Master’s thesis

Master’s Programme in Computer Science

DevOps in Medical Device Software Development: A Multivocal Literature Review

Niko Häppölä

May 24, 2024

Faculty of Science

University of Helsinki

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Study programme Master's Programme in Computer Science

Tekijä — Författare — Author

Niko Häppölä

Työn nimi — Arbetets titel — Title

DevOps in Medical Device Software Development: A Multivocal Literature Review

Ohjaajat — Handledare — Supervisors

D. Sc. Mikko Raatikainen

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year May 24, 2024	Sivumäärä — Sidoantal — Number of pages 53 pages, 16 appendix pages

Tiivistelmä — Referat — Abstract

Introduction: EU medical device regulation (MDR) sets requirements for medical device software (MDSW) development. Following international standards, such as IEC 62304 and IEC 82304-1, is considered best practice to ensure compliance with regulation. At first glance, MDR and standards seem counter-intuitive to the DevOps approach. DevOps has been successful in regular software development, and it could improve MDSW development. In addition, standalone software is more prevalent as a medical device and as software does not need to be embedded into a physical device, the DevOps approach should be more feasible.

Methods: In this thesis, a systematic approach of multivocal literature review was conducted. The goal is to find the state-of-the-art of DevOps in MDSW development, what DevOps techniques and practices are suggested by academic literature and industry experiences, and what the challenges and benefits of DevOps are in MDSW. 18 scientific articles and 10 sources of gray literature were analyzed.

Results: The DevOps benefits of improved quality and faster release cycle can be achieved up to a certain point. Regulations prevent Continuous Deployment, but Continuous Integration (CI) and Continuous Delivery (CD) are possible. The most promising improvements can be made by automated documentation creation and bringing tasks of regulatory experts and developers closer together by streamlining the regulatory process. Existing DevOps tools can be extended to support compliance requirements. Third-party platforms and AI/ML solutions remain problematic due to regulations.

ACM Computing Classification System (CCS)
General and reference → Document types → Surveys and overviews
Software and its engineering → Software creation and management → Software development process management → Software development methods → Agile software development
Applied computing → Life and medical sciences → Health informatics

Avainsanat — Nyckelord — Keywords

DevOps, medical device software, medical device regulation, standard, compliance, RegOps

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software study track

Contents

1 Introduction **1**

1.1 Context . . . 1

1.2 Research goal and scope . . . 2

1.3 Contribution . . . 3

1.4 Thesis structure . . . 3

2 Background **4**

2.1 Regulatory domain . . . 4

2.2 International standards . . . 4

2.3 EU Medical Device Regulation . . . 7

2.4 DevOps . . . 10

3 Research methods **13**

3.1 Method selection . . . 13

3.2 Research questions . . . 13

3.3 Study protocol . . . 15

3.3.1 Search process . . . 15

3.3.2 Source selection criteria . . . 18

3.3.3 Quality assessment criteria . . . 19

3.3.4 Snowballing . . . 21

3.3.5 Data extraction . . . 22

3.3.6 Data synthesis . . . 22

4 Results **24**

4.1 Overview . . . 24

4.2 The state-of-the-art of DevOps in MDSW (RQ1) . . . 27

4.3 Suggestions on utilizing DevOps in MDSW (RQ2) . . . 29

4.3.1 Standards-based approach . . . 29

4.3.2 DevOps practices . . . 32

4.3.3 DevOps tools 33
4.3.4 Third party platforms 34
4.3.5 Organizational culture 34
4.3.6 MLOps 35
4.4 Challenges and benefits of DevOps in MDSW (RQ3) 36
4.4.1 Compliance 37
4.4.2 Software development process 38
4.4.3 Organizational culture 39
4.4.4 AI/ML 40

5 Discussion 41
5.1 The state-of-the-art of DevOps in MDSW (RQ1) 41
5.2 Implementing the DevOps within MDSW (RQ2) 42
5.3 Challenges and benefits of DevOps in MDSW (RQ3) 43
5.4 Step-by-step guidance on adopting DevOps in MDSW domain 44
5.5 Limitations and validity 45
5.5.1 Limitations of the source selection process 45
5.5.2 Limitations in data extraction 46

6 Conclusions 47

Bibliography 49

A Literature identified in the search process and their inclusion/exclusion criteria

B Literature reviewed in the quality criteria step

C Selected data sources

1 Introduction

The introduction of this thesis is structured to provide a context for the topic, define the research goal and scope, and present the contribution and thesis structure.

1.1 Context

Several industries, such as healthcare and aerospace, pose risks to human health. Therefore, software developed in these industrial fields must have high quality and trustworthiness, and be safe for users [34]. Governments and authorities regulate such industrial fields to ensure reliable, transparent, and traceable software projects. Regulation authorities set the requirements that manufacturers must fulfill to get their products to the market. International standards are established norms or requirements that can be seen as formulas that describe the best way to do something [27]. Relevant standards are usually harmonized with the regulations. Thus, complying with the standards is a safe way to ensure compliance with regulations.

A medical device is a product or equipment intended for medical purposes, introducing benefits and possible risks to user health [10]. Medical devices are heavily regulated in the healthcare industry. Medical devices include a wide range of equipment and items, ranging from spectacles to pacemakers. As technology has evolved, medical devices may contain embedded software, or even be standalone software. Regulations have been expanded to handle software as part of medical devices, identifying the concept of *Medical Device Software* (MDSW). Within MDSW, embedded software is a part of the physical device, usually controlling the device or acting as a functional part of the device, for example, calculating results. Standalone software, such as mobile phone applications, is not bound to a physical device but is used for medical purposes regardless of the platform. This thesis uses the term MDSW to describe both embedded and standalone software.

As software has a larger part in medical devices, and more standalone software is developed for medical purposes, attention should be paid to the software development models used in healthcare. Although regulations and standards do not specify any particular development model, a waterfall-oriented model is generally considered the easiest to follow. This is because detailed documentation is required to demonstrate the quality and safety of a product. Modern software practices, such as agile development, may seem difficult or counterintuitive to comply with. However, agile methods such as DevOps are generally considered beneficial for software development practitioners [5]. In addition, modern technologies, such as artificial intelligence and machine learning, are becoming more prevalent in MDSW [39].

DevOps is a software development methodology that aims to shorten the development cycle and improve software quality by releasing small improvements continuously [2]. DevOps relies on automated tools and pipelines, and collaboration between developers and operations. The DevOps approach may have potential benefits in the development of MDSW.

1.2 Research goal and scope

The primary objective of this research is to determine how DevOps can benefit regulated MDSW development, and what concrete actions manufacturers can take to implement DevOps. To understand this, it must be determined how the DevOps approach is currently utilized in medical device software development, what means are recommended by experts in the field, and what can be learned from established practices and practical experiences. In addition, it is imperative to understand the requirements of regulations, and the guidance offered by international standards.

To achieve this goal, a multivocal literature review was conducted to gather information from both formal and grey literature sources. Experiences from industry practitioners are important to look into, as practices evolve quickly, and not everything is noted in the formal literature.

Before proceeding with the primary objectives, the framework of MDSW must be understood: what regulations and laws concern medical devices, and what is the role of international standards? Regulations are more or less regional, therefore the scope of the research must be defined. This thesis focuses on the European Union (EU) region. EU has recently updated older directives concerning medical devices with two regulations: Medical Device Regulation (MDR) 2017/745 [42] and In Vitro Diagnostic Medical Device Regulation (IVDR) 2017/746 [43]. At the time of writing, the transition period is ending, and all medical devices marketed in the EU must comply with MDR or IVDR [44]. International standards guide the manufacturing and development of medical devices and regulations are generally based on these standards. The most notable standards for medical devices are ISO 13485 [25], IEC 62304 [23] and IEC 82304-1 [21]. Standards are not mandatory, but it is often the most convenient way to comply with regulations. Standards may appear very detailed and rigid but often allow flexibility in how their guidance is implemented. In addition to examining regulations and standards, we must explain DevOps. DevOps, a combination of words *development* and *operations*, is a broad term with a variety of meanings, so we have to define what approaches and techniques we mean when discussing DevOps in this thesis. Chapter 2 investigates these questions and provides definitions to set a basis for discussing DevOps in software development.

1.3 Contribution

This thesis aims to provide an understanding of the current state-of-the-art and offer step-by-step guidance on implementing DevOps practices in MDSW development. Guidance will be drawn from existing best practices and recommendations from researchers and practitioners. Ultimately, this study sought to facilitate the adoption of modern software development practices in the field of medical devices.

This thesis shows that DevOps benefits of improved software quality and faster release cycles can be achieved in MDSW development. Continuous Integration (CI) and Continuous Delivery (CD) pipelines can be set up. These and extended DevOps tools can help automate documentation creation and support compliance requirements. Bringing developers and regulatory experts closer and streamlining the regulatory process enables faster completion of compliance tasks. However, regulations prevent Continuous Deployment and pose challenges for third-party platforms and AI/ML solutions.

1.4 Thesis structure

The remainder of this thesis is organized as follows. In Chapter 2 we provide a background on the regulations and standards concerning medical devices and explore DevOps in general. Chapter 3 presents the study methodology and research questions. Chapter 4 presents the results of a multivocal literature review. In Chapter 5, the results are discussed, step-by-step guidance on applying DevOps to MDSW development is presented, and the research questions are answered. Finally, the conclusions are drawn in Chapter 6.

2 Background

This chapter presents standards relevant to medical devices, EU medical device legislation, and an overview of DevOps and agile development in the regulated field.

2.1 Regulatory domain

The regulatory domain of medical device software can be interpreted as a layer model [19]. In the European context, the layers are EU legislation, national legislation, guidance documents, and international standards (Figure 2.1). From the manufacturer's perspective, EU legislation, MDR, or IVDR sets the baseline rules and is applied in all cases. EU member states have national legislation that can set additional requirements for marketing products in specific countries. MDR and IVDR are not easy to implement, thus, several guidance documents that help with the effective and harmonized implementation of the regulations have been implemented by the Medical Device Coordination Group (MDCG) [47]. Relevant international standards established by the International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO) are either harmonized with MDR and IVDR or at least consistent with them. The following standards are usually the best way to comply with legislation layers.

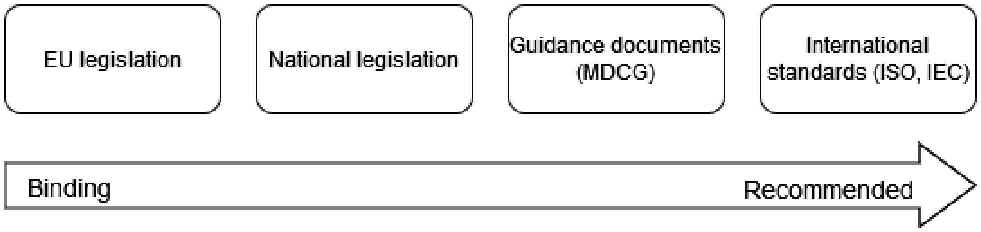


Figure 2.1: Layers of EU regulatory framework, adapted from [19].

2.2 International standards

International standards are norms or requirements, established by recognized and entrusted bodies, such as the International Electrotechnical Commission (IEC) and International Organization for Standards (ISO) [22, 27]. They are written and maintained by experts in the field and can be seen as a globally agreed formula for the best way of doing something [29]. Standards are comprised of rules, guidelines, processes, and characteristics that allow repeatable outcomes [22]. Relevant standards for medical devices are presented in Table 2.1. The most notable ones are ISO 13485 [25], IEC 62304 [23], and IEC 82304-1 [21]. In addition, ISO 14791 [26] and IEC 62366-1 [24] overlap at MDSW development, as illustrated in Figure 2.2.

Table 2.1: List of standards applicable on MDSW [21, 23, 24, 25, 26].

Standard name	Title
ISO 13485:2016	Medical devices – quality management systems – requirements for regulatory purposes
ISO 14971:2019	Medical devices – application of risk management to medical devices
IEC 62304:2006	Medical device software – Software life cycle processes
IEC 62366-1:2015	Medical devices – Part 1: Application of usability engineering to medical devices
IEC 82304-1:2016	Health software – Part 1: General requirements for product safety

Standards guide manufacturing at the general level [29]. They present the circumstances and activities that manufacturers should prepare for and what inputs must be considered for the activity. Standards tell us what needs to happen within certain activities, and what personnel have to be involved. Finally, the required outputs of these activities are presented. However, these standards do not explain detailed low-level actions to meet the requirements. In addition, standards do not prevent practicing more activities or having even stricter internal requirements.

ISO 13485 defines the requirements of a quality management system (QMS) [25]. A QMS is used to prove that an organization is capable of producing medical devices or services that consistently meet both customer and regulatory requirements. Establishing the QMS is seen as the minimum act for organizations working in the medical device domain because EU MDR requires it [41]. Thus, ISO 13485 is relevant for MDSW development organizations although it does not directly guide the software development process.

ISO 14971 specifies the process required for medical device risk management [26]. The purpose of risk management is to eliminate unacceptable risks when producing medical devices that are safe to use. Risk management is a continuous process throughout the life

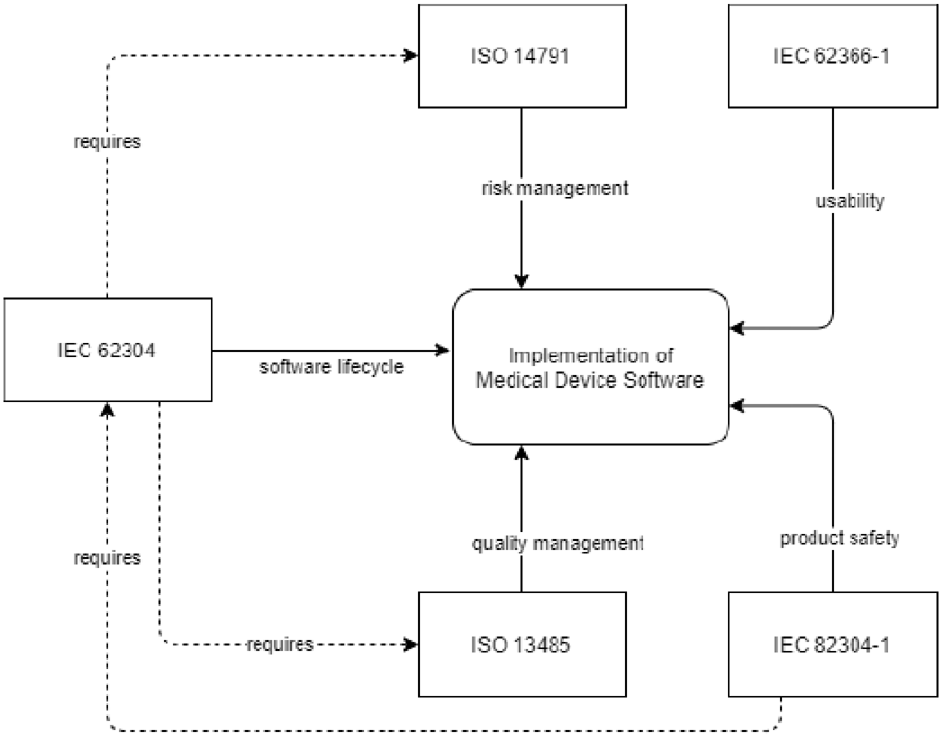


Figure 2.2: Relationship of MDSW and the related standards (Figure source [47]).

cycle of a device and, in most cases, is connected to other QMS processes, although the standard does not specifically require it. The standard was connected to other relevant standards, such as IEC 62304 and IEC 62366, which are cross-referenced with ISO 14971.

IEC 62304 establishes a framework for the entire life cycle of medical device software [23]. It defines the processes, activities, and tasks that must be completed during the MDSW life cycle. This standard describes software development and maintenance processes, risk management, configuration management, and problem resolution [41]. Validation and final release are not covered in IEC 62304, but are nevertheless required in ISO 13485 and MDR/IVDR. The standard does not exactly state how to implement its requirements, so manufacturers can choose modern software development practices, as long as they can implement the required tasks from the standard. However, the standard establishes a sequential order of activities that might require the output from the previous step as an input, thus restricting implementation at some level. IEC 62304 introduces the following software safety classes [23]:

- Class A: no injury or damage to health is possible.
- Class B: non-serious injury is possible.

- Class C: death or serious injury is possible.

The software safety class determines the requirements of IEC 62304 that apply to MDSW [41]. Class C must comply with all requirements, Class B is exempted from some, and Class A has multiple requirements.

Usability engineering is an important part of medical device product development, and IEC 62366 is considered a state-of-the-art method for planning usability engineering activities [41]. The standard defines requirements to mitigate use-related risks by focusing on the risk management of the user interface, ergonomic risks, and environmental risks [24].

IEC 82304 is a standard specifically concerning standalone software that is not embedded in physical devices but operates on general computing platforms [21]. It references the life cycle requirements of IEC 62304 and adds health software usage requirements to the process. The standard goes further than IEC 62304 by introducing requirements for validation, product identification, and post-market activities, such as maintenance and disposal [47].

Standards are not mandatory to follow, but as they contain so-called condensed wisdom about how something should be done to retain high quality, in practice it is considered the best way to ensure compliance and a good working process [47]. The relevant standards are harmonized with regulations, meaning that a standard is incorporated into EU legislation [29]. In essence, the EU MDR and IVDR set the requirements, and the standards guide the manufacturer to use the best practices in the industry to comply with requirements. However, it should be noted that standards still allow the manufacturer to have leeway to decide how to implement these practices.

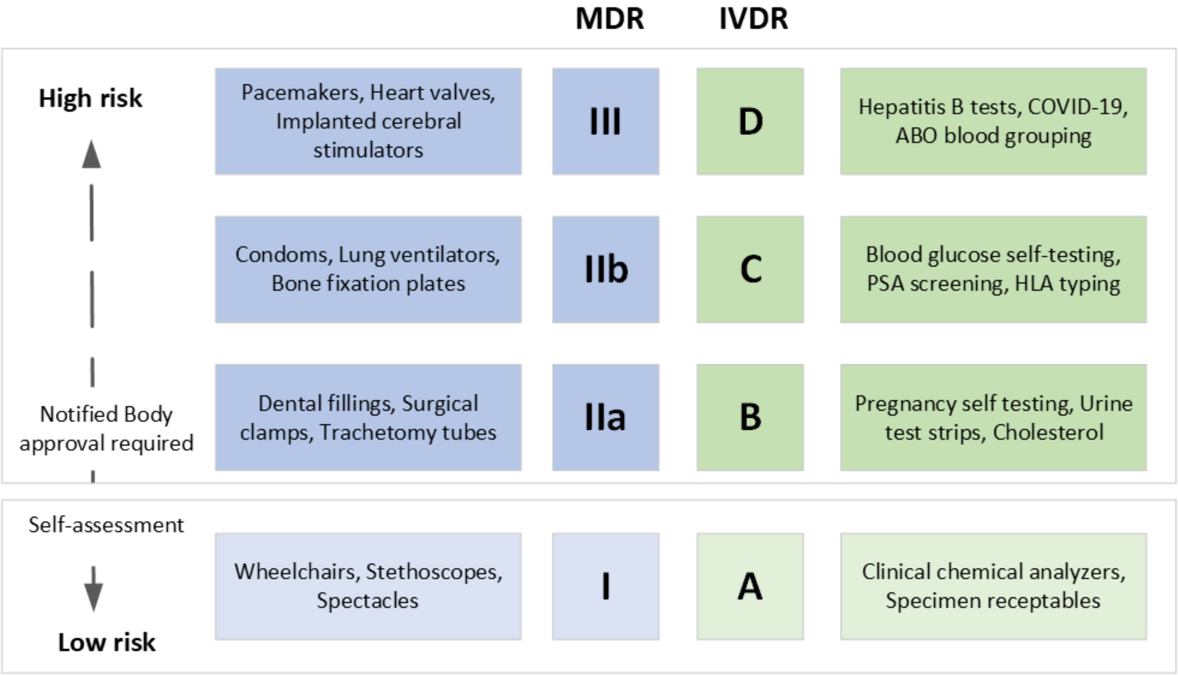
2.3 EU Medical Device Regulation

Within the European Union (EU), two regulations concerning medical devices and software are in place: Medical Device Regulation (MDR) 2017/745 [42] and In Vitro Diagnostic Medical Device Regulation (IVDR) 2017/746 [43]. Regulations have been transitioned from the old Council Directive 93/42/EEC on Medical Devices (MDD) [3] and Directive 98/79/EC on In Vitro Diagnostic Medical Devices (IVDD) [6] to the new MDR and IVDR, and the transition period is now ending [44]. Earlier versions were directives, and such were implemented as national laws when regulations were legal acts applying automatically and uniformly to all EU countries [9]. MDR and IVDR aim to improve the quality, safety, and reliability of medical devices by harmonizing legislation in the EU region and removing the national interpretation of the law [42, 43].

