# Model-Driven Approaches for DevOps: A Systematic Literature Review

Hugo da Gião
DEI, Faculty of Engineering,
University of Porto &
HASLab/INESC TEC
larst@affiliation.org

Rui Pereira
OutSystems
rui.alexandre.pereira@outsystems.com

Jácome Cunha
DEI, Faculty of Engineering,
University of Porto &
HASLab/INESC TEC
hugo.a.giao@inesctec.pt

## ABSTRACT

DevOps is a combination of tools and methodologies which aims to improve software development, build, and deploy processes by shortening this lifecycle and improving software quality. Despite its many benefits, it still presents many challenges regarding its ease of use and accessibility. One of the reasons is the tremendous proliferation of tools, languages, and syntax, which makes the field quite challenging to learn and keep up to date.

On the other hand, as happens in other fields, model-driven engineering approaches can cope with some of these challenges.

This work aims to understand how these two realms currently co-exist. We do so by executing a systematic literature review where we analyze several digital libraries and identify publications whose work focuses on applying various model-driven engineering approaches to the field of DevOps and its different variants (e.g. DevSecOps, MLOps).

We have found and categorized 37 publications, ranging eight different computing domains, from cloud to machine learning, but also found some domain-agnostic proposals. We have also found that deployment is the most addressed phase of DevOps and that most works focus on the cloud domain. Moreover, we also discuss the goals of these works, having found that most proposals try to abstract the specific cloud vendor technologies.

## CCS CONCEPTS

• **Software and its engineering** → *System modeling languages*; *Design languages*; *Specification languages*; • **General and reference** → **Empirical studies**.

## KEYWORDS

DevOps, model-driven engineering, systematic studies

**ACM Reference Format:**
Hugo da Gião, Rui Pereira, and Jácome Cunha. 2023. Model-Driven Approaches for DevOps: A Systematic Literature Review. In *Proceedings of The 24th International Conference on Evaluation and Assessment in Software Engineering (EASE 2023)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX


*EASE 2023, 14–16 June, 2023, Oulu, Finland*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

DevOps is a means of software development and delivery that uses various tools and techniques to integrate the worlds of development and operations [18]. Indeed, the word DevOps comes from the combination of Development with Operations. This term describes a culture where these two realms are no longer separated as in the past but are now just one. Integrating these two fields involves achieving a high level of automated development, build, deployment, and monitoring of software. The goals include achieving reliable, secure, fast, and continuous software delivery to improve business productivity and workers' well-being. Moreover, software engineers that use DevOps are more resilient to fast changes [29].

However, organizations and developers face many obstacles when adopting DevOps, including changes in the architectural organization, dealing with problems in older infrastructures, and the integration of different DevOps tools [22]. Other problems developers face are learning and using specific DevOps tools, which can be overwhelming and can decrease the improvements expected with the adoption of DevOps [25].

Model-driven engineering (MDE) approaches help cope with numerous problems in software development, from aiding in the design process to generating code for different platforms to improving the communication between different teams and team members [50]. As a partial system representation, models can ignore details and focus on the relevant parts in each software development and maintenance stage. Thus, it seems a very reasonable approach to the complex problem of DevOps use.

This paper presents the first systematic literature review (SLR) reporting the MDE proposals for DevOps. Our goal with this study is to summarize the work done at the intersection of these fields. Our goal is also to identify some possible contributions in this space. In particular, we seek to answer the following research questions (RQs):

**RQ1** Which computing domains are covered by model-driven approaches for DevOps?
**RQ2** Which phases of DevOps are covered by model-driven approaches?
**RQ3** What are the goals of the model-driven approaches for DevOps?

In our work, we seek to i) know the domains in which model-driven approaches for DevOps are used and where they are still lacking, ii) which phases of DevOps are covered by MDE approaches, and again, the ones that are still unexplored, and iii) which are the goals of these works.

With RQ1, we want to understand the field better, and MDE already covers which kinds of domains and approaches for DevOps and, as a consequence, where they still need to be improved. The 37 works found cover eight computing domains, from cloud to blockchain, but we also found some domain-agnostic proposals.

DevOps is applied in different phases [59] of software construction and maintenance. Thus, with RQ2, we seek to understand which of these phases are covered by model-driven approaches and, again, the ones that are still unexplored. We have found that MDE covers all seven phases of DevOps approaches, although deployment is the most addressed.

We also propose to understand the goals of these works better. In RQ3, we found that several objectives exist, from improving the development or deployment phases of DevOps. However, these works cope with the different characteristics, technologies, and languages available in the cloud vendors' solutions.

We organized the remainder of this paper as follows. We present related work in 2. Afterward, in 3, we present the methodology we followed in this work. In 4, we present the results of the systematic study, from the papers found to the analysis of the different dimensions of the papers, answering the RQs. We then in 5, present threats to the validity of our work. Finally, in 6, we present some concluding remarks.

## 2 RELATED WORK

Several literature reviews exist for both the fields of DevOps and MDE, but none address the use of MDE for DevOps.

Relating to DevOps, one survey focuses on finding and understanding different challenges from the perspective of engineers, managers, and researchers [33]. Their process was to search scientific libraries, select papers, and then snowball the papers found. They organized the results into conceptual maps. Another work showcases a list of security challenges and practices for DevOps software development gathered from a systematic literature review [34]. Other researchers have been working on understanding the DevOps culture by doing an SLR of various DevOps works [45]. Others have done work to understand the best practices [4], pitfalls [40], and benefits [12] of DevOps. Some aim at studying the impact DevOps has on software quality [13]. However, it has yet to address the use of MDE in DevOps.

In MDE, various surveys exist to gather and summarize current contributions to the field. Two of the publications review 140 research papers to understand the uses of MDE across the development lifecycle of cyber-physical systems and identify research gaps and areas that need more investigation [39]. Another study reviews the use of meta-models of elicitation techniques [26]. Researchers have also been working to understand the benefits of MDE in the software industry and how existing processes can be reused [27]. The use of MDE for reverse engineering was also studied [43]. The application and challenges of MDE in cloud computing have also been studied [41]. A systematic mapping study exists to understand the role and challenges of MDE for the development of safety and security systems [36].

A systematic review of the use of requirements engineering in model-driven development has been made [35]. One publication did the work of understanding how DevOps is used to improve MDE methods [42]. The authors analyzed several case studies from their life experiences and literature. In the paper, the authors also describe possible automation areas in the MDE process. Finally, the authors identify possible future work and research areas in this field.

However, to our knowledge, no survey or systematic study exists to understand the use of MDE in DevOps.

