International Journal of Scientific Research and Engineering Development-– Volume 6 Issue 6, Nov- Dec 2023

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Semantic Knowledge Modeling in DevOps: An ontological Framework Aimed at Aiding the Comprehension of DevOps within Academia and the Software Business

Sarthak Srivastava[1]

Visa Inc

[1]sarthaksrivastava44@gmail.com

Abstract: *Presently, there exists a limited understanding of DevOps and its components, leading to informal and often incorrect implementations. The DevOps landscape is filled with various proposals, resulting in widespread confusion and terminological conflicts. To address this challenge, this article introduces the concept of DevOps Ontology, a semi-formal framework. This ontology aims to establish a uniform, coherent, and comprehensible language for disseminating knowledge related to DevOps implementation in software development.*

DevOps Ontology serves as a valuable tool by elucidating the relationships between software process elements and agile principles/values. It was developed using the REFSENO formalism, represented in UML, and employs the OWL language with tools like Prótegé and HermiT Reasoner to assess its structural integrity. The ontology underwent successful evaluation in various scenarios, including theoretical validation and the instantiation of continuous integration and deployment practices from GitLab. Furthermore, a mobile application was created to access information from the DevOps Ontology using SPARQL and RDF. The results of these evaluations confirm that DevOps Ontology is consistent, comprehensive, and concise. It demonstrates the ability to infer knowledge, ensuring completeness and well-established definitions and inferences. Moreover, the ontology is assessed for conciseness, free from redundant or unnecessary definitions, and possesses the potential for expansion to incorporate new concepts and relationships. Overall, this ontology offers a systematic and structured approach to organize knowledge in the DevOps domain, alleviating confusion and enhancing consistency in terminology and concepts within the field.

Keywords: Development and Operations (DevOps), Semantic Web Technology, Infrastructure as Code (IaC), Software Development Lifecycle (SDLC), Configuration Management.

I. Introduction

The software development industry has continuously evolved over time, primarily driven by the imperative to enhance the quality of software products and services through ongoing refinement of processes and methodologies [1]–[3]. As clients' requirements, industry standards, and models in software development have grown in complexity, various solutions have emerged. These solutions can broadly be categorized into two main approaches: agile and traditional methods. Traditional methodologies include well-known frameworks like CMMI, RUP, Microsoft Solution Framework, ISO 9001, and ISO 29110, while agile methodologies encompass approaches like Scrum, Lean Software, TDD, and XP. There are also hybrid approaches that blend elements from both traditional and agile practices, such as Scrum & XP, Scrumban, and Scrum & CMMI.

However, these frameworks primarily focus on the development aspect (Dev) of software products, leaving out critical operations-related aspects (Ops) such as infrastructure provisioning, continuous monitoring, collaboration, experimentation, deployment, and product maintenance [16]–[18]. In the realm of operations and infrastructure, frameworks like ITIL, COBIT, and standards like ISO/IEC 20000 have been developed to manage IT services and processes.

The industry now recognizes the need for collaboration between software development and infrastructure to enhance productivity and competitiveness [22]–[24]. However, the current frameworks tend to complement each other rather than providing a seamless integration between Dev and Ops. This division has created confusion among professionals in both areas.

In response to these challenges, the concept of DevOps emerged around 2009 [28]–[30]. DevOps aims to bridge the gap between development and operations by encompassing all elements essential for implementing high-quality software [31], [32]. It emphasizes human expertise, technology, and streamlined processes to foster a culture of innovation and collaboration throughout the software development lifecycle. DevOps practices focus on continuous deployment, monitoring, testing, and maintenance, with automation tools playing a crucial role [37]–[39].

However, adopting DevOps is not straightforward, and it requires a clear understanding of concepts, practices,

tools, definitions, benefits, and challenges [30]. To reduce ambiguity and promote knowledge sharing in the DevOps domain, this article proposes DevOps Ontology, a formal mechanism that provides clarity and structure to concepts and terminology, facilitating collaboration among stakeholders [41], [42]. DevOps Ontology has the potential to assess DevOps compliance in software organizations by identifying connections between software development processes, agile principles, and software measurement. It has been adapted to suit the unique attributes of software processes within different organizations and incorporates concepts from various ontological solutions [43].

