Received 10 May 2023; revised 4 March 2024; accepted 18 April 2024. Date of publication 23 April 2024;
date of current version 17 May 2024. The review of this article was arranged by Editor V. Piuri.

Digital Object Identifier 10.1109/OJSE.2024.3392691

Exploring Current Practices and Challenges of HIPAA Compliance in Software Engineering: Scoping Review

FARAH ELKOURDI, CHENHAO WEI, LU XIAO, ZHONGYUAN YU, AND ONUR ASAN

School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030 USA

CORRESPONDING AUTHOR: ONUR ASAN (e-mail: oasan@stevens.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/OJSE.2024.3392691, provided by the authors.

ABSTRACT Healthcare systems and applications are increasingly used to improve patient care. However, these applications face data security, privacy, and regulatory compliance challenges. The health insurance portability and accountability act (HIPAA) regulates the use and disclosure of patient health information. Ensuring HIPAA compliance in the software engineering process poses critical challenges to software engineering practitioners. This review focuses on understanding the state-of-the-art in the current literature for ensuring HIPAA compliance in the software development life cycle, namely, requirement gathering, software design, implementation, software testing, and evolution. The findings of this study shed light on software engineers in creating HIPAA compliance healthcare systems and applications. This literature review presents the key themes and trends in this research area. Also, it provides recommendations for future research in the intersection of software engineering methods and HIPAA compliance.

INDEX TERMS Health insurance portability and accountability act, healthcare regulation, healthcare system, health insurance portability and accountability act (HIPAA), privacy, protected health information, regulatory compliance, security, software development life cycle, software engineering.

I. INTRODUCTION

The health insurance portability and accountability act (HIPAA)—a federal law that required the creation of national standards to protect sensitive patient health information from being disclosed without the patient's consent or knowledge—was passed in 1996 to take advantage of technology to lower costs, enhance quality, and guarantee the portability and continuance of related systems [1]. While the original focus of HIPAA was not directly on addressing privacy concerns, the advent of technology in healthcare and the digitization of health information eventually highlighted the importance of protecting Patient Health Information (PHI). Prior studies show that medical data breaches are the second highest risk [8], exposing patients to financial damage, mental stress, and social stigma [13]. As a result, 75% of consumers using health websites hesitate to share their personal information without their consent [18]. In response, the US Congress included privacy rules and security rules as part of HIPAA to address these concerns [11]. HIPAA plays a crucial role in protecting the privacy and security of PHI. Compliance with HIPAA is an ethical obligation and a legal requirement to protect patients' personal health information [16].

Balancing security with business operations is a critical challenge in managing information security risks, which requires proactive investment strategies [12]. Software engineers play a critical role in ensuring that healthcare applications and systems comply with HIPAA regulations as the significance of technology in healthcare continues to grow. HIPAA requires that all software systems and applications that handle PHI implement appropriate security controls, such as access controls, authentication, and encryption, to ensure the confidentiality and integrity of PHI. In addition to security controls, HIPAA requires that software engineers implement specific technical safeguards, such as data backup and disaster recovery procedures, to ensure that PHI remains available and protected against loss, damage, or theft [15]. Furthermore, HIPAA requires that software engineers implement audit controls to track and monitor PHI access and ensure that PHI is

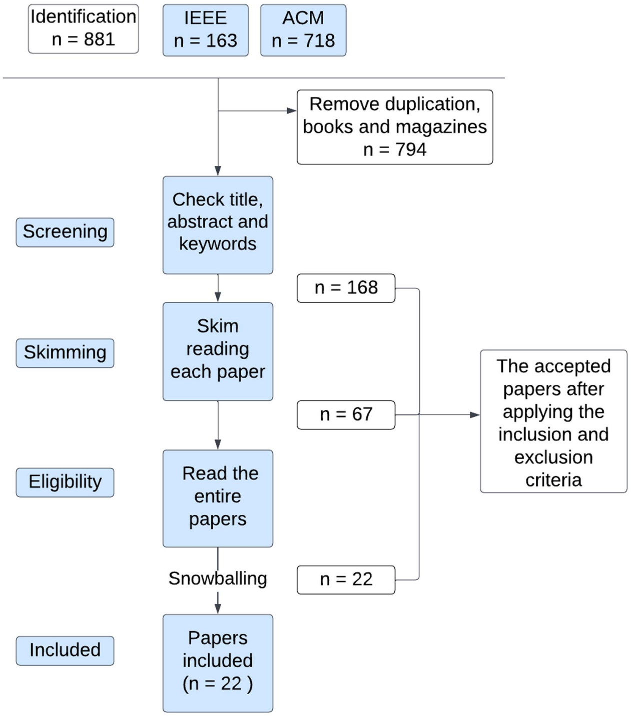


FIGURE 1. Overview of the study process.

used only for authorized purposes. All these aspects of HIPAA require software engineers to deliver HIPAA-compliant systems that are secure and reliable to maintain the privacy of patients' information. Software engineers must be knowledgeable of HIPAA regulations and take the necessary engineering steps to ensure that healthcare software systems are HIPAA compliant [6].

The impact of software engineering practices on HIPAA compliance cannot be overstated, with its expertise directly contributing to PHI's safeguarding and ethical management. Software engineering ensures the privacy, integrity, and accessibility of PHI by strictly following software development best practices and design patterns. From ensuring electronic health record systems are constructed in a manner that is legally compliant [14] to embedding HIPAA privacy rules within healthcare applications through the creation of decision engines [2], software engineering showcases its indispensable capability to navigate and bridge the complex realms of technology and regulatory compliance, thereby ensuring the operational integrity and compliance of health information systems.

Both HIPAA and software engineering are well-studied in their own field. However, there still exists a knowledge gap at the intersection of HIPAA and software engineering, which brings challenges for software engineers as they need to balance the technical and regulatory requirements with the practical demands of delivering effective software systems. It is critical to address questions such as how to ensure compliance with HIPAA regulations while maintaining functionality and usability, the best practices for implementing HIPAA-compliant systems, and how to ensure the security and reliability of HIPAA-compliant systems over time. These questions highlight the need to better understand the gap between HIPAA and software engineering and how to bridge this gap to deliver secure and effective healthcare systems.

In order to answer these questions, we conducted a scoping review to explore the trending literature on the application of software engineering techniques in the area of HIPAA. The purpose of this review is to understand how software engineering techniques are being used to address HIPAA- related problems and solutions in different software life cycle phases, as this is a new and under-researched area.