## 3 RESEARCH METHOD

Our research method consisted in realizing a SLR where we queried four major digital libraries, namely the *ACM Digital Library*[1], the *IEEE Xplore*[2], *ScienceDirect*[3] and *SpringerLink*[4], for papers containing both topics of interests, namely MDE and DevOps. We did this work on January 2023. Thus, we started by defining an initial query for searching these libraries for relevant publications as follows:

DevOps AND ("model-driven" OR "model driven")

After running the query (the number of papers found is in the second column of Table 1), after some initial screening of the papers, we realized several other terms related to DevOps exist, namely 'MLOps,' 'DevSecOps,' 'CloudOps,' 'AIOps,' and 'DataOps,' which represent various fields of application of DevOps. Thus, we decided also to include them in the search query. Regarding the model part, we did not include terms such as "model-based" or similar. Since our focus is on model-driven approaches, that is, approaches for which models are paramount in contrast with model-based approaches where models are used in a more lightweight way. Thus, we extended the original query as follows:

((DevOps OR MLOps OR ChainOps OR DevSecOps OR CloudOps OR AIOps OR DataOps)
AND
("model-driven" OR "model driven"))

Running this query, we found the number of papers reported in the third column of Table 1. Note that we have run the query against the complete text of the publications (title, abstract, full text, etc.) to increase the chances of finding all the relevant papers.

**Table 1: Number of papers found in each library.**

| Venue | First query | Second query |
|---|---|---|
| **ACM Digital Library** | 135 | 138 |
| **IEEE Xplore** | 34 | 34 |
| **ScienceDirect** | 104 | 105 |
| **SpringerLink** | 518 | 549 |
| **Total** | 718 | 744 |

Indeed, the addition we made to the query produced more results.

After gathering the papers containing the search terms, we manually filtered, by screening the content, the papers whose work involved the application of MDE techniques in various areas of DevOps using the following inclusion and exclusion criteria:

**Inclusion criteria**

[1]https://dl.acm.org
[2]http://ieeexplore.ieee.org
[3]https://www.sciencedirect.com
[4]https://link.springer.com

- IC1 The work in the paper applies MDE to the field of DevOps. We accepted in our review any work that uses MDE techniques to improve the use of DevOps. This includes any publication whose work creates or improves upon techniques, tools, modeling languages and frameworks for using MDE in any aspect of DevOps. We included both domain-agnostic and specific approaches in our literature review.
- IC2 The paper exposes the requirements for using MDE in DevOps.
  We accepted in our review any work that tries to understand how MDE techniques can improve the state of DevOps. This includes any publication whose work gathers requirements from industrial or academic case studies for using MDE in any aspect of DevOps. We included both domain-agnostic and specific approaches in our literature review.

**Exclusion criteria**

- EC1 The work does not address the topics of DevOps and MDE. We rejected any publications whose work did not contribute to any of the fields of DevOps or MDE.
- EC2 The paper applies DevOps techniques to MDE, but not the other way around.
  Several works exist that apply DevOps techniques to improve several aspects of MDE. Since those works are not the focus of this literature review, we dismissed them.
- EC3 Secondary, tertiary, or survey works.
  These works were also dismissed from this literature review since our goal is to evaluate original primary works and report on the state of the art of MDE and DevOps. We feature these works in the related work section of this paper.
- EC4 (Extended) abstracts.
  We dismissed abstracts or extended abstracts since they did not provide enough information to evaluate their contributions to this field.
- EC5 Proceedings.
  We also dismissed proceedings of conferences since they contain several publications, most of which are irrelevant to our query. The ones that are should appear in the search on their own.

After applying the inclusion and exclusion criteria to the initial 718 papers, we ended up with 37 publications. Overall, we found a stable but growing interest in this research topic, as seen in Figure 1, with the publications starting in the year 2015. 2020 is an exception, with quite some new papers (10).

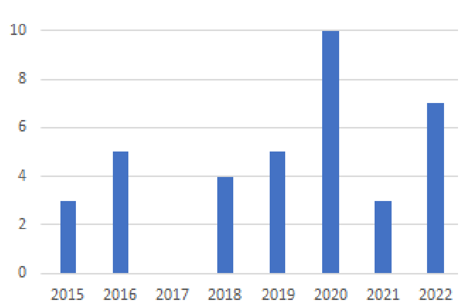


**Figure 1: Publications per year.**

Finally, we read the 37 papers and organized them according to their technical domain and DevOps phase. In the next section, we detail our findings.

## 4 RESULTS

In this section, we present the results we found in detail, starting by answering RQ1.

### 4.1 RQ1: What computing domains are covered by model-driven approaches for DevOps?

To answer this RQ, we read in detail each of the 37 papers and collected the domains which the papers address. We gathered this information by analyzing the papers collected, and it was not always explicit. When the domain of the paper was not explicit, we interpreted the contents of the paper and classified it according to the contributions made by the paper. We started with no domains, and for each paper, we either devised a new domain or classified the work according to one of the already collected domains. We classified papers whose work contributed to several domains as the most relevant to their work. In the following, we describe these works grouped by domain.

*Cloud.* Of the 37 publications under study, 17 (45.9%) address the cloud domain, being this the most coveted domain.

Regarding the modeling language used, most of these projects adapted existing languages already used in the cloud computing domain, such as TOSCA [3, 8, 11, 14, 47–49, 58]. We also found that the use of meta-models was prevalent in these publications [3, 8, 14, 20, 47–49, 58].

We also found that the generation of infrastructure code was prevalent across these papers since most of them aim at solving concrete technical problems across this domain. In this domain, saving time and reducing errors associated with using specific cloud tools was a motivation to adopt MDE and automate processes.

*Internet of Things.* We found only four publications (10.8% of all papers) in the Internet of Things (IoT) space. One of these publications proposes a meta-model capable of standardizing current and future IoT architectures [19]. Another project tackles the problem of deploying multiple gateways and connected IoT devices automatically using a model-driven approach [52]. Another group of authors created a modeling infrastructure for the systematic engineering of IoT applications [31]. The other publication uses MDE to improve liability, quality, security, and privacy in the DevOps process of IoT systems [21]. The authors extend an existing modeling language to support the continuous deployment of IoT security and privacy mechanisms on the Edge.

*Safety-Critical Systems.* For safety-critical systems, we found just 1 article (2.7% of the studied papers) [38]. In this work, the authors propose a modeling framework compatible with the DevOps principles of continuous testing and continuous integration to design safety-critical systems while obeying industry standards. The authors define their modeling language using a meta-model in this work. The authors also express that the ability to reason and make proofs over the models was one factor for their use of MDE.

*Cyber-Physical Systems.* In terms of cyber-physical systems (CPS), we found three publications (8.1%) whose focus was to model the whole development lifecycle of these systems [9, 17, 28].

Two of these publications created their languages through meta-models. Once more, in both papers, the authors express that the ability to reason and make proofs over the models was one factor in their use of MDE. Another publication uses MDE to provide AI-augmented automation capabilities to the continuous development of cyber-physical systems [9].

*Blockchain.* In the blockchain domain, we found one publication whose work consisted in developing a model-driven approach to the creation of intelligent contracts [55]. The authors designed their language, making use of a meta-model. Once again, the ability to reason and make proofs over the models was one factor for their use of MDE. With their approach, non-expert users can generate the code for the distributed blockchain applications and the necessary operations.

*Machine Learning.* In this space, we found four publications [5, 6, 30, 54].

One of the works treats model abstractions of machine learning models as first-class citizens and promotes end-to-end transparency and portability from raw data detection to model verification and policy-driven model management [54]. The authors created their modeling language, used meta-models, and produced code.

Another work uses MDE to facilitate runtime updates of the architecture and orchestration of autonomous machine learning applications according to the desired specification and changes in the environment [5]. The authors of this work use models at runtime to update to generate the infrastructure necessary for the model to fit the given specifications in the changing environment.