In summary, the software development industry faces the challenge of integrating development and operations effectively, and DevOps, supported by DevOps Ontology, aims to address this challenge by providing a structured approach to knowledge sharing and collaboration.

II. Related Work

Following a comprehensive systematic literature review [37], an extensive examination was conducted concerning studies and initiatives related to DevOps terminology, definitions, concepts, and their interrelationships. This examination encompassed:

Exploratory Studies on DevOps Terminology: Over the years, there has been a dedicated effort by the scientific community to understand the essence of DevOps, including its definition, objectives, scope, and associated elements. Exploratory studies aimed to uncover "what" DevOps represents, both in literature and within business contexts. For example, [45] explored common fallacies and mistakes made by software development companies when adopting DevOps. Systematic literature reviews were conducted to identify key concepts, practices, tools, and essential elements that contribute to defining a DevOps knowledge base [41], [46]. Exploratory studies in [47] and [37] delved into trends and perspectives surrounding DevOps, while [48] outlined a process for DevOps adoption. Additionally, studies like [27] and [49] focused on identifying concepts and challenges in DevOps adoption.

Taxonomies Supporting DevOps: Several studies proposed taxonomies and their practical applications, often validated through empirical case studies or external evaluations within software development companies. These taxonomies aimed to address functional challenges in DevOps adoption. For instance, [50] and [51] introduced taxonomies describing the relationships required to tackle functional challenges in DevOps adoption. [52] presented a taxonomy for addressing challenges related to SecDevOps, while [53] investigated the consequences of applying DevOps during software delivery. An empirical taxonomy, based on input from 11 DevOps experts, was defined in [54].

Ontologies Supporting DevOps: The exploration also identified solutions that leverage ontologies to support specific DevOps elements. Examples include [55], which proposed a software configuration management process, [56] presented the OSDAS ontology unifying multiple agile approaches, including DevOps, and [57] introduced a generic ontology for assessing the maturity level of DevOps. However, these ontologies often addressed specific aspects of DevOps and lacked a comprehensive conceptual framework encompassing its values, principles, practices, roles, and processes.

In summary, numerous initiatives have aimed to standardize and clarify DevOps terminology and concepts. Despite significant progress, challenges remain due to the lack of consensus and uniformity in definitions and concepts within the DevOps domain, necessitating ongoing efforts to establish a unified DevOps framework. Initiatives such as the IEEE 2675-2021 standard [58] have attempted to provide a definition of DevOps based on various facets but lack a comprehensive reference model for standardizing DevOps terminologies and concepts

III. Methodology

Various methodologies are available to facilitate ontology creation, including knowledge management relying on ontologies [59], Methontology [60], and other approaches, such as a translation method for developing versatile ontology solutions [61], and ontologies based on first-order logic [62]. Additionally, the REFSENO methodology [63] is used to define ontologies within the field of software engineering.

The REFSENO methodology, chosen by the authors to create the DevOps Ontology, is specifically designed for software engineering ontology development. It offers constructs for defining concepts, attributes, and relationships while providing a user-friendly alternative to methods reliant on predicate logic, simplifying complex representations. REFSENO is an adaptation of Methontology, used to build ontologies from scratch, and it offers techniques to assess ontology consistency and instances at the implementation level. It enables knowledge representation at two levels: the knowledge level, describing "what to represent" independently of implementation, and the symbols level, specifying "how to represent." REFSENO introduces three knowledge levels: the epistemological level, describing primitives like concepts, attributes, and relationships; the conceptual level, defining a standard vocabulary; and the linguistic level, proposing mechanisms for defining concrete instances of constructs [64].

The process of defining the DevOps Ontology using the REFSENO approach involves three stages: construction, evolution, and validation. Additionally, descriptive logic (DL) is employed for its expressiveness compared to propositional logic [65]. The construction stage

encompasses defining the epistemological level, which outlines concepts, attributes, and relationships, the conceptual level, which establishes a standard vocabulary, and the linguistic level, which defines mechanisms for representing concrete instances. The evolution stage entails updating and refining the ontology as the DevOps domain evolves, while the validation stage ensures the ontology's consistency and correctness at the implementation level.

Regarding integration in the ontology domain, it can encompass various meanings, including identifying correspondences between ontological solutions, identifying common elements between ontologies, mapping elements between ontologies, merging existing ontologies, and incorporating one ontology within another along with the statements that define their relationship [67].