II. BACKGROUND

A. SOFTWARE ENGINEERING PHASES

Software engineering is the process of designing, creating, testing, and maintaining software applications. Software engineering aims to produce high-quality software that meets the needs of its users and stakeholders [17]. To achieve this goal, software engineers follow a structured process with several phases, each with a specific set of activities. The phases of software engineering are as follows.

1) REQUIREMENT ANALYSIS

The first phase of software engineering involves gathering and documenting the requirements of the software system. This phase involves working with stakeholders to understand their needs, identify the software system's goals and objectives, and define the requirements for the software.

2) SOFTWARE DESIGN

In this phase, software engineers focus on the high-level design of the software system by determining the overall structure, relationships between components, and communication patterns. They also make decisions on the choice of algorithms and data structures and the user interface design, which must be intuitive and user-friendly. These decisions significantly impact the software system's performance and efficiency and set the foundation for the rest of the software development process.

3) SOFTWARE IMPLEMENTATION

The software implementation phase is where the actual code is written, and the software is built. Software engineers use the design specifications and requirements to guide software development.

4) SOFTWARE TESTING

In the testing phase, the software is tested to ensure that it meets the requirements and works as intended. This phase includes various types of testing, such as unit, integration, and system testing.

5) SOFTWARE MAINTENANCE

The final phase of software engineering is software maintenance. This phase involves fixing bugs, making updates and improvements, and providing on-going support for the software system.

Each phase of software engineering is essential and contributes to the overall quality and success of the software system. A well-executed software engineering process results in a software system that is reliable, efficient, and meets the needs of its users and stakeholders. We use the software development phase to clarify the papers into five different software engineering categories.

B. SOFTWARE QUALITY ATTRIBUTES

Quality attributes in software engineering refer to the non-functional characteristics of a software system that are used to evaluate its overall performance and satisfaction [4]. These attributes, such as reliability, usability, scalability, security, maintainability, performance, portability, and testability, provide a framework for assessing the fitness of a software system for its intended purpose and environment.

1) RELIABILITY

Refers to the ability of a software system to perform its intended functions correctly and consistently, even in the presence of errors or unexpected conditions.

2) USABILITY

Describes how easily and effectively users can interact with a software system, including the user interface, navigation, and overall user experience.

3) SCALABILITY

Refers to a software system's ability to handle the increased load, such as more users or increased data, without a significant decline in performance.

4) SECURITY

Concerns the measures taken to protect a software system from unauthorized access, use, disclosure, disruption, modification, or destruction.

5) MAINTAINABILITY

Refers to the ease with which a software system can be modified, corrected, or updated over time to address changing requirements or to improve performance.

6) PERFORMANCE

Indicates a software system's speed, responsiveness, and efficiency in meeting its functional requirements.

7) PORTABILITY

Refers to the ability of a software system to run on different hardware or software platforms without modification.

TABLE 1. **Inclusion and Exclusion Criteria**

Index	Inclusion and Exclusion Criteria
1	Capture peer-revived journals and full conference proceeding papers
2	Exclude literature review papers and previous versions of extended papers
3	Including HIPAA and Software Engineering techniques
4	Being published between 2012 to 2022 and written in English

8) TESTABILITY

Refers to the degree to which a software system can be tested effectively and efficiently to verify that it meets its requirements and specifications.

C. HIPAA REGULATION

The HIPAA of 1996 primarily aimed to enhance insurance portability and standardize the electronic management of health care transactions. Over time, as the management and exchange of health information increasingly shifted to electronic formats, the U.S. Department of Health and Human Services (HHS) recognized the critical need to protect this information. Consequently, HIPAA evolved to include robust privacy and security regulations [1]. As a special committee in HHS prepared several recommendations based upon extensive expert witness testimony from academia, industry, and government, deriving the following enforcement.

The *Privacy Rule* sets national standards for the protecting of individually identifiable health information by three types of covered entities: 1) health plans, 2) health care clearinghouses, and 3) health care providers who conduct the standard health care transaction electronically [3]. The *Security Rule* sets national standards for protecting electronically protected health information's confidentiality, integrity, and availability [10]. The *Enforcement Rule* states the actions that HHS must take to ensure compliance and accountability under HIPAA [19]. The full description of HIPAA can be found at the HHS's official website [1]. This article focuses on how recent articles reported or used various software engineering techniques in the studies covering HIPAA security and privacy rule compliance.

III. METHODOLOGY

A. STUDY DESIGN / PROTOCOL REVIEW

We performed a scoping review to explore the trending literature on the application of software engineering techniques in the HIPAA area. We conducted this scoping review following the reporting guidance from the preferred reporting items for systematic reviews and meta-analyses extension for scoping reviews statement [21]. Scoping reviews are an effective and useful strategy for synthesizing emerging concepts and topics in a specific domain [21]. In this study, we explore how software engineering techniques are used to address HIPAA-related problems and solutions, a new area that has not been comprehensively reviewed. Table 1 presents our inclusion and exclusion criteria in the scoping review.

B. SEARCH STRATEGY AND DATABASE

We utilized two primary databases, including IEEE and ACM, the two major venue publishing studies in software engineering. The search criteria were also limited to journals and full conference proceedings papers published in English between 2012 to 2022 to capture the latest trend. We excluded review papers as well as grey literature and preprint publications. We defined the search string and retrieved the initial set of papers. The initial search was based only on the "HIPAA" keyword to be included in the abstract, title, or keywords. to capture all HIPAA-related papers in these two databases. With consecutive filtering, we applied software engineering related concepts in the search and reviewed each paper to determine if the paper is related to software engineering. F. Elkourdi and C. Wei conducted the initial review of the papers for selection, and the other three authors subsequently confirmed the selections, resolving any conflicts or disagreements.

We conducted three rounds of selection to determine the included papers in our final study data set, followed by final snowballing technique [5], [9], [20]. Section III-B shows all three rounds of the study process. First, we screened the papers by reviewing the titles, abstracts, and keywords and removing duplications. Then, we skimmed through the introduction and conclusion. Finally, we read the full text for the remaining papers and finalized the selection. The inclusion and exclusion criteria were applied (see Table 1) on all three rounds. The first two authors extracted the papers' information from the search results into an Excel file shared with the other authors for revision. The two authors confirmed each other's decision in excluding papers to mitigate personal bias during the three rounds. If the two authors disagreed, the other authors were invited to read the conflict papers to make a final decision. Other authors also confirmed the papers who are agreed by the two authors initially. We recorded our results from each round, including initial search results, notes for the selection process, accepted papers from each round, and the data annotation to maintain a clear chain of evidence for our findings.