MDR defines a medical device as any instrument, apparatus, appliance, software, implant, reagent, material, or other article intended to be used (alone or in combination) for medical purposes by the manufacturer [42]. Medical devices intended to be used in vitro for the examination of specimens, such as blood and tissue, derived from the human body, are defined as in vitro medical devices, and regulated by IVDR [43]. MDR and IVDR do not differentiate between the physical device, embedded software, or standalone software (nor do MDD and IVDD) [42]. This means that standalone software is considered a medical device as long as it is intended for medical use. Therefore, developing such software must follow the same directives, regulations, national laws, and technical standards as manufacturing physical medical devices. This practice leads to challenges for organizations developing standalone medical software [17]. As a result, medical software manufacturers are required to have a QMS in place, and placing products on the market requires a certification process by the Notified Body. Although implementing QMS has been an established route to comply with regulations for medical device manufacturers, such a solution might not be easy for organizations working only with software [12].

Compliance with MDR or IVDR is necessary to sell or distribute the device [41]. MDR and IVDR have high safety, reliability, and quality standards for use in medical devices. They cover the medical device life cycle from clinical investigation to sale and post-market actions. The intended purpose of the device was at the core of regulations. In addition to defining whether the device is medical, it defines the classification of medical devices. This classification reflects the potential harm that could be caused by the risk being realized. MDR and IVDR do not differentiate between the composition and type of device. Software, a physical device, or any other article was classified according to the same rules. It should be noted that this classification differs from the safety classification introduced in the IEC 62304 standard.

Medical devices are classified into three classes: Class I, Class II, and Class III [42]. Classification affects the requirements with which the device must comply. Class III is the highest risk-level class, and includes devices that can cause death or irreversible injury. Class II is divided into two sub-classes: IIa and IIb. Devices that can cause harm or danger are classified under IIb. All devices or software that monitor physiological processes or that are used for diagnostic or therapeutic purposes are classified as IIa. Other devices and software were classified as Class I. As devices and software for medical purposes are



	MDR	IVDR	
Pacemakers, Heart valves, Implanted cerebral stimulators	III	D	Hepatitis B tests, COVID-19, ABO blood grouping
Condoms, Lung ventilators, Bone fixation plates	IIb	C	Blood glucose self-testing, PSA screening, HLA typing
Dental fillings, Surgical clamps, Trachetomy tubes	IIa	B	Pregnancy self testing, Urine test strips, Cholesterol
Wheelchairs, Stethoscopes, Spectacles	I	A	Clinical chemical analyzers, Specimen receptables

Figure 2.3: Overview of classification of medical devices [33].

usually used for monitoring or diagnostic purposes, most fall into Class IIa or higher.

In vitro medical devices are classified into four classes, from A to D [43]. Class D is the highest risk-level class, and contains devices that detect human organic materials that cause life-threatening diseases. Devices used, for example, self-testing glucose or screening prostate cancer, are classified into Class C. Class B contains devices used for self-testing pregnancy, fertility, cholesterol levels, etc. They also contain devices that cannot be classified into other classes. Class A is the lowest risk-level class and contains general-purpose devices used in in vitro diagnostics.

Figure 2.3 presents an overview of the medical device classifications. Classifying a medical device is usually not an easy task, and long guidance documents have been published by MDCG [37, 38]. Third-party organizations also offer services for device classification.

MDR and IVDR require activities before a product is placed on the market [41]. The device must be evaluated using either a clinical evaluation (MDR) or performance evaluation (IVDR). The goal of these activities is to prove that the medical device is working as intended and is safe to use for patients. The evaluation is performed systematically during the planning phase of the life cycle and requires comprehensive documentation. In the case of in vitro devices, scientific evidence is required [43]. After development validation activities are required before launching the product [41].

The CE mark is the EU's mandatory mark for regulated goods sold in EU countries [11]. CE marking indicates that the manufacturer has assessed a product and it meets EU safety, health, and environmental protection requirements. In addition to the manufacturer's assessment, a declaration of conformity from a Notified Body (NB) is required. Several other tasks were also required. The device should be given a Unique Device Identifier (UDI) and should be registered for the national authority or the EUDAMED database [8]. In addition, the manufacturer needs to ensure that all required technical documentation is verified and in place, and that conformity assessment for the product is done in a proper manner, involving the NB if required.

After placing the product on the market, a new type of activity called post-market surveillance (PMS) is required [41]. PMS is a systematic procedure within the QMS, and its purpose is to ensure continuous patient safety. PMS includes vigilance activities, feedback handling, post-market clinical follow-up (PMCF), post-market performance follow-up (PMPF), and various types of reports. In essence, the manufacturer is required to collect information related to the product and closely monitor its safety and performance. The outcome of PMS depends on the classification of the device. The higher the risk level, the more detailed and frequent the monitoring and reporting required. The product is also continuously evaluated by PMCF for medical devices and PMPF for in vitro devices.

2.4 DevOps

DevOps, a combination of words *development* and *operations*, is a broad term that has a variety of meanings ranging from organizational philosophy to simply the use of automated tools for software deployment and maintenance [4]. DevOps philosophy is commonly seen as a cultural shift toward seamless collaboration between development (e.g., programmers and testers) and operation (e.g., system administrators and network technicians), usually referred to as breaking the silos. DevOps has its roots in agile software development and from a practical point of view, it can be described as a set of practices that aims to simultaneously reduce the time between changes to production and ensure high quality at the same time [2]. The impact of agile software development is visible in the definition of DevOps goal: to improve software quality and shorten the development life cycle, also referred to as delivering value to end users [45].

DevOps can improve several areas of software development [4]. This could improve col-

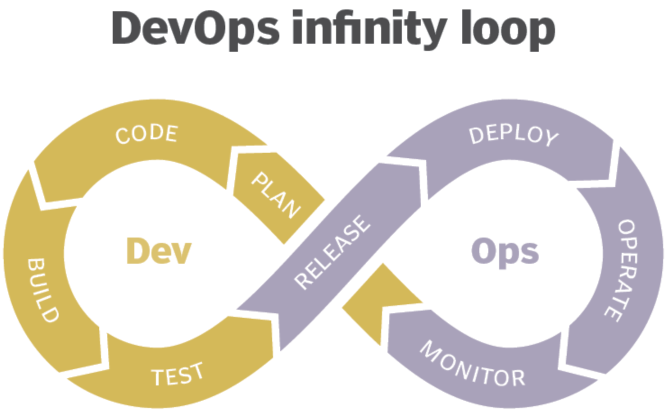


Figure 2.4: DevOps infinity loop describing continuous development process (Figure source [30]).

laboration and communication, as different roles work closely together. DevOps adopt iterative development, which can improve development outcomes and product quality as defects are found early and the business is more responsive to product demands. Deployment management is easier because of the automated pipelines and tools. The iterations follow eight steps that are repeated throughout the iterations: plan, code, build, test, release, deploy, operate, and monitor [30]. This cycle is illustrated in Figure 2.4.

However, the challenge in defining DevOps is that there is no established framework or clear overview of DevOps procedures [4, 31]. DevOps contain a broad range of diverse methods and guidelines which are typically implemented in specific environments [36]. This leads to the meaning of DevOps and the agreement of requested techniques varying between stakeholders, and it is easy to call almost any type of approach as DevOps [31]. A simple example is tools to automate processes: there are many tools available, and questions include how to find a proper choice, how it should be configured, and how it should be used. Because DevOps can be practiced in several ways, it is not straightforward to adopt it.

Nevertheless, certain practices considered as DevOps can be described. These include version control, Continuous Integration and Continuous Delivery, Continuous Deployment, automated testing, proactive monitoring of production environments, and a culture of high trust [30]. Continuous Integration (i.e., the practice of constantly merging working copies to a shared mainline [45]) and Continuous Delivery (i.e., the practice of ensuring that an application is always in a production-ready state [45]) together are known as CI/CD. Production artifacts should be stored in a centralized version control system that should be integrated into the CI/CD pipeline. The CI/CD pipeline and automated tests ensure that small incremental changes to the software are tested and added to the software continuously, rather than at predetermined times. In Continuous Deployment, changes are deployed to the end users as soon as they pass the pipeline. Changes are then monitored in a production environment, which provides feedback for planning future changes. In addition, organizational culture should allow debates, hard questions, and failures to enable continuous development.

DevOps is a buzzword and the wording has been applied to other approaches that have similar ideas of combining two or more conventionally separated expertise. These are typically referred to as "xOps", such as MLOps or DevSecOps. MLOps is a combination of machine learning and operations, and its purpose is to apply the DevOps approach to artificial intelligence and machine learning (AI/ML) development [16]. MLOps embraces automation and monitoring in all steps of ML system development. AI/ML enables the design of innovative systems in healthcare [39], and MLOps faces similar problems in MDSW development as DevOps does [16]. DevSecOps is an extension of the DevOps approach that emphasizes the role of security in software development [1]. MDSW needs to be secure, and cybersecurity is addressed in MDR, so security compliance is as much concern as other regulatory requirements for medical device manufacturers [20]. The term RegOps, combining regulatory and operations, is sometimes used within MDSW to describe the general collaboration between software developers and regulatory experts [15]. However, RegOps can also describe collaborations unrelated to software development in any regulated field.

One definition of DevOps within MDSW is *a development processes that reduce repetitive tasks in development, quality assurance, and deployment with the help of automation tools and workflows* [34]. This acts as a basis for the DevOps practices that we study in this thesis. We focus on automation tools and workflows for both development and quality assurance, the CI/CD pipeline and Continuous Deployment, and organizational culture and collaboration between developers and compliance officers. In addition, we acknowledge other "xOps" approaches, such as MLOps and DevSecOps.

3 Research methods

This chapter presents research the methods and questions, and the study protocol. The study protocol describes the search process, criteria for source selection and quality assessment, snowballing, and data extraction and synthesis.

3.1 Method selection

In this thesis, a multivocal literature review (MLR) was conducted. MLR is a systematic literature review that includes both academic and grey literature (GL). In addition to academic peer-reviewed papers, GL, such as blogs, technical reports, and standards is gathered systematically and used as input for the review.

The guidelines proposed in [13] were followed when considering an appropriate method for this research. As the goal was to obtain an overview of the state-of-the-art and understand the possibilities for DevOps in MDSW development in general, a literature review was a basis. Instead of a conventional systematic literature review, the decision to include GL was made based on the MLR guideline 3 [13]. There were seven questions to consider, and we answered "Yes" easily for four of them. Table 3.1 presents the questions and their answers. The inclusion of GL in the study was justified according to the guidelines.

3.2 Research questions

In general, the research problem was how DevOps could benefit regulated medical device software development. To understand this, we had to determine how the DevOps approach was already utilized in medical device software development, what means were suggested by experts in the field, and what could be learned from established practices and practical experiences. The following research questions were derived from these questions:

RQ1: What is the state-of-the-art of DevOps in regulated medical device software development?

RQ2: What is the suggested way to utilize DevOps in medical device software development?

Table 3.1: Questions and answers to decide whether to include the GL in the research [13].

#	Question	Answer
1	Is the subject "complex" and not solvable by considering only the formal literature?	Yes. The DevOps state-of-the-art in MDSW development is not found solely in the formal literature.
2	Is there a lack of volume or quality of evidence, or a lack of consensus of outcome measurement in the formal literature?	Yes. The formal literature about DevOps in the context of MDSW is scarce.
3	Is the contextual information important to the subject under study?	Yes. Regulations in the medical device domain are unique to the field.
4	Is it the goal to validate or corroborate scientific outcomes with practical experiences?	Partially. The goal is to find out how practical experiences relate to the scientific findings.
5	Is it the goal to challenge assumptions or falsify results from practice using academic research or vice versa?	Partially. The goal is to find out if there are some differences between practicalities and suggestions based on scientific papers.
6	Would a synthesis of insights and evidence from the industrial and academic community be useful to one or even both communities?	Yes. Software engineering is a highly practical field, so practitioners usually have solid approaches. On the other hand, the medical device domain is highly regulated, so approaches have to be systematically considered and justified.
7	Is there a large volume of practitioner sources indicating a high practitioner interest in a topic?	No. The DevOps approach is uncommon among medical device manufacturers. However, some of them praise the benefits of DevOps, so there is something to be discovered.

RQ3: What are the challenges and benefits of DevOps practices in medical device software development?

RQ1 focused on experiences from the industry, case studies, and other information about the current status of the DevOps approach in MDSW development. RQ2 focused on methods and techniques that are possible or suggested but not widely used, or require a specific setting to be utilized. RQ3 focused on the evidence of the challenges and benefits of DevOps in MDSW development.

3.3 Study protocol

The study protocol followed the MLR guidelines [13]. The protocol described how the literature was gathered, the criteria for including or excluding data sources, and the processes for data extraction and synthesis. Some adjustments were made to the proposed process, such as repositioning snowballing and accommodating single-author work. The protocol is illustrated in Figure 3.1. The guidelines were focused especially on handling GL, therefore, some guidance for SLR [32] was followed where applicable.

3.3.1 Search process

Formulating the search phrase is an iterative process that involves preliminary searches, trial searches, and consultations with experts [32]. We considered preparatory work performed during course exercises and a similar MLR [12]. In addition, the thesis supervisor was consulted and several search phrases were trialed in the most popular scientific databases. The basis was the terms "devops" and "medical device", and this was expanded by the names of the standards, the names of regulations and authorities, and with terms like "software", "healthcare", "regulation", "regulated", "development", and "environment". In addition, "medical systems" were added to broaden the search. The search phrase was iteratively formatted during the trial search.

The databases were considered simultaneously when formulating the search phrase. The databases IEEE Xplore, ACM Digital Library, ScienceDirect, Wiley Online Library, SpringerLink, Google Scholar, and Google Web Search were used in [12]. All scientific databases except the Wiley Online Library, were listed in the University of Helsinki Computer Science database list. The author of this thesis did not have access to the Wiley Online Library, so it was left out. The Scopus database was listed in the Computer Science database list and recommended by the supervisor, so it was included. Google Scholar and SpringerLink provided several hundred results during trial searches, so they were excluded, and snowballing was used instead to find relevant papers potentially missed by this decision.

In the previous MLR [12], the dates of the sources were limited between 2015 and 2020. We did similar limitations, but due to the uncertainty of database updates, the dates limit

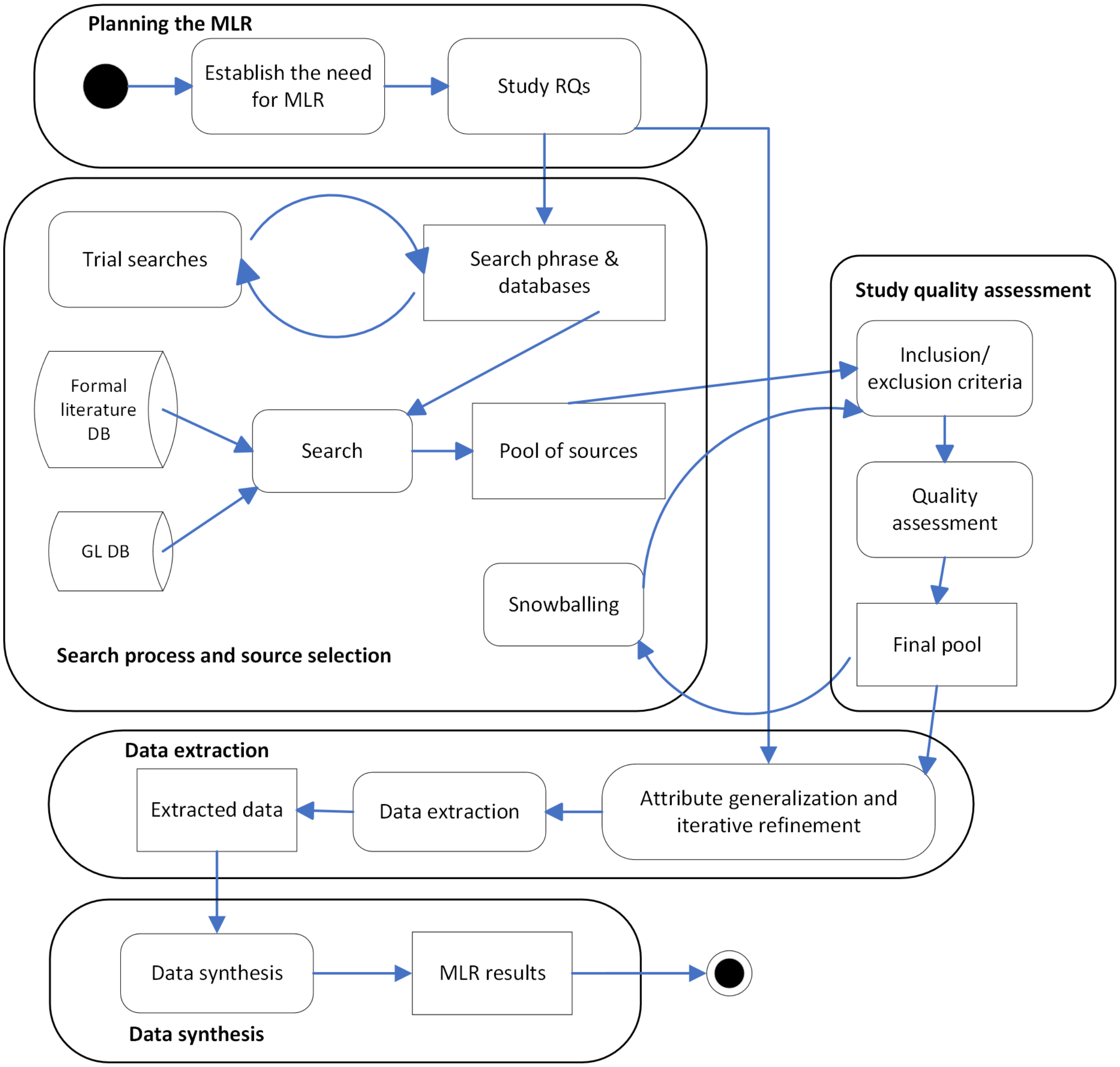


Figure 3.1: An overview of the study process, adapted from [13].

was set between 2018 and 2024.

Based on this process, the following search phrase was used.

(devops OR regops) AND ("medical system" OR "medical systems" OR "medical device" OR "medical devices" OR "medical software") AND (regulated OR regulation)

The search was done in the following databases:

- ACM Digital Library
- IEEE Xplore
- ScienceDirect
- Scopus
- Google Web Search (for GL)

The concept of "quasi-gold standard" (QGS) was proposed in [49] to evaluate search performance. Preparatory work showed that papers [18, 35, 46] were fundamental to this research, so they were used to determine the QGS. All of these were found in the trial search.

Search-stopping criteria are needed for GL, as there could be hundreds of thousands or even more results. Theoretical saturation (no new concepts emerge from the results), effort-bounded (including only the top N results), and evidence exhaustion (extracting all evidence) are the criteria proposed in Guideline 8 [13]. We used effort bounding to stop searching Google Web Search for the first 100 results.

The search was conducted in March 2024. In total, 245 formal literature results were found, and approximately 13 900 results of GL.