IV. Results

The systematic review of literature on the implementation of DevOps in software development companies [37] uncovered two main objectives for creating DevOps Ontology: (i) identifying DevOps-related terminology and (ii) standardizing that terminology. DevOps Ontology has been purposefully crafted to achieve these aims by offering clear definitions and establishing relationships among DevOps concepts. This makes it a valuable resource for the development of assessment models and processes in the field. As discussed in [68], ontologies in software engineering can be categorized based on their domain or software artifacts. DevOps Ontology falls into the domain category because it organizes knowledge related to fundamental DevOps concepts and their interconnections within the software development industry. In the following section, we provide a comprehensive overview of DevOps Ontology, including its intended purpose, a glossary of its concepts and relationships, its graphical representation, and its integration with other ontologies.

4.1. Purpose of the ontology

The concepts within DevOps Ontology have practical applications in two distinct settings. Firstly, in an academic context, it serves as a valuable resource for individuals interested in studying or researching DevOps adoption strategies within the software development industry. Secondly, in an industrial context, it offers software development companies a tool to implement and streamline their adoption processes, aiding them in defining the specific terms and relationships relevant to DevOps.

Furthermore, DevOps Ontology serves as a foundational framework for the development of instruments that enable the automated assessment and evaluation of DevOps adoption processes. These instruments can incorporate various technologies, including machine learning, to enhance the efficiency of evaluating DevOps implementation.

In summary, DevOps Ontology finds practical utility in academic research, industrial application, and the creation of assessment tools for evaluating DevOps adoption processes through automation, contributing to the understanding and implementation of DevOps strategies.

4.2. Integration of DevOps ontology with other ontologies

DevOps Ontology plays a pivotal role in facilitating the evaluation of the DevOps adoption process within software development companies. A crucial aspect of this is the incorporation of a precise and standardized terminology to support software measurement practices. This objective is achieved through the introduction of the Ontology of Process-reference Models (PrMO) [43], which serves as the cornerstone for defining fundamental elements across various approaches, encompassing both traditional and agile methodologies such as Scrum, CMMI, SWEBOK, ITIL, ISO 9001, ISO 27001, ISO 29110, and ISO 31000.

Moreover, the integration of DevOps Ontology extends to the Software Measurement Ontology (SMO) subontology introduced in [69]. SMO plays a vital role in elucidating and establishing the core components required for defining software measures and the associated terminology integral to software measurement activities.

For a visual representation of the terms and relationships embedded within DevOps Ontology, Figure 1 offers a comprehensive depiction utilizing the Unified Modeling Language (UML) representation.

4.3. DevOps ontology concepts

The information presented in the ontology regarding definitions and concepts is organized systematically. It consists of a structured presentation with four columns. The first column enumerates the specific concept employed within the ontology. In the second column, it is indicated to which overarching concept this particular term is affiliated. The third column offers a concise definition of the term within the ontology. Lastly, the fourth column provides a reference source, either from which the term was directly taken or adjusted.

4.4. Discussion of some concepts and their relationships

In this section, an in-depth exploration and clarification of key terms within the DevOps Ontology are provided, encompassing principles, practices, and dimensions. The term "value" serves as the foundational basis from which the core essence of DevOps derives. For instance, existing literature commonly emphasizes certain elements.

Collaboration [70] is highlighted as a value that should permeate various facets of software development, fostering the exchange of knowledge, skill expansion, and shifts in responsibilities among teams. Additionally, automation [70] stands out as a critical element, enabling the repetitive and rapid execution of diverse tasks. Measurement [70] plays a pivotal role throughout the process, facilitating enhanced efficiency through the incorporation of metrics. Lastly, monitoring [70] aids in resource allocation, detecting, reporting, and rectifying issues that may arise during or after system updates.

Practices translate these principles into concrete actions, with integration, construction, deployment, configuration, and continuous monitoring being prominent in numerous studies [37]. Dimensions encompass diverse business areas where practices and related elements find classification, including processes, people, culture, and technology [26].

4.5. Assessment of DevOps ontology

According to [73], assessing an ontology involves three fundamental aspects: conciseness, completeness, and consistency. Consistency is achieved when the ontology's knowledge is deducible without ambiguity. Completeness is met when the ontology contains no missing information, ensuring clear definitions or inferable content. Additionally, conciseness is achieved when the ontology is free from redundancy and unnecessary definitions.