1) BASIC INFORMATION READING – SCREENING (FIRST ROUND)

We defined the search string to be "HIPAA" and retrieved the initial set of papers from ACM (n = 718) and IEEE (n = 163) to have a total number of papers (n = 881). We removed all duplication, magazines, and books, resulting in (n = 794) papers. In this round, we excluded papers that are not written in English, previous versions of extended papers, and secondary studies, such as literature reviews. We excluded all the papers that do not explicitly cover or mention "HIPAA". In addition, we aimed to include papers that are relevant to software engineering, and we checked all the papers (n = 794) based on the title, abstract, and keywords. We removed all the papers unrelated to software engineering, such as those that only focused on HIPAA regulation. For example, we excluded the "security standards for electronic health records" paper, which apparently focused on regulations and standards without including software engineering topics. Note that the authors passed papers that might be related to software engineering to the next round because it was not possible to apply all the inclusion and exclusion criteria simply based on the title and abstract. As shown in Section III-B, we had (n = 168) accepted papers after the first round. Fig. 1 illustrates the overall screening process.

2) SKIMMING (SECOND ROUND)

In this round, we skimmed through the abstract, introduction, keywords, and conclusion of each paper, applying the inclusion and exclusion criteria. We removed all the papers that are not relevant to software engineering, such as papers focusing on networking, and communication engineering. For example, the "security vulnerabilities, attacks, countermeasures, and regulations of networked medical devices" paper was excluded as it is related to networking engineering.

3) ELIGIBILITY-INTENSIVE READING (THIRD ROUND)

In the final round, we carefully reviewed the remaining papers (n = 67). We read the entire paper and applied all the exclusion criteria to each paper. We decided on each paper and finalized our data set with all the authors. Finally, during this round, we also applied the snowballing search. The snowballing search strategy involves reviewing the remaining papers' references to gather any additional papers that are relevant to the literature review topic and were skipped by the search string. Accordingly, we further examined the references of the finalized papers from the third round to identify any additional papers to be added to our data set that might not have been captured previously at the first stage by our string term. We did not find any additional papers in snowballing search that passed our inclusion and exclusion criteria. All the papers found during the Snowballing search were already included in our initial search, published before Jan. 2012, or unrelated to HIPAA and software engineering. We finalized the final number of papers as 22 to be included in this scoping review, which satisfies all the inclusion criteria.

C. DATA EXTRACTION AND ANALYSIS

The first two authors created an Excel document for data extraction and shared it with the other authors for revision and confirmation. The Excel document contains all the data extracted from the three rounds and the data annotation. We recorded each paper's publication title, authors, library name, and year. The Excel sheet included a comment section to justify each paper's exclusion based on the predefined inclusion and exclusion criteria. We extracted the data to group and categorize similar papers besides analyzing general data such as the evolution of the publication over the years. We included data related to the software engineering topics, techniques, and HIPAA content in the final papers set. Our goal is to identify a common pattern between papers on software engineering topics and the techniques used to comply with HIPAA.

TABLE 2. Final Set of Papers

Index	Software Engineering phase
[P1], [P2], [P3], [P4], [P5]	Requirements
[P1], [P6], [P7], [P8], [P9], [P10], [P11], [P12], [P13]	Design
[P14], [P15], [P16], [P17], [P18], [P19], [P20], [P21]	Testing
[P22]	Implementation

After finalizing the papers, we annotated information related to software engineering topics and HIPAA from each paper. We started our annotation by reading all the papers carefully and assigning tags to each paper. After finishing the tagging process, we matched all the papers' tags to create the final categories set. We created high-level categorization of the papers based on their primary objectives. Each paper in our final set fell into one or more categories. The high-level categories were related to the software engineering life cycle and HIPAA content.

IV. RESULTS

We identified total of 22 papers after thorough application of inclusion and conclusion criteria, which was published between 2012–2022. The number of publications per year varies from zero to four with maximum publications in the year of 2017. We primarily classified the themes related to software engineering techniques used in reported HIPAA studies. Secondarily, we also reported details on how papers covered HIPAA in detail.

A. SOFTWARE ENGINEERING THEMES

We initially classified the papers based on their focus on software engineering phases, including requirements, design, testing, and implementation. Table 2 illustrates the classification of the final papers across four software phases. One paper may fall into more than one software engineering phase. The design phase has the highest number of papers (n = 8), while there is minimum focus on the implementation phase with only one paper.

We also developed a master table detailing the key contribution of each paper's focus on different software development phases including the purpose of the studies and recommendations from the studies (see Table 3).

1) REQUIREMENTS PHASE

The requirement phase is the starting point in defining a system's functional and nonfunctional requirements describing users' needs. In the first study, the requirements for a cloud computing system "MedCloud" were proposed based on the need for storing electronic medical records (EMRs) in one place to overcome the delay of transferring EMRs back and forth between different healthcare providers [P1]. This will help the developers create different healthcare applications that share the same data [P1]. Besides, it saves time gathering patient data from different sources [P1]. The "MedCloud" requirement specifications are used to comply with HIPAA, such as choosing the access level for users in a medical institution from the already defined list of access categories [P1]. Also, permitting only authorized users to access electronic patient health information in the medical information system, and defining security data transmission as a requirement [P1]. Modeling regulations provide significant benefits in the understanding of regulatory requirements. The second study [P2] focuses on requirements modeling, a technique for creating graphical representations of system behavior to document requirements and improve system understandability. Understanding the policies and regulations is necessary to write specifications that comply with them. The requirements engineers encounter challenges due to the complexity of regulations [P2]. Representing the compliance requirements assist in decision-making and verifying that the system operates according to the domain regulations [P2]. From this standpoint, a visual business policy modeling language called "CoReL" represents compliance requirements, allowing enterprises to manage decision-making related to regulatory compliance flexibly [P2]. In the [P2] paper, HIPAA was used as a case study, specifically, modeling a HIPAA regulation paragraph that addresses "what information on individuals' covered entities (health insurance entities) may be disclosed and under which condition". There are challenges in modeling this regulation such as difficulties in gathering the needed data, and regulation refers to exceptions to the rules [P2]. Most of CoReL's useful features cannot be easily applied to this regulation without further refinement, which requires further research in future work on CoReL to overcome these challenges [P2].