One of those works defines a list of industrial requirements for the support of AI-driven MDE. The authors worked on an industrial project in which researchers and practitioners worked in AI-augmented automation to support modeling, coding, testing, monitoring, and continuous development in CPS. Through elicitation and refinement, 78 high-level requirements were defined and generalized into 30 generic requirements [6].

Finally, one work uses MDE to develop machine learning systems focusing on IoT. In their work, the authors illustrate how two state-of-the-art open-source modeling tools are used for this purpose using a case study. Their case study involves using artificial neural networks to automate the recognition of digits integrated into IoT systems. The authors focus on developing the software model, and their approach does not generate code [30].

*Data Science.* In this field, one project applied MDE techniques to improve the reproducibility and replicability of data science projects. The authors created a model-driven framework that allows users to define data science models in a platform-independent way by separating the definition of the modeling and operation layers [37].

Using their approach, users can generate code and DevOps pipelines in a way that improves the process applicability, automates the process execution, and enhances process reproducibility. This allows for the definition of model verification rules and provides intentionality restrictions defined by the user [37].

*Big Data.* Regarding big data, we found one publication [10]. The language created by the authors is of their design, and they use a meta-model to define its abstract syntax. This publication aims to use MDE techniques for data-intensive applications. To do so, the authors present their research agenda for DICE, an MDE methodology for designing and orchestrating data-intensive applications. They aim to develop a quality engineering toolchain that offers simulation, verification, and architectural optimization for big data applications [10].

*Domain-Agnostic.* We found 5 publications (13.5%) whose work is not restricted by any particular domain [7, 15, 16, 44, 53]. In the approach of one of these works [16], developers can use the proposed framework to create processes and platforms, a platform, a sequence of processes, and a process of an application or service. However, their approach depends on the tools and the platform used and does not offer a generic approach for each DevOps phase and the entire lifecycle. This work is based on a previous work [7] that derived requirements for modeling DevOps from an industrial case study. These requirements include general, description and analysis, and simulation requirements. Both have their languages and use meta-models [7, 16].

Another paper [15] introduces an approach that uses MDE and blended modeling to simulate DevOps processes using an industrial case study. In the paper, the authors present their work in progress in which they create a framework that allows tools and DevOps models to be defined using JSON. Domain experts and users who do this have access to a graphical interface to define and simulate DevOps pipelines.

We also found a project [53] whose goal was to aid the management of software projects by applying model-driven techniques. In this work, the authors use MDE to predict the impact of adopting new technologies into the developer workflow. Their work touches on the operations phase of the DevOps lifecycle.

Another project uses UML to automate the deployment [44]. The authors use UML to increase software delivery frequency and quality in their approach. They do so by using MDE to automate deployment processes and specifying a software architecture and the deployment behind it. The authors do so by extending UML with features capable of capturing the semantics and requirements of software components' installation, configuration, and update.

### 4.1.1 *Summary.*

In the 37 papers analyzed, we found eight computing domains, plus some works that are domain-agnostic and can potentially be used in different domains. Indeed, 32 papers (86.5%) present domain-specific approaches. This may indicate that creating a model-driven generic approach for DevOps is not the best approach, being a more specific proposal a better path.

The cloud is the domain to which most researchers dedicate their time, accounting for almost half of all work. In line with the previous reasoning that a more domain-specific approach is more fruitful, many domains are still not yet covered or are just slightly addressed (e.g. safety-critical systems, blockchain). This indicates that there is still space for proposals, especially in domains other than the cloud.

Several works reused or adapted existing modeling languages, but many others created their own. Moreover, many of the papers presented a meta-model which makes extending the language and the approach easier. One of the main reasons claimed by the authors for using a model-driven approach is that it is possible to reason using the models and make proofs about the underlying approaches.

## 4.2 RQ2: Which phases of DevOps are covered by model-driven approaches?

In this section, we analyzed the various articles and organized them according to the phases of DevOps addressed by each one. In this case, we used the DevOps phases proposed by Zhu *et al.* [59]: development, integration, testing, monitoring, feedback, deployment, and operations. Note that each paper can cover more than one phase.

*Development.* The phase of continuous development involves the tasks of planning and developing software. This process is an application of the agile methodology to the world of software development. The software is separated into smaller components and updated in smaller batches after being tested [59].

We found several works relating to this phase. These works span almost all domains (except safety-critical systems): cloud [47], IoT [19, 31, 52], cyber-physical systems [17, 28], blockchain [55], machine learning [5, 6, 30, 54], data science [37], big data [10], as well as domain-agnostic approaches [7, 15, 16].

*Integration.* This phase is at the core of the DevOps lifecycle and affects the process of committing new changes to the source code. When doing continuous integration, developers need to do more frequent software releases in conjunction with various testing techniques, such as unit testing and integration testing. Developers also use other quality assurance practices such as code reviews to make the software more secure and stable [59].

We found several works addressing this phase across various domains: cloud [47], IoT [19, 31, 52], cyber-physical systems [17, 28], blockchain [55], data science [37], machine learning [6, 30], big data [10], as well as domain-agnostic approaches [7, 15, 16].

*Testing.* Continuous testing involves the use of automatic testing tools such as Selenium or JUnit [59]. This process saves organizations time and increases productivity [59].

Several of the works studied address this phase. These works span various fields of computing including cloud computing [32], IoT [31], safety-critical systems [38], cyber-physical systems [17, 28], blockchain [55], data science [37], machine learning [6, 30], as well as domain-agnostic approaches [7, 15, 16, 44].

*Monitoring.* This phase tackles performance monitoring and recording of the application, including potential errors such as high memory usage and networking problems. This monitoring allows development teams to find and correct errors earlier and substantially reduce costs [59].

We found several works relating to this phase in different domains such as IoT [31], cyber-physical systems [9, 17, 28], safety-critical systems [38], blockchain [55], data science [37], machine learning [6, 30], big data [10]. Again, some domain-agnostic approaches also address this phase [7, 15, 16].

*Feedback.* This phase comprises gathering, analyzing, and using the client's software usage feedback. This information includes performance and errors and customer feedback [59].

Some of the works analyzed cope with this phase, spanning various domains, including cloud [24], IoT [31], safety-critical systems [38], cyber-physical systems [17, 28], blockchain [55], data science [37], machine learning [6, 30], big data [10], as well as domain-agnostic approaches [7, 15, 16].

*Deployment.* Continuous deployment affects the process of deploying new code to the production environment. This process is made automatically and reduces the need for manual and planned releases since new changes to the source code are sent into production after integration and testing. This phase extensively uses container technology such as Docker to make use of standardized environments [59].

Most of the works addressing this phase are in the cloud domain [2, 3, 8, 11, 14, 20, 23, 47–49, 51, 56–58]. However, we have also found works in other areas including IoT [31], safety-critical systems [38], cyber-physical systems [17, 28], blockchain [55], machine learning [6, 30], data science [37], big data [10], as well as domain-agnostic approaches [7, 15, 16].

*Operations.* This phase includes automating all operation processes that help developers automate releases, detect issues faster and improve software products. This practice allows for faster delivery times and higher quality software [59].