The summarized ontology, subsequently presented, underwent evaluation using the OWL ontology language [56], along with Protégé [61] and the HermiT Reasoner [62], with a primary focus on assessing its structural consistency. The analysis confirmed that the DevOps Ontology exhibited no structural inconsistencies.

V. Conclusions

DevOps Ontology is a semi-formal ontology designed to facilitate the adoption and utilization of DevOps principles within the software industry. This ontology establishes a standardized terminology, effectively organizing existing knowledge from relevant literature, thereby enhancing clarity and mitigating inconsistencies in the DevOps domain. In the creation of DevOps Ontology, the REFSENO formalism was employed, incorporating UML representation. Additionally, the utilization of the OWL language was chosen to enhance accessibility for both academic and industry professionals.

Furthermore, the primary objective of DevOps Ontology is to provide an initial solution for disambiguating and clarifying the terminology associated with DevOps. To achieve this goal, an MSL (Model Specification Language) was implemented. The evaluation of DevOps Ontology encompassed three practical implementation scenarios:

(i) A theoretical validation that demonstrated how the ontology's concepts and relationships can effectively describe various aspects of a DevOps adoption process.

(ii) Representation of integration and continuous deployment practices proposed by GitLab, aligning with the terms defined within the ontology.

(iii) Development of a mobile application that visualizes the relationship between DevOps principles and practices through the use of SPARQL queries.

The outcomes derived from the development of DevOps Ontology are slated for future utilization in specific tasks, including conducting case studies in software companies, establishing a reference model for DevOps adoption and evaluation, and employing inference mechanisms to automate DevOps assessment, allowing for comparisons with manual assessments.

Future research directions include extending the ontology's evaluation through comparative analysis, instantiation with various models and workflows, and the development of a tool for seamless integration with existing processes through conceptual and terminological verification using natural language.

References

[1] H. Conradi and A. Fuggetta, “Improving software process improvement,” IEEE Softw, vol. 19, no. 4, pp. 92–99, Jul. 2002.

[2] C. E. Mokhlis, A. Elmortada, M. Sbihi, and K. Mokhlis, “The impact of ISO 9001 quality management on organizational learning and innovation: Proposal for a conceptual framework,” Periodicals of Engineering and Natural Sciences, vol. 7, no. 2, pp. 944–951, 2019.

[3] Y. Andalib and C. E. Mokhlis, “The contribution of the quality certification process to the improvement of human resources management practices,” Periodicals of Engineering and Natural Sciences, vol. 8, no. 3, pp. 1880–1887, 2020.

[4] CMMI Institute, “CMMI V2.0 model,” Second edition, 2020. https://cmmiinstitute.com/cmmi (accessed Mar. 21, 2023).

[5] Davor Gornik, “IBM Rational Unified Process: Best Practices for Software Development Teams. TP026B, Rev 11/01,” Somers, New York, 2020.

[6] G. Lory, D. Campbell, A. Robin, G. Simmons, and P. Rytkonen, “Microsoft Solutions Framework v3 Overview,” 2003.

[7] ISO, “ISO 9001:2015 Quality Management Systems,” Geneva, Switzerland, 2020.

[8] ISO, “ISO/IEC TR 29110-1:2016 Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs),” Geneva, Switzerland, 2021.

[9] K. Schwaber and J. Sutherland, “The scrum guide the definitive guide to scrum: The rules of the game,” USA, 2020.

[10] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit. USA: Addison Wesley Longman Publishing Co., Inc., 2003.

[11] K. Beck, Test Driven Development: By Example, 1st ed. Boston: Addison-Wesley Professional, 2002.

[12] K. Beck and C. Andres, Extreme Programming Explained: Embrace Change, 2nd ed. Boston: Addison Wesley Professional, 2004.

[13] H. Kniberg, Scrum and XP from the Trenches, 2nd ed. USA: InfoQ, 2015.

[14] C. Ladas, Scrumban-essays on kanban systems for lean software development, 1st ed. Seattle: Modus Cooperandi Press, 2009.PEN Vol. 11, No. 2, April 2023, pp.207-220 218

[15] J. Sutherland, C. R. Jakobsen, and K. Johnson, "Scrum and CMMI level 5: The magic potion for code warriors," in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008, pp. 466–466.