It can be difficult to determine whether requirements comply with regulations due to their complexity. Involving regulation experts in the requirements phase can assist in confirming regulation compliance [P3]. In the third study, the Wideband Delphi method is used to make decisions by reaching a consensus among a group of software engineers. However, Wideband Delphi alone does not significantly improve decision-making for legal requirements [P3]. It could be used with supportive tools and methods to assist engineers in requirement compliance decision-making [P3].

When software engineers attempt to extract requirements from regulations, they often encounter issues related to ambiguity, redundancy, and traceability maintenance [P4]. Muyideen Mustapha et al. [P4] presented a systematic approach and algorithms for requirement analysts to acquire, extract, and analyze requirements. Regulation changes require changing the software requirements to maintain legal compliance, which might impact the system's stability [P5]. Maxwell et al. [P5] presented a framework to assist requirements engineers in predicting regulation rules that are likely to change, which helps the software engineers work towards more stable areas of regulation. The framework was developed through a formative case study on the HIPAA Security Rule and applied in a summative study on the EHR Certification Rule, correctly predicting 75 percent of the changes [P5].

TABLE 3. Key Takeaways: Software Engineering Theme

Software Engineering Phase	Citation	Purpose of the study	Recommendations
Requirements and Design	[P1]	Purposing a cloud computing system for storing Electronic Medical Records (EMRs) complies with HIPAA privacy and security rules.	Defining authorized users and access levels to support patient privacy and disclosure the minimum patient information to comply with HIPAA.
Requirements	[P2]	CoReL is a visual business policy modelling language. This paper studies the modeling support that CoReL provides to HIPAA.	Adopting visual business policy modeling language to represent HIPAA requirements. This strategy assists in dealing with the complexity of requirements.
	[P3]	Examining how graduate-level software engineering students assess whether the software requirements met or exceeded the HIPAA obligations (case study)	Involving experts in HIPAA obligations is a necessity for formulating software requirements that align with HIPAA.
	[P4]	Presenting a systematic approach and algorithms for requirement analysts to acquire, extract, and analyze requirements. The approach was applied to three HIPAA requirements.	Using a systematic approach to prioritize regulatory compliance requirements related to HIPAA while addressing software requirements. This is essential to overcome issues of redundancy and ambiguity in software development early stages.
	[P5]	Purposing a framework that can assist software engineers in predicting what areas of proposed rules are most likely to evolve, allowing engineers to build towards the more stable sections of HIPAA.	Using a framework to help requirements engineers predict potential changes in regulatory rules such as HIPAA. This strategy can result in time and cost savings for organizations.
Design Phase	[P6]	Proposing a HIPAA-compliant privacy access control model for web services.	Designing the HIPAA-compliant privacy access control model with a modular structure, promoting reusability of components to easily adapt and integrate with various web services.
	[P7]	Proposing data exchange platform to improve and support the communication between healthcare information systems. This system also considers security and privacy under the standard of HIPAA.	Leveraging cloud computing technology for data exchange since it has the advantage of reliability, scalability and cost effective.
	[P8]	Proposing an architecture for secured medical information exchange protocols and maintaining the technical safeguards of the HIPAA security rule.	Incorporating an audit system into the architecture to maintain records of all incoming and outgoing exchange requests and support technical safeguards of the HIPAA security rule. The audit system maintains records for every healthcare provider to ensure effective monitoring and transparency within the healthcare system.
	[P9]	Proposing a healthcare framework named "ChainSDI" to render high performance and real-time responsiveness for home-based healthcare services. It was built to be compliant with regulations such as HIPAA while still providing high data interoperability.	Using blockchain technology to support the specification of HIPAA regulation-compliant data sharing/processing requirements. It assists in solving issues related to data interoperability and security, such as enabling effective authorized interactions between patients and medical applications and delivering patient data securely to a variety of organizations and devices.
	[P10]	Using an existing ISO QMS protocol to formulate a framework for HIPAA compliance protocol. The HIPAA Security Rule requirements were mapped against ISO 9001 requirements to identify similarities to achieve HIPAA compliance and examine the possibility if the existing protocol could be tailored to achieve HIPAA compliance.	Overall HIPAA, there is a good match in a few cases when some ISO 9001 guideline clauses have been mapped to the HIPAA as a whole entity and vice versa.
	[P11]	Proposing a secure end-to-end data transmission mechanism and an advanced access control scheme to be compliant with the HIPAA technical safeguard.	Design the system to contain public and private domains.
	[P12]	Proposing a cloud architecture to help clinicians and researchers to have accessibility to data sets from multiple sources, while adhering to security standards such as HIPAA.	Using cloud architecture to facilitate access to data sets from multiple sources with consideration of HIPAA security standards.
	[P13]	Developing cloud-based platform for the storage and analysis of large amount of Personal Health Information while adhering to security standards such as HIPAA.	Using a secured gateway that has the implementation of appropriate security controls to ensure alignment with HIPAA and guarantee secured transfer of data.
Testing Phase	[P14]	Developing an abbreviated HIPAA test suite and applied it to three open-source electronic health record systems.	Checking and evaluating the software system in a repeatable and traceable manner using Behavior- Driven-Development (BDD) testing practice.
	[P15]	Evaluating modeling and analyzing a subset of HIPAA rules using sequence diagrams.	Sequence diagrams can be utilized to assist in verifying the system design of an organization in the domain of HIPAA.
	[P16]	Proposing a method based on HIPAA privacy and security rule to analyze mHealth applications.	Using testing tools and methods such as privacy policy analysis, static analysis, dynamic analysis and HTTP analysis to ensure HIPAA compliance and detect security vulnerabilities such as SQL Injection.
	[P17]	Proposing a method based on HIPAA privacy and security rule to analyze mHealth applications.	Developers should avoid incorporating user input in raw SQL queries to prevent SQL injection vulnerabilities while developing mHealth apps using SQLite database.
	[P18]	Evaluating an open source EHR application for security vulnerabilities and mapping identified vulnerabilities to HIPAA technical requirements.	Using a code analysis tool (RIPS) to detect vulnerabilities that are impacting HIPAA technical requirements.
	[P19]	Identifying security and privacy requirements for HIPAA technical requirements and then evaluate EHR applications for security vulnerabilities.	Ensuring that all user inputs, especially those received through $\$\_GET$ or $\$\_POST$ parameters, undergo strict validation to reject any input that does not adhere to an expected format or content.
	[P20]	Analyzing HIPAA privacy and security rules for 24 mobile personal health records.	Prioritize the implementation of user-friendly and secured authentication mechanisms, such as biometric techniques, to address user concerns about privacy while ensuring an efficient user experience.
	[P21]	Developing a framework to assist in creating and verifying prototypical Electronic Medical Record System following HIPAA rules.	The system should enable easy authorization revocation, and prevent release of records post-revocation.
Implementation Phase	[P22]	Reviewing the security capabilities of MongoDB and singularity container focusing on four main HIPAA data security requirements.	MongoDB and Singularity security features integrate to build a more secure framework to analyze data securely than through using their individual methods and techniques for authentication, authorization, encryption, and auditing.