We found several works relating to this phase. These works span various domains from cyber-physical systems [17, 28], to blockchain [55], data science [37], machine learning [6, 30], as well as domain-agnostic approaches [7, 15, 16].

*4.2.1 Summary.* We found considerably more papers for the deployment phase, 27 to be exact (73.0%) than for the other phases. This is due to the considerable work done to improve compatibility between different cloud providers.

We have also found several projects, nine (24.3%), that tackled the whole DevOps lifecycle.

Finally, we found that the monitoring, integration, and operations phases have the least amount of work and, thus, more space for new contributions.

## 4.3 RQ3: Which are the goals of the model-driven approaches for DevOps?

In this section, we discuss the goals of the works under analysis. For each paper, we present its goal and the domain and the DevOps phases the work tackles so one can better understand the goal. We intend to view better objectives already conquered by applying MDE to DevOps. These results can be overviewed in Table 2 and detailed analysis.

*Abstract cloud vendors.* One of the main objectives of these works is to *abstract cloud vendors* particularities so the management of could(-deployed) applications can be made in an agnostic way of the particular cloud supplier. These particularities include lock-in issues with auto-scaling configurations [1, 2] and infrastructure provisioning using a domain-specific language that converts to code for different providers [49]. However, most of the works focus

**Table 2: Goals found in the different papers under analysis. Dev abbreviates the Development phase, Int the Integration, Test the Testing, Mon the Monitoring, Feed the Feedback, Dep the Deployment, Op the Operations, and All for all phases. We represented certain domains as abbreviations, including Machine learning as ML, Cyber-physical systems as CPS, and Safety-critical systems as SCS Each publication is represented by its first author's initials and the year of publication.**

| Goals | Domain | Phases | Paper |
|---|---|---|---|
| Facilitate fast service deployment | Cloud | Dev | H.A, 2016 [1] |
| Abstract and automate a continuous delivery process of cloud resources | Cloud | Dep, Int | J.S, 2018 [47] |
| Optimization and runtime capacity allocation of cloud applications | Cloud | Feed, Dep | M.G, 2015 [23] |
| Continuous deployment for quality DevOps | Cloud | Dep | M.A, 2016 [3] |
| Improve compatibility between different cloud resources | Cloud | Dep | D.W, 2016 [56] |
| Orchestration for cloud resources | Cloud | Dep | H.B, 2019 [8] |
| Manage infrastructure as code | Cloud | Dep | J.S, 2019 [49] |
| Low priority of data security and privacy assurance in agile environments | Cloud | Test | R.K,2020 [32] |
| DevSecOps using open-source software over the cloud | Cloud | Feed, Dep | M.G, 2015 [24] |
| Managing cloud resources | Cloud | Dep | D.W, 2016 [57] |
| Manage multi-cloud applications | Cloud | Dep | N.F, 2018 [20] |
| Auto-scaling services on multiple clouds | Cloud | Dep | H.A, 2018 [2] |
| Cloud resource management | Cloud | Dep | S.C, 2020 [14] |
| Automatically assigning multiple software deployments | Cloud | Dep | H.S, 2020 [51] |
| Streamlining DevOps automation for cloud applications | Cloud | Dep | J.W, 2016 [58] |
| Infrastructure provisioning for multiple clouds | Cloud | Dev | J.S, 2019 [48] |
| Rational decomposition and orchestration for serverless computing | Cloud | Dep | G.C, 2020 [11] |
| Fleet deployment in the IoT | IoT | Dep | S.H,2020 [52] |
| Continuous deployment of IoT systems | IoT | Dep | N.F, 2019 [21] |
| Standardize existing IoT architectures | IoT | Dev, Int | B.E, 2020 [19] |
| Development and deployment of reliable IoT applications | IoT | Dev, Dep | J.K,2022 [31] |
| Design of safety-critical systems | SCS | Test, Mon, Feed | B.M, 2019 [38] |
| Unified model-based engineering, digital twins, and DevOps practice | CPS | All | J.H, 2020 [28] |
| DevOps environment for multi-cloud applications | CPS | All | B.C, 2020 [17] |
| Digital twins for the engineering of CPS | CPS | Mon | H.B, 2022 [9] |
| Allow non-experts to develop smart contract-based distributed applications | Blockchain | All | W.J, 2021 [55] |
| MLOps for intelligent enterprise applications | ML | Dev | W.J, 2020 [54] |
| Modeling autonomic systems for ML and microservices | ML | Int; Dep | N.B, 2021 [5] |
| Reporting on industrial requirements & machine learning for supporting AI-enhanced MDE | ML | All | J.B, 2022 [6] |
| Adopt MDE for the development of machine learning-enabled IoT software | ML | All | J.K, 2022 [30] |
| Reproducibility and replicability of data science projects | Data Science | All | F.M,2022 [37] |
| Quality-driven development of data-intensive cloud applications | Big Data | Dev,Int,Mon,Feed,Dep | G.S, 2015 [10] |
| Gathering requirements for a modeling framework for domain agnostic DevOps | Domain agnostic | All | F.B, 2020 [7] |
| Blended modeling and scenario simulation of continuous delivery pipelines | Domain agnostic | All | F.B, 2020 [15] |
| Modeling framework for DevOps that can be reusable across different domains | Domain agnostic | All | A.C, 2020 [16] |
| Acceptance tests to support software measurement | Domain agnostic | Test, Op | L.F, 2018 [44] |

on the management of cloud resources or infrastructure [8, 14, 20, 23, 48, 48, 49, 56, 57]. In general, most of these articles aim at creating a cloud platform-agnostic way of managing code and doing different operations in different cloud services, improving reusability between those services, and hiding technical details from the users. We also found that generating code and having and generating a concrete DevOps pipeline is a goal for those works.

*Improve deployment.* Several works set as goal to *improve the deployment* phase [1, 3, 20, 21, 31, 44, 51]. These different span domains, but most of them focus on the cloud. These works often aim to abstract from several technical tasks relating to deploying software to the cloud. This abstraction is useful to improve compatibility between different cloud providers. Other works focus on the deployment phase and provide an integrated approach for doing DevOps, either generic or for a particular field.

*Quality.* Four works explicitly refer to *quality* as an objective[5]. One of the papers uses MDE to create a framework that allows the development of data-intensive software to support reliability, efficiency, and safety requirements [10]. One other uses continuous deployment to increase the quality of the software [3]. Two works explicitly address reliability, which can be seen as a characteristic of quality [10, 31].

*Serverless computing.* Two works try to cope with *serverless* technologies, either by aiding in creating and managing applications based on such paradigm [11] or by aiding in the decomposition and orchestration in such an environment [58].

*Multi-cloud applications.* The management of *multi-cloud applications* is addressed in two works [17, 20]. In the work in the cyber-physical systems' domain [17], the author's motivation is the fact that the various components of cyber-physical systems often run in several heterogeneous environments during their lifespan, and the author aims to provide a layer of abstraction for those different environments to facilitate the development process. The other paper [20] tries to solve the problem of dealing with various cloud providers when developing large-scale distributed applications due to the need for more compatibility between the said providers. Their solution was to create a framework and environment capable of facilitating the deployment of multi-cloud applications.