[16] E. Diel, S. Marczak, and D. S. Cruzes, "Communication Challenges and Strategies in Distributed DevOps," in Proceedings of the 11th International Conference on Global Software Engineering (ICGSE), 2016, pp. 24–28.

[17] M. Rajkumar, A. K. Pole, V. S. Adige, and P. Mahanta, "DevOps culture and its impact on cloud delivery and software development," in Proceedings of the International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), 2016, pp. 1–6.

[18] M. de Bayser, L. G. Azevedo, and R. Cerqueira, "ResearchOps: The case for DevOps in scientific applications," in Proceedings of the International Symposium on Integrated Network Management (IM), 2015, pp. 1398–1404.

[19] A. Hochstein, R. Zarnekow, and W. Brenner, "ITIL as common practice reference model for IT service management: formal assessment and implications for practice," in Proceedings of the International Conference on e-Technology, e-Commerce and e-Service, 2005, pp. 704-710.

[20] G. Ridley, J. Young, and P. Carroll, "COBIT and its utilization: a framework from the literature," in Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2014, pp. 1–8.

[21] ISO/IEC, "ISO/IEC 20000:2018 Information Technology," London, United Kingdom, 2021.

[22] M. Virmani, "Understanding Devops & Bridging The Gap From Continuous Integration To Continuous Delivery," in Proceedings of the Fifth International Conference on the Innovative Computing Technology (INTECH 2015). IEEE, 2015, pp. 78–82.

[23] S. S. Samarawickrama and I. Perera, "Continuous scrum: A framework to enhance scrum with DevOps," in Proceedings of the 7th International Conference on Advances in ICT for Emerging Regions (ICTer), Sep. 2017, pp. 19–25.

[24] Cristian Barz, C. K. Jalba, Z. Erdei, and S. M. L. Hahn, "Approaches for the planning and implementation of Industry 4.0," Periodicals of Engineering and Natural Sciences, vol. 7, no. 1, pp. 375–380, 2019.

[25] G. de França, B., Jeronimo, H., & Travassos, "Characterizing DevOps by Hearing Multiple Voices," in Proceedings of the 30th Brazilian Symposium on Software Engineering - SBES '16, 2016, pp. 53–62.

[26] I. Bucena and M. Kirikova, "Simplifying the DevOps Adoption Process," in Proceedings of the 16th International Conference on Perspectives in Business Informatics Research, 2017, pp. 1–15.

[27] T. Riungu, L; Mäkinen, S; Lwakatare, L; Tiihonen, J & Männistö, "DevOps Adoption Benefits and Challenges in Practice: A Case Study," in Proceedings of the Product-Focused Software Process Improvement, 2016, pp. 590–597.

[28] B. S. Farroha and D. L. Farroha, "A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment," in Proceedings of the Military Communications Conference, 2014, pp. 288–293.

[29] H. Chen, R. Kazman, S. Haziyev, V. Kropov, and D. Chtchourov, "Architectural Support for DevOps in a Neo-Metropolis BDaaS Platform," in Proceedings of the 34th Symposium on Reliable Distributed Systems Workshop (SRDSW), 2015, pp. 25–30.

[30] J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications," in Proceedings of the International Conference on Cloud Engineering, 2015, pp. 60–65.

[31] S. Nagpal and A. Shadab, "Literature Review: Promises and Challenges of DevOps," Waterloo, Canada, 2017.

[32] R. N. Jaffar, A. A. A. M. Hussain, and W. Chiad, "A new model for study of quality attributes to components based development approach," Periodicals of Engineering and Natural Sciences, vol. 7, no.3, pp. 1177–1185, 2019, doi: http://dx.doi.org/10.21533/pen.v7i3.686.

[33] J. Michelsen, Dysfunction Junction: A Pragmatic Guide to Getting Started with DevOps. Boston: CA Technologies, 2014.

[34] J. Moore, G. Kortuem, A. Smith, N. Chowdhury, J. Cavero, and D. Gooch, "DevOps for the Urban IoT," in Proceedings of the Second International Conference on IoT in Urban Space, 2016, pp. 78–81.

[35] M. Soni, "End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery," in Proceedings of the International Conference on Cloud Computing in Emerging Markets (CCEM), 2015, pp. 85–89.

