Instrument for the Survey

<설문작성 예시>

※ 질문 문항에 해당되는 것에 "○" 및 "√" 표시 등으로 표기하여 주시기 바랍니다.

절대 아니다	아니다	약간 아니다	보통이다	약간 그렇다	그렇다
①	②	③	④	⑤	⑥

English	Korean	Source
Technology Complexity (TCX) TCX1. I do not have enough time to study and upgrade my technology skills. TCX2. I find that my colleagues know more about software technologies than I do. TCX3. I often find it too complex for me to understand and use new technologies. TCX4. I often find it difficult to understand my organization's software policies.	TCX1. 나는 나의 기술 능력을 공부하고 업그레이드하는데 충분한 시간을 투자할 수 없다. TCX2. 동료들은 나보다 더 많은 소프트웨어 관련 기술들을 알고 있다는 것을 알고 있다. TCX3. 나는 종종 새로운 기술을 이해하고 사용하는 것이 너무 복잡하다고 생각한다. TCX4. 내가 소속된 조직의 소프트웨어 정책을 이해하기가 어려운 경우가 많다.	(D'Arcy et al., 2014; Nasirpouri & Biros, 2020)
Technology Overload (TOV) TOV1. Technology is making the work procedure faster than before, and l am forced to cope with that pace. TOV2. Technology is making the work procedure faster than before, and l am forced to work more than I can handle.	TOV1. 나는 빠르게 변화하는 기술들로 빠르게 업무를 처리하도록 압박감을 받는다. TOV2. 나는 빠르게 변화하는 기술들로 내가 감당할 수 있는 것보다 더 많은 작업을 처리하도록 압박을 받는다. TOV3. 나는 다양한 유형의 기술들로 빡빡한 스케줄 속에	(Ragu-Nathan et al., 2008; Nasirpouri & Biros, 2020)

TOV3. Technology is making the work procedure faster than before, and l am forced to work with very tight time schedules. TOV4. I have a higher workload because of increased technology complexity.	업무처리에 대한 압박을 받는다. TOV4. 나는 증가하는 기술 복잡성으로 인해 많은 업무 과부하를 경험한다.	
Technology Uncertainty (TUC) TUC1. There are always new developments in technologies we use in our organization. TUC2. There are regular changes in computer software/hardware in our organization. TUC3. There are frequent upgrades in computer systems and applications in our organization.	TUC1. 우리 회사(조직)에서 사용하고 있는 기술은 항상 새로운 변화가 발생한다. TUC2. 우리 회사(조직)에서 사용하고 있는 컴퓨터 소프트웨어 및 하드웨어는 끊임없이 정기적으로 변화가 이루어지고 있다. TUC3. 우리 회사(조직)에서는 컴퓨터 시스템과 어플리케이션의 업그레이드가 빈번하게 이루어지고 있다.	(Nasirpouri & Biros, 2020)
Technology Invasion (TNV) TNV1. I spend less time with my family due to using different types of technologies. TNV2. I have to be in touch with my work even during my vacation due to using different types of technologies. TNV3. I have to sacrifice my vacation and weekend time to keep up to date with new technologies. TNV4. I feel my personal life is being invaded because of technology.	TNV1. 다른 종류의 기술 사용으로 인해 가족과 보내는 시간이 적다. TNV2. 나는 다른 유형의 IT 기술사용을 위해 휴가 중에도 업무 연락을 취하고 있다. TNV3. 나는 새로운 최신 기술 사용을 유지하기 위해 주말과 휴가와 주말 시간을 희생해야 한다. TNV4. 나는 기술 때문에 내 개인생활이 침해되고 있다고 느낀다.	(Ragu-Nathan et al., 2008)

Technology Insecurity (TNS) TNS1. I feel a constant threat to my job security due to new technologies that I do not fully master. TNS2. I have to constantly update my technology skills to avoid being replaced. TNS3. I am threatened by coworkers who master newer technology skills. TNS4. I do not share my knowledge with my coworkers for fear of being replaced. TNS5. I feel there is less sharing of knowledge among coworkers for fear of being replaced.	TNS1. 나는 내가 완벽히 사용할 수 없는 신기술로 인해 나의 직업 안정성에 지속적으로 위협을 느낀다. TNS2. 나는 내 자리가 다른 사람으로 대체되는 것을 피하기 위해 지속적으로 IT 기술 역량을 업데이트해야 한다고 생각한다. TNS3. 나는 새로운 IT 기술을 완벽히 습득한 동료들에게 위협을 느끼고 있다. TNS4. 나는 다른 사람으로 내 자리를 대체될 수 있다는 두려움으로 동료들과 내 지식을 공유하지 않는다. TNS5. 나는 다른 사람으로 내 자리를 대체될 수 있다는 두려움으로 동료들과 지식공유를 줄이고 있다고 생각한다.	(Ragu-Nathan et al., 2008)
Strain (ST) ST1. I feel drained from activities that require me to use technologies. ST2. I feel tired from my technology-related activities. ST3. Working all day with different types of technologies is a strain for me. ST4. I feel burned out from my technology-related activities.	ST1. 나는 나에게 요구하는 기술 사용과 관련된 활동에 지친다. ST2. 나는 기술 관련 활동으로 인해 피곤함을 느낀다. ST3. 언제나 다양한 종류의 기술을 사용하여 작업하는 것은 나에게 스트레스(부담)이다. ST4. 나의 기술관련 활동에 지친다고 생각된다.	(Nasirpouri & Biros, 2020; Moore, 2000)