*Security.* Two works focus on *security*. One consists of a conceptual security model to facilitate the adoption of DevSecOps[6].) for the business processes over the cloud [24]. Another work focuses on improving teams' data security and privacy assurance using agile methodologies [32].

*Design.* Although model-driven is widely used for designing software, we could only find two papers explicitly setting *design* as a goal, one for designing safety-critical system [38] Moreover, another allows users to graphically design cloud applications and deploy and manage them at runtime [14].

*Improving development.* A single publication also addresses the *development* process [37]. The authors aim to improve the development process by providing a pipeline definition of data science projects agnostic of different technologies.

*Aiding non-experts.* One paper focuses on *aiding non-experts* to develop smart contract-based distributed applications [55].

*Standardization.* Finally, one paper has a primary goal to *standardize* current and future IoT architectures [19].

*4.3.1 Summary.* Answering RQ3, we found several goals in the analyzed papers, namely the creation of could vendor agnostic framework to allow abstracting each vendor's specificities and improving DevOps phases such as development and deployment. We also found the goals of increasing software quality, coping with multi-cloud applications and the serverless paradigm, improving software security, aiding non-experts, and working through standardization.

Some of these objectives are domain-specific, for instance, those more related to cloud vendors, but others can exist in different domains (e.g., deployment, development).

Critical in online and distributed applications, security is only directly addressed in one case. This indicates that model-driven approaches for DevOps to improve security are still very much unexplored.

With the current trend of low/no-code platforms, which rely on modeling, we found only one paper about helping non-experts. This also indicates that DevOps is still seen to be used only by experts, which creates opportunities to research how it can be made available to people with less expertise in DevOps.

Some of the main benefits of DevOps include improving developers' productivity and reducing costs. On the other hand, an MDE approach includes various benefits, from automation to reusability and replicability, to better cohesion between team members. Indeed, most of the work's goals are to improve further aspects that motivate and are the main benefits of using DevOps.

## 4.4 Relation between domains and phases

In this section, we perform an analysis orthogonal to the research questions. In particular, we identify several facts in the intersection of the domains and the DevOps phases. The distribution of the publications according to the identified domains and the DevOps phases is presented in Table 3.

For the papers relating to the cloud computing domain, we can observe from the table that most contributions focus on enhancing and managing existing cloud computing processes. Those proposals often involve using MDE to improve or provide reusability of processes across various platforms and also the creation of platform-agnostic models to do so. The contributions made in this domain are primarily focused on the deployment phase. The vast majority of the contributions are also intended for technical users with previous experience in improving developers' productivity.

Other projects offer contributions outside of the deployment phase. These include proposing a model-driven approach to abstract and automate a continuous delivery process of cloud resources in development, testing, and production environments, which touches the development, integration, testing, feedback, deployment, and operations phases [47]. Another project that reached the same phases consists of a conceptual security model to facilitate the adoption of DevSecOps for the business processes over the cloud [32].

In the IoT domain, most of the contributions focus on improving the development process. Those contributions focus mainly on the deployment and development phases of the DevOps process.

When it comes to safety-critical systems, the work done in this area focuses on the various phases of the DevOps process and aims at developing a model-driven approach that allows users to perform their tasks while abstracting from concrete technologies.

In terms of cyber-physical systems, we found two publications whose focus is to model the whole development lifecycle of these systems [17, 28]. This is also the case of the work in blockchain [55] and the one in data science [37]. These three domains are the

[5]One can admit that DevOps and model-driven approaches can implicitly help to achieve several goals, from complexity management to fast deployment to better security. However, we focus on the goals researchers explicitly expressed in the papers.
[6]DevSecOps is a variant of DevOps, but including security controls to provide continuous security assurance [46]

**Table 3: Publications organized according to their domains and DevOps lifecycle phases.**

| Domain/Phase | Develop. | Integration | Testing | Monitoring | Feedback | Deployment | Operations |
|---|---|---|---|---|---|---|---|
| **Cloud** | J.S, 2018 [47] | J.S, 2018 [47] | R.K, 2020 [32] | | M.G, 2015 [24] | H.A, 2016 [1]<br>M.G, 2015 [23]<br>D.W, 2016 [56]<br>J.S, 2019 [49]<br>D.W, 2016 [57]<br>N.F, 2018 [20]<br>H.A, 2018 [2]<br>S.C, 2020 [14]<br>M.A, 2016 [3]<br>H.B, 2019 [8]<br>G.C, 2020 [11]<br>J.W, 2016 [58]<br>J.S, 2019 [48]<br>H.S, 2020 [51] | |
| **IoT** | B.E, 2020 [19]<br>J.C,2021 [31]<br>H.S,2022 [52] | B.E, 2020 [19]<br>J.C,2021 [31] | J.C,2021 [31] | J.C,2021 [31] | J.C,2021 [31] | J.C,2021 [31]<br>SN.F, 2019 [21] | |
| **Safety-Critical Systems** | | | B.M, 2019 [38] | B.M, 2019 [38] | B.M, 2019 [38] | | |
| **Cyber-Physical Systems** | B.C, 2020 [17]<br>J.H, 2020 [28] | B.C, 2020 [17]<br>J.H, 2020 [28] | B.C, 2020 [17]<br>J.H, 2020 [28] | B.C, 2020 [17]<br>J.H, 2020 [28]<br>H.B, 2022 [9] | B.C, 2020 [17]<br>J.H, 2020 [28] | B.C, 2020 [17]<br>J.H, 2020 [28] | B.C, 2020 [17]<br>J.H, 2020 [28] |
| **Blockchain** | W.J, 2021 [55] | W.J, 2021 [55] | W.J, 2021 [55] | W.J, 2021 [55] | W.J, 2021 [55] | W.J, 2021 [55] | W.J, 2021 [55] |
| **Machine Learning** | J.B, 2022 [6]<br>J.K, 2022 [30]<br>W.J, 2020 [54]<br>N.B, 2021 [5] | J.B, 2022 [6]<br>J.K, 2022 [30] | J.B, 2022 [6]<br>J.K, 2022 [30] | J.B, 2022 [6]<br>J.K, 2022 [30] | J.B, 2022 [6]<br>J.K, 2022 [30] | J.B, 2022 [6]<br>J.K, 2022 [30] | J.B, 2022 [6]<br>J.K, 2022 [30] |
| **Data Science** | F.M, 2022 [37] | F.M, 2022 [37] | F.M, 2022 [37] | F.M, 2022 [37] | F.M, 2022 [37] | F.M, 2022 [37] | F.M, 2022 [37] |
| **Big Data** | G.S, 2015 [10] | G.S, 2015 [10] | | G.S, 2015 [10] | G.S, 2015 [10] | G.S, 2015 [10] | |
| **Domain Agnostic** | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15]<br>L.F, 2018 [44] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15] | F.B, 2020 [7]<br>A.C, 2020 [16]<br>A.C, 2022 [15]<br>K.T, 2022 [53] |

only ones where all the DevOps phases are covered by the same work (except for the domain-agnostic approaches).