- ACM Digital Library: 57
- IEEE Xplore: 97
- ScienceDirect: 22
- Scopus: 69
- Google Web Search: ca. 13 900

After applying the search-stopping criteria, the results were collected using the tool Paperpile [40]. Formal literature was collected by exporting citations to Paperpile. GL was collected by saving web pages or links to videos as PDF files and uploading them to Paperpile. In the case of videos, a link to the video was saved in Paperpile. Paperpile was used to automatically remove duplicate results by comparing the meta data of the sources. Finally, 216 sources of formal literature and 75 sources of GL were collected. The lists of sources are available in Appendix A.

3.3.2 Source selection criteria

Guidelines 9 and 10 [13] state that the criteria for source selection and quality assessment should overlap, and that the source selection process for GL and formal literature should be integrated in a coordinated manner. The inclusion and exclusion criteria were based on research questions [32]. We were looking for sources that discussed the current state of the DevOps approach, suggested practices to implement DevOps, or considered obstacles to DevOps in MDSW development. In addition, practical issues, such as language, were used as criteria.

The following inclusion criteria were applied:

1. All outlet types in the search results. These included articles, blogs, books, magazines, news articles, presentations, reports, theses, white papers, and videos.
2. Literature relevant to medical devices and discussing DevOps or regulations and MDSW compliance.

The following exclusion criteria were applied:

1. Inaccessible sources.
2. Sources not in English or Finnish.
3. Sources dated before the year 2018.
4. Sources without date or published year or author or organization.
5. Sources discussing solely requirements of The U.S. FDA.
6. Duplicates and otherwise similar data sources.

7. Secondary studies, such as systematic literature reviews, or mapping studies.
8. Vendor advertisements or job announcements.
9. Search result pages in other search engines.
10. Courses or event invitations.

The criteria were applied manually by skimming through all sources. The title, abstract, and first few paragraphs were read from the sources of formal literature. From the sources of GL, the author or organization, and date were checked, and the title and first few paragraphs were read, or in the case of videos or presentations, the first few frames were watched.

After applying the inclusion and exclusion criteria, 18 formal literature and 15 GL sources remained.

3.3.3 Quality assessment criteria

Assessing the quality of formal literature and GL has the same goal, of determining sources that are valid and free of bias [13, 32]. However, to achieve this goal, different actions are required depending on the literature type. Formal literature follows a controlled review and publication process, so it was not meaningful to assess whether authors have the expertise, or if there is any evidence to support inferences. GL requires a more thorough assessment, and the diversity of GL makes it more laborious. Even the basic details, such as the name of the author, had to be checked. For these reasons, separate quality assessment criteria were created for the formal literature and GL.

The quality assessment criteria for formal literature were based on the criteria used in [7] and [28]. For the answers, a 2- or 3-point scale was used. The criteria used were as follows:

1. Did the source present an empirical study (or is it an overview or "lessons learned" report)? (*Yes=1.0, No=0.0*)
2. Were the aims and motivations of the research clearly stated? (*Yes=1.0, to some extent=0.5, No=0.0*)
3. Was there an adequate description of the research context? (*Yes=1.0, to some extent=0.5, No=0.0*)

4. Was the chosen methodology appropriate for addressing the aims of this research? (*Yes=1.0, to some extent=0.5, No=0.0*)

5. Was the data collection process documented and addressed the research issue? (*Yes=1.0, to some extent=0.5, No=0.0*)

6. Was the data analyzed using rigorous and suitable methods? (*Yes=1.0, to some extent=0.5, No=0.0*)

7. Were possible biases evaluated, and were validity and limitations concerned and discussed? (*Yes=1.0, to some extent=0.5, No=0.0*)

8. Were the findings clearly stated and justified with credible results? (*Yes=1.0, to some extent=0.5, No=0.0*)

9. What is the level of value of research or practice? (*High=1.0, Moderate=0.5, Low=0.0*)

For GL, the following quality assessment criteria were created based on the guideline 11 [13]. Depending on the question, a 2- or 3-point scale was used.

1. Was the publishing organization identifiable and reputable? (*Yes=1.0, No=0.0*)

2. Was the author associated with a publishing organization? (*Yes=0.5, No=0.0, No author=0.0*)

3. Did the author have expertise in this area? (e.g., job title Software Engineer or Quality Manager)? (*Yes=1.0, No=0.0*)

4. Did the source have a stated methodology? (*Yes=1.0, No=0.0*)

5. Did the work cover a specific question? (*Yes=1.0, No=0.0*)

6. Was the work presentation balanced? (*Yes=1.0, No=0.0*)

7. Was there any evidence to support the statements in the work? (*Yes=1.0, No=0.0*)

8. Did it strengthen or refute a current position? (*Yes=1.0, No=0.0*)

9. Were the statements of a subjective opinion? (*Yes=0.0, No=1.0*)

10. Which outlet type was the material? (*high=1.0 / moderate=0.5 / low=0.0, where high=books, magazines, theses, government reports, whitepapers, international standards; moderate=annual reports, news articles, presentations, videos, Q/A sites, Wiki articles; and low=blogs, emails, tweets*)

A quality assessment score of 0-1 was calculated for all sources by summarizing and normalizing the set of answers. Sources with a normalized quality assessment criterion score greater than 0.5 were included. This scoring system has been used in both formal literature and GL. Microsoft Excel was used to collect quality assessment data and Paperpile was used to classify the sources based on the quality assessment to find the included sources easily.

After applying the quality assessment criteria, 15 sources of formal literature and nine sources of GL remained. The quality assessment is presented in Appendix B.

3.3.4 Snowballing

Snowballing (a technique to follow references from a source) was utilized in the study process to find relevant sources not present in the searches [48]. Snowballing was performed at one level for the formal literature, and both backward and forward snowballing were performed.

References from all formal literature sources that passed the quality assessment criteria, were reviewed (backward snowballing). In addition, all papers referring to this source were reviewed (forward snowballing). The Scopus database was used. Only if the source was not found in the Scopus, Google Scholar was used. Inclusion/exclusion criteria were applied to all the reviewed sources. The quality assessment was done on sources fulfilling the inclusion criteria, and sources with a criterion score over 0.5 were included in the sources.

Snowballing led to the inclusion of three additional formal literature sources and 1 GL source. The final literature consisted of 18 sources of formal literature and 10 GL sources. The number of sources collected and included in the data-collection steps is shown in Figure 3.2.

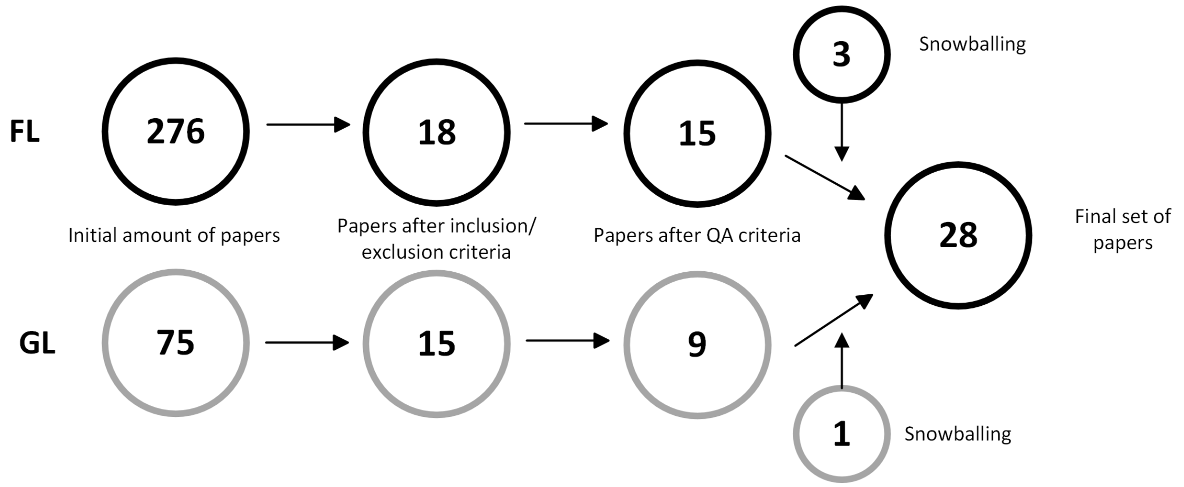


Figure 3.2: Number of sources by data collection steps.

3.3.5 Data extraction

Both [13] and [32] focused on the design of the data-extraction form. The form must be designed such that *the chain of evidence* is visible, in other words, *traceability* is maintained. In addition, Guideline 12 [13] emphasizes that data extraction must be designed to address each research question sufficiently. As the thesis was conducted by a single researcher, the proposed data extraction procedures [32] requiring two or more researchers could not be followed. Instead, the proposed test-retest process was utilized. In the test-retest process, the second extraction was done from a random selection of sources to ensure consistency in data extraction.

The design of the data extraction form was based on the proposed guidelines [13, 32] and an example from an earlier MLR study [14]. The structure of this form is shown in Table 3.2.

Paperpile was used to manage all sources and their metadata, and the sources were stored on the OneDrive University account. Microsoft Excel was used to collect the extracted data and link them to the sources.

3.3.6 Data synthesis

GL is diverse, therefore a suitable data synthesis method depends on the sources, as stated in guideline 13 [13]. Both the formal literature and GL sources found in this research were mostly qualitative and experience-based in nature. The guidelines for SLR [32] present

Table 3.2: Structure of data extraction form, adapted from [13, 14].

Field	Concern/research question	Details
Number	Documentation	-
Source	Documentation	Title of the source
Demographic info	Documentation	Year, publication forum, and author affiliation.
DevOps state-of-the-art in MDSW development	RQ1	Topics where DevOps approach is taken, DevOps techniques and practices in use, tools, implementation.
Suggested techniques and approaches	RQ2	Suggested DevOps tools (what kind of tool), suggested DevOps approaches, CI/CD, Continuous Deployment, IaC, AI/ML, standard related, organizational culture.
Challenges of DevOps in MDSW development	RQ3	Compliance requirements, Notified Bodies and authorities, organizational culture, third-party platforms, AI/ML systems, DevOps technique related, standard related.
Benefits of DevOps in MDSW development	RQ3	Faster development cycle, higher product quality, easier understanding of compliance requirements for non-regulatory developers, feedback, documentation.
Other	RQ1, RQ2, RQ3	Any other comments and insights.

qualitative and descriptive data synthesis. They have similarities and are suitable for data synthesis in this research, therefore we conducted qualitative coding inspired by these techniques.

Qualitative coding was performed by labeling and organizing the extracted data to identify themes and relationships between them. Labels were based on details fields in data extraction form. References to extracted data were collected to the data synthesis sheet under the respective labels. Data were organized under the research questions and were clumped further down by the labels. The significance of certain labels was determined by the number of sources providing the data and the quality assessment scores of the sources. Microsoft Excel stored in the author's OneDrive was used to manage data synthesis.

4 Results

In this chapter, the findings from the synthesized data are presented. The data sources are listed in Appendix C. First, we provide a brief overview of the sources, and the results are presented according to the research questions.

4.1 Overview

The metadata of the final set of sources is listed in Table 4.1.

The sources were selected from the entire period defined by the search. Most of the sources (eight) are from 2021. Sources do not contain formal literature from 2024 (the writing year of this thesis). Figure 4.1 illustrates the number of sources categorized by year.

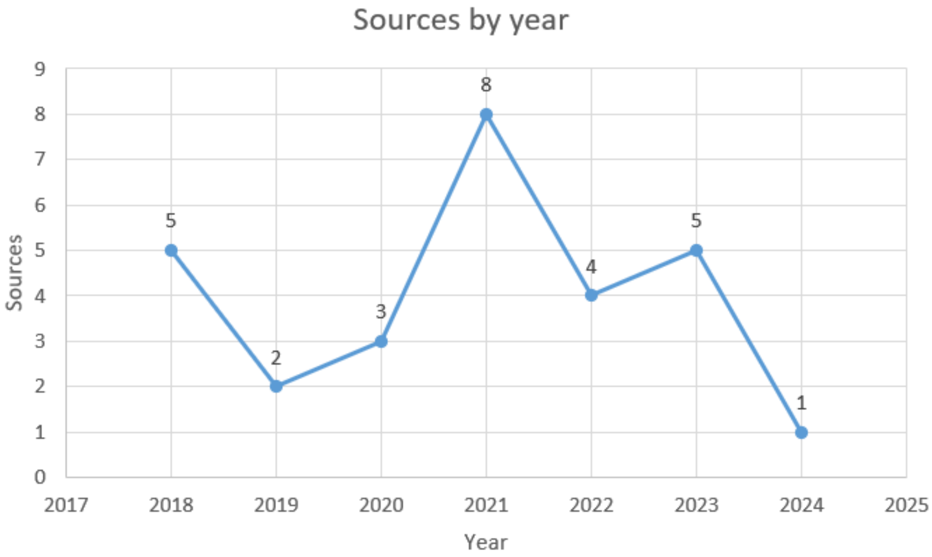


Figure 4.1: Sources by year. Full data is shown in Table 4.1.

There were eight source types: articles, workshop papers, conference papers in the formal literature and research reports, white papers, blogs, web articles, and theses in GL. From the formal literature, an article published in a journal was the most common with ten articles in the source set. In GL, a blog was the most common source type (five sources). Types of sources are presented in Figure 4.2.

Table 4.1: Publishing year, publication forum, author affiliation, and research questions discussed in the sources.

ID	Year	Publication forum/type of source	Author affiliation	Contributes for research questions
S63	2023	Journal	Academic	RQ1, RQ2, RQ3
S64	2023	Journal	Collaboration	RQ1, RQ2, RQ3
S81	2022	Journal	Collaboration	RQ2, RQ3
S82	2022	Journal	Collaboration	RQ1, RQ2
S139	2021	Workshop	Collaboration	RQ2, RQ3
S143	2021	Journal	Collaboration	RQ3
S146	2021	Journal	Collaboration	RQ2, RQ3
S152	2021	Journal	Collaboration	RQ2, RQ3
S153	2021	Journal	Collaboration	RQ1, RQ2, RQ3
S168	2020	Conference	Collaboration	RQ2, RQ3
S170	2020	Conference	Collaboration	RQ1, RQ2, RQ3
S189	2019	Conference	Industry	RQ1, RQ3
S207	2018	Journal	Academic	RQ1, RQ2
S209	2018	Conference	Collaboration	RQ2, RQ3
S210	2018	Journal	Academic	RQ2
S217	2021	Workshop	Collaboration	RQ2
S218	2019	Workshop	Industry	RQ1, RQ2, RQ3
S220	2018	Conference	Academic	RQ2, RQ3
G5	2023	Research report	Collaboration	RQ2, RQ3
G15	2022	White paper	Industry	RQ2, RQ3
G25	2024	Blog	Collaboration	RQ1, RQ2, RQ3
G27	2021	Web article	Industry	RQ2, RQ3
G31	2020	Blog	Collaboration	RQ1, RQ2, RQ3
G32	2022	Blog	Industry	RQ1, RQ2
G64	2023	Blog	Industry	RQ1, RQ2, RQ3
G66	2023	Blog	Industry	RQ1, RQ2, RQ3
G71	2021	Master's thesis	Academic	RQ2, RQ3
G76	2018	Web article	Industry	RQ1, RQ2, RQ3

Most papers discussed topics relevant to more than one research question. Almost every source (26 sources) had some suggestions for implementing the DevOps approach. Many of the sources have discussed the challenges and benefits of DevOps. Presenting state-of-the-art was uncommon, and only half of the sources provided some information about this. Sources contribution for research questions is shown in Figure 4.3.

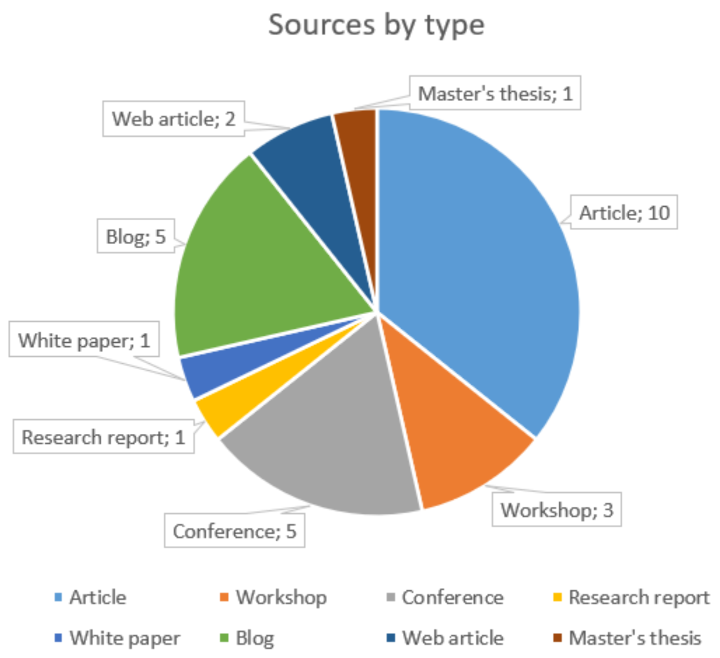


Figure 4.2: Sources by type. Full data is shown in Table 4.1.

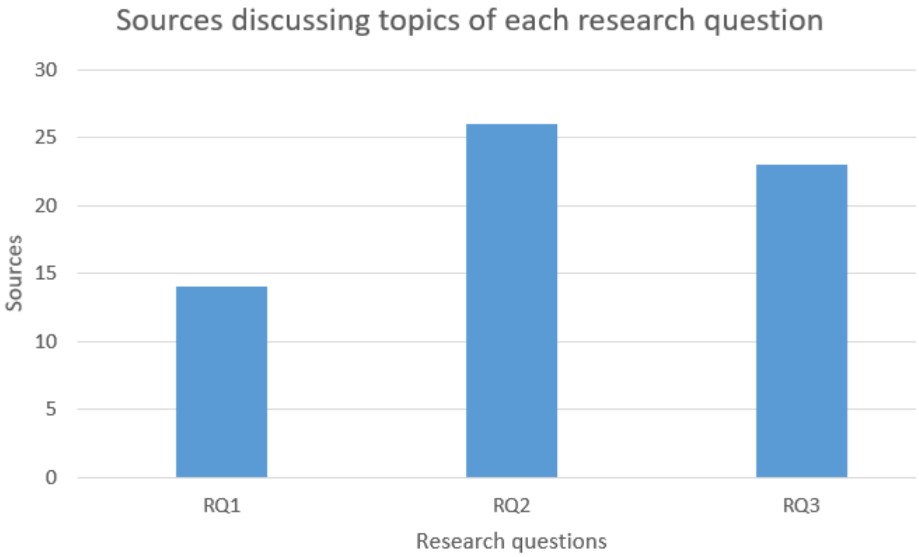


Figure 4.3: Sources by type. Full data is shown in Table 4.1.

Author affiliations look at whether the authors of the source solely represent the academic field or industry. Over half of the sources had authors affiliated with both, resulting in "collaboration" affiliation. Most commonly, collaborative author affiliation resulted from one author representing a university and another representing a technology company.

Figure 4.4 illustrates author affiliation distribution.

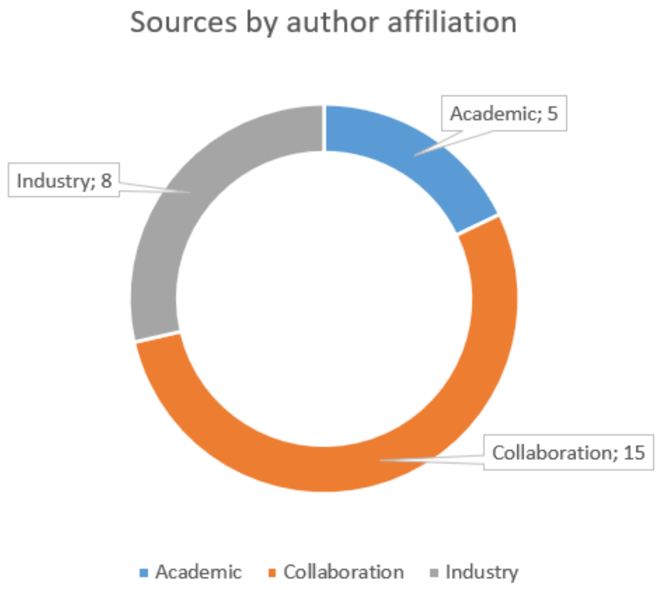


Figure 4.4: Sources by type. Full data is shown in Table 4.1.