[36] P. Lees, K; Gardner, J & Eaton, "VMware, DevOps and Agile Development," 2017.PEN Vol. 11, No. 2, April 2023, pp.207-220 219

[37] J. Guerrero, K. Zuñiga, C. Certuche, and C. Pardo, "A systematic mapping study about DevOps," Journal de Ciencia e Ingeniería, vol. 12, no. 1, pp. 48–62, 2020.

[38] F. Ahmadighohandizi and K. Systä, "ICDO: Integrated Cloud-based Development Tool for DevOps," in Proceedings of the 14th Symposium on Programming Languages and Software Tools (SPLST), 2015, pp. 76–90.

[39] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "An Exploratory Study of DevOps: Extending the Dimensions of DevOps with Practices," in Proceedings of the Eleventh International Conference on Software Engineering Advances (ICSEA), 2016, pp. 1–9.

[40] Floris Erich, C. Amrit, and M. Daneva, "Report: DevOps Literature Review," Enschede, Netherlands, 2014.

[41] A. Ghantous, G.B & Gill, "DevOps: Concepts, Practices, Tools, Benefits and Challenges," in Proceedings of the Pacific Asia Conference on Information Systems (PACIS), 2017, pp. 1–12.

[42] V. Babenko, L. Lomovskykh, A. Oriekhova, L. Korchynska, M. Krutko, and Y. Koniaieva, "Features of methods and models in risk management of IT projects," Periodicals of Engineering and Natural Sciences, vol. 7, no. 2, pp. 629–636, 2019.

[43] C. Pardo, O. García, M. Piattini, F. Pino, and M. T. Baldassarre, "A reference ontology for harmonizing process-reference models," Revista Facultad de Ingeniería Universidad de Antioquia, vol. 1, no. 73, pp. 29–42, 2014.

[44] Devopsdays, "Devopsdays - Organizing Guide," 2009. https://bit.ly/3i7DKUb (accessed Jul. 27, 2021).

[45] A. Caprarelli, E. Di Nitto, and D. A. Tamburri, "Fallacies and Pitfalls on the Road to DevOps: A Longitudinal Industrial Study," in Proceedings of the DEVOPS 2019: Software Engineering Aspects of Continuous Development and New