We found four publications whose work is not dedicated to any particular domain. Two of these publications span the whole DevOps lifecycle [7, 16]. The goal of one of the other articles [16] is to build upon those requirements to create a conceptual framework for modeling and combining DevOps processes and platforms. In their approach, developers can use the proposed framework to create processes and platforms, however, their approach is dependent on specific tools and technologies. Another approach [15] also does a similar job using a JSON interface for specifying tools and platforms instead of a visual platform. Another project's goal is to aid the management side of software projects by applying model-driven techniques. Their work touches on the operations phase of the DevOps lifecycle [53].

*4.4.1 Summary.* Only 9 papers (24.3%) cross all DevOps phases, 2 in the CPS domain, 1 in blockchain, 2 in machine learning, 1 in data science, and 3 domain-agnostic. This may seem to indicate that creating an approach that copes with all the phase is too difficult or not interesting as a solution.

We can also conclude that several intersections are not covered. For instance, the operations phase for cloud, IoT, safety-critical systems, and big data are not addressed by any work. In fact, operations are the most neglected phase.

Finally, we can see that the safety-critical systems' domain is only addressed by one work and only three phases. This means that researchers can focus their work on this domain and in (almost) all the DevOps phases.

## 5 THREATS TO VALIDITY

Despite our intention and efforts to include all publications related to the application of MDE for DevOps, there is always the risk of missing some relevant papers. To mitigate this risk, we have run our query against the publications' entire content (text), including the citations. Thus, a publication addressing DevOps and MDE would need no reference to such terms that do not appear in our search, which is not likely. Moreover, we searched the four main digital libraries in computer science. Note that some of them (e.g. ACM) also index publications that are not published by themselves, and thus the number of listed papers is even higher.

Another possible threat to the validity of our approach is that not all publications have explicit domains or DevOps phases. To mitigate this risk, we read the articles and implicitly found the phases and domains for the articles in which those were not explicit. The same author always did this to guarantee that the criteria were consistent.

Another possible threat is the application of the inclusion and exclusion criteria, as this is inherently subjective. To mitigate this, we defined each criterion clearly, as seen in Section 3. Once more, this was also always done by the same author to ensure consistency.

## 6 CONCLUDING REMARKS AND FUTURE WORK

Based on our research, most papers proposing MDE approaches for DevOps tackle the cloud domain, specifically in the deployment phase. Nevertheless, we also found that some papers address the other phases of the same domain but in less quantity. We also found that most of the papers relating to the deployment phase of cloud computing focus on interoperability between cloud services.

We also learned that using meta-models was prevalent and found that the ability to reason and create proofs over models was a factor in using model-driven approaches in safety-critical systems, cyber-physical systems, and blockchain.

Regarding domain-agnostic approaches, we found that some authors acknowledge the need for a model-driven approach to the DevOps process and that existing frameworks still rely on specific technologies.

## REFERENCES

[1] Hanieh Alipour and Yan Liu. 2016. A Model Driven Method to Deploy Auto-Scaling Configuration for Cloud Services. In *Proceedings of the 4th International Workshop on Release Engineering* (Seattle, WA, USA) *(RELENG 2016)*. Association for Computing Machinery, New York, NY, USA, 23. https://doi.org/10.1145/2993274.3011285
[2] Hanieh Alipour and Yan Liu. 2018. Model Driven Deployment of Auto-Scaling Services on Multiple Clouds. In *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*. 93–96. https://doi.org/10.1109/ICSA-C.2018.00033
[3] Matej Artač, Tadej Borovšak, Elisabetta Di Nitto, Michele Guerriero, and Damian A. Tamburri. 2016. Model-Driven Continuous Deployment for Quality DevOps. In *Proceedings of the 2nd International Workshop on Quality-Aware DevOps* (Saarbrücken, Germany) *(QUDOS 2016)*. Association for Computing Machinery, New York, NY, USA, 40–41. https://doi.org/10.1145/2945408.2945417
[4] Sher Badshah, Arif Ali Khan, and Bilal Khan. 2020. Towards Process Improvement in DevOps: A Systematic Literature Review. In *Proceedings of the Evaluation and Assessment in Software Engineering* (Trondheim, Norway) *(EASE '20)*. Association for Computing Machinery, New York, NY, USA, 427–433. https://doi.org/10.1145/3383219.3383280
[5] Nelly Bencomo. 2021. Modeling Autonomic Systems in the time of ML, DevOps and Microservices. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 731–731. https://doi.org/10.1109/MODELS-C53483.2021.00118
[6] Johan Bergelin and Per Erik Strandberg. 2022. Industrial Requirements for Supporting AI-Enhanced Model-Driven Engineering. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Montreal, Quebec, Canada) *(MODELS '22)*. Association for Computing Machinery, New York, NY, USA, 375–379. https://doi.org/10.1145/3550356.3561609
[7] Francis Bordeleau, Jordi Cabot, Juergen Dingel, Bassem S. Rabil, and Patrick Renaud. 2020. Towards Modeling Framework for DevOps: Requirements Derived from Industry Use Case. In *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Jean-Michel Bruel, Manuel Mazzara, and Bertrand Meyer (Eds.). Springer International Publishing, Cham, 139–151.
[8] Hayet Brabra, Achraf Mtibaa, Walid Gaaloul, Boualem Benatallah, and Faiez Gargouri. 2019. Model-Driven Orchestration for Cloud Resources. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. 422–429. https://doi.org/10.1109/CLOUD.2019.00074
[9] Hugo Bruneliere, Vittoriano Muttillo, Romina Eramo, Luca Berardinelli, Abel Gómez, Alessandra Bagnato, Andrey Sadovykh, and Antonio Cicchetti. 2022. AIDOaRt: AI-augmented Automation for DevOps, a model-based framework for continuous development in Cyber–Physical Systems. *Microprocessors and Microsystems* 94 (2022), 104672. https://doi.org/10.1016/j.micpro.2022.104672
[10] Giuliano Casale, Danilo Ardagna, Matej Artac, Franck Barbier, Elisabetta Di Nitto, Alexis Henry, Gabriel Iuhasz, Christophe Joubert, Jose Merseguer, Victor Ion Munteanu, Juan Fernando Perez, Dana Petcu, Matteo Rossi, Craig Sheridan, Ilias Spais, and Daniel Vladuic. 2015. DICE: Quality-Driven Development of Data-Intensive Cloud Applications. In *2015 IEEE/ACM 7th International Workshop on Modeling in Software Engineering*. 78–83. https://doi.org/10.1109/MiSE.2015.21
[11] G. Casale, M. Artač, W.-J. van den Heuvel, A. van Hoorn, P. Jakovits, F. Leymann, M. Long, V. Papanikolaou, D. Presenza, A. Russo, S. N. Srirama, D. A. Tamburri, M. Wurster, and L. Zhu. 2020. RADON: rational decomposition and orchestration for serverless computing. *SICS Software-Intensive Cyber-Physical Systems* 35, 1 (01 Aug 2020), 77–87. https://doi.org/10.1007/s00450-019-00413-w
[12] Daniel Céspedes, Paula Angeleri, Karin Melendez, and Abraham Dávila. 2020. Software Product Quality in DevOps Contexts: A Systematic Literature Review. In *Trends and Applications in Software Engineering*, Jezreel Mejia, Mirna Muñoz, Álvaro Rocha, and Jose A. Calvo-Manzano (Eds.). Springer International Publishing, Cham, 51–64.
[13] Daniel Céspedes, Paula Angeleri, Karin Melendez, and Abraham Dávila. 2020. Software Product Quality in DevOps Contexts: A Systematic Literature Review. In *Trends and Applications in Software Engineering*, Jezreel Mejia, Mirna Muñoz, Álvaro Rocha, and Jose A. Calvo-Manzano (Eds.). Springer International Publishing, Cham, 51–64.
[14] Stéphanie Challita, Fabian Korte, Johannes Erbel, Faiez Zalila, Jens Grabowski, and Philippe Merle. 2020. Model-Based Cloud Resource Management with TOSCA and OCCI. https://doi.org/10.48550/ARXIV.2001.07900
[15] Alessandro Colantoni, Luca Berardinelli, Antonio Garmendia, and Johannes Bräuer. 2022. Towards Blended Modeling and Simulation of DevOps Processes: The Keptn Case Study. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Montreal, Quebec, Canada) *(MODELS '22)*. Association for Computing Machinery, New York, NY, USA, 784–792. https://doi.org/10.1145/3550356.3561597
[16] Alessandro Colantoni, Luca Berardinelli, and Manuel Wimmer. 2020. DevOpsML: Towards Modeling DevOps Processes and Platforms. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Virtual Event, Canada) *(MODELS '20)*. Association for Computing Machinery, New York, NY, USA, Article 69, 10 pages. https://doi.org/10.1145/3417990.3420203
[17] Benoit Combemale and Manuel Wimmer. 2020. Towards a Model-Based DevOps for Cyber-Physical Systems. In *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Jean-Michel Bruel, Manuel Mazzara, and Bertrand Meyer (Eds.). Springer International Publishing, Cham, 84–94.
[18] Christof Ebert, Gorka Gallardo, Josune Hernantes, and Nicolas Serrano. 2016. DevOps. *IEEE Software* 33, 3 (2016), 94–100. https://doi.org/10.1109/MS.2016.68
[19] Badr El Khalyly, Abdessamad Belangour, Mouad Banane, and Allae Erraissi. 2020. A new metamodel approach of CI/CD applied to Internet of Things Ecosystem. In *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. 1–6. https://doi.org/10.1109/ICECOCS50124.2020.9314485
[20] Nicolas Ferry, Franck Chauvel, Hui Song, Alessandro Rossini, Maksym Lushpenko, and Arnor Solberg. 2018. CloudMF: Model-Driven Management of Multi-Cloud Applications. *ACM Trans. Internet Technol.* 18, 2, Article 16 (jan 2018), 24 pages. https://doi.org/10.1145/3125621
[21] Nicolas Ferry and Phu H. Nguyen. 2019. Towards Model-Based Continuous Deployment of Secure IoT Systems. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 613–618. https://doi.org/10.1109/MODELS-C.2019.00093