4.2 The state-of-the-art of DevOps in MDSW (RQ1)

The data on the state-of-the-art of DevOps is divided into topics, practices, and tools. Table 4.2 shows an overview of the results. Topics describe the part or aspect of the MDSW development life cycle, practices represent specific DevOps techniques and approaches used, and tools describe tools used to achieve DevOps methods.

In one project [S63], MDSW maintenance is handled by a DevOps approach called software maintenance architecture (SWMA). Within the same project, software documentation is created and exported automatically to maintain medical device files (entities containing all regulatory documentation of medical devices). Automated documentation creation is in place in the project of a large technology company Siemens [S189]. The gap between developers and regulatory experts was reduced for the same project. Automated testing is performed in several MDSW development cycles [G66]. The DevOps approach is used to handle risk analysis for software of unknown provenance (SOUP) [S82].

Continuous Integration is practiced in several sources, and although it is not explicitly mentioned in all cases, it is done to achieve DevOps goals in certain parts of the software

Table 4.2: State-of-the-art of DevOps in MDSW development.

Topics	Software maintenance, testing, documentation, risk analysis, integrating regulatory experts to the software pipeline
Practices	Agile methods (documentation sprint, Domain-Specific Language, Model-Driven Engineering, Test-Driven Development), Continuous Integration and automated documentation creation, Continuous Delivery, Post Mortem, automated tests, Compliance-as-Code, Solita RegProof
Tools	Jira and Confluence, CompliancePal and pull requests, conventional tools (Polarion, Jetbrains, MDevSPICE) used in agile settings

life cycle [S63, S82, S189, G64, G66, G76]. Documentation is automatically created with CI by triggering changes in documentation when a code change is committed [S63, S189, G76]. Continuous Delivery has rarely been mentioned [G64, G76], and its exact implementation is vague. Automated testing is common and is typically performed at the unit, integration, and system levels [S218, G64, G66, G76]. The DevOps method *post mortem* (a process to review failures or incidents) is used in one source [S153]. The Medical Compliance as Code -approach was taken in [S82]. It is similar to other "x as Code", and the software pipeline is utilized to create artifacts required for medical compliance. Company Solita used the *RegProof* approach [S153, G25, G31], which combines the software development cycle with regulatory activities. In addition to DevOps approaches, some sources utilize agile methods such as documentation sprint [S170], Domain Specific Language [S189], and Test Driven Development [G76].

The sources list several tools used to achieve the DevOps approach. Jira and Confluence are customized in SWMA [S63]. Tools Polarion (Application Life cycle Management tool) [G5], JetBrains [S189], and MDevSPICE [S207] are not pure DevOps tools, but they are either customized or used in a way that enables a DevOps approach. The Git and its features, such as pull requests, are utilized in the CompliancePal tool [S81, S82, S170, G66].

The state of the MLOps has been discussed in several studies [S64, S82, S139, S153]. Model cards for ML and CD4ML approach are in use. All ML models in production are in a locked state (unable to train and improve) owing to the regulatory requirements.

4.3 Suggestions on utilizing DevOps in MDSW (RQ2)

Suggestions for utilizing the DevOps approach to improve MDSW development can be divided into the following topics: standards-based approach, practices, tools, third-party platforms, organizational culture, and MLOps.

4.3.1 Standards-based approach

The standard-based approach looks at which clauses in relevant standards can be fulfilled using the DevOps approach. This approach has been suggested in sources S63, S146, and G71. Following international standards in MDSW development is considered best practice, and the most notable standards affecting the use of the DevOps approach are IEC 62304 and IEC 82304-1 [S63, S146, G71]. S63 utilizes SWMA to fulfill several clauses of IEC 62304, and S146 and G71 comprise a regulatory CI/CD pipeline that implements 46 regulatory requirements from both notable standards, either fully or partially. These clauses are presented in Table 4.3.

IEC 62304 clause 5 describes the software development process, clause 6 describes the software maintenance process, clause 7 describes the software risk management process, clause 8 describes the software configuration management process, and clause 9 software problem-resolution process [23]. Clauses 1–4 concern general topics considered irrelevant to the DevOps approach [34].

- In IEC 62304 clause 5, S63 finds four specific sub-clauses for DevOps-based SWMA. Clauses 5.1.1 Software development plan and 5.1.9 Software configuration management planning are handled by a QMS based on the ISO 13485 standard. The compliance with clauses 5.6.8 Use software problem resolution process and 5.7.2 Use software problem resolution process (these clauses are named the same) is achieved by the software problem resolution process based on the Jira tool. The regulatory CI/CD pipeline fully implements 17 sub-clauses and partially 13 sub-clauses from clause 5. The criteria for the requirements are that activities must be fulfilled after the code has been checked into the version control system, and before the software is released [G71]. Some requirements require specific tooling or manual work in the pipeline. However, documentation, verification, and testing can be automated at some level.

Table 4.3: Summary of the IEC 62304 and IEC 82304-1 clauses implemented in SWMA solution or regulatory CI/CD pipeline [S63, S146, G71].

Clause	Title	In SWMA	In regulatory pipeline (partially implemented in italic)
IEC 62304 5	Software development process	5.1.1, 5.1.9, 5.6.8, 5.7.2	5.3.6, 5.4.4, 5.5.5, 5.6.1-5.6.7, 5.7.4, 5.7.5, 5.8.1, 5.8.3, 5.8.4, 5.8.6, 5.8.7, *5.1.12, 5.3.1-5.3.4, 5.4.1, 5.5.2, 5.5.3, 5.7.1, 5.7.3, 5.8.2, 5.8.5, 5.8.8*
IEC 62304 6	Software maintenance process	6.1, 6.2.2, 6.2.3, 6.2.4, 6.2.5, 6.3.1, 6.3.2	*6.3.2*
IEC 62304 7	Software risk management process	N/A	7.3.1, 7.3.3, *7.1.3, 7.4.3*
IEC 62304 8	Software configuration management process	8.2.1, 8.2.2, 8.2.3, 8.2.4	8.1.2, 8.1.3, *8.2.3*
IEC 62304 9	Software problem resolution process	9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8	9.8
IEC 82304-1 4	Health software product requirements	N/A	*4.4, 4.7*
IEC 82304-1 6	Health software product validation	N/A	6.2, 6.3
IEC 82304-1 7	Health software identification and accompanying documents	N/A	7.1, *7.2*
IEC 82304-1 8	Post-market activities for health software	N/A	8.3

- Conformity with clause 6 can be achieved using SWMA [S63]. Each raised problem was reported to the responsible person in the maintenance process, and an appropriate change request was initiated. However, modifications and fixes to the problem must be released separately, either as part of a full re-release or as a modification kit, when all tasks are completed. The regulatory CI/CD pipeline implicitly implements the requirements of clause 6.3.2 [G71].

- The regulatory CI/CD pipeline verifies the risk control measures (clause 7.3.1) and documents traceability requirements by using the version control system (7.3.3) [G71]. Evaluating SOUPs (7.1.3) and performing risk management activities (7.4.3) requires human input.

- Clause 8 requires the management of configuration items. The pipeline automatically identifies the SOUPs and system configuration documentation (clauses 8.1.2 and 8.1.3) [G71]. The SWMA solution focuses on the change request process [S63], change requests undergo proper phases and activities and are traced, complying with clause 8.2.

- SWMA achieves conformity with clause 9 by collecting detected problems, guiding their solution with workflow, and keeping track of all items and documentation [S63]. The pipeline runs automated tests and provides test documentation (clause 9.8) [G71].

The relevant IEC 82304-1 clauses are 4, 6, 7, and 8.

- IEC 82304-1 clause 4 concerns product requirements taken care of outside the regulatory CI/CD pipeline, but they might need to be updated during development (clauses 4.4 and 4.7) [G71].

- Software validation is guided by Clause 6. Clause 8 contains the requirements for revalidation. The pipeline contains a Deployment Pipeline stage in which the software is deployed into a specific environment and is released to the end-users only by human decision (clauses 6.2, 6.3, 8.3) [S146].

In addition to standards, guidance documentation provided by authorities, such as the Notified Bodies Operations group best practice guidance, should be followed [S168]. S210 argues that standards and regulations are lacking when it comes to DevOps, and suggests that not only should development comply with prevailing requirements, but requirements should also be accommodated to take into account modern software development approaches.

4.3.2 DevOps practices

The most common DevOps practices in the sources were Compliance as Code, CI/CD, and Continuous Deployment. This is presented in the following sections. Scaled agile framework (SAFe) [G5], SCRUM techniques as a replacement for certain regulatory activities [G32], architecture-focused approach [S218], post mortem [G66], and hybrid approach with agile accommodated waterfall and V-models [S207] were practices suggested by single sources.

Compliance as Code

The idea of *something as a Code* was repeated in the sources. G71 discusses *Infrastructure as Code* (IaC) and *Configuration as Code* which are important in the regulatory CI/CD pipeline suggested by the source. Adopting *Documentation as Code* practice has been suggested directly [G66, S81, S82], and implicitly as a proposed tool feature [S63, S64, S146, S152, S170, S189, S210, S218]. Documentation requirements are considered exhaustive, and the role of documentation is more important than in regular software development. Creating and updating documentation automatically is the most proposed single topic among sources.

S82 goes further and discusses *(Medical) Compliance as Code.* Compliance should be integrated into the MDSW development life cycle by applying software development principles to the compliance requirements [S82]. In addition to automated documentation, an example scenario involves handling third-party software (SOUPs). Adding SOUP to the product requires risk analysis, and in *Compliance as Code* mentality, the tools notice new SOUP and manage risk analysis via automatic workflow. Regulatory experts are integrated into the same pipelines as developers, and compliance tasks are handled seamlessly as the software development itself.

Continuous Integration, Continuous Delivery, and Continuous Deployment

The practice of continuous software development has been discussed in several sources [S64, S81, S146, S152, S170, S189, S209, S210, S218, G5, G71, G76]. Many researchers and practitioners have suggested Continuous Integration [S81, S146, S170, S189, S209, S210, G5, G71]. There are no hindrances in the regulations for practicing CI, and it is required practice for the automated documentation presented in the previous section. Demands for

CI in MDSW development are broader than in regular development, consisting of SOUP analysis [S146, S170, S209, G71] and documentation creation [S81, S146, S170, S189, S210, S218, G71] in addition to regular CI pipeline tasks, such as automated testing.

As a next step following CI, Continuous Delivery has been suggested by several sources [S81, S146, S189, S210, S218, G5, G71, G76]. There are additional requirements for CD in MDSW development. To achieve a deployable state, the software must be assessed and approved as required in the regulations [S189, S210, G71]. In addition, a human should make the final deployment decision, therefore the most convenient way is to keep the CD internal only [S210, G71].

Continuous Deployment is recommended for use in test environments [S146, G71]. It cannot be practiced in a production environment, because changes to MDSW require approval from the Notified Body. In practice, it is impossible to build a pipeline for Continuous Deployment [S146, S168, G71].

Other continuous practices are also observed. Company Siemens uses *Continuous Planning*, which includes activities for roadmaps, portfolio Kanban, backlog, and a concept "software as a program" instead of a project [S218]. S64 proposed *a Continuous Design Control* to improve regulatory design control activities in a machine learning environment.

4.3.3 DevOps tools

Tools Jira and Confluence from the company Atlassian are suggested to be used to inform all relevant parties in stages of the MDSW life cycle [S63]. G5 suggests them too. In addition, application life cycle management tools (ALM), such as Polarion, can be utilized. Jira allows for measuring and improving processes, and these tools can be integrated into other platforms [S63, G5].

A common version control system is adopted in DevOps approach suggestions, in most cases, the proposed system is GitHub [S81, S82, S152, S170, S209, G5, G66]. The suggested GitHub features are pull requests [S81, S82, S170] and issues [S152]. G5 discusses the use of GitHub extensions. The CompliancePal tool utilizes and extends GitHub features [S170]. S152 and S170 present it, and S81 suggests the CompliancePal to reach *the Calm Compliance* (see 4.3.5).

S219 suggests installing virtual machines that should be used for all workstations to ensure that development is carried out in a similar environment. G5 discussed containerization and microservices without a clear action plan.

S210 lists four important features of DevOps tools in the MDSW domain: item tracking across tools, standard templates that comply with regulations, hierarchical tools, and guiding the developer to follow the workflow in a compliant manner. S209 added that tools should be able to identify risks for every code commit. The wheel may not be reinvented, S170 suggests leveraging application programming interfaces (API) of existing DevOps tools to complement their functionality for compliance.

4.3.4 Third party platforms

The sources provided mixed signals about third-party platforms (in practice, cloud platforms). Utilizing cloud platforms is seen as very challenging due to regulations [S168, S217], but G15 proposes them without much comments on compliance aspects. On the other hand, third-party cloud platform providers, such as Amazon and Microsoft, respect regulatory requirements [S168, G15]. As long as the shared responsibility model is understood, G15 suggests that cloud platforms could be used, and proposes implementing automated tools, reusing established architectural patterns, and continuous monitoring within them.

However, S168 and S217 point out that regulations do not free the manufacturer from responsibility when relying on third-party platforms. Although cloud platforms and IaC practice are popular in software development, clinical evaluation containing technical validation must be conducted in the MDSW domain. In practice, this leads to a situation in which the manufacturer must provide detailed technical information about the third-party cloud platform and ensure that the platform remains unchanged after the validation of the MDSW [S168]. Another option would be to define the cloud platform as a SOUP. However, this requires the manufacturer to document evidence for SOUP requirements and is similarly challenging to technical validation.

4.3.5 Organizational culture

Approaches concerning organizational culture and mindset have been suggested among DevOps techniques and tools [S81, S153, S168, S219, G25, G31].

S81 introduces the concept of *Calm Compliance*. This source suggests that regulatory processes should be streamlined. Thus, the balance between regulatory requirements and regular software development must be understood. Not all modules in the software usually manage medical purposes. They can be separated from medical modules and left out of medical regulations. Another suggestion is that compliance must be viewed as everyone's business. Whenever a new change to the code is committed, regulatory activities should be invoked and addressed. In other words, compliance officers' tasks should be aligned with those of the developers.

Solita implemented the DevOps approach for MDSW development [S153, G25, G31]. The *RegProof method* is illustrated in Figure 4.5. Daily agile software development tasks are performed within workflows containing automated processes to manage compliance [G25, G31]. However, unlike in regular DevOps, releases are performed several times per year and not continuously. They are building on top of the Calm Compliance, automating everything possible, and adopting the streamlined regulatory processes presented in S81. This method is utilized in the Oravizio product (see 4.3.6).

S168 indicates that authorities and Notified Bodies play an imperative role in the MDSW life cycle. The Design Change Approval Process assumes that a Notified Body approves the product, and it forms a huge barrier to Continuous Deployment. Therefore, the availability of resources and services of Notified Bodies is a critical environmental property that MDSW manufacturers find difficult to improve. A full DevOps approach would require cultural changes in authority organizations. S219 acknowledges that, in some cases, a partial DevOps approach is the only option, and the source proposes DevOps assessment and benchmarking to determine how DevOps can be utilized.

4.3.6 MLOps

MLOps is the suggested approach for AI/ML MDSW [S139, S143, S153]. Regulations do not address AI/ML properly and manufacturers must deploy products in a "locked state" in which ML training is disabled [S168]. AI/ML-based systems can also be interpreted as risk Class III devices, bringing more requirements to the manufacturer.

Despite this, Solita developed the product Oravizio using the DevOps approach as much as possible [S64]. S64 suggests implementing a continuous training pipeline and utilizing ML model cards to manage design control documentation. Continuous Delivery for Machine Learning (CD4ML) is the proposed MLOps implementation, which aims to produce an ML application in small and safe increments. These increments can be reproduced and released in short adaptation cycles, at any given time. S143 discusses extending SOUP to ML models to guide their regulatory activities.

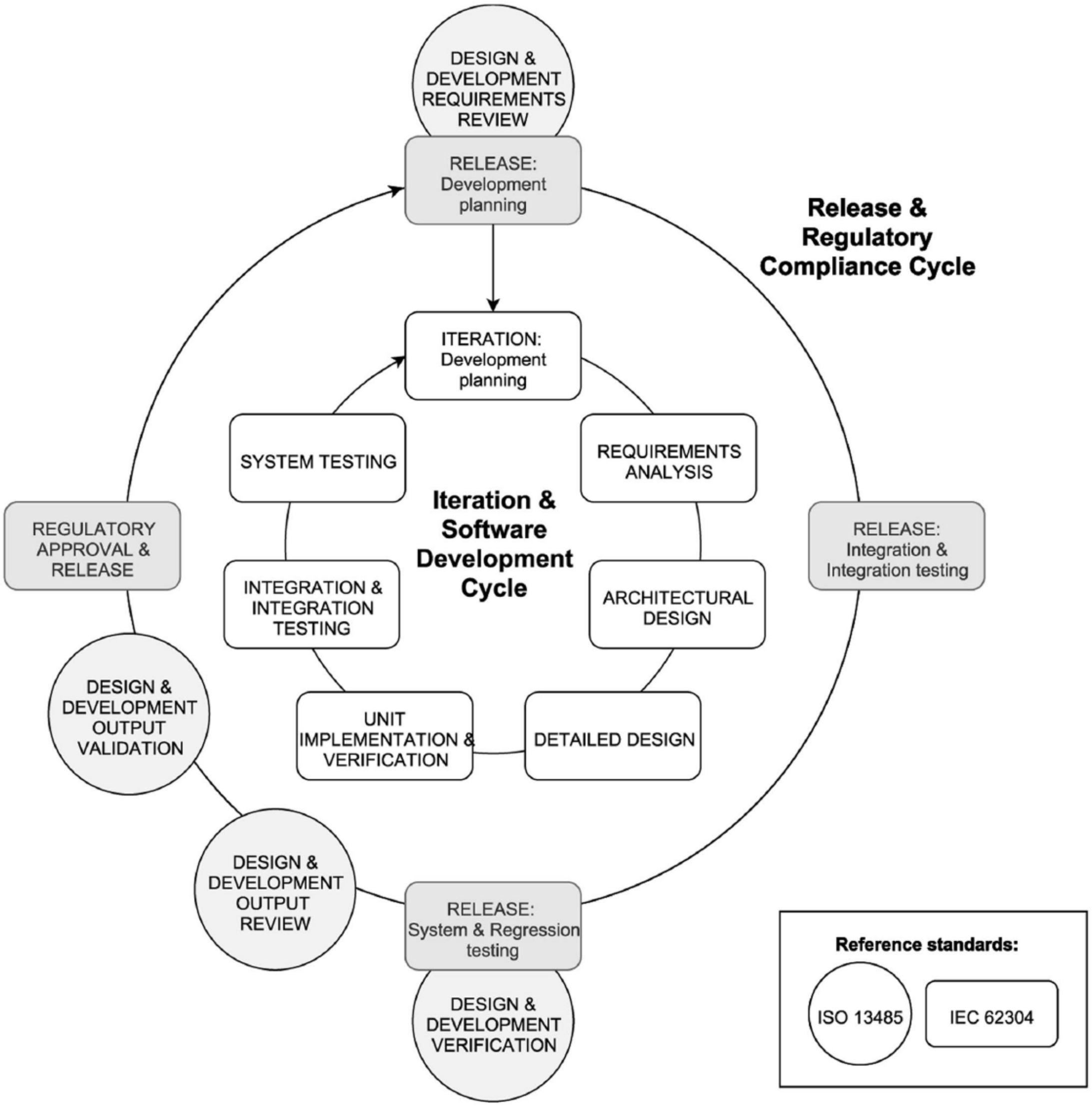

DESIGN & DEVELOPMENT REQUIREMENTS REVIEW
RELEASE: Development planning
Release & Regulatory Compliance Cycle
ITERATION: Development planning
SYSTEM TESTING
REQUIREMENTS ANALYSIS
Iteration & Software Development Cycle
REGULATORY APPROVAL & RELEASE
RELEASE: Integration & Integration testing
INTEGRATION & INTEGRATION TESTING
ARCHITECTURAL DESIGN
DESIGN & DEVELOPMENT OUTPUT VALIDATION
UNIT IMPLEMENTATION & VERIFICATION
DETAILED DESIGN
DESIGN & DEVELOPMENT OUTPUT REVIEW
RELEASE: System & Regression testing
DESIGN & DEVELOPMENT VERIFICATION
Reference standards:
ISO 13485
IEC 62304

Figure 4.5: Regulatory-compliant MDSW development process by Solita [S153].