Neutralization (NT) NT1. It is ok to bypass software related policies if no harm is done. NT2. It is ok to violate software related policies if it is too restrictive to accomplish the work. NT3. It is ok to bypass software related policies to get a job done faster. NT4. It is ok to bypass software related policies when you are under a tight deadline. NT5. It is ok to violate software related policies if everybody is doing so.	NT1. 아무런 피해가 없다면, 소프트웨어 관련 정책을 무시하는 것은 괜찮다고 생각한다. NT2. 소프트웨어 관련 정책이 작업 수행에 너무 엄격하고 제한적인 경우, 이를 위반할 수 있다고 생각한다. NT3. 작업을 더욱 빨리 완료하기 위해 소프트웨어 관련 정책을 무시해도 된다고 생각한다. NT4. 마감일이 촉박할 때는 소프트웨어 관련 정책을 생략하는 것은 괜찮다고 생각한다. NT5. 모든 사람이 소프트웨어 관련 정책을 위반하고 있다면, 이를 위반하는 것은 문제가 없다고 생각한다.	(Siponen & Vance, 2010)
Intention to Use Workarounds (IUW) IUW1. I had the intention to use alternative ways to accomplish my required tasks. IUW2. I had the intention to bypass the work procedures when they are complicated. IUW3. I had the intention to skip the work procedures when they do not fit with my work practices. IUW4. I had the intention to look	IUW1. 내게 요구되는 업무를 완수하기 위해 다른 대체방안 사용하려는 의도가 있었다. IUW2. 절차가 복잡한 경우, 정해진 절차를 생략할 수 있다는 의도가 있었다. IUW3. 정해진 절차가 내 업무 관행에 맞지 않을 때는 생략하려는 의도가 있었다. IUW4. 절차가 나의 업무에 방해될 때 다른 대안을 찾으려는 의도가 있었다.	(Ajzen, 1991; M.-F. Chen et al., 2009)

for other alternatives when the work procedure obstructs my work.		
Workaround Behavior (WB) WB1. I often bypass some procedures in order to accomplish required tasks. WB2. I usually use different ways to achieve the same goal if the procedure is complicated. WB3. If the prescribed procedure obstructs my work, I look for other alternatives. WB4. I bypass prescribed procedures if they do not fit with my work practices.	WB1. 필요한 작업을 수행하기 위해서는 종종 몇 가지 절차를 생략한다. WB2. 나는 동일한 목표를 달성하기 위해 보통 다른 방법들을 사용한다. WB3. 만약 정해진 절차가 업무에 방해가 된다면, 나는 보통 다른 대안들을 찾는다. WB4. 나는 정해진 절차가 업무 관행에 적합하지 않은 경우에는 정해진 절차를 생략한다.	(Ejnefjäll & Ågerfalk, 2019; Laumer et al., 2017; Alter, 2014)
Autonomy (AUT) AUT1. In my organization, I have freedom over the technical decisions. AUT2. In my organization, I have control over the technical decisions. AUT3. My organization gives me independence and freedom in the way I accomplish my tasks.	AUT1. 나는 우리 조직에서 기술관련 의사결정에 있어 자유롭게 의사결정을 한다. AUT2. 나는 우리 조직에서 기술적 의사결정은 통제하고 있다. AUT3. 우리 조직은 내게 업무 수행 방법에의 독립성과 자유를 보장한다	(Coeckelbergh, 2006)
Perceived Behavioral Control (PBC) PBC1. I could easily use workarounds if I wanted. PBC2. Nothing outside of my control could prevent me from implementing workarounds.	PBC1. 내가 원한다면, 나는 쉽게 대체방안(회피책)을 사용할 수 있다. PBC2. 나의 통제 밖의 어떤 것도 내가 대체방안(회피책)을 수행하는 것을 막을 수 없다.	(Ajzen, 1991)

PBC3. It would be up to me whether or not I use workarounds.	PBC3. 내가 대체 방안(회피책)을 쓸 지 안 쓸지는 대부분 나에게 달려있다.	