[22] Georges Bou Ghantous and Asif Qumer Gill. 2017. DevOps: Concepts, Practices, Tools, Benefits and Challenges. In *PACIS*.

[23] Michele Guerriero, Michele Ciavotta, Giovanni Paolo Gibilisco, and Danilo Ardagna. 2015. A Model-Driven DevOps Framework for QoS-Aware Cloud Applications. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. 345–351. https://doi.org/10.1109/SYNASC.2015.60

[24] Michele Guerriero, Michele Ciavotta, Giovanni Paolo Gibilisco, and Danilo Ardagna. 2015. SPACE4Cloud: A DevOps Environment for Multi-Cloud Applications. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps* (Bergamo, Italy) *(QUDOS 2015)*. Association for Computing Machinery, New York, NY, USA, 29–30. https://doi.org/10.1145/2804371.2804378

[25] Mubin Ul Haque, Leonardo Horn Iwaya, and M. Ali Babar. 2020. Challenges in Docker Development: A Large-Scale Study Using Stack Overflow. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '20)*. Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. https://doi.org/10.1145/3382494.3410693

[26] Chuan He and Gunter Mussbacher. 2016. Model-Driven Engineering and Elicitation Techniques: A Systematic Literature Review. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*. 180–189. https://doi.org/10.1109/REW.2016.041

[27] Regina Hebig and Reda Bendraou. 2014. On the Need to Study the Impact of Model Driven Engineering on Software Processes. In *Proceedings of the 2014 International Conference on Software and System Process* (Nanjing, China) *(ICSSP 2014)*. Association for Computing Machinery, New York, NY, USA, 164–168. https://doi.org/10.1145/2600821.2600846

[28] Jerome Hugues, Anton Hristosov, John J. Hudak, and Joe Yankel. 2020. TwinOps - DevOps Meets Model-Based Engineering and Digital Twins for the Engineering of CPS. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Virtual Event, Canada) *(MODELS '20)*. Association for Computing Machinery, New York, NY, USA, Article 94, 5 pages. https://doi.org/10.1145/3417990.3421446

[29] G. Kim, J. Humble, P. Debois, and J. Willis. 2016. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press. https://books.google.pt/books?id=ui8hDgAAQBAJ

[30] Jörg Christian Kirchhof, Evgeny Kusmenko, Jonas Ritz, Bernhard Rumpe, Armin Moin, Atta Badii, Stephan Günnemann, and Moharram Challenger. 2022. MDE for Machine Learning-Enabled Software Systems: A Case Study and Comparison of MontiAnna and ML-Quadrat. In *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (Montreal, Quebec, Canada) *(MODELS '22)*. Association for Computing Machinery, New York, NY, USA, 380–387. https://doi.org/10.1145/3550356.3561576

[31] Jörg Christian Kirchhof, Bernhard Rumpe, David Schmalzing, and Andreas Wortmann. 2022. MontiThings: Model-Driven Development and Deployment of Reliable IoT Applications. *Journal of Systems and Software* 183 (2022), 111087. https://doi.org/10.1016/j.jss.2021.111087

[32] Rakesh Kumar and Rinkaj Goyal. 2020. Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). *Computers & Security* 97 (2020), 101967. https://doi.org/10.1016/j.cose.2020.101967

[33] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. 2019. A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv.* 52, 6, Article 127 (nov 2019), 35 pages. https://doi.org/10.1145/3359981

[34] Tiina Leppänen, Anne Honkaranta, and Andrei Costin. 2022. Trends for the DevOps Security. A Systematic Literature Review. In *Business Modeling and Software Design*, Boris Shishkov (Ed.). Springer International Publishing, Cham, 200–217.

[35] Grzegorz Loniewski, Emilio Insfran, and Silvia Abrahão. 2010. A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development. In *Model Driven Engineering Languages and Systems*, Dorina C. Petriu, Nicolas Rouquette, and Øystein Haugen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 213–227.