Paradigms of Software Production and Deployment, 2020, pp. 200– 210.
[46] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is DevOps? A Systematic Mapping Study on Definitions and Practices," in Proceedings of the Scientific Workshop Proceedings of XP2016, 2016, pp. 1–11.
[47] J. Guerrero, C. Certuche, K. Zúñiga, and C. Pardo, "Trends in devops: A systematic mapping of the literature," RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, vol. 2020, no. E32, 2020.
[48] K. Zúñiga and C. Certuche, "Proceso para soportar DevOps en la integración, entrega y despliegue continuo en Pymes de software," University of Cauca, 2021.
[49] F. Jones, S., Noppen, J & Lettice, "Management challenges for DevOps adoption within UK SMEs," in Proceedings of the 2nd International Workshop on Quality-Aware DevOps - QUDOS 2016, 2016, pp. 7–11.
[50] A. A. Khan and M. Shameem, "Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process," Journal of Software: Evolution and Process, vol. 32, no. 10, pp. 1– 26, Oct. 2020.
[51] M. A. Akbar et al., "Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis," IEEE Access, vol. 8, no. 1, pp. 202487–202507, 2020.
[52] S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, and A. Gumaei, "Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE," IEEE Access, vol. 8, no. 1, pp. 105426–105446, 2020.
[53] M. A. Rothenberger, J. Humble, J. B. Thatcher, D. Smith, and N. Forsgren, "A Taxonomy of Software Delivery Performance Profiles: Investigating the Effects of DevOps Practices," in Proceedings of the Americas Conference on Information Systems, Aug. 2020, pp. 1–6.
[54] R. W. Macarthy and J. M. Bass, "An Empirical Taxonomy of DevOps in Practice," in Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020, pp. 221–228.
[55] C. Orozco, C. Pardo, and S. Vásquez, "SCMOnto: Una ontología para soportar la gestión de la configuración de software," Iberian Journal of Information Systems and Technologies, vol. 38, no. 12, pp. 75–90, 2020.
[56] D. Parsons, "Agile software development methodology, an ontological analysis," in Proceedings of the 9th International Conference on Applications and Principles of Information Science, 2010, pp. 1–4.
[57] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere, "Composable DevOps: Automated Ontology Based DevOps Maturity Analysis," in Proceedings of the International Conference on Services Computing, Jun. 2015, pp. 600–607.
[58] I. S. Committee, "IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment: IEEE Standard 2675-2021," 2021.PEN Vol. 11, No. 2, April 2023, pp.207-220 220
[59] D. Fensel, "Ontology-based knowledge management," Computer (Long Beach Calif), vol. 35, no. 11, pp. 56–59, Nov. 2002.
[60] M. Fernandez-lopez, Gomez-Perez.A, and Juristo. N, "METHONTOLOGY: from Ontological Art towards Ontological Engineering," Proceedings of the AAAI97 Spring Symposium, Stanford, USA, pp. 33–40, 1997.
[61] T. R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, vol. 5, no. 2. pp. 199–220, 1993.
[62] T. Hikita and M. J. Matsumoto, "Business process modelling based on the ontology and first-order logic," in Proceedings of the 3rd International Conference on Enterprise Information Systems, 2001, pp. 717–723.
[63] C. Tautz and C. G. Wangenheim, "REFSENO: a representation formalism for software engineering ontologies," Fraunhofer, 1998.
[64] U. Reimer, "Schema-based and network-based rendering formats," in Proceedings of the Einführung in die Wissensrepräsentation, 1991, pp. 28–78.
[65] L. F. Sikos, Description Logics in Multimedia Reasoning. Cham: Springer International Publishing, 2017.
[66] H. S. Pinto, A. Gómez-Pérez, and J. P. Martins, "Some Issues on Ontology Integration," in 16th International Joint Conference on Artificial Intelligence (IJCAI'99), 1999, vol. 18, pp. 7–12.
[67] J. Euzenat and P. Shvaiko, Ontology Matching. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
[68] F. Ruiz and J. R. Hilera, "Using Ontologies in Software Engineering and Technology," in Ontologies for Software Engineering and Software Technology, Springer, Berlin, Heidelberg, 2006, pp. 49–102.
[69] F. García et al., "Towards a consistent terminology for software measurement," Inf Softw Technol, vol. 48, no. 8, pp. 631–644, Aug. 2006.
[70] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "Dimensions of DevOps," in Proceedings of the Agile Processes in Software Engineering and Extreme Programming - XP, 2015, pp. 212–217.
[71] S. Lohmann, S. Negru, and D. Bold, "The ProtégéVOWL Plugin: Ontology Visualization for Everyone," 2014, pp. 395–400.
[72] W3C, "OWL 2 Web Ontology Language Document Overview," Second edition, 2020. https://bit.ly/3l0GCnB (accessed Mar. 21, 2023).
[73] R. Almeida, I. Percheiro, C. Pardo, and M. M. da Silva, "An Ontology-Based Model for ITIL Process Assessment Using TIPA for ITIL," in Software Process Improvement and Capability Determination, 2018, pp. 104–118.
[74] GitLab, "Introduction to CI/CD with GitLab," 2020. https://bit.ly/2BiX7Dq (accessed Mar. 21, 2023).
[75] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere, "Composable DevOps: automated ontology based DevOps maturity analysis," in Proceedings of the International Conference on Services Computing, 2015, pp. 600–607.
[76] A. A. Khan and M. Shameem, "Multicriteria decision□making taxonomy for DevOps challenging factors using analytical hierarchy process," Evolution and Process, vol. 32, no. 10, p. e2263, 2020.
[77] M. Akbar et al., "Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis," IEEE Access, vol. 8, pp. 426–446, 2020.
[78] S. Rafi, W. Yu, M. Akbar, A. Alsanad, and A. Gumaei, "Prioritization Based Taxonomy of DevOps Security Challenges Using PROMETHEE," IEEE Access, vol. 8, pp. 426–446, 2020.
[79] N. Forsgren, M. A. Rothenberger, J. Humble, J. B. Thatcher, and D. Smith, "A Taxonomy of Software Delivery Performance Profiles: Investigating the Effects of DevOps Practices," 2020.
[80] R. W. Macarthy and J. M. Bass, "An empirical taxonomy of DevOps in practice," in Proceedings of the 46th Euromicro Conf. on Software Eng. and Advanced Applications, 2020, pp. 221–228.