Demographic Questions**:**

English	Korean
1. Gender: ① Male, ② female **2. Age** : ① <=20s ② 30s ③ 40s ④ 50s ⑤ >=60s **3. Education:** ① High school or equivalent ② Junior college graduate ③ Bachelor degree ④ Master degree ⑤ Doctoral degree or above **4. Profession**: ① Software Engineering, system analysis, consulting ② System, network, security ③ System programming ④ Web programming ⑤ Application programming **5. Industry** ① Energy② Materials ③ Industrials ④ Consumer Discretionary ⑤ Consumer Staples ⑥ Educational Services ⑦ Healthcare Financials ⑧ Information Technology 9) Communication Services 10) Public Service 11) Manufacturing 12) Utilities 13) Real Estate **6. Years of Experience:** ① 3< 5 ② 5 < 7 ③ 7 < 9 ④ 9 < 11 ⑤ 11 < 13 ⑥ > 13 **7. Size of organization (number of employees)** ① <10 ② 10 < 50 ③ 50 < 250 ④ > 250	**1. Gender 귀하의 성별은?** ① 남 성 ② 여 성 **2. Age 귀하의 나이는?** ① 20 대 이하 ② 30 대 ③ 40 대 ④ 50 대 ⑤ 60 대 이상 **3. Education 귀하의 교육수준은?** ① 고등학교 졸업 이하 ② 전문대학 졸업 ③ 대학교 졸업 ④ 대학원(석사) 졸업 ⑤ 대학원 박사 이상 **4. Profession 귀하의 전문분야는?** ① SW 엔지니어링, 시스템 분석, 컨설팅 ② 시스템, 네트워크, 보안 ③ 시스템 프로그래밍 ④ 웹 프로그래밍 ⑤ 앱 프로그래밍 **5. Industry 귀하가 소속된 조직의 소속 산업은?** ① 에너지② 재료 ③ 공업용④ 소비자 재량 ⑤ 소비자용 스테이플 ⑥ 교육⑦ 의료보건 금융 ⑧ 정보기술 (IT) 9) 통신 서비스 10) 공공서비스 11) 제조 12) 공익 사업 13) 부동산 **6. Career 귀하의 업무경력은?** ① 3-5년 ② 5-7년 ③ 7-9년

	④ 9-11년 ⑤ 11-13년 13년 이상 **7. Firm size 귀사의 기업규모(종업원 수)는?** ① 10명 미만 ② 10명~50명 ③ 50명-250명 ④ 250명 이상

Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Human Resource Development Project for Global R&DB Program (IITP-2019-0-01328) supervised by the IITP (Institute for information and communications Technology, Planning, and Evaluation).

Abstract (Korean)

본고는 56%의 소프트웨어 공격이 내부자에 의해 발생한다는 점에서 비롯한다. 또한, 소프트웨어 공격이나 사고를 경험한 조직의 수는 지난 2년간 7% 증가하여 67%를 기록했다. 단기적 소프트웨어 공학 해결책은 중요한 기업적 문제이지만 비교적 논의되지 않은 분야이기 때문에, 본고는 해당 해결책으로 인한 결과 또한 논의한다. 단기적 소프트웨어 공학 해결책을 통한 단기적 이익은 시간이 지남에 따라 회복할 수 없는 기술적 부채를 초래할 수 있다. 더하여, 단기적 해결책과 임시 수정은 향후 소프트웨어 릴리스와 전체 보안 및 유지에도 영향을 미칠 수 있다. 단기적 해결책은 인력 낭비 중 25% 이상을 차지하기 때문에, 해결책 개선을 위한 요소를 조사하고, 원인을 이해하고 해결하기 위해 해결책 개선을 위한 요소를 조사할 필요가 있다. 본고는 다음의 두가지 주요 연구를 제시한다.