[36] Atif Mashkoor, Alexander Egyed, and Robert Wille. 2020. Model-driven Engineering of Safety and Security Systems: A Systematic Mapping Study. https://doi.org/10.48550/ARXIV.2004.08471

[37] Fran Melchor, Roberto Rodriguez-Echeverria, José M. Conejero, Álvaro E. Prieto, and Juan D. Gutiérrez. 2022. A Model-Driven Approach for Systematic Reproducibility and Replicability of Data Science Projects. In *Advanced Information Systems Engineering*, Xavier Franch, Geert Poels, Frederik Gailly, and Monique Snoeck (Eds.). Springer International Publishing, Cham, 147–163.

[38] Bart Meyers, Klaas Gadeyne, Bentley Oakes, Matthias Bernaerts, Hans Vangheluwe, and Joachim Denil. 2019. A Model-Driven Engineering Framework to Support the Functional Safety Process. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 619–623. https://doi.org/10.1109/MODELS-C.2019.00094

[39] Mustafa Abshir Mohamed, Moharram Challenger, and Geylani Kardas. 2020. Applications of model-driven engineering in cyber-physical systems: A systematic mapping study. *Journal of Computer Languages* 59 (2020), 100972. https://doi.org/10.1016/j.cola.2020.100972

[40] Mirna Muñoz, Mario Negrete, and Jezreel Mejía. 2019. Proposal to Avoid Issues in the DevOps Implementation: A Systematic Literature Review. In *New Knowledge in Information Systems and Technologies*, Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo (Eds.). Springer International Publishing, Cham, 666–677.

[41] Abdul Wahab Muzaffar, Shumyla Rasheed Mir, Muhammad Waseem Anwar, and Adeela Ashraf. 2017. Application of Model Driven Engineering in Cloud Computing: A Systematic Literature Review. In *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing* (Cambridge, United Kingdom) *(ICC '17)*. Association for Computing Machinery, New York, NY, USA, Article 137, 6 pages. https://doi.org/10.1145/3018896.3036380

[42] Christophe Ponsard and Valery Ramon. 2022. Survey of Automation Practices in Model-Driven Development and Operations. In *2022 IEEE/ACM 4th International Workshop on Bots in Software Engineering (BotSE)*. 14–17. https://doi.org/10.1145/3528228.3528405

[43] Claudia Raibulet, Francesca Arcelli Fontana, and Marco Zanoni. 2017. Model-Driven Reverse Engineering Approaches: A Systematic Literature Review. *IEEE Access* 5 (2017), 14516–14542. https://doi.org/10.1109/ACCESS.2017.2733518

[44] Luis F. Rivera, Norha M. Villegas, Gabriel Tamura, Miguel Jiménez, and Hausi A. Müller. 2018. UML-Driven Automated Software Deployment. In *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering* (Markham, Ontario, Canada) *(CASCON '18)*. IBM Corp., USA, 257–268.

[45] Mary Sánchez-Gordón and Ricardo Colomo-Palacios. 2018. Characterizing DevOps Culture: A Systematic Literature Review. In *Software Process Improvement and Capability Determination*, Ioannis Stamelos, Rory V. O'Connor, Terry Rout, and Alec Dorling (Eds.). Springer International Publishing, Cham, 3–15.

[46] Mary Sánchez-Gordón and Ricardo Colomo-Palacios. 2020. Security as Culture: A Systematic Literature Review of DevSecOps. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. Association for Computing Machinery, New York, NY, USA, 266–269. https://doi.org/10.1145/3387940.3392233

[47] Julio Sandobalin. 2018. A Model-Driven Approach to Continuous Delivery of Cloud Resources. In *Service-Oriented Computing – ICSOC 2017 Workshops*, Lars Braubach, Juan M. Murillo, Nima Kaviani, Manuel Lama, Loli Burgueño, Naouel Moha, and Marc Oriol (Eds.). Springer International Publishing, Cham, 346–351.

[48] J. Sandobalin, E. Insfran, and S. Abrahao. 2019. Towards Model-Driven Infrastructure Provisioning for Multiple Clouds. In *Advances in Information Systems Development*, Bo Andersson, Björn Johansson, Chris Barry, Michael Lang, Henry Linger, and Christoph Schneider (Eds.). Springer International Publishing, Cham, 207–225.

[49] Julio Sandobalin, Emilio Insfran, and Silvia Abrahão. 2019. ARGON: A Model-Driven Infrastructure Provisioning Tool. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. 738–742. https://doi.org/10.1109/MODELS-C.2019.00114

[50] D.C. Schmidt. 2006. Guest Editor's Introduction: Model-Driven Engineering. *Computer* 39, 2 (2006), 25–31. https://doi.org/10.1109/MC.2006.58

[51] Hui Song, Rustem Dautov, Nicolas Ferry, Arnor Solberg, and Franck Fleurey. 2020. Model-Based Fleet Deployment of Edge Computing Applications. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (Virtual Event, Canada) *(MODELS '20)*. Association for Computing Machinery, New York, NY, USA, 132–142. https://doi.org/10.1145/3365438.3410951

[52] Hui Song, Rustem Dautov, Nicolas Ferry, Arnor Solberg, and Franck Fleurey. 2022. Model-based fleet deployment in the IoT–edge–cloud continuum. *Software and Systems Modeling* 21, 5 (01 Oct 2022), 1931–1956. https://doi.org/10.1007/s10270-022-01006-z

[53] Konstantinos Tsilionis and Yves Wautelet. 2022. A model-driven framework to support strategic agility: Value-added perspective. *Information and Software Technology* 141 (2022), 106734. https://doi.org/10.1016/j.infsof.2021.106734

[54] Willem-Jan van den Heuvel and Damian A. Tamburri. 2020. Model-Driven ML-Ops for Intelligent Enterprise Applications: Vision, Approaches and Challenges. In *Business Modeling and Software Design*, Boris Shishkov (Ed.). Springer International Publishing, Cham, 169–181.

[55] Willem-Jan van den Heuvel, Damian A. Tamburri, Damiano D'Amici, Fabiano Izzo, and S. Potten. 2021. ChainOps for Smart Contract-Based Distributed Applications. In *Business Modeling and Software Design*, Boris Shishkov (Ed.). Springer International Publishing, Cham, 374–383.

[56] Denis Weerasiri, Moshe Chai Barukh, Boualem Benatallah, and Jian Cao. 2016. A Model-Driven Framework for Interoperable Cloud Resources Management. In *Service-Oriented Computing*, Quan Z. Sheng, Eleni Stroulia, Samir Tata, and Sami Bhiri (Eds.). Springer International Publishing, Cham, 186–201.

[57] Denis Weerasiri, Moshe Chai Barukh, Boualem Benatallah, and Cao Jian. 2016. CloudMap: A Visual Notation for Representing and Managing Cloud Resources. In *Advanced Information Systems Engineering*, Selmin Nurcan, Pnina Soffer, Marko Bajec, and Johann Eder (Eds.). Springer International Publishing, Cham, 427–443.

[58] Johannes Wettinger, Uwe Breitenbücher, Oliver Kopp, and Frank Leymann. 2016. Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel. *Future Generation Computer Systems* 56 (2016), 317–332. https://doi.org/10.1016/j.future.2015.07.017

[59] L. Zhu, L. Bass, and G. Champlin-Scharff. 2016. DevOps and Its Practices. *IEEE Software* 33, 03 (may 2016), 32–34. https://doi.org/10.1109/MS.2016.81