2) DESIGN PHASE

Software architectures focus on providing high-level design solutions, such as architectural and access-control models, to achieve HIPPA compliance. Sobhy et al. [P1] and Alshugran and Dichter [P6] proposed models to ensure data privacy. Cloud-oriented approach to medical system "Med-Cloud" architecture was built to focus on scalability and privacy quality attributes by deploying a adoop cluster and designing the system based on HIPAA requirements [P1]. An access control model can be implemented in software architecture to achieve information security and privacy. The access control model permits the authenticated user to operate on specific data based on the user's permission [P6]. Jessadapattharakul et al. [P7] and Ibrahim and Singhal [P8] presented an architecture for medical data exchange. Interoperability refers to the system's ability to share patient information electronically among different healthcare systems. There are difficulties in sharing data between different institutions [P7]. Accordingly, Jessadapattharakul et al. [P7] proposed a data exchange framework architecture for healthcare services designed to use a cloud-based service platform with activity diagrams demonstrating data exchange scenarios. There are challenges in providing data security during data collection, transmission, and sharing [P9].

Exchanging data among different healthcare providers raises issues in integration, security, and privacy [P8]. Ibrahim and Singhal [P8] proposed an architecture that assists in maintaining audibility and supporting technical safeguards of the HIPAA security rule. The audit system is used to maintain information about each transaction by keeping records of all incoming and outgoing exchange requests for each healthcare provider [P8]. HIPAA regulation can be integrated into quality audit protocol based on the ISO 9001 standard by cross-mapping HIPAA requirements with ISO requirements to identify similarities [P10]. Uzma et al. [P10] indicated that healthcare organizations can still use ISO 9001 guidelines and processes to achieve HIPAA compliance. Uzma et al. [P10] demonstrated that customer satisfaction has increased and one of the reasons for improved satisfaction is that HIPAA requirements were mapped against ISO 9001 standards.

Li et al. [P9] and [P11] presented regulation compliance in home-based healthcare services. Home-based healthcare is a health-care service that is provided to patients in their homes. There are challenges in the design of home-based healthcare services running on software-defined infrastructure, including data sharing challenges. Li et al. [P11] presented the ”CareNet” framework, which consists of a set of APIs and secures data transmission mechanisms for healthcare services running on software-defined infrastructure (SDI). Li et al. [P9] proposed a framework that supports the specification of HIPAA regulation-compliant called ”ChainSDI”. The framework handles issues related to health data interoperability and security, such as authorized interactions between patients and medical applications and sharing data securely, besides improving the overall efficiency of medical applications [P9]. The framework uses blockchain techniques to manage secure data sharing and computing sensitive patient data [P9].

Valluripally et al. [P12] and Dean et al. [P13] centered around analyzing large amounts of health data, commonly referred to as health big data analytics. Implementing big data analytics in healthcare applications is challenging due to the sensitivity of healthcare data [P12]. Accordingly, Valluripally et al. [P12] proposed a cloud architecture to help clinicians and researchers in accessing data sets from multiple sources aligning with security standards such as HIPAA. A large amount of healthcare data requires protecting health information confidentiality's ability, integrity, and privacy under HIPAA-regulated environment [P13]. Software engineers might encounter challenges that are related to Cloud-based analytics such as data isolation while processing protected health information (PHI) due to HIPAA regulations [P13]. Accordingly, Dean et al. [P13] developed Watson Health Cloud, which is a cloud-based platform used for storing and analyzing a large amount of PHI, to overcome these challenges.

3) TESTING PHASE

Compliance testing evaluates the software in accordance with legal requirements using various testing tools and methods. The behavior-driven-development (BDD) is a test suit that could be built to support the checking of regulatory requirements [P14]. The system can be evaluated using this automated test in a repeatable and traceable approach [P14]. In addition, sequence diagrams can be used to verify the system process interactions. The sequence diagram is a unified modeling language to graphically express privacy policies [P15]. Sequence diagrams allow decision-makers such as security architects to specify the system design easily and verify the expected behavior [P15].

Security vulnerabilities and compliance with privacy polities should be tested in EHR and Mhealth applications. Zhao et al. [P16] and Migiro et al. [P17] used MobSF, a mobile security framework, to detect security vulnerabilities. Both papers show that Mhealth applications have high risks of PHI leakage [P16], [P17]. Zhao et al. [P16] presented testing methods such as privacy policy analysis, static analysis, dynamic analysis, and HTTP analysis that could be used to detect vulnerabilities and analyze Mhealth applications based on HIPAA. The analysis indicates that some Mhealth applications are vulnerable to SQL Injection [P16]. Also, Migiro et al. [P17] showed that some contact tracing applications have poor security features, such as allowing dangerous permissions, sharing of data, and storing it within third-party entities. Farhadi et al. [P18] and [P19] evaluated an open-source EHR application for security vulnerabilities using an open-source scanner tool (RIPS), and map identified vulnerabilities to HIPAA technical requirements. In addition, Farhadi et al. [P19] used an analysis tool called "PHP VulnHunter". It is a static analysis tool that scans PHP vulnerabilities automatically to evaluate EHR applications for security vulnerabilities.