첫번째 연구에서는 체계적인 문헌 연구를 통해 소프트웨어 컴플라이언스의 현재 연구 동향, 이론과 개념의 발전, 잠재적인 차이와 방향을 검토한다. 본 연구에서는 검토를 위한 질문에 답하기 위해 증거 기반 사고를 활용하였다. 검토 프로토콜과 포함 및 제외 기준에 기초하여, 84개의 관련 연구를 검토하였다. 검토를 통해 다양한

범위에서 행동 컴플라이언스에 영향을 미치는 55개의 요인과 이들과 관련된 컴플라이언스 문제에 관한 20가지 정책을 확인했다. 주요한 검토 결과는 다음과 같다. (1) 최종 사용자 보안은 법률 및 개인정보 보호 문제와 연결된 주요 문제다. (2) 보안 인식 및 규정 준수 자동화가 가장 많이 인용된 정책이다. (3) 도메인 및 컴플라이언스 전문가와 소프트웨어 엔지니어 간의 견해 차이에도 강조점이 있다. (4) 계획된 행동 이론이 통설이나, 단기적 해결책 이론도 등장하고 있다. (5) 도메인에는 진화한 개념들이 있는데, 설계에 의한 컴플라이언스 및 개인정보 보호, 코드, 보안 스트레스, 홈 오피스 사용자로서의 정책이 그것이다. 본 연구는 연구자와 실무자에게 연구 방향과 정책 지침에 관한 제언을 함으로서 일련의 이론적 및 실제적 함의를 갖는다.

두번째 연구에서는 연역적, 양적 방법을 통해 소프트웨어 공학 해결책에 영향을 미치는 요인과, 기술 스트레스 요인이 단기적 해결책으로 이어지는 정도를 조사한다. 또한 해당 영향의 조절자로 기능하는 중립화 전략, 전문적 자율성, 인식된 행동 통제의 역할을 조사한다. 본 연구는 소프트웨어 공학 해결책이 등장하기까지의 요인을 평가하고, 단기적 해결책의 관점에서 기술 스트레스에 대한 새로운

이해를 제공하는 동시에 이것이 소프트웨어 공학 해결책에 미치는 영향이 얼마나 큰지를 강조하는 것을 목표로 한다. 본 연구는 기술 스트레스를 해결책의 새로운 선행 사건으로 제시한다. 소프트웨어 엔지니어가 어떤 소프트웨어 업종 종사자보다 기술적인 복잡성에 대한 이해도가 높기 때문에, 소프트웨어 엔지니어를 중점으로 단기적 해결책을 설명한다. 문헌 연구에 따르면 단기적 해결책의 원인은 마감일 준수, 작업 관행의 잘못된 적합성, 불충분한 리소스, 압도적인 기술의 복잡성에 대한 압박에서 비롯된다고 나타나지만, 본 연구에서는 전술한 원인 중 기술 스트레스를 단기적 해결책의 원인으로 가정한다. 그 외에도, 본 연구는 중립화, 기술적 결정에 관해 엔지니어에게 주어진 전문적 자율성의 정도, 인식된 행동 통제가 그 영향을 강화할 수 있다고 주장한다. 본 연구는 306개 소프트웨어 엔지니어를 대상으로 진행한 단면 연구 데이터를 기반으로, 공분산 기반(CB) 및 부분 최소 제곱(PLS) 구조 방정식 모델(SEM)을 적용하여 제안된 연구 모델을 평가한다. 본 연구는 CB-SEM과 PLS-SEM의 연구결과를 상세하게 분석하고 비교하였다.

결론적으로 기술 스트레스 (과부하 및 침입)이 압박을 통해 간접적으로 단기적 해결책을 예측하는 반면, 복잡성, 과부하, 침입은 압

박에만 직접적인 영향을 미치는 것으로 나타났다. 또한, 연구 결과 기술 스트레스 (과부하 및 보안 미흡)가 단기적 해결책을 구현하고자 하는 의도에 직접적인 영향을 미치는 것으로 나타났다. CB-SEM과 PLS- SEM의 연구 결과 중립화가 기술 과부하를 제외하고는 조절에 유의미한 영향을 미치지 못한다는 것으로 나타났다. CB-SEM 분석에서는 자율성과 인식된 행동 통제가 압박과 단기적 해결책을 구현하려는 의도 사이의 관계 조절에 있어 유의미한 효과를 갖는다고 나타난 반면, PLS-SEM 분석에서는 유의미한 효과가 없는 것으로 나타났다. 본 연구는 해결책 이론을 확장하고, 소프트웨어 공학의 관점에서 기술 스트레스와, 엔지니어의 단기적 해결책 탐색에 미치는 중립화, 자율성, 행동 제어라는 조절자에 대한 새로운 이해를 제시한다. 본 연구의 결과는 실무자가 단기적 해결책을 통제할 정책을 개발하고, 연구자가 향후 연구를 위한 추가적인 인사이트를 얻는 데 도움이 될 것이다.

주요어 : 소프트웨어 컴플라이언스, 소프트웨어 정책, 기술 스트레스, 중립화, 자율성, 단기적 해결책, 기술 부채

학 번 : 2019-39915