4.4 Challenges and benefits of DevOps in MDSW (RQ3)

The challenges and benefits of the DevOps approach are grouped according to the type of challenge/benefit. Table 4.4 presents the number of challenges and benefits and a number of their sources.

Table 4.4: Number of challenges and benefits discussed by sources.

Type	Challenges	Sources	Benefits	Sources
Compliance	19	14	4	5
Development process	5	4	8	16
Organizational	8	15	6	11
AI/ML	6	6	1	1

4.4.1 Compliance

Compliance requirements pose the greatest challenges to the DevOps approach in MDSW development. Many legal and regulatory bindings contradict DevOps practices [S152, G64, G71].

The design change approval process requires approval from a Notified Body whenever the application operation changes [S168]. This forms a barrier to continuous deployment, maintenance, and agile development [S168, G5, G31, G71]. Notified Bodies need time to certify the application [S153], so any significant changes cannot be made in an automated and fast manner.

SOUP modules make MDSW more complex from a regulatory viewpoint, although it is common practice in modern software development to utilize common libraries and modules [S143]. Manufacturers must specify and evaluate all used SOUPs and keep a record of them [S143, S146, S168, G71]. This is challenging because SOUP might be undocumented and decades old. Similar requirements concern third-party cloud platforms, manufacturers should conduct technical validation in a platform where they typically have no control [S168, G15, G71]. Data location and retention times may also pose challenges if manufacturers do not have control over them.

Information handling concerns medical device manufacturers in the form of HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation) [S170, S209]. Most devices collect data from users, and in the MDSW domain, the data is usually sensitive patient data. Therefore, manufacturers must establish procedures to handle the data. Patient data may not be used in a staging environment, thus creating an extra layer of complexity in the software development process [S146].

These challenges have been discussed in several sources. In addition, the following challenges were identified from single sources.

- The documentation required is comprehensive and more burdensome than in a non-regulated domain [G71].
- Some compliance activities cannot be automated but require human decision [G31].
- The concept of a *single-fault condition* in regulations is based on physical devices and is challenging to adopt in the software context [S168].
- Patient safety security, data integrity, risk management, and clinical effectiveness are non-negotiable, leaving no room for flexibility [G25].
- Hardly any software-only product is below Class IIa. Therefore, a quality management system must be in place [S168].
- Standalone and embedded software are concerned the same in regulations [S168].
- Post-market surveillance is required [S153].
- Standard harmonization to the regulations is lacking [S217].

Although compliance requirements pose many challenges to the DevOps approach, there are some benefits. Documentation requirements can be alleviated through automated documentation creation and updating [G5, G66]. Integrating documentation into the software development process also establishes a single source of truth for documentation [S63]. G71 and S170 mentioned that the traceability and transparency of processes are core principles of DevOps, which may help comply with regulations. S63 states that the DevOps approach made it easy to follow standards.

4.4.2 Software development process

The main benefit of the DevOps approach for MDSW development is its faster development cycle and improved software quality. DevOps practices enable a fast feedback cycle, which reduces development time and cost [S63, G5], and shortens time-to-market [G15]. Near-to-instant feedback from automated tests is the most significant factor, as it leads to quick reactions to bugs and compliance issues [S139, S146, S153, S170, S189, G27, G64, G71], and keeps developers focused on the recent commit [G76]. Automation also improves the development efficiency [S146, S189, G5]. S209 discusses satisfying customer needs faster, but S210 notifies that rapid customer feedback is not always applicable within MDSW.

A faster development cycle is related to improved software quality. Discovering compliance issues early means that the changes can be made when they are easier and quicker [G25, G27, G76]. DevOps practices may also improve code quality and secure coding [S63, G64, G76], and reduce the number of defects [S207] and user errors by normalizing the deployment process [G71]. G66 and G71 stated that automating trivial tasks enables employees' limited mental capacity to focus on non-trivial tasks.

The challenges found in the sources are related to the "vagueness" of DevOps. Documentation or process control are not mentioned in regular DevOps cycles, and there is no easy way to know what tools are suitable for the organization or environment [G5]. S170 warns not to interpret the agile manifesto clause "working software over comprehensive documentation" too strictly. DevOps assets, such as centralized document repositories, may be missing in established organizations, and implementing them is not straightforward [S220].

4.4.3 Organizational culture

The main benefit of DevOps for medical device manufacturers' organizational culture is bringing developers and regulatory experts together. DevOps can alleviate the disconnection between the compliance team and developers [S63, S189, G25], and set compliance as a shared goal [S170, S209]. Thus, developers find it less challenging to understand and cope with regulations [G25, G64, S81]. They can be confident that compliance problems are detected and handled, and can use DevOps patterns they are familiar with [S170, S209, S210]. Common tooling for developers and regulatory officers has further improved this [S152]. The integrated software development cycle provides feedback for all stakeholders at all stages [S64]. The downside is that a multi-disciplinary team is required [S81] and cross-domain terminology poses a risk of misunderstanding [S64, G25].

Challenges lie in attitudes, resources, and complex systems. The DevOps approach requires attitude change, collaborative culture is needed and non-developers must enter the development pipeline [S82, S189]. Established organizations might hesitate to implement DevOps [S81, S152, G5, G71]. Some may see conventional up-front planning as safer in the MDSW domain [S207, S210]. There is a risk that the compliance activities and the realities of the software implementation will collide [S209, G5, G27]. G66 warns that developers may not value the compliance aspects integrated into development processes. G71 notifies that customers might not want continuous updates to the software.

DevOps tools require resources that some organizations lack [S189, S220]. In addition, tools for compliance may be incompatible with DevOps tools [G66]. The DevOps approach may be difficult to adopt in large, complex systems [S218, G27]. DevOps may grow product and organization complexity, and in large, embedded software systems, parts of the product move at different "paces of change". However, G76 states that DevOps improves team velocity and S153 suggests that DevOps practices bring rigor to development.

4.4.4 AI/ML

Several challenges related to the MLOps approach were found in the sources. The effectiveness of the AI/ML device is difficult to prove because the functionality of the device should remain the same throughout the life cycle [S64, G5]. Legislation has shortcomings in ML medical devices [S143, S153], which has led to deploying ML systems in a locked state [S139, S153, S168]. Data engineers, data scientists, and software developers are needed, and their skill sets may vary [S153]. Data availability [S139] and personal data handling [S143] pose significant challenges. MLOps practice of continuous design control can help mitigate the design problems [S64].

5 Discussion

The discussion section is divided into subsections according to the research questions and step-by-step guidance.

5.1 The state-of-the-art of DevOps in MDSW (RQ1)

The research on DevOps in the MDSW domain is limited. In particular, sources discussing the state-of-the-art are difficult to obtain. Only half of the sources mentioned how DevOps is currently being used in MDSW development. Many scientific sources described pilot projects or small organizations and practices that had been in use for a short period. Instead, most of the GL did not go much into the details and vaguely discussed practices. Both formal and gray literature focused on what could and should be done. Only two significant organizations were mentioned as having already adopted the DevOps approach: Siemens and Solita.

The lack of a proper presentation of current DevOps practices suggests that most medical device manufacturers do not utilize DevOps. This was expected because the common claim is that regulated software development requires a waterfall approach [34]. The results also indicate that established organizations are hesitant or slow to adopt DevOps. In addition, physical device manufacturing cannot be iterative, therefore, embedded software might be developed with the same up-front planning process as the device.

However, the results showed that the state-of-the-art consists of Continuous Integration among automated testing, Compliance as Code, partially automated documentation creation, and DevOps tools (the most common being Jira and GitHub). Solita's RegProof approach implementing all practices, can be seen as a forerunner. CI and automated testing are easy steps to implement as there are no regulatory requirements to consider, and they are the basics of modern software development.

The terminology used varied between sources. DevOps was not always explicitly mentioned in the sources, although the adopted practice was commonly associated with DevOps. Many sources have focused on a single practice, so we assumed that the authors did not see it as relevant to connect it to DevOps. In addition to DevOps, continuous development and agile methods are used as umbrella terms.

5.2 Implementing the DevOps within MDSW (RQ2)

The results indicate that within MDSW, DevOps has expanded from combining development and operations to combining software-related tasks with regulatory activities. The term RegOps was used to describe this in a few sources. Regulations require special knowledge, similar to high-level software development. It is not feasible to demand that developers acquire regulation skills, or compliance officers acquire development skills. Instead, seamless collaboration between them is crucial. Thus, developers' regulatory burden is alleviated and compliance officers' understanding of the software is improved.

Compliance as Code was the proposed methodology to be adopted to achieve this. Compliance can be integrated into the software development process by adopting DevOps practices, such as CI/CD pipelines and suitable tools, and adding compliance tasks to the pipeline. Tool CompliancePal is an example of this approach. Every code commit initiates compliance officer activities automatically. Smarter tools tailored to MDSW requirements are needed. The tools could follow predetermined workflows guiding developers, and create as much documentation and regulatory material as possible. Existing DevOps tools offer APIs that can be used to customize them to meet MDSW needs.

Interpreting the agile manifesto clause "working software over comprehensive documentation" too strictly was noted in the results. In the MDSW domain, an agile mindset requires adjustments. The main goal is to provide value to the customers. In a regular setting, comprehensive documentation is not valuable to the customer, but the working software is valuable. However, with medical devices, the software cannot be deployed and marketed without sufficiently comprehensive documentation to comply with the regulations. Thus, documentation becomes valuable, and providing good documentation creates (indirect) value for the customer.

Everything cannot be automated in MDSW development. Certain decisions must be human-made and design changes must be approved by the Notified Body. This means that proper Continuous Deployment is impossible. It can be done in an internal testing environment, but not in production. However, as indicated in the results, continuously updating the software might not be the end user's interest. Disrupting the use of software for updates may be challenging for devices used in hospital or laboratory settings. Also, many devices simply do not require frequent updates. Understanding DevOps practices that do not fit well into the development process is imperative.

5.3 Challenges and benefits of DevOps in MDSW (RQ3)

Entry into the market is usually slow for medical devices because of several compliance requirements. The device must impact health issues, and the manufacturer must provide evidence. Standalone software is becoming the more prevalent type of medical device, and the same requirements are applied. DevOps, popular among software development organizations, can reduce release times and complexity. Product quality could improve because of a faster feedback loop, as the errors and bugs will be fixed faster. The medical device industry is a specialized field, and DevOps can help developers without prior knowledge to work in the field.

Continuous Integration combined with collaboration between developers and regulatory experts, and common tooling and patterns for both parties improve the visibility and understanding of the development process and its status. Regulatory experts can intervene early if they notice non-compliant software changes. Developers, on the other hand, are confident that they will get fast feedback and guidance on how to ensure compliant development. No time is lost making changes to long-developed features that turned out to be non-compliant. Additionally, automated documentation creation reduces compliance burden. This results in higher quality and faster development.

However, failure is not an option in the MDSW domain. Medical devices cannot be shipped with the mindset that bugs can be fixed on the next release if they can cause harm to the patient. The more physical the device and the more risks involved, the lower the possibility of implementing a full DevOps cycle.

The results showed factors affecting DevOps practices, that were not controllable by manufacturers. Utilizing third-party cloud platforms and IaC is challenging because the manufacturer is responsible for showing evidence that the platform is compliant. This is a difficult task with modern third-party cloud providers. However, cloud operators are already involved in mission-critical sectors, such as finance, and they would be open to collaboration, should the regulations be adjusted. Technology and software development practices have evolved rapidly, and legislation, guidance documentation, standards, and authority resources have not been maintained at every level. For example, the results indicate that AI/ML is a blind spot in regulations, and most of the benefits of AI/ML systems cannot be achieved with current regulations.

5.4 Step-by-step guidance on adopting DevOps in MDSW domain

Based on the results and summary of related work [12], we propose the following steps to successfully implement the DevOps approach in MDSW development.

1. Think about your environment, software product, and organization.
   - Does the DevOps approach bring benefit to you?
   - What is your device like? With absolutely safety-critical software, the waterfall model or V-model may be better suited. DevOps may be the best approach for standalone software.
   - What is your organization like? Is there a mindset for DevOps? Do you have the resources to establish DevOps practices and let employees familiarize themselves with them?
2. If possible, introduce regulators and auditors as stakeholders in the release cycle. This should ensure that no compliance surprises occur.
3. Review the QMS system (or establish it). Does it require adjustments to enable DevOps practices?
4. Follow standards. Ensure that you comply with all the clauses that concern your product. Specific DevOps practices must not be enforced if compliance challenges arise.
5. Establish DevOps tooling for issue tracking. Ensure that a full audit trail is available.
6. Create a CI/CD pipeline and follow Compliance as Code.
   - Continuous Integration should be easy to implement. Continuous Delivery can be performed partially or only in the testing environment.
   - Compliance activities must be triggered on every code commit.

- Documentation must be created as automatically as possible. Use tools and formats that both developers and compliance officers can understand.

7. Focus on a faster development cycle and update compliance status whenever code changes.

8. Emphasize collaboration between developers and compliance officers.

5.5 Limitations and validity

This study maintains its validity through several means. A multivocal literature review is beneficial on the topic heavily linked to practice as the latest knowledge is most likely found on GL. The method was chosen after considering justification in established guidelines [13]. The study process followed the guidelines for GL and formal literature [32], and the methodology of previous research. The search process was conducted systematically and the study process was thoroughly documented.

Interpretive methods, such as applying inclusion/exclusion criteria, assessing quality, and extracting data, should be conducted individually by two or more researchers [32]. This was not possible in this study, forming a major limitation and threat to validity. Other limitations include the potential inadequacy of the search process and the scarcity of relevant data. This topic has not been widely studied, and the research is focused mainly on a few researchers. This poses the risk that the results become one-sided.

5.5.1 Limitations of the source selection process

A quasi-gold standard, iterative trial search process, and consultation with the supervisor were used to ensure that the search phrase found a comprehensive set of sources.

During the trial searches, the names of the standards were not beneficial in the search phrase, they did not provide more results. It seems that a single standard is rarely a keyword. The same applied to the names of regulations and authorities, terms like "eu mdr" or "fda qms" did not provide more results. The term "software" yielded too many results, but when combined with "medical" to form "medical software", it provided results that could not be found using just "medical device". The term "healthcare" increased the results excessively, and it brought many results related to healthcare information systems that are not within the scope of this thesis. Terms "development" and "environment" used in [12] expanded the search much without bringing in relevant results. However, the terms "regulation" and "regulated", the latter used in [12], dropped unnecessary results. The term "regops" was noticed to be impactful in preparatory work, especially in GL. In addition, we noticed that a plural impacted the search, most likely due to quotation marks for exact terms (where wildcards are not available).

Although the search phrase was iterated multiple times, it is possible that it could not capture all relevant sources. Terms "devops" and "regops" were required, and even some selected sources did not emphasize them, but used terms such as "continuous development" and "agile methods". This poses the risk that some relevant sources may be missed. However, snowballing was performed for the selected sources to alleviate this risk. Thus, we note that source selection was performed sufficiently.

5.5.2 Limitations in data extraction

The data extraction form was created according to the guidelines [13], and the extraction process was documented. There are two major limitations to this process. First, quality assessment and extraction of relevant information were more or less subjective, especially within GL, as there was no established structure of the scientific literature. Second, the process had a single author, and it was not feasible to replicate or cross-reference data extraction. This leaves a risk of researcher bias.

However, the sources were read multiple times during the study process. They were skimmed when applying the inclusion/exclusion criteria, then read through for quality assessment, and again for data extraction. In addition, the sources were returned multiple times during the writing process. This is in line with [32] suggestion of a re-test process for a single researcher.

6 Conclusions

In this thesis, DevOps in MDSW development was studied by conducting a multivocal literature review. The research problem was how DevOps could benefit regulated medical device software development. The state-of-the-art and ways of adopting DevOps in an MDSW setting were studied to answer this. Step-by-step guidance on applying DevOps practices was also proposed, and these combined offer support to practitioners and researchers.

Regarding the state-of-the-art (RQ1), the results indicate that the DevOps approach is not widely used in the MDSW domain. DevOps is mostly adopted in pilot projects or small organizations. However, two companies, Solita and Siemens were forerunners in adopting DevOps practices and mindset for MDSW development. The state-of-the-art consists of Continuous Integration with automated testing and partially automated documentation creation, DevOps tools such as issue trackers, adapted application life cycle management tools, and Compliance as Code mindset.

Several suggestions were found (RQ2). The most significant suggestion was to adopt a Compliance as Code mindset to integrate compliance activities into the software development process. This could be achieved by CI/CD pipeline with human decisions behind certain delivery activities. Documentation creation could be automated, and code commit could trigger compliance tasks. Leveraging the existing DevOps tools is suggested to accommodate them to support regulatory requirements. Calm Compliance emphasizes seamless collaboration between developers and regulatory experts, sometimes called RegOps. In addition, the results highlighted that following standards is recommended despite the development methods.

Regarding the challenges and benefits (RQ3), the results showed that most challenges were related to regulatory requirements. Continuous Deployment in a production environment is impossible due to the change design approval process, where a Notified Body must approve significant changes to the software before launching. Additionally, regulatory requirements are troublesome for third-party cloud platforms and AI/ML solutions. The DevOps approach could improve software quality and enable faster entry into the market. Collaboration between developers and regulatory experts could strengthen the confidence of both parties, thus leading to an improved development process.

Step-by-step guidance (see 5.4) proposes steps based on the results. The environment should be assessed and regulators should be introduced as stakeholders. The QMS system needs to be reviewed and standards should be followed. DevOps practices start with establishing tooling and pipelines, followed by integrating compliance into the development process and emphasizing collaboration.

DevOps benefits MDSW development by shortening time to market, improving software quality, streamlining and integrating compliance activities into the development process, and fostering an organizational culture that increases confidence.

Future research

Information on this topic is scarce, and more research is needed. The DevOps benefits for MDSW development could be supported by research comparing devices developed using the DevOps approach and conventional methods. One possible source of information could be the databases of device-related incidents. The experiences of experts working with MDSW could be gathered to determine how developers see collaboration with compliance officers, and especially, how compliance officers feel about being integrated into the development process. Choosing the correct DevOps tools for MDSW development may be challenging, and more research on these tools could alleviate this.

Regulations and standards determine many activities in the development of MDSW. The results showed that modern technologies and software development methodologies are not always considered in regulations, standards, and authority processes. Research could find and propose sensible adjustments for authorities to better support modern software development in the medical device domain.

Bibliography

[1] Amazon Web Services. *What is DevSecOps?* URL: https://aws.amazon.com/what-is/devsecops/ (visited on 23.4.2024).

[2] L. Bass, I. Weber, and L. Zhu. *DevOps: A software architect's perspective.* Addison-Wesley Professional, 2015.

[3] "COUNCIL DIRECTIVE 93/42/EEC of 14 June 1993 concerning medical devices". In: *OJ* L 169 (12.7.1993), pp. 1–43.

[4] M. Courtemanche, E. Mell, and A. S. Gills. *What Is DevOps? The Ultimate Guide.* 2021. URL: https://www.techtarget.com/searchitoperations/definition/DevOps (visited on 20.10.2023).

[5] DevOps Research and Assessments, Google Cloud. *Accelerate State of DevOps.* Tech. rep. 2023-12.

[6] "Directive 98/79/EC of the European Parliament and of the Council of 27 October 1998 on in vitro diagnostic medical devices". In: *OJ* L 331 (7.12.1998), pp. 1–37.

[7] T. Dybå and T. Dingsøyr. "Strength of evidence in systematic reviews in software engineering". In: *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement.* 2008, pp. 178–187.

