DevOps as a culture of interaction and deployment in an insurance company.

Angel Salvatierra Melgar (corresponsal author)[1*], Jacinto Joaquín Vértiz Osores[3], Ricardo Iván Vértiz Osores[3], Héctor Raúl Santa María Relaiza[4], Jessica Acevedo Flores[5], Vladimir Hilario Quispe Orihuela[6], Raúl Alberto Rengifo Lozano[7]

[1*]Universidad Nacional Mayor de San Marcos, asalvatierram@unmsm.edu.pe https://orcid.org/0000-0003-2817-630X
[2]Universidad Nacional Tecnológica de Lima Sur, Perú https://orcid.org/0000-0003-2774-1207
[3]Universidad César Vallejo, Perú https://orcid.org/0000-0003-1223-2784
[4]Universidad César Vallejo, Perú https://orcid.org/0000-0003-4598-7032
[5]Universidad Privada San Juan Bautista, Perú https://orcid.org/0000-0002-0050-7886
[6]Universidad Nacional Federico Villarreal, Perú https://orcid.org/0000-0003-0700-1834
[7]Universidad Nacional Mayor de San Marcos, Perú https://orcid.org/0000-0002-6545-6442

***Article History**: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021*

Abstract

This study provides the contributions of the implementation and development of DevOps in the integration and deployment of software in the organization due to problems previously detected regarding the speed and quality of reports. The arguments assumed from a DevOps culture, respond to: techniques and practices of continuous integration, continuous deployment, test automation, collaboration, automation, measurement and monitoring; these facts, was thanks to the implementation of the pre-experimental research design of applied type, in a sample of 48 weekly sprints, with a total of 96 user stories of the project taken from the organization. The study allowed to demonstrate the decrease of the Cycletime of days on average, increase the frequency of code release and increase the percentage of attention ratio, based on these results, it was possible to demonstrate the improvement of the speed during the deployment and quality of software processes, these results were thanks to the adoption of DevOps as a culture of interaction and deployment in an organization in the insurance sector.

Keywords: DevOps, integration, continuous deployment, culture.

1. Introduction

This study reports on the progress achieved on the cultural organization within companies that use software, in order to optimize their databases and provide an adequate service for public and private entities. The cultural organization in such scenarios uses the use of layers in the cloud [1], where the development of DevOps represents a transversal perspective that allows the reduction of the gap between operations and product development, deliveries and integration of software continuity, which is a practice of the Scrum-DevOps culture that allows the viability of deliverables constantly and also the control of management in the planning phases through the design of case diagrams [2].

The implementation of DevOps is due to the existence of the silos of the Software Development area (DEV) and the Infrastructure and Security area (OPS) during the sequence of implementation, integration and sequencing of software for companies with higher demand, as is the case of the insurance sector. Such silos are developed and show presence because they safeguard the security and stability of the company's IT assets at key points in order to sequence, absolve delays in deliverables, identify the consequences of poor quality, security and lack of stability of the software in the company [3].

The adoption of DevOps is key for high performance organizations [4]. Some large companies have implemented DevOps, such as Netflix, Amazon, Google, among others; finding a gap with high performance in code deployments and optimal delivery times. An implementation experience carried out in Peru, is the case of the BBVA bank that around 2016 implemented DevOps which allowed it to shorten the development cycle with new applications. Such action allowed improving the development of new DevOps applications [5].

The application of DevOps culture in incident management, allows the improvement in automated testing and code infrastructure, this has allowed the improvement in deployment times [6]. The application of DevOps supports impacts on the processes of serial and continuous delivery of software integration platforms [7]. Also the DevOps strategy tends to attenuate the ratio of errors in the applications and improves the development time to MVP (Cycletime) [8]. DevOps techniques manage to increase the quantity and quality of applications going to production and deployment automation that favorably impact the production deployment [9, 10].

Approximately 29% of the information technology developers that were applied to the Global Developer Report study, proposed to invest in DevOps technology, due to the efficiency and effectiveness in their processes, which allows accelerating the processes of software developers [11]. This proposal of technological culture of development operations explores and creates strategic solutions in order to unify and accelerate the sequences and development process in software operations. In the report of the specialized TechRepublic portal, it was found that approximately 71% of developers use DevOps technology to a great extent, since it represents the automation of software development, representing a high priority in the main successful companies.

2. Methods and analysis

In recent times, emerging and successful companies are constantly searching and exploring how to improve their processes by applying new ways to optimize the development of software technologies [12]. Organizations are frequently experimenting with new and successful software development practices that provide fast and efficient performance. DevOps-based practices enable software development under the culture of interaction, constantly adapting to changes and implementing new successful processes for developers, who will be able to meet the demands of the most demanding companies in the market [13]. These demands allow to improve and rethink the workflows in Information Technology (ICT).