TABLE 4. HIPAA Rules

Index	HIPAA specific rule number	Target
[P1]	Overall HIPAA	Privacy and security rules
[P2]	164.512(f)	Privacy rules
[P3]	164.312	Technical safeguards
[P4]	Overall HIPAA	Security rules
[P5]	164.308(a)(7)(i), 164.310(c),164.308(a)(1), 164.306(b),164.308(a)(3)(ii)(A)	Security rules
[P6]	Overall HIPAA	Privacy rules
[P7]	Overall HIPAA	Privacy and security rules
[P8]	Overall HIPAA	Security rules
[P9]	Overall HIPAA	Security rules
[P10]	Overall HIPAA	Security rules
[P11]	164.312(a), 164.312(c),164.312(d), 164.312(e)	Technical safeguards
[P12]	Overall HIPAA	Security rules
[P13]	Overall HIPAA	Security rules
[P14]	170.302(o), 170.302(p), 170.302(q),170.302(r), 170.302(s), 170.302(t),170.302(u)	Security rules
[P15]	164.512(j)(2), 164.524(b)(2)(i) - (ii),164.508(c)(2)(i), 164.508(b)(2)(i),164.508(a)(3)	Privacy rules
[P16]	Overall HIPAA	Privacy and security rules
[P17]	Overall HIPAA	Privacy rules
[P18]	164.312(a)(1), 164.312(a)(2)(i),164.312(a)(2)(ii), 164.312(a)(2)(iii), 164.312(a)(2)(iv), 164.312(b), 164.312(c)(1), 164.312(c)(2),164.312(d), 164.312(e)(1),164.312(e)(2)(i), 164.312(e)(2)(ii)	Security rules
[P19]	164.312(a)(1), 164.312(a)(2),164.312(b), 164.312(c)(1),164.312(c)(2), 164.312(d),164.312(e)	Security and privacy rules
[P20]	Overall HIPAA	Privacy rules
[P21]	Overall HIPAA	Privacy rules
[P22]	Overall HIPAA	Security rules

Farhadi et al. [P18], [P19], and Cruz Zapata et al. [P20] are evaluating security and privacy requirements in healthcare technology applications. Farhadi et al. [P19] showed that there is a gap between HIPAA technical requirements and traditional security vulnerabilities. Cruz Zapata et al. [P20] presented a questionnaire containing six questions in order to analyze the privacy policy of 24 mobile personal health records (mPHRs) for Android and iOS systems. Those questions can be used by software developers when assessing the privacy of their future mPHRs [P20]. The findings show that no mPHR scores more than 3.5 points out of a maximum of 6 [P20]. Cruz Zapata et al. [P20] suggested complying with a healthcare privacy law such as HIPAA to improve mPHRs. Also, Farhadi et al. [P18] demonstrated that the EHR application does not comply with some HIPAA security requirements, meaning there is a gap between traditional security vulnerabilities and HIPAA security requirements. Based on the results, Farhadi et al. [P18] recommend storing patients' information in encrypted form and fixing the vulnerabilities in EHR applications as soon as possible once they are discovered. Johnson et al. [P21] proposed a framework to create and verify a prototypical electronic medical record system. Johnson et al. [P21] developed the design of the history aware programming language into the framework for creating systems that can be automatically checked against privacy specifications. This will assist in verifying and enforcing the HIPAA privacy policy [P21].

4) IMPLEMENTATION PHASE

Choosing the proper development tools is essential to create healthcare software that complies with the requirements. Mailewa Dissanayaka et al. [P22] introduced EXPOSOME Project, which includes sensitive healthcare data. The data privacy in the EXPO- SOME project is located in the database, the network infrastructure, web applications, and physical security. Mailewa Dissanayaka et al. [P22] examined the security capabilities of free and open-source MongoDB community edition and Singularity within the EXPOSOME Project. MongoDB is a NoSQL-based document-oriented database [P22]. Singularity is a Linux container that has strong security features and the ability to provide users full autonomy over working environments to package scientific workflows, software, libraries, and data [P22]. The project focuses on four HIPAA requirements: data security, authorization, encryption, and auditing [P22]. MongoDB provides security through authentication, authorization, encryption mechanisms, and Singularity provides a secure environment to analyze data when applications are executed [P22].

B. THEMES ON HIPAA RULES

We have several papers addressing and covering HIPAA as a whole (n = 14), while some other papers just focused on specific rules in HIPAA (n = 8). For instance, Kharbili et al. [P2] addressed 164.512(f) HIPAA rule, which is one of the HIPAA rules focusing on limiting data disclosure by defining the data and disclosure conditions [P2]. Disclosure is defined as revealing and providing permission to access healthcare data outside the entity holding the information [7].

Shen et al. [P15] discussed multiple rules in HIPAA. The 164.512(j)(2) defines a safety property by expressing forbidden behaviors [P15]. Section 164.524(b)(2)(i)-(ii) expresses the liveness property (time session) and exception cases [P15]. Section 164.508(b)(2)(i) describes the invalidity of authorizations after the expiration date [P15]. Both 164.508(b)(2)(i) and 164.508(a)(3) are related to authorization [P15]. Authorization refers to giving a user permission to access or perform a process [P15].

Massey et al. [P3]], Li et al. [P11], and Farhadi et al. [P19 focused on the 164.312 rule. Section 164.312 governs technical safeguards, which primarily focus on technical measures for protecting health-care information [P3]. Maxwell et al. [P5] focused on multiple rules, such as 164.306, 164.308, and 164.310. The 164.310 rule restricts access to authorized users [P5]. Both 164.306 and 164.308 rules state that risk assessment must be used to ensure the confidentiality, integrity, and availability of electronic patient health information [P5]. Additionally, both 164.306 and 164.308 rules provide security measures that should protect against anticipated threats [P5]. Papers that address overall HIPAA, such as [P1], [P4], and [P9], focus on the privacy and/or security quality attributes. Table 4 presents the 22 papers and their targeted HIPAA rules in which they applied software engineering methods.

V. CONCLUSION

This article presents the applications of software engineering methods in the domain of HIPAA application. Enabling HIPAA compliance from a software engineering perspective is a growing area. The review analyzed 22 papers between 2012 and 2022 after three selection rounds based on the exclusion and inclusion criteria and reported what type of software engineering methods have been used so far to address HIPAA compliance or specific HIPAA rules during the software development life cycle. The majority of papers are related to the design and testing phases from software engineering life cycle. Privacy and security are the targeted quality attributes in the HIPAA-compliance software design phase. The findings emphasize the necessity of using code analysis tools such as the RIPS tool during testing to detect the gap between HIPAA technical requirements and security vulnerabilities. The security vulnerabilities include SQL injection, allowing risky permissions, sharing of data, and storing it within third-party entities. The examined papers reveal significant privacy and security concerns in mHealth applications, particularly vulnerability to issues like SQL Injection. Fewer papers were classified under the requirement and implementation phases. The requirements extraction process is challenging due to the ambiguity and complexity of specific rules. The implementation phase focuses on understanding the capabilities and limitations of each component in health-care software to ensure HIPAA compliance. In addition, we found a gap in addressing the topics related to the abstract view of software engineering approaches to achieve HIPAA compliance, such as traceability. Traceability is tracking and verifying software artifacts through the Software Development Life Cycle. Finally, the listed recommendations in Table 3 presents insightful considerations emerged from the papers in this area and should be used as a guide for future studies.