[8] European Commission. *EUDAMED - European Database on Medical Devices.* URL: https://ec.europa.eu/tools/eudamed/#/screen/home (visited on 8.3.2024).

[9] European Commission. *Types of EU law.* URL: https://commission.europa.eu/law/law-making-process/types-eu-law_en (visited on 8.3.2024).

[10] European Medicine Agency. *Medical devices.* 2024. URL: https://www.ema.europa.eu/en/human-regulatory-overview/medical-devices (visited on 6.2.2024).

[11] European Union. *CE marking.* URL: https://europa.eu/youreurope/business/product-requirements/labels-markings/ce-marking/index_en.htm (visited on 8.3.2024).

[12] M. Forsberg Lie, M. Sánchez-Gordòn, and R. Colomo-Palacios. “DevOps in an ISO 13485 Regulated Environment: A Multivocal Literature Review”. In: *ESEM ’20: Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement’*. IEEE. 2020, pp. 1–11.

[13] V. Garousi, M. Felderer, and M. V. Mäntylä. “Guidelines for including grey literature and conducting multivocal literature reviews in software engineering”. In: *Information and software technology* 106 (2019), pp. 101–121.

[14] V. Garousi and M. V. Mäntylä. “When and what to automate in software testing? A multi-vocal literature review”. In: *Information and Software Technology* 76 (2016), pp. 92–117.

[15] T. Granlund. *RegOps – diving into the dilemma of agile software development in regulated industry*. 2020. URL: https://www.solita.fi/blogs/regops-diving-into-the-dilemma-of-agile-software-development-in-regulated-industry/ (visited on 27.10.2023).

[16] T. Granlund, A. Kopponen, V. Stirbu, L. Myllyaho, and T. Mikkonen. “MLOps challenges in multi-organization setup: Experiences from two real-world cases”. In: *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*. IEEE. 2021, pp. 82–88.

[17] T. Granlund, T. Mikkonen, and V. Stirbu. “On Medical Device Software CE Compliance and Conformity Assessment”. In: *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE. 2020, pp. 185–191.

[18] T. Granlund, T. Mikkonen, and V. Stirbu. “On Medical Device Software CE Compliance and Conformity Assessment”. In: *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE. 2020, pp. 185–191.

[19] T. Granlund, V. Stirbu, and T. Mikkonen. “Medical software needs calm compliance”. In: *IEEE Software* 39.1 (2021), pp. 19–28.

[20] T. Granlund, J. Vedenpää, V. Stirbu, and T. Mikkonen. “On Medical Device Cybersecurity Compliance in EU”. In: *2021 IEEE/ACM 3rd International Workshop on Software Engineering for Healthcare (SEH)*. IEEE. 2021, pp. 20–23.

[21] International Electrotechnical Commission. *Health software - part 1: General requirements for product safety*. IEC 82304-1:2012. Geneva, Switzerland, 2012.

[22] International Electrotechnical Commission. *International Standards (IS)*. URL: https://www.iec.ch/publications/international-standards (visited on 5.3.2024).

[23] International Electrotechnical Commission. *Medical device software – software life cycle processes*. IEC 62304:2006. Geneva, Switzerland, 2006.

[24] International Electrotechnical Commission. *Medical devices Part 1: Application of usability engineering to medical devices*. IEC 62366-1:2015. Geneva, Switzerland, 2015.

[25] International Organization for Standardization. *Medical devices - Quality management systems - Requirements for regulatory purposes*. ISO 13485:2016. Geneva, Switzerland, 2016.

[26] International Organization for Standardization. *Medical devices Application of risk management to medical devices*. ISO 14971:2019. Geneva, Switzerland, 2019.

[27] International Organization for Standardization. *Standards*. 2024. URL: https://www.iso.org/standards.html (visited on 6.2.2024).

[28] M. Ivarsson and T. Gorschek. "A method for evaluating rigor and industrial relevance of technology evaluations". In: *Empirical Software Engineering* 16 (2011), pp. 365–395.

[29] I. Juuso. *Developing an ISO 13485-Certified Quality Management System: An Implementation Guide for the Medical-Device Industry*. CRC Press, 2022.

[30] W. Kelly. *Demystify the DevOps process, step by step*. URL: https://www.techtarget.com/searchitoperations/tip/Demystify-the-DevOps-process-step-by-step (visited on 12.3.2024).

[31] M. S. Khan, A. W. Khan, F. Khan, M. A. Khan, and T. K. Whangbo. "Critical challenges to adopt DevOps culture in software organizations: a systematic review". In: *IEEE Access* 10 (2022), pp. 14339–14349.

[32] B. Kitchenham and S. Charters. "Guidelines for performing Systematic Literature Reviews in Software Engineering". In: *EBSE Technical Report* EBSE-2007-01 (2007).

[33] H. Korhonen. *Lecture notes in Regulatory Essentials in Health Tech*. Oct. 2022.

[34] T. Laukkarinen, K. Kuusinen, and T. Mikkonen. "DevOps in Regulated Software Development: Case Medical Devices". In: *2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track*. IEEE. 2017, pp. 15–18.

[35] T. Laukkarinen, K. Kuusinen, and T. Mikkonen. "Regulated software meets DevOps". In: *Information and Software Technology* 97 (2018), pp. 176–178.

[36] L. E. Lwakatare, P. Kuvaja, and M. Oivo. “Relationship of devops to agile, lean and continuous deployment: A multivocal literature review study”. In: *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17*. Springer. 2016, pp. 399–415.

[37] Medical Device Coordination Group. *Guidance on Classification Rules for in vitro Diagnostic Medical Devices under Regulation (EU) 2017/746*. Tech. rep. 2023.

[38] Medical Device Coordination Group. *MDCG 2021-24 Guidance on classification of medical devices*. Tech. rep. 2021.

[39] U. J. Muehlematter, P. Daniore, and K. N. Vokinger. “Approval of artificial intelligence and machine learning-based medical devices in the USA and Europe (2015–20): a comparative analysis”. In: *The Lancet Digital Health* 3.3 (2021), e195–e203.

[40] *Paperpile: Modern reference and PDF management*. URL: https://paperpile.com/welcome (visited on 20.10.2023).

[41] H. Pitkänen, L. Raunio, I. Santavaara, and S. Tom. *European Medical Device Regulations MDR & IVDR – A Guide to Market*. Tech. rep. Business Finland, 2020. URL: https://www.leanentries.com/wp-content/uploads/european-medical-device-regulations-mdr-ivdr-a-guide-to-market.pdf (visited on 8.3.2024).

[42] “REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC”. In: *OJ* L 117 (5.5.2017), pp. 1–175.

[43] “REGULATION (EU) 2017/746 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 5 April 2017 on in vitro diagnostic medical devices and repealing Directive 98/79/EC and Commission Decision 2010/227/EU”. In: *OJ* L 117 (5.5.2017), pp. 176–332.

[44] “REGULATION (EU) 2023/607 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 15 March 2023 amending Regulations (EU) 2017/745 and (EU) 2017/746 as regards the transitional provisions for certain medical devices and in vitro diagnostic medical devices”. In: *OJ* L 80 (20.3.2023), pp. 24–29.

[45] M. Shahin, M. Ali Babar, and L. Zhu. “Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices”. In: *IEEE Access* 5 (2017), pp. 3909–3943. DOI: 10.1109/ACCESS.2017.2685629.

[46] V. Stirbu and T. Mikkonen. “Towards agile yet regulatory-compliant development of medical software”. In: *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE. 2018, pp. 337–340.

[47] H. Toivakka. *Integration of EU medical device regulatory requirements into a CI/CD pipeline*. Master’s thesis. University of Tampere, Finland, Sept. 2021.

[48] C. Wohlin. “Guidelines for snowballing in systematic literature studies and a replication in software engineering”. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10.

[49] H. Zhang, M. A. Babar, and P. Tell. “Identifying relevant studies in software engineering”. In: *Information and Software Technology* 53.6 (2011), pp. 625–637.

Appendix A Literature identified in the search process and their inclusion/exclusion criteria

Table A.1: List of studies and initial filtering step.

Study ID	Author(s)	Year	Title	Included	Reason
S1	Cederbladh J., Cicchetti A., Suryadevara J.	2024	Early Validation and Verification of System Behaviour in Model-based Systems Engineering: A Systematic Literature Review	No	I2
S2	-	2024	ISEC '24: Proceedings of the 17th Innovations in Software Engineering Conference	No	I2
S3	IEEE	2024	IEEE Draft Standard for Measures of the Software Aspects of Dependability	No	I2
S4	Pelliccione P., Laranjeiro N.	2024	Insights From the Software Reliability Research Community	No	I2
S5	Hafi H., Brik B., Frangoudis P. A., Ksentini A., Bagaa M.	2024	Split Federated Learning for 6G Enabled-Networks: Requirements, Challenges, and Future Directions	No	I2
S6	Schwarz M., Hinske L. C., Mansmann U., Albashiti F.	2024	Designing an ML Auditing Criteria Catalog as Starting Point for the Development of a Framework	No	I2
S7	Scriven A., Kedziora D. J., Musial K., Gabrys B.	2024	The Technological Emergence of AutoML: A Survey of Performant Software and Applications in the Context of Industry	No	E1
S8	Timinger H., Schmidtner M., Reiche F.	2024	A Framework for the Construction and Tailoring of Engineering Development Process Models	No	I2
S9	Gill S. S., Patros P., Ottaviani C., Arora P., Pujol V. C., Haunschild D., Parlikad A. K., Cetinkaya O., Lutfiyya H., Stankovski V., Li R., Ding Y., Qadir J., Abraham A., Ghosh S. K., Song H. H., Sakellariou R., Rana O., Rodrigues J. J., Kanhere S. S., Dustdar S., Uhlig S., Ramamohanarao K., Buyya R.	2024	Modern computing: Vision and challenges	No	I2
S10	Kara K., Yalcin G. C., Simic V., Önden I., Edinsel S., Bacanin N.	2024	A single-valued neutrosophic-based methodology for selecting warehouse management software in sustainable logistics systems	No	I2
S11	Terzi S., Stamelos I.	2024	Architectural solutions for improving transparency, data quality, and security in eHealth systems by designing and adding blockchain modules, while maintaining interoperability; the eHDSI network case	No	I2
S12	Ciancarini P., Giancarlo R., Grimaudo G.	2024	Digital Transformation in the Public Administrations: A Guided Tour for Computer Scientists	No	I2
S13	Lu Q., Zhu L., Xu X., Whittle J., Zowghi D., Jacquet A.	2023	Responsible AI Pattern Catalogue: A Collection of Best Practices for AI Governance and Engineering	No	I2
S14	-	2023	EuroUSEC '23: Proceedings of the 2023 European Symposium on Usable Security	No	I2
S15	-	2023	WSSE '23: Proceedings of the 2023 5th World Symposium on Software Engineering	No	I2
S16	-	2023	ESEC/FSE 2023: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering	No	I2
S17	-	2023	SCSW '23 Companion: Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing	No	I2
S18	-	2023	WSC '23: Proceedings of the Winter Simulation Conference	No	I2

Study ID	Author(s)	Year	Title	Included	Reason
S19	-	2023	ASIA CCS '23: Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security	No	I2
S20	-	2023	CSAI '23: Proceedings of the 2023 7th International Conference on Computer Science and Artificial Intelligence	No	I2
S21	-	2023	ICSIM '23: Proceedings of the 2023 6th International Conference on Software Engineering and Information Management	No	I2
S22	-	2023	ICAIL '23: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law	No	I2
S23	-	2023	MISNC '23: Proceedings of the 10th Multidisciplinary International Social Networks Conference	No	I2
S24	-	2023	EuroPLoP '23: Proceedings of the 28th European Conference on Pattern Languages of Programs	No	I2
S25	-	2023	SC-W '23: Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis	No	I2
S26	-	2023	CPS-IoT Week '23: Proceedings of Cyber-Physical Systems and Internet of Things Week 2023	No	I2
S27	-	2023	ISSTA 2023: Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis	No	I2
S28	-	2023	ARES '23: Proceedings of the 18th International Conference on Availability, Reliability and Security	No	I2
S29	-	2023	SBQS '23: Proceedings of the XXII Brazilian Symposium on Software Quality	No	I2
S30	-	2023	AfriCHI '23: Proceedings of the 4th African Human Computer Interaction Conference	No	I2
S31	-	2023	CompSysTech '23: Proceedings of the 24th International Conference on Computer Systems and Technologies	No	I2
S32	-	2023	SAC '23: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing	No	I2
S33	-	2023	ICSE '23: Proceedings of the 45th International Conference on Software Engineering: Companion Proceedings	No	I2
S34	-	2023	ICISS '23: Proceedings of the 2023 6th International Conference on Information Science and Systems	No	I2
S35	-	2023	Onward! 2023: Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software	No	I2
S36	Knop U., Hofman P., Mihatsch M., Siegmund M.	2023	Balancing Variability and Costs in Software Product Lines: An Experience Report in Safety-Critical Systems	No	I2
S37	Matsui B. M., Goya D. H.	2023	MLOps: a guide to its adoption in the context of responsible AI	No	I2
S38	IEEE	2023	IEEE Draft Standard for Measures of the Software Aspects of Dependability	No	I2
S39	-	2023	ISO/IEC/IEEE International Standard – Systems and software engineering – System life cycle processes	No	I2
S40	-	2023	ITNOW Volume 65 Issue 1, Full Issue	No	E1
S41	FSSC	2023	The Functional Safety Terminology Landscape	No	I2
S42	Quigley J. M.	2023	SAE International's Dictionary of Testing, Verification, and Validation	No	E1
S43	Stiennon R., Ernst R. B., Forslund F.	2023	Net Zeros and Ones: How Data Erasure Promotes Sustainability, Privacy, and Security	No	E1
S44	Antic A., Thompson J. K.	2023	Creators of Intelligence: Industry secrets from AI leaders that you can easily apply to advance your data science career	No	E1

Study ID	Author(s)	Year	Title	Included	Reason
S45	Seferlis C., Nellis C., Roberts A.	2023	Practical Guide to Azure Cognitive Services: Leverage the power of Azure OpenAI to optimize operations, reduce costs, and deliver cutting-edge AI solutions	No	E1
S46	Wardzinski A., Jarzebowicz A.	2023	Development of the System Assurance Reference Model for Generating Modular Assurance Cases	No	I2
S47	Kulkarni P.	2023	Antennas for IoT	No	I2
S48	Smeenk H. G., Petock M.	2023	Internet of Things for Smart Buildings: Leverage IoT for smarter insights for buildings in the new and built environments	No	I2
S49	Yarali A.	2023	Networks of the Future	No	I2
S50	Vasylieva K., Kuhrmann M., Xavier K., Klünder J.	2023	How Agile Are you? Discussing Maturity Levels of Agile Maturity Models	No	I2
S51	Bratsis I.	2023	The AI Product Manager's Handbook: Develop a product that takes advantage of machine learning to solve AI problems	No	I2
S52	Ghosh S.	2023	Building Low Latency Applications with C++: Develop a complete low latency trading ecosystem from scratch using modern C++	No	I2
S53	Kapoor A., Chatterjee S.	2023	Platform and Model Design for Responsible AI: Design and build resilient, private, fair, and transparent machine learning models	No	I2
S54	Subasini C. A., Swetha P. N., Ravishankar S., Sheeba A.	2023	Dynamic Healthcare System using Cloud Computing	No	I2
S55	Mulder J.	2023	Multi-Cloud Strategy for Cloud Architects: Learn how to adopt and manage public clouds by leveraging BaseOps, FinOps, and DevSecOps	No	E1
S56	Saulaiman M. N., Kozlovsky M., Csilling A.	2023	Side Benefits of Cybersecurity Measures an Empirical Study	No	I2
S57	Conrad E., Misenar S., Feldman J.	2023	CISSP Study Guide (Fourth Edition) – Chapter 9 - Domain 8: Software Development Security	No	E1
S58	Batarseh F. A., Freeman L. J.	2023	AI Assurance	No	I2
S59	Mallik A. K.	2023	The future of the technology-based manufacturing in the European Union	No	I2
S60	Letafati M., Otoum S.	2023	On the privacy and security for e-health services in the metaverse: An overview	No	I2
S61	Pauzi Z., Capiluppi A.	2023	Applications of natural language processing in software traceability: A systematic mapping study	No	I2
S62	Fahmy H., Samir S., Nasr M.	2023	Improving INtegrity, Security, and Accuracy During DevOps Process	No	I2
S63	Martina M. R., Bianchini E., Sinceri S., Francesconi M., Gemignani V.	2023	Software medical device maintenance: DevOps based approach for problem and modification management	Yes	
S64	Stirbu V., Granlund T., Mikkonen T.	2023	Continuous design control for machine learning in certified medical systems	Yes	
S65	Hernandez R., Moros B., Nicolas J.	2023	Requirements management in DevOps environments: a multivocal mapping study	No	I2
S66	Blüher T., Maelzer D., Harrendorf J., Stark R.	2023	DevOps for manufacturing systems: Speeding up software development	No	I2
S67	Byrne K., Cevenini A.	2023	Aligning DevOps Concepts with Agile Models of the Software Development Life Cycle (SLDC) in Pursuit of Continuous Regulatory Compliance	No	E1
S68	Shahin M., Rezaei Nasab A., Ali Babar M.	2023	A qualitative study of architectural design issues in DevOps	No	I2
S69	Uddin M. M., Ge M.	2023	Data Analytics Framework for Identifying Relevant Adverse Events in Medical Software	No	I2
S70	Christoph M.-D., Vierhauser M., Bichler S., Keplinger F., Cleland-Huang J., Egyed A., Mehofer T.	2023	ProCon: An automated process-centric quality constraints checking framework	No	I2
S71	Castillo-Olea C., Rojas-Mendizabal V., Zuniga C.	2023	A Technological Platform to Support Monitoring of Patients with Schizophrenia	No	I2

Study ID	Author(s)	Year	Title	Included	Reason
S72	Souto S., Barbosa P., Oliveira L., Gaeta E., Batistel A., Bastida L., Barbosa L.	2023	Reference Architectures for Health	No	E1
S73	Blanco D. F., Le Mouel F., Lin T., Escudie M. P.	2023	A Comprehensive Survey on Software as a Service (SaaS) Transformation for the Automotive Systems	No	I2
S74	Hemon-Hildgen A., Rowe F.	2022	Conceptualising and defining DevOps: A review for understanding, not a framework for practitioners	No	I2
S75	Klotins E., GOrschek T., Sundelin K., Falk E.	2022	Towards cost-benefit evaluation for continuous software engineering activities	No	I2
S76	Fraustino J., Adriano D., Amaro R., Pereira R., da Silva M. M.	2022	DevOps benefits: A systematic literature review	No	I2
S77	Dakkak A., Bosch J., Olsson H. H.	2022	Controlled Continuous Deployment: A Case Study From The Telecommunications Domain	No	I2
S78	Furrer F. J.	2022	Safety and Security of Cyber-Physical Systems: Engineering dependable Software using Principle-based Development	No	I2
S79	Peldszus S. M.	2022	Security Compliance in Model-driven Development of Software Systems in Presence of Long-Term Evolution and Variants	No	I2
S80	Yasar H., Teplov S. E.	2022	DevSecOps in Embedded Systems: An Empirical Study of Past Literature	No	E7
S81	Granlund T., Stirbu V., Mikkonen T.	2022	Medical Software Needs Calm Compliance	Yes	
S82	Stirbu V., Raatikainen M., Röntynen J., Sokolov V., Lehtonen T., Mikkonen T.	2022	Toward Multiconcern Software Development with Everything as Code	Yes	
S83	Otta M.	2022	Towards a health software supporting platform for wearable devices	Yes	
S84	Plant O. H., van Hillegersberg J., Aldea A.	2022	Rethinking IT governance: Designing a framework for mitigating risk and fostering internal control in a DevOps environment	No	I2
S85	Schneider P., Xhafa F.	2022	Anomaly Detection and Complex Event Processing over IoT Data Streams – Chapter 8 - Machine learning: ML for eHealth systems	No	E1
S86	Campagner A., Stemini F., Cabitza F.	2022	Decisions are not all equal—Introducing a utility metric based on case-wise raters' perceptions	No	I2
S87	Fotopoulos A., Lappas P. Z., Mellitsiotis A.	2022	Wearable Sensing and Intelligent Data Analysis for Respiratory Management – Chapter 8 - The edge-cloud continuum in wearable sensing for respiratory analysis	No	I2
S88	Rajapakse R. N., Zahedi M., Babar M. A., Shen H.	2022	Challenges and solutions when adopting DevSecOps: A Systematic review	No	E7
S89	Thomasian A.	2022	Storage Systems – Organization, Performance, Coding, Reliability, and Their Data Processing – Chapter 2 - Storage technologies and their data	No	I2
S90	Ranganath S.	2022	Chapter Two - Edge Computing: Types and Attributes	No	I2
S91	De Silva D., Alahakoon D.	2022	An artificial intelligence life cycle: From conception to production	No	I2
S92	Zeydan E., Baranda J., Mangues-Bafalluy J., Turk Y., Ozturk S. B.	2022	Blockchain-Based Service Orchestration for 5G Vertical Industries in Multicloud Environment	No	I2
S93	-	2022	ISO/IEC/IEEE Draft International Standard - Systems and Software Engineering -Life Cycle Management - Part 2: Guidelines for the Application of ISO/IEC/IEEE 15288 (System Life Cycle Processes)	No	E1
S94	-	2022	ISO/IEC/IEEE Draft Standard - Systems and Software Engineering – System Life Cycle Processes	No	E1
S95	DeFranco J.	2022	Should Cyberphysical Systems and the Internet of Things Get Married?	No	I2
S96	-	2022	ISSE 2022 Technical Program	No	I2
S97	Ebert C., Bajaj D., Weyrich M.	2022	Testing Software Systems	No	I2
S98	Mani V. S., Ebert C.	2022	Medical Software	Yes	