DevOps is understood as an adaptable support and not as a predetermined workflow or tool. It can also be understood as a conglomerate or amalgam of these, which is characterized by breaking down the walls that obstruct and separate the development of operations, in order to direct them to common goals based on collaboration and continuous improvements [14]. DevOps can be understood as a culture and philosophy for a successful organization [15, 16]. Such a culture allows the reduction in delivery time in each development flow and, in addition, ensures the quality of the development, demonstrating the benefits in the data record in a detailed way.

DevOps-based implementations are adopted and implemented according to the realities and needs of each organization, since there is no single formula or process standard for replication, as each organization is differentiated in terms of its culture and philosophy by its own organizational rules and strategies [17]. For others, DevOps can be understood as a framework [18]. It can also be defined as a set of technologies and practices associated with an organized and collaborative work culture and philosophy [19].

DevOps seeks to break down or transform the existing organizational wall between the developer teams, eliminating obstacles and deficiencies in the processes, in order to improve the speed of change implementation. Possible causes of organizational walls between Dev and Ops are: (a) the Dev supports change and adapts agile practices, however Ops is oriented to the operational stability of the infrastructure; (b) varied metrics are applied to weight the performance and performance of achievements between Dev and Ops, and (c) there is an organizational culture of collaboration, consequently each area must ensure the optimal functioning in their communications of the different flows in accordance with the other areas towards a common goal.

On the other hand, the adoption of DevOps shortens the time between the developer and software changes, allowing deploying the production associated with the quality and stability for its development [20, 21]. For the achievement, it is necessary the decrease of common or recurring problems in the following aspects: (i) fear and natural expectations to changes resulting from the implementation of technologies, (ii) the supposedly tedious production-oriented implementation process that leads to uncertainties and, (iii) feelings of responsibility due to the non-fulfillment of common goals and objectives [22].

The implementation of DevOps, responds to key components such as: (a) adoption of a culture, (b) process automation, (c) measurement associated with values for each stage, (d) practice of a collaborative culture, (e) monitoring of flow sequences. It also requires an organized and committed work with predisposition to aspects such as: the assignment of shared responsibilities of those who are part of the process and common objectives for the various areas of development and operations, the generation of a culture of synergy among team members and the availability of a team called placebo that acts as a control against experimentation [23, 24].

The DevOps culture is based on collaborative work, with an eminently communicative, fluid and agile practice, which facilitates a circuit of automated processes, leading to the minimization of time within the flow. It also tends to the easy detection of metrics leading to the restructuring of the process, to optimize the work of software developers.

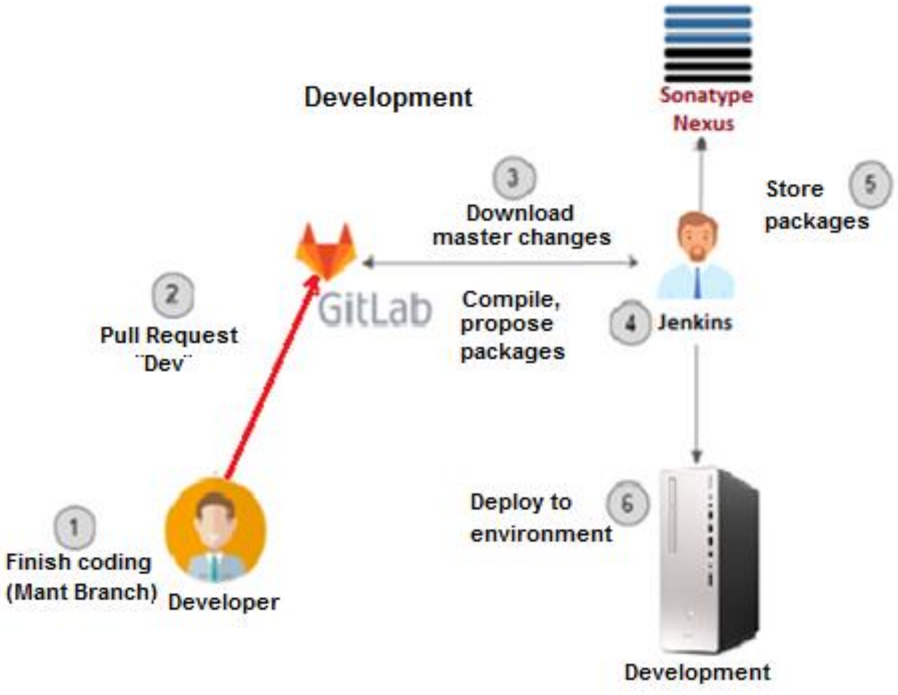

Figure1. Sequence of creating, building and deploying applications

Figure 1 represents the dynamism of software development and deployment. The flow allows identifying the respective code input, which are captured by request detection, then filtered for reporting and forwarded for pipeline testing in Jenkins, allowing real-time development and code release for software deployment productivity.