We limited our review to two main software engineering databases due to the nature of the scoping review. Future studies should run more comprehensive systematic literature reviews to better frame some guidance in this area. Finally, the software engineering management topics should investigate the potential drawbacks or advantages of using Agile frameworks over Waterfall methodology in achieving HIPAA compliance in future studies.

V. LITERATURE DATASET

[P1] D. Sobhy, Y. El-Sonbaty, and M. Abou Elnasr, “Medcloud: Healthcare cloud computing system,” in *Proc. Int. Conf. Internet Technol. Secured Trans.*, 2012, pp. 161–166.

[P2] M. El Kharbili, “Applying corel to an excerpt of HIPAA: A critical discussion,” in *Proc. 6th Int. Workshop Requirements Eng. Law*, 2013, pp. 61–64.

[P3] A. K. Massey, P. N. Otto, and A. I. Antón, “Evaluating legal implementation readiness decision-making,” *IEEE Trans. Softw. Eng.*, vol. 41, no. 6, pp. 545–564, Jun. 2015.

[P4] A. Muyideen Mustapha, O. ’Tale Arogundade, A. Abayomi-Alli, O. John Adeniran, K. Adesemowo, and C. Yetunde Alonge, “A systematic method for extracting and analyzing cloud-based compliance requirements,” in *Proc. Int. Conf. Math., Comput. Eng. Comput. Sci.*, 2020, pp. 1–7.

[P5] J. C. Maxwell, A. I. Anton, and P. Swire, “Managing changing compliance requirements by predicting regulatory evolution,” in *Proc. IEEE 20th Int. Requirements Eng. Conf.*, 2012, pp. 101–110.

[P6] T. Alshugran and J. Dichter, “Toward a privacy preserving HIPAA- compliant access control model for web services,” in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, 2014, pp. 163–167.

[P7] R. Jessadapattharakul, S. Prom-on, C. Tanprasert, and T. Achalakul, “Data exchange protocol for healthcare service in Thailand,” in *Proc. 4th Int. Conf. Future Gener. Communication Technol.*, 2015, pp. 1–6.

[P8] A. Ibrahim and M. Singhal, “An abstract architecture design for medical information exchange,” in *Proc. Int. Conf. Ind. Inform. Comp. Syst.*, 2016, pp. 1–6.

[P9] P. Li et al., “Chainsdi: A software-defined infrastructure for regulation-compliant home-based healthcare services secured by blockchains,” *IEEE Syst. J.*, vol. 14, no. 2, pp. 2042–2053, Jun. 2020.

[P10] S. Uzma Gardazi, A. Ali Shahid, and C. Salimbene, “HIPAA and QMS based architectural requirements to cope with the OCR audit program,” in *Proc. 3rd FTRA Int. Conf. Mobile, Ubiquitous, Intell. Comput.*, 2012, pp. 246–253.

[P11] P. Li, C. Xu, Y. Luo, Y. Cao, J. Mathew, and Y. Ma, “Carenet: Building regulation-compliant home-based healthcare services with software-defined infrastructure,” in *Proc. IEEE/ACM Int. Conf. Connected Health: Appl., Syst., Eng. Technol.*, 2017, pp. 373–382.

[P12] S. Valluripally et al., “Community cloud architecture to improve use accessibility with security compliance in health Big Data applications,” in *Proc. 20th Int. Conf. Distrib. Comput. Netw.*, 2019, pp. 377–380.

[P13] D. J. Dean et al., “Engineering scalable, secure, multi-tenant cloud for healthcare data,” in *Proc. IEEE World Congr. Services*, 2017, pp. 21–29.

[P14] P. Morrison, C. Holmgreen, A. Massey, and L. Williams, “Proposing regulatory-driven automated test suites for electronic health record systems,” in *Proc. 5th Int. Workshop Softw. Eng. Health Care*, 2013, pp. 46–49.

[P15] H. Shen, R. Krishnan, R. Slavin, and J. Niu, “Sequence diagram aided privacy policy specification,” *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 381–393, May/Jun. 2016.

[P16] W. Zhao, H. Shahriar, V. Clincy, and Z. Alam Bhuiyan, “Security and privacy analysis of mhealth application: A case study,” in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, 2020, pp. 1882–1887.

[P17] L. Migiro, H. Shahriar, and S. Sneha, “Analyzing security and privacy concerns of contact tracing applications,” in *Proc. IEEE Int. Conf. Digit. Health*, 2021, pp. 283–292.

[P18] M. Farhadi, H. Haddad, and H. Shahriar, “Static analysis of hippa security requirements in electronic health record applications,” in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf.*, 2018, vol. 2, pp. 474–479.

[P19] M. Farhadi, H. Haddad, and H. Shahriar, “Compliance checking of open source EHR applications for HIPAA and onc security and privacy requirements,” in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf.*, 2019, vol. 1, pp. 704–713.

[P20] B. Cruz Zapata, A. Hernández Niñirola, J. Luis Fernández- Alemán, and A. Toval, “Assessing the privacy policies in mobile personal health records,” in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014, pp. 4956–4959.

[P21] C. Johnson, T. MacGahan, J. Heaps, K. Baldor, J. von Ronne, and J. Niu, “Verifiable assume-guarantee privacy specifications for actor component architectures,” in *Proc. 22nd ACM on Symp. Access Control Models Technol.*, 2017, pp. 167–178.

[P22] A. Mailewa Dissanayaka, R. Ramprasad Shetty, S. Kothari, S. Mengel, L. Gittner, and R. Vadapalli, “A review of mongodb and singularity container security in regards to HIPAA regulations,” in *Proc. 10th Int. Conf. Utility Cloud Comput.*, 2017, pp. 91–97.

REFERENCES

[1] U.S. department of health and human services. Health insurance portability and accountability act(HIPAA).

[2] T. Alshugran, J. Dichter, and M. Faezipour, “Formally expressing HIPAA privacy policies for web services,” in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, 2015, pp. 295–299.