Study ID	Author(s)	Year	Title	Included	Reason
S99	-	2022	SCSE 2022 Conference Proceedings	No	I2
S100	Chatterjee S., Deshpande S., Been H., van der Gaag M.	2022	Designing and Implementing Microsoft DevOps Solutions AZ-400 Exam Guide: Prepare for the certification exam and successfully apply Azure DevOps strategies with practical labs	No	I2
S101	Shrivastava S., Srivastav N., Sheth R., Karmarkar R., Arora K.	2022	Solutions Architect's Handbook: Kickstart your career as a solutions architect by learning architecture design principles and strategies	No	I2
S102	Kaufmann M., Dohmke T., Brown D.	2022	Accelerate DevOps with GitHub: Enhance software delivery performance with GitHub Issues, Projects, Actions, and Advanced Security	No	E1
S103	Chatterjee A., Gerdes M. W., Khatiwada P., Prinz A.	2022	SFTSDH: Applying Spring Security Framework With TSD-Based OAuth2 to Protect Microservice Architecture APIs	No	I2
S104	Koutsopoulos K., Simon A., Ertl B., Tompros S., Kapusta K., Coatrieux G., Gavras A., Ledakis G., Toscano O., Govaci S., Thümmler C.	2022	Federated machine learning through edge ready architectures with privacy preservation as a service	No	I2
S105	Ferreira R.	2022	Policy Design in the Age of Digital Adoption: Explore how PolicyOps can drive Policy as Code adoption in an organization's digital transformation	No	I2
S106	Mulder J., Mulder H.	2022	Transforming Healthcare with DevOps: A practical DevOps4Care guide to embracing the complexity of digital transformation	No	E1
S107	Mahapatra A., May D.	2022	Simplifying Data Engineering and Analytics with Delta: Create analytics-ready data that fuels artificial intelligence and business intelligence	No	I2
S108	Care J.	2022	Mastering Technical Sales: The Sales Engineer's Handbook, Fourth Edition	No	I2
S109	Costa Negro F.	2022	Simplifying Hybrid Cloud Adoption with AWS: Realize edge computing and build compelling hybrid solutions on premises with AWS Outposts	No	I2
S110	Di Federico G., Barcaroli F.	2022	Cloud Identity Patterns and Strategies: Design enterprise cloud identity models with OAuth 2.0 and Azure Active Directory	No	I2
S111	Teter J. A., Tobin B.	2022	Technical Program Manager's Handbook: Empowering managers to efficiently manage technical projects and build a successful career path	No	I2
S112	Malmqvist L.	2022	Salesforce Anti-Patterns: Create powerful Salesforce architectures by learning from common mistakes made on the platform	No	I2
S113	Tarun R.	2022	The Convergence of Cyber and Physical	No	I2
S114	Birch M.	2022	CompTIA CASP+ CAS-004 Certification Guide: Develop CASP+ skills and learn all the key topics needed to prepare for the certification exam	No	I2
S115	Ratan U.	2022	Applied Machine Learning for Healthcare and Life Sciences Using AWS: Transformational AI implementations for biotech, clinical, and healthcare organizations	No	I2
S116	Abdelaziz M.	2022	Designing Production-Grade and Large-Scale IoT Solutions: A comprehensive and practical guide to implementing end-to-end IoT solutions	No	I2
S117	Fisher D.	2022	Application Security Program Handbook: A guide for software engineers and team leaders	No	E1
S118	Keys G., Whiting D.	2022	Machine Learning at Scale with H2O: A practical guide to building and deploying machine learning models on enterprise systems	No	I2
S119	Raibulet C., Fontana F. A., Pigazzini I.	2022	Hints on Designing and Running Project-based Exams for a Software Engineering Course	No	I2

Study ID	Author(s)	Year	Title	Included	Reason
S120	Shein E.	2022	Neurotechnology and the law	No	I2
S121	-	2022	ICIT '22: Proceedings of the 2022 10th International Conference on Information Technology: IoT and Smart City	No	I2
S122	-	2022	AICCC '22: Proceedings of the 2022 5th Artificial Intelligence and Cloud Computing Conference	No	I2
S123	-	2022	QP4SE 2022: Proceedings of the 1st International Workshop on Quantum Programming for Software Engineering	No	I2
S124	-	2022	ESEM '22: Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement	No	I2
S125	-	2022	ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security	No	I2
S126	-	2022	ASE '22: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering	No	I2
S127	-	2022	SBQS '22: Proceedings of the XXI Brazilian Symposium on Software Quality	No	I2
S128	-	2022	EuroUSEC '22: Proceedings of the 2022 European Symposium on Usable Security	No	I2
S129	-	2022	ICIMMI '22: Proceedings of the 4th International Conference on Information Management & Machine Intelligence	No	I2
S130	-	2022	SE4RAI '22: Proceedings of the 1st Workshop on Software Engineering for Responsible AI	No	I2
S131	-	2021	CHI '21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems	No	I2
S132	-	2021	CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems	No	I2
S133	Haas R., Elsner D., Juergens E., Pretschner A., Apel S.	2021	How can manual testing processes be optimized? developer survey, optimization guidelines, and case studies	No	I2
S134	Zhou P., Ali Khan A. A., Liang P., Badshah S.	2021	System and Software Processes in Practice: Insights from Chinese Industry	No	I2
S135	Lazuardi M., Raharjo T., Hardian B., Simanungkalit T.	2021	Perceived Benefits of DevOps Implementation in Organization: A Systematic Literature Review	No	I2
S136	Stewart J. M.	2021	CompTIA Security+ Review Guide: Exam SY0-601	No	I2
S137	Book A.	2021	AWS Certified DevOps Engineer - Professional Certification and Beyond: Pass the DOP-C01 exam and prepare for the real world using case studies and real-life examples	No	E1
S138	Lee J., Leonardo G., Milgram J., Rendon D.	2021	Azure Strategy and Implementation Guide: The essential handbook to cloud transformation with Azure	No	E1
S139	Granlund T., Kopponen A., Stirbu V., Myllyaho L., Mikkonen T.	2021	MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases	Yes	
S140	Rethi A.-B., Antal T., Mathe O., Foszto M., Koncz T., Simon K.	2021	medR: Software System for Managing Medical History and Patient Examination Data	No	I2
S141	Zhu Q., Rass S., Dieber B., Vilches V. M.	2021	Cybersecurity in Robotics: Challenges, Quantitative Modeling, and Practice	No	I2
S142	Franssens N., Gopalakrishnan S., Lenz G.	2021	Hands-on Kubernetes on Azure: Use Azure Kubernetes Service to automate management, scaling, and deployment of containerized applications	No	I2
S143	Stirbu V., Granlund T., Helen J., Mikkonen T.	2021	Extending SOUP to ML Models When Designing Certified Medical Systems	Yes	
S144	Sadkhan S. B., Al Refaai N.	2021	Crypto Warfare Techniques- Status, Challenges, and Future Trends	No	I2
S145	Baron C., Louis V.	2021	Towards a continuous certification of safety-critical avionics software	No	I2
S146	Toivakka H., Granlund T., Poranen T., Zhang Z.	2021	Towards RegOps: A DevOps Pipeline for Medical Device Software	Yes	

Study ID	Author(s)	Year	Title	Included	Reason
S147	Kempe E., Massey A. K.	2021	Regulatory and security standard compliance throughout the software development lifecycle	No	E7
S148	Nawaz M., Nazir T., Islam S., Masood M., Mehmood A., Kanwal S.	2021	Agile software development techniques: A survey	No	I2
S149	Mayr-Dorn C., Vierhauser M., Bichler S., Kepllinger F., Cleland-Huang J., Egyed A., Mehofer T.,	2021	Supporting quality assurance with automated process-centric quality constraints checking	No	I2
S150	Oshida Y.	2021	Artificial intelligence for medicine: People, society, pharmaceuticals, and medical materials	No	I2
S151	Perez A. J., Zeadally S.	2021	Recent advances in wearable sensing technologies	No	I2
S152	Stirbu V., Mikkonen T.	2021	Introducing Traceability in GitHub for Medical Software Development	Yes	
S153	Granlund T., Stirbu V., Mikkonen T.	2021	Towards Regulatory-Compliant MLOps: Oravizio's Journey from a Machine Learning Experiment to a Deployed Certified Medical Product	Yes	
S154	Ganeshan M., Vigneshwaran P.	2021	A Survey on DevOps Techniques Used in Cloud-Based IOT Mashups	No	E1
S155	Zhou Y., Su Y., Chen T., Huang Z., Gall H., Panichella S.	2021	User Review-Based Change File Localization for Mobile Applications	No	I2
S156	Lee S. J., Kim G. B.	2021	K-FFRaaS: A Generic Model for Financial Forensic Readiness as a Service in Korea	No	I2
S157	Parker M.	2021	Risk Considerations for MobileDevice implementations	No	I2
S158	Lie Forsberg M., Sanchez-Gordon M., Colomo-Palacios R.	2020	DevOps in an ISO 13485 regulated environment: A multivocal literature review	No	E7
S159	Malamas V., Kotzanikolaou P., Dasaklis T. K., Burmester M.	2020	A Hierarchical Multi Blockchain for Fine Grained Access to Medical Data	No	I2
S160	Oesterle S., Jöhnk J., Keller R., Urbach N., Yu X.	2020	A contingency lens on cloud provider management processes	No	I2
S161	Faustino J., Pereira R., Alturas B., da Silva M. M.	2020	Agile information technology service management with DevOps: An incident management case study	No	I2
S162	Teixeira D., Pereira R., Henriques T. A., Silva M., Faustino J.	2020	A systematic literature review on DevOps capabilities and areas	No	I2
S163	Mishra A., Otaiwi Z.	2020	DevOps and software quality: A systematic mapping	No	I2
S164	Shahin M., Babar M. A.	2020	On the role of software architecture in DevOps transformation: An industrial case study	No	I2
S165	Mora M., Gomez J. M., O'Connor R. V., Buchalcevova A.	2020	Balancing agile and disciplined engineering and management approaches for IT services and software products	No	E1
S166	Teixeira D., Pereira R., Henriques T., da Silva M. M., Faustino J., Silva M. M.	2020	A maturity model for DevOps	No	I2
S167	Han S., Sinha R., Lowe A.	2020	Assessing Support for Industry Standards in Reference Medical Software Architectures	No	E7
S168	Granlund T., Mikkonen T., Stirbu V.	2020	On Medical Device Software CE Compliance and Conformity Assessment	Yes	
S169	Santos D. F., Rodriguez A. F., Filho W. O., Pereira M. F.	2020	Adapting agile practices during the evolution of a healthcare software product	No	E1
S170	Stirbu V., Mikkonen T.	2020	CompliancePal: A Tool for Supporting Practical Agile and Regulatory-Compliant Development of Medical Software	Yes	
S171	-	2020	Chapter 4: Content of the cybersecurity curricular framework	No	I2
S172	CC20 Task Force	2020	Computing Curricula 2020: Paradigms for Global Computing Education	No	I2
S173	Gupta R. K., Balaji B., Mekanathan V., Ferose Khan J.	2020	Challenges in scaling AI-powered distributed software product: a case study of a healthcare organization	No	I2

Study ID	Author(s)	Year	Title	Included	Reason
S174	Prenner N., Unger-Windeler C., Schneider K.	2020	How are Hybrid Development Approaches Organized?: A Systematic Literature Review	No	I2
S175	-	2020	ISO/IEC/IEEE Draft International Standard - Software and Systems Engineering – Software Testing – Part 1: Concepts and Definitions	No	I2
S176	Anderson R.	2020	In Security Engineering: A Guide to Building Dependable Distributed Systems	No	E1
S177	Malisow B.	2020	(ISC)2 CCSP Certified Cloud Security Professional Official Practice Tests – Practice Exam 1	No	I2
S178	Docter Q., Fuchs C.	2020	CompTIA Cloud Essentials+ Study Guide: Exam CLO-002	No	I2
S179	Sullivan D.	2020	Official Google Cloud Certified Professional Cloud Architect Study Guide – Analyzing and Defining Technical Processes	No	E1
S180	Coursey C.	2020	The Practitioner's Guide to Cellular IoT	No	E1
S181	Gupta R., Tanwar S., Tyagi S., Kumar N.	2020	Machine Learning Models for Secure Data Analytics: A taxonomy and threat model	No	I2
S182	Islam G., Storer T.	2020	A case study of agile software development for safety-Critical systems projects	No	I2
S183	Özcan-Top Ö., Demirors O.	2019	Application of a software agility assessment model – AgilityMod in the field	No	I2
S184	Li B., Dong Q., Downen R. S., Tran N., Jackson J. H., Pillai D., Zaghloul M., Li Z.	2019	A wearable IoT aldehyde sensor for pediatric asthma research and management	No	I2
S185	Leite L., Rocha C., Kon F., Milojicic D., Meirelles P.	2019	A Survey of DevOps Concepts and Challenges	No	I2
S186	Jacobson I., Lawson H., Ng P.-W., McMahon P. E., Goedicke M.	2019	The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!	No	I2
S187	Szabo D. M., Steghöfer J. P.	2019	Coping strategies for temporal, geographical and sociocultural distances in agile GSD: a case study	No	I2
S188	Wolschke C., Becker M., Schneickert S., Adler R., MacGregor J.	2019	Industrial Perspective on Reuse of Safety Artifacts in Software Product Lines	No	I2
S189	Nehls H., Ratiu D.	2019	Towards continuous delivery for domain experts: using MDE to integrate non-programmers into a software delivery pipeline	Yes	
S190	Viggiato M., Oliveira J., Figueiredo E., Jamshidi P., Kastner C.	2019	Understanding similarities and differences in software development practices across domains	No	I2
S191	IEEE	2019	P7010/D1, Jun 2019 - IEEE Draft Standard for Well-being Metrics for Autonomous and Intelligent Systems	No	I2
S192	-	2019	2019 Global IoT Summit (GIoTS)	No	I2
S193	-	2019	2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)	No	I2
S194	Mikkonen T., Taivalsaari A.	2019	Software Reuse in the Era of Opportunistic Design	No	I2
S195	Taivalsaari A., Mikkonen T., Mäkitalo N.	2019	Programming the Tip of the Iceberg: Software Reuse in the 21st Century	No	I2
S196	Kurhmann M., Diebold P., Munch J., Tell P., Trektere K., McCaffery F., Garousi V., Felderer M., Linssen O., Hanser E., Prause C. R.	2019	Hybrid Software Development Approaches in Practice: A European Perspective	No	I2
S197	Buyya R., Narayana Srirama S.	2019	Fog and Edge Computing: Principles and Paradigms – Testing Perspectives of Fog-Based IoT Applications	No	I2
S198	Wijaya P. E., Rosyadi I., Taryana A.	2019	An attempt to adopt DevOps on embedded system development: Empirical evidence	No	I2
S199	Kim D., Lee B., Lee J.-W.	2019	Building a rule-based goal-model from the IEC 62304 standard for medical device software	No	I2
S200	Van Belzen M., Trienekens J., Kusters R.	2019	Critical success factors of continuous practices in a DevOps context	No	I2
S201	Rodriguez P., Mäntylä M., Oivo M., Lwakatare L. E., Seppänen P., Kuvaja P.	2019	Advances in Using Agile and Lean Processes for Software Development	No	I2

Study ID	Author(s)	Year	Title	Included	Reason
S202	Morales-Trujillo M. E., Garcia-Mireles G. A., Matla-Cruz E. O., Piattini M.	2019	A Systematic Mapping Study of Privacy by Design in Software Engineering	No	I2
S203	Giorgi F., Paulisch F.	2019	Transition towards Continuous Delivery in the Healthcare Domain	Yes	
S204	Hanssen G. K., Stålhane T., Myklebust T.	2018	SafeScrum® – Agile Development of Safety-Critical Software	No	E1
S205	Demissie S., Keenan F., Özcan-Top Ö., McCaffery F.	2018	Agile usage in embedded software development in safety critical domain–A systematic review	No	E7
S206	Silva M. A., Faustino J. P., Pereira R., da Silva M. M.	2018	Productivity gains of DevOps adoption in an IT team: A case study	No	I2
S207	Özcan-Top Ö., McCaffery F.	2018	A hybrid assessment approach for medical device software development companies	Yes	
S208	Zaitsev A.	2018	Agile Methods as a Risk Management Strategy Tool-A FinTech Case Study	No	I2
S209	Stirbu V., Mikkonen T.	2018	Towards Agile Yet Regulatory-Compliant Development of Medical Software	Yes	
S210	Laukkarinen T., Kuusinen K., Mikkonen T.	2018	Regulated Software Meets DevOps	Yes	
S211	-	2018	2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)	No	I2
S212	-	2018	2018 IEEE 5G World Forum (5GWF)	No	I2
S213	-	2018	2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)	No	I2
S214	Vermesan O., Bacquet J.	2018	7 IoT European Security and Privacy Projects: Integration, Architectures and Interoperability	No	I2
S215	Kuusinen K., Balakumar V., Jepsen S. C., Larsen S. H., Lemqvist T. A., Muric A., Nielsen A. O., Vestergaard O.	2018	A Large Agile Organization on Its Journey Towards DevOps	No	I2
S216	-	2018	Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity	No	I2

Table A.2: List of GL and initial filtering step.

GL ID	Author(s)/Organization	Year	Title	Included	Reason
G1	Parasoft	-	16 Must-Haves to Get Started With Medical Device Software Compliance	No	E1
G2	AAMI	2021	AAMI/CR510:2021; Appropriate Use of Public Cloud Computing for Quality Systems and Medical Devices	No	E1
G3	Ahola J.	2020	AHMED: Agile and Holistic Medical Development -presentation	No	E6
G4	Taipuva	2020	AHMED project started creating RegOps	No	E6
G5	Lähteenmäki J., Ahola P., Baraian A., Förger K., Granlund T., Hopia J., Kaikkonen R., Mikkonen T., Niemirepo T., Pajula J., Partanen J., Pellinen T., Stirbu V., Torhola M.	2023	Agile and Holistic Medical Software Development – Final report of AHMED project	Yes	
G6	Byrne K., and Cevenini A.	2022	Aligning DevOps Concepts with Agile Models of the Software Development Life Cycle (SLDC) in Pursuit of Continuous Regulatory Compliance	No	E1
G7	Borad A.	2023	An Overview of FDA Regulations for Medical Devices	No	E5
G8	Storytel	-	Books from ASQ Quality Press	No	I2
G9	BCB Medical	2023	Instagram post	No	I2
G10	Inget J.	2021	7 Key Challenges in Medical Device Design and How to Solve Them	No	I2
G11	Nord Hero	-	Building infrastructure and DevOps processes for 5-star patient experience	No	E4
G12	Oxlund H.	-	Can Microsoft Dynamics 365 FO be used in heavily regulated Life Science	No	E4

GL ID	Author(s)/Organization	Year	Title	Included	Reason
G13	Civil Service Jobs	-	-	No	E8
G14	Kelly P.	2024	IVDR and in-house devices – Video presentation	No	I2
G15	Müller U., Steiner P., Bäck P., Berger C.	2022	Medical Clouds: A Case for Continuous Validation in Medtech & Pharma	Yes	
G16	Neebal Technologies	-	Why Adopting DevOps in highly regulated industries is need of the hour	No	E4
G17	Altynpara E., Chabanovska D.	2023	What is Software Development for Medical Devices [Expert Guide]	No	I2
G18	Selleo	2023	What is Software As A Medical Device (SaMD)?	No	E5
G19	Koskimies O.	2019	What is MedDevOps?	Yes	
G20	Waikar A. M.	2023	Understanding the significance of Human Factors in Medical Devices	No	I2
G21	Shah P.	2023	Understanding IEC 62304 in Medical Device Software Development	Yes	
G22	Modern Requirements	-	Tools designed for Healthcare and Medical Devices Companies	No	E4
G23	TOPRA	-	Professional development specialties	No	E4
G24	Kivirauma K.	2021	Solita Health RegOps	No	E6
G25	Granlund T.	2024	Solita RegProof blog series, part 1: Typical pain points in medical software development and how calm compliance helps	Yes	
G26	Yalantis	-	Medical Device Software Engineering Solutions from Yalantis	No	E4
G27	Hartung W., Schalago J., Rossi C., Pavkov R.	2021	Software as a Medical Device Fundamentals	Yes	
G28	-	-	Software as a Medical Device Embracing the Digital Healthcare Revolution	No	E4
G29	Scispace	-	How to implement devsecops in medical devices?	No	E4
G30	Laukkarinen T., Kuusinen K., Mikkonen T.	2018	Regulated software meets DevOps	No	E6
G31	Granlund T.	2020	RegOps – diving into the dilemma of agile software development in regulated industry	Yes	
G32	Leppämäki J.	2022	RegOps – agile development of medical software	Yes	
G33	Shah S. N.	-	Quantitative Regulatory Science ("RegOps") for FDA Digital Health/Therapeutics Compliance	No	E4
G34	TestingXperts	2024	Quality Assurance in Software as a Medical Device	Yes	
G35	Method Park	-	Product Design Processes for Medical Devices	No	E4
G36	Kruuti A.	2023	Planning a Change to an EU Medical Device Regulation Certified Transcranial Magnetic Stimulation Device	No	E1
G37	De Vos D.	2022	Microsoft Dynamics 365 in the heavily-regulated Pharma & Life Sciences industry: is it possible?	No	I2
G38	Method Park	-	Medical Device Software	No	E4
G39	Antelope	-	Medical Device Software Engineer	No	E4
G40	Gaiffe N.	-	Medical devices and Healthcare: accelerate innovation and meet ever-changing regulations	No	E4
G41	Amazon	-	Medical Devices on AWS	No	E4
G42	Pinja	-	Medical device regulation and software development – Key points to consider	No	E4
G43	-	-	Medical Devops manufacturer	No	E4
G44	-	-	Medical Devops Development and Operation Services	No	E4
G45	MedDevOps	-	MedDevOps DevOps and Medical Regulation	No	E4
G46	Qarea	2023	Medical Device Software Development: From Software Design to Launch	Yes	
G47	BCB Medical	-	MDR certification granted to BCB Medical's software	No	E4
G48	Nearform	2021	Making DevOps work for highly regulated industries	Yes	
G49	Pinja	-	Keyword: Software development	No	I2
G50	Ketryx	2024	Ketryx builds safer FDA compliance software faster	No	E8
G51	Google Scholar	-	Kati Kuusinen	No	E4
G52	Lähteenmäki J.	2023	Linkedin post	No	I2

GL ID	Author(s)/Organization	Year	Title	Included	Reason
G53	Mitrofanskiy K.	2024	Medical Device Regulation EU Compliance in 2024	No	I2
G54	Mphasis	-	Integrating engineering excellence in medical devices with intelligent insights	No	E4
G55	Indeed	-	Avoimet työpaikat	No	I2
G56	Modern Requirements	-	How to Facilitate the Medical Device Design Controls in Azure DevOps	No	E4
G57	Lemberg Solutions	-	Healthcare Software Development	No	I2
G58	Matrix requirements	-	Get your medical device MDR compliant faster	No	I2
G59	Labquality	-	Framework of modern medical software development lifecycle	No	E4
G60	Shah S.	2021	FDA Digital Health Regulation presentation	No	E10
G61	Howard T.	2020	How the Role of Regulatory Operations Professionals Will Evolve	No	I2
G62	Innokas Medical	-	Master the quality and regulations of your medical device	No	E4
G63	Laukkarinen T., Kuusinen K., Mikkonen T.	2017	DevOps in Regulated Software Development: Case Medical Devices	No	E3
G64	Bhat S.	2023	DevOps in the Healthcare Industry: Ensuring Quality, Security, and Safety	Yes	
G65	Makkonen M.	2022	DevOps in regulated environment	No	E1
G66	Leitão J. C., Lynge H., Kocak V.	2023	DevOps in medical software	Yes	
G67	Norton S.	2022	DevOps Framework	No	I2
G68	Helvetica-Partners	2023	DevOps Engineer (Medical Device Sector) – SEM59568A	No	E8
G69	Ketryx	-	Developer-first Solution for Regulated Software	No	E4
G70	Advamed	2024	Creating an Automated Test Strategy and Framework for Regulated Medical Device Software	No	E10
G71	Toivakka H.	2021	Integration of EU medical device regulatory requirements into a CI/CD pipeline	Yes	
G72	Dragann S., Valentine J.	2022	Confidential VMs – a security breakthrough for medical device software	Yes	
G73	Nir R.	2024	Combine Xray test management with Jira Snapshots for a perfect regulated DevOps flow	No	E8
G74	ERNI	2024	Case study: From Cobot to medical device	No	I2
G75	UCSC	2024	Courses	No	E10

Appendix B Literature reviewed in the quality criteria step

Table B.1: Quality assessment for studies.

Study ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Score
S63	Martina M. R., Bianchini E., Sinceri S., Francesconi M., Gemignani V.	2023	Software medical device maintenance: DevOps based approach for problem and modification management	0	1	1	0	0,5	0,5	0	1	1	**0.56**
S64	Stirbu V., Granlund T., Mikkonen T.	2023	Continuous design control for machine learning in certified medical systems	0	1	1	1	0.5	1	1	1	1	**0.83**
S81	Granlund T., Stirbu V., Mikkonen T.	2022	Medical Software Needs Calm Compliance	0	1	1	1	0.5	0.5	0	1	1	**0.67**
S82	Stirbu V., Raatikainen M., Röntynen J., Sokolov V., Lehtonen T., Mikkonen T.	2022	Toward Multiconcern Software Development with Everything as Code	0	1	1	1	0.5	0.5	0	1	1	**0.67**
S83	Otta M.	2022	Towards a health software supporting platform for wearable devices	0	1	0.5	0	0.5	0	0	0.5	0	0.28
S98	Mani V. S., Ebert C.	2022	Medical Software	0	0.5	1	0	0.5	1	0.5	0.5	0.5	0.5
S139	Granlund T., Kopponen A., Stirbu V., Myllyaho L., Mikkonen T.	2021	MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases	1	1	1	1	0.5	1	1	1	1	**0.94**
S143	Stirbu V., Granlund T., Helen J., Mikkonen T.	2021	Extending SOUP to ML Models When Designing Certified Medical Systems	0	1	1	1	0.5	1	0.5	1	1	**0.78**
S146	Toivakka H., Granlund T., Poranen T., Zhang Z.	2021	Towards RegOps: A DevOps Pipeline for Medical Device Software	1	1	1	1	0.5	0.5	0	1	1	**0.78**
S152	Stirbu V., Mikkonen T.	2021	Introducing Traceability in GitHub for Medical Software Development	1	1	1	1	0.5	0.5	0.5	1	1	**0.83**
S153	Granlund T., Stirbu V., Mikkonen T.	2021	Towards Regulatory-Compliant MLOps: Oravizio's Journey from a Machine Learning Experiment to a Deployed Certified Medical Product	1	1	1	1	1	1	1	1	1	**1.00**
S168	Granlund T., Mikkonen T., Stirbu V.	2020	On Medical Device Software CE Compliance and Conformity Assessment	1	1	1	1	0.5	1	1	1	1	**0.94**
S170	Stirbu V., Mikkonen T.	2020	CompliancePal: A Tool for Supporting Practical Agile and Regulatory-Compliant Development of Medical Software	1	1	1	0.5	1	0.5	0	1	1	**0.78**
S189	Nehls H., Ratiu D.	2019	Towards continuous delivery for domain experts: using MDE to integrate non-programmers into a software delivery pipeline	0	1	1	0.5	1	0.5	1	1	0.5	**0.72**
S203	Giorgi F., Paulisch F.	2019	Transition towards Continuous Delivery in the Healthcare Domain	0	0.5	0.5	0.5	0	0	0	0.5	1	0.33

Study ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Score
S207	Özcan-Top Ö., McCaffery F.	2018	A hybrid assessment approach for medical device software development companies	1	0.5	1	0.5	0.5	0.5	0	0.5	0.5	**0.56**
S209	Stirbu V., Mikkonen T.	2018	Towards Agile Yet Regulatory-Compliant Development of Medical Software	1	1	1	1	0.5	0.5	0	1	1	**0.78**
S210	Laukkarinen T., Kuusinen K., Mikkonen T.	2018	Regulated Software Meets DevOps	1	1	1	1	0.5	0.5	0	0.5	1	**0.72**

Table B.2: Quality assessment for formal literature found by snowballing.

Study ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Score
S217	Granlund T., Vedenpää J., Stirbu V., Mikkonen T.	2021	On Medical Device Cybersecurity Compliance in EU	0	1	1	0.5	0,5	1	0	0.5	1	**0.61**
S218	Jachmann T.	2019	Transforming a Large Medical Organization towards Speed and Flow	1	0.5	1	0.5	0,5	0.5	0	0.5	1	**0.61**
S219	Kim D., Lee B., Lee J. W.	2019	Methods of Extracting and Providing R&D Documentation Guideline for Licensing Medical Device Software	0	1	0.5	0.5	0	0.5	0	0.5	0.5	0.39
S220	Morales J. A., Yasar H., Volkman A.	2018	Implementing DevOps Practices in Highly Regulated Environments	0	1	1	1	0.5	1	0.5	1	0.5	**0.72**
S221	Agyei E. E., Pohjolainen S., Oinas-Kukkonen H.	2022	Impact of Medical Device Regulation on Developing Health Behavior Change Support Systems	0	1	1	0.5	0.5	0.5	0	0.5	0	0.44

Table B.3: Quality assessment for GL.

GL ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
G5	Lähteenmäki J., Ahola P., Baraian A., Förger K., Granlund T., Hopia J., Kaikkonen R., Mikkonen T., Niemirepo T., Pajula J., Partanen J., Pellinen T., Stirbu V., Torhola M.	2023	Agile and Holistic Medical Software Development – Final report of AHMED project	1	0.5	1	1	1	1	1	1	1	1	**1**
G15	Müller U., Steiner P., Bäck P., Berger C.	2022	Medical Clouds: A Case for Continuous Validation in Medtech & Pharma	1	0.5	1	1	1	0	1	1	1	1	**0.89**
G19	Koskimies O.	2019	What is MedDevOps?	0	0	1	0	1	0	0	0	1	0	0.32
G21	Shah P.	2023	Understanding IEC 62304 in Medical Device Software Development	1	0.5	0	0	1	0	1	0	1	0	0.47
G25	Granlund T.	2024	Solita RegProof blog series, part 1: Typical pain points in medical software development and how calm compliance helps	1	0.5	1	1	1	1	1	1	1	0	**0.89**
G27	Hartung W., Schalago J., Rossi C., Pavkov R.	2021	Software as a Medical Device Fundamentals	1	0.5	1	1	0	1	1	1	1	0.5	**0.84**

GL ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
G31	Granlund T.	2020	RegOps – diving into the dilemma of agile software development in regulated industry	1	0.5	1	0	1	1	1	1	1	0	**0.79**
G32	Leppämäki J.	2022	RegOps – agile development of medical software	1	0.5	1	0	1	0	0	1	1	0	**0.58**
G34	TestingXperts	2024	Quality Assurance in Software as a Medical Device	1	0	0	0	0	1	0	0	1	0	0.32
G46	Qarea	2023	Medical Device Software Development: From Software Design to Launch	1	0	0	0	0	1	1	0	0	0	0.32
G48	Nearform	2021	Making DevOps work for highly regulated industries	1	0	0	0	1	1	0	1	0	0	0.42
G64	Bhat S.	2023	DevOps in the Healthcare Industry: Ensuring Quality, Security, and Safety	1	0.5	1	0	1	0	0	1	1	0	**0.58**
G66	Leitão J. C., Lynge H., Kocak V.	2023	DevOps in medical software	1	0.5	1	1	1	1	1	1	1	0	**0.89**
G71	Toivakka H.	2021	Integration of EU medical device regulatory requirements into a CI/CD pipeline	1	0.5	0	1	1	1	1	1	1	1	**0.89**
G72	Dragann S., Valentine J.	2022	Confidential VMs – a security breakthrough for medical device software	1	0.5	1	0	1	0	0	0	1	0	0.47

Table B.4: Quality assessment for GL found by snowballing.

GL ID	Authors	Year	Title	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
G76	Farley D., Ukis V.	2018	Adopting Continuous Delivery at teamplay, Siemens Healthineers	1	0.5	1	0	1	1	1	1	1	1	**0.84**
G77	Medical Device Coordination Group	2022	Ongoing guidance development and deliverables of MDCG Subgroups	1	0	0	0	0	1	0	1	1	0.5	0.47

Appendix C Selected data sources

Table C.1: List of the final set of data sources.

ID	Author(s)	Year	Title	Published
S63	Martina M. R., Bianchini E., Sinceri S., Francesconi M., Gemignani V.	2023	Software medical device maintenance: DevOps based approach for problem and modification management	Journal of Software: Evolution and Process (2023): e2570
S64	Stirbu V., Granlund T., Mikkonen T.	2023	Continuous design control for machine learning in certified medical systems	Software Quality Journal 31.2 (2023): 307-333
S81	Granlund T., Stirbu V., Mikkonen T.	2022	Medical Software Needs Calm Compliance	IEEE Software 39.1 (2021): 19-28
S82	Stirbu V., Raatikainen M., Röntynen J., Sokolov V., Lehtonen T., Mikkonen T.	2022	Toward Multiconcern Software Development with Everything as Code	IEEE Software 39.4 (2022): 27-33
S139	Granlund T., Kopponen A., Stirbu V., Myllyaho L., Mikkonen T.	2021	MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases	2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN). IEEE, 2021
S143	Stirbu V., Granlund T., Helen J., Mikkonen T.	2021	Extending SOUP to ML Models When Designing Certified Medical Systems	2021 IEEE/ACM 3rd International Workshop on Software Engineering for Healthcare (SEH). IEEE, 2021
S146	Toivakka H., Granlund T., Poranen T., Zhang Z.	2021	Towards RegOps: A DevOps Pipeline for Medical Device Software	International Conference on Product-Focused Software Process Improvement. Cham: Springer International Publishing, 2021
S152	Stirbu V., Mikkonen T.	2021	Introducing Traceability in GitHub for Medical Software Development	Product-Focused Software Process Improvement: 22nd International Conference, PROFES 2021, Turin, Italy, November 26, 2021, Proceedings 22. Springer International Publishing, 2021
S153	Granlund T., Stirbu V., Mikkonen T.	2021	Towards Regulatory-Compliant MLOps: Oravizio's Journey from a Machine Learning Experiment to a Deployed Certified Medical Product	SN computer Science 2.5 (2021): 342
S168	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). IEEE, 2020
S170	Stirbu V., Mikkonen T.	2020	CompliancePal: A Tool for Supporting Practical Agile and Regulatory-Compliant Development of Medical Software	2020 IEEE International Conference on Software Architecture Companion (ICSA-C). IEEE, 2020
S189	Nehls H., Ratiu D.	2019	Towards continuous delivery for domain experts: using MDE to integrate non-programmers into a software delivery pipeline	2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE, 2019
S207	Özcan-Top Ö., McCaffery F.	2018	A hybrid assessment approach for medical device software development companies	Journal of Software: Evolution and Process 30.7 (2018): e1929
S209	Stirbu V., Mikkonen T.	2018	Towards Agile Yet Regulatory-Compliant Development of Medical Software	2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2018
S210	Laukkarinen T., Kuusinen K., Mikkonen T.	2018	Regulated Software Meets DevOps	Information and Software Technology 97 (2018): 176-178
S217	Granlund T., Vedenpää J., Stirbu V., Mikkonen T.	2021	On Medical Device Cybersecurity Compliance in EU	2021 IEEE/ACM 3rd International Workshop on Software Engineering for Healthcare (SEH). IEEE, 2021
S218	Jachmann T.	2019	Transforming a Large Medical Organization towards Speed and Flow	2019 IEEE/ACM 1st International Workshop on Software Engineering for Healthcare (SEH). IEEE, 2019
S220	Morales J. A., Yasar H., Volkman A.	2018	Implementing DevOps Practices in Highly Regulated Environments	Proceedings of the 19th International Conference on Agile Software Development: Companion. 2018
G5	Lähteenmäki J., Ahola P., Baraian A., Förger K., Granlund T., Hopia J., Kaikkonen R., Mikkonen T., Niemirepo T., Pajula J., Partanen J., Pellinen T., Stirbu V., Torhola M.	2023	Agile and Holistic Medical Software Development – Final report of AHMED project	VTT Technical Research Centre of Finland. VTT Research Report No. VTT-R-01079-22
G15	Müller U., Steiner P., Bäck P., Berger C.	2022	Medical Clouds: A Case for Continuous Validation in Medtech & Pharma	Zühlke Engineering AG

ID	Author(s)	Year	Title	Published
G25	Granlund T.	2024	Solita RegProof blog series, part 1: Typical pain points in medical software development and how calm compliance helps	https://www.solita.fi/blogs/typical-pain-points-in-medical-software-development/
G27	Hartung W., Schalago J., Rossi C., Pavkov R.	2021	Software as a Medical Device Fundamentals	https://ispe.org/pharmaceutical-engineering/july-august-2021/software-medical-device-fundamentals
G31	Granlund T.	2020	RegOps – diving into the dilemma of agile software development in regulated industry	https://www.solita.fi/blogs/regops-diving-into-the-dilemma-of-agile-software-development-in-regulated-industry/
G32	Leppämäki J.	2022	RegOps – agile development of medical software	https://atostek.com/en/regops-agile-development-of-medical-software/
G64	Bhat S.	2023	DevOps in the Healthcare Industry: Ensuring Quality, Security, and Safety	https://www.ltts.com/blog/DevOps-healthcare
G66	Leitão J. C., Lynge H., Kocak V.	2023	DevOps in medical software	https://jorgecarleitao.medium.com/devops-in-medical-software-54d408596c64
G71	Toivakka H.	2021	Integration of EU medical device regulatory requirements into a CI/CD pipeline	Tampere University – Faculty of Information Technology and Communication Sciences (ITC)
G76	Farley D., Ukis V.	2018	Adopting Continuous Delivery at teamplay, Siemens Healthineers	https://www.infoq.com/articles/continuous-delivery-teamplay/