[3] HIPAA Compliance Assistance, “Summary of the HIPAA privacy rule,” Office Civil Rights, 2003.

[4] P. Berander et al., “Software quality attributes and trade-offs,” *Blekinge Inst. Technol.*, vol. 97, no. 98, 2005, Art. no. 19.

[5] D. S. Debono et al., “Nurses’ workarounds in acute healthcare settings: A scoping review,” *BMC Health Serv. Res.*, vol. 13, pp. 1–16, 2013.

[6] B. A. Fiedler, “Challenges of new technology: Securing medical devices and their software for hippa compliance,” in *Managing Medical Devices Within a Regulatory Framework*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 315–329.

[7] J. C. Garner, “Final HIPAA security regulations: A review,” *Managed Care Quart.*, vol. 11, no. 3, pp. 15–27, Summer 2003.

[8] R. Hasan and W. Yurcik, “A statistical analysis of disclosed storage security breaches,” in *Proc. 2nd ACM workshop Storage Secur. Survivability*, 2006, pp. 1–8.

[9] A. Hasselgren, K. Kralevska, D. Gligoroski, S. A. Peder- sen, and A. Faxvaag, “Blockchain in healthcare and health sciences—A scoping review,” *Int. J. Med. Inform.*, vol. 134, 2020, Art. no. 104040.

[10] US HHS, “Summary of the HIPAA security rule,” Office for Civil Rights, 2003.

[11] S. Hoffman and A. Podgurski, “Securing the HIPAA security rule,” *J. Internet Law*, pp. 6–26, Spring 2007.

[12] M. E. Johnson and E. Goetz, “Embedding information security into the organization,” *IEEE Secur. Privacy*, vol. 5, no. 3, pp. 16–24, May/Jun. 2007.

[13] N. L. Martin, T. Imboden, and D. T. Green, “HIPAA security rule compliance in small healthcare facilities: A theoretical framework,” *Issues Inf. Syst.*, vol. 16, no. 1, pp. 180–188, 2015.

[14] A. K. Massey, P. N. Otto, and A. I. Antón, “Evaluating legal implementation readiness decision-making,” *IEEE Trans. Softw. Eng.*, vol. 41, no. 6, pp. 545–564, Jun. 2015.

[15] M. R. Mia et al., “A comparative study on HIPAA technical safeguards assessment of android mhealth applications,” *Smart Health*, vol. 26, 2022, Art. no. 100349.

[16] W. Moore and S. Frye, “Review of HIPAA, part 1: History, protected health information, and privacy and security rules,” *J. Nucl. Med. Technol.*, vol. 47, no. 4, pp. 269–272, 2019.

[17] N. M. A. Munassar and A. Govardhan, “A comparison between five models of software engineering,” *Int. J. Comput. Sci. Issues*, vol. 7, no. 5, 2010, Art. no. 94.

[18] A. Raman, “Enforcing privacy through security in remote patient monitoring ecosystems,” in *Proc. 6th Int. Special Topic Conf. Inf. Technol. Appl. Biomed.*, 2007, pp. 298–301.

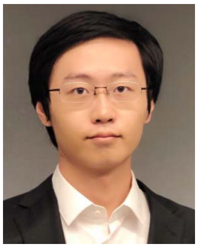
[19] G. M. Stevens, “Enforcement of the HIPAA privacy and security rules. Library of Congress,” *Congressional Res. Service*, 2008. [Online]. Available: http://assets.opencrs.com/rpts/RL33989_20080811.pdf

[20] R. Streeton, M. Cooke, and J. Campbell, “Researching the researchers: Using a snowballing technique,” *Nurse Researcher*, vol. 12, no. 1, pp. 35–47, 2004.

[21] A. C. Tricco et al., “PRISMA extension for scoping reviews (PRISMA-ScR): Checklist and explanation,” *Ann. Intern. Med.*, vol. 169, no. 7, pp. 467–473, Oct. 2018.

FARAH ELKOURDI received the B.Sc. degree in computer engineering from Princess Sumaya University for Technology, Jordan, in 2018, and the M.Sc. degree in software engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2022. She is currently working toward the Ph.D. degree in systems engineering with Stevens Institute of Technology, Hoboken, NJ, USA (*advised by Dr. O. Asan*). Farah’s research focuses on utilziing health information technologies to improve care for complex pediatrics patients.

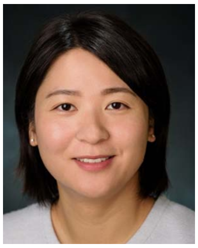
Her research interests include software engineering, human-computer interaction, and digital health.



CHENHAO WEI received the M.Sc. degree in information and data engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2019, where he is currently working toward the Ph.D. degree in system engineering with the School of Systems and Enterprises (*advised by Dr. L. Xiao*).

His research interests include software testing and artificial intelligence for software engineering.

Mr. Wei was the recipient of the third-place prize at the ASE Student Research Competition in 2022.



LU XIAO received the bachelor's degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009 and the Ph.D. degree from Drexel University, Philadelphia, PA, USA, in 2016 (*advised by Dr. Y. Cai*), both in computer science.

She is an Assistant Professor with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA. She has authored and coauthored her work in different conferences and journals, including TSE, ICSE, FSE, and ICSA. Her research interests include the broad area of software engineering, particularly in software architecture, software economics, cost estimation, and software ecosystems.

Prof. Xiao is an awardee of NSF CAREER project in 2021. She was the recipient of the first-place prize at the ACM Student Research Competition in 2015.

ZHONGYUAN YU received the M.S. degree in operations research and in industrial engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2010 and 2012, respectively, and the Ph.D. degree in system engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2014.

She is a Teaching Associate Professor and Software Engineering Program Director with the School of Systems and Enterprises, Stevens Institute of Technology. She builds smart and interactive data-driven decision support systems and has extensive software developing expertise. Her research interests include multiple disciplines including applied statistics, data visualization, simulation, natural language processing, and network analysis.

ONUR ASAN received the bachelor degree from Industrial Engineering of Marmara University in Istanbul, Turkiye, in 2008. He received the master degree from Industrial and Systems Engineering of UW-Madison, in 2010, and Ph.D. degree from UW-Madison, in 2013.

He is an Associate Professor with the School of Systems and Enterprises, Stevens Institute of Technology. His research interests include on application of theory, methods, and design from the discipline of human factors engineering and human computer interaction to improve socio-technical change in health care.