The adoption of DevOps, embraces a set of practices aimed at reducing the time to rapid deployment; as they represent changes to the production system ensuring high quality. Organizations that are highly dependent on organizational culture standards have been assuming different cultures thanks to DevOps-based implementation [25]. Successful practices have been identified within this model, since they represent an environment adapted to different scenarios that have been evidenced through optimal adoptions of DevOps, of which there are: (a) shared goals and objectives of the team, (b) collaborative culture among team members and between teams, (c) change of cultures and permanent models to respond to immediate standards, (d) a culture based on continuous learning, with experiences from successes and failures, (e) providing a space or scenario where members are involved and demonstrate their importance within the team or organization and, (f) a communicative culture based on empathy, loyalty and transparency.

The cultural change with DevOps, values and identifies the common goals of the team members, who stimulate collaborative work and detachment activities, demonstrating that joint actions allow the benefit for the company or organization, based on its philosophy of implementing the reduction of time for the confirmation of a system change, while ensuring high quality [26]. The sum of the experiences put into consideration in previous lines about the adoption of DevOps in the company, allows establishing the practice of a collaborative culture minimizing the conflicts that could derive in the rupture of the developers' work flow.

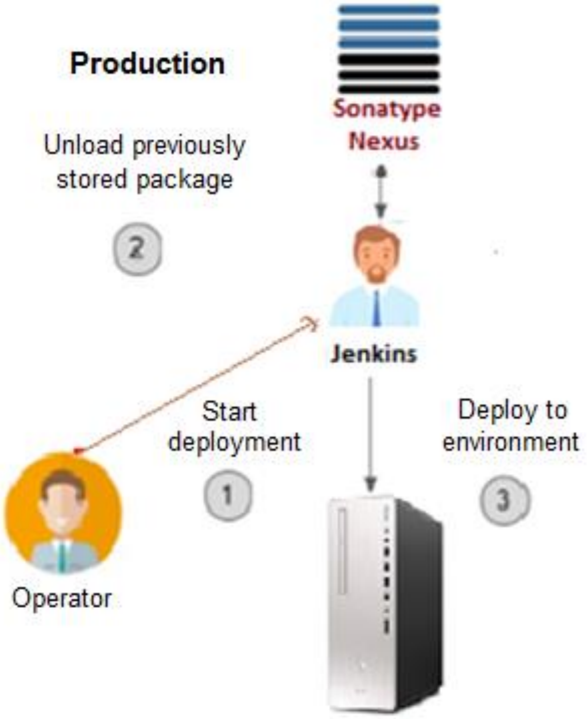

Figure 2. Productivity and deployment process

According to figure 2 representing the flowchart, the operational processes of productivity deployment are shown; who interacts in the test spaces of pipelines in Jenkins, interacts in various areas and absolve some anomalies detected, allowing to raise in the minimum times and eliminate failed components, for the development in a timely manner and the release of codes in the productivity of software deployment in the environments for specific development in the company in reference.

The sequential model of DevOps guides in taking advantage of the benefits and advantages of cloud-connected services, such as: (IaaS) the structure as service, (PaaS) the various platforms made available for deployment and (SaaS) the software made available as services. The deployment and intrinsic plasticity is due to the culture imposed on the model, which requires components and a different way of acting in the organization [27]. Once implemented, the arguments of micro services are put into practice, which allow the adoption of a true definition of architecture with DevOps technified support [28], sustained on three bases, pondering three supports:

(a) Stability, in order to measure and establish the KPI's of scalability, response times, performance and availability.

b) Speed in terms of: timely delivery, development, and the time it takes to correct production errors in the metrics, such as time of reports and delivery of stories to the interested party, frequency of code detachments and average resolution time.

c) Quality in the delivery of reports periodically with Continuous Integration (CI), deployment report and metrics related to the processes, detecting: application error, deployment success ratios, pending errors taken into consideration and the severity of the incident within the developers.

The present study was approached from the perspective of the pre-experimental design under the guidelines of the hypothetico-deductive method [29]. The data for the analysis were obtained from the indicators of the study. For the Cycletime indicator processes, 48 sprints distributed in 96 user stories were taken; however, for the indicators of code release frequency and success rate, 47 sprints were taken. These data were recorded with an observation sheet on observation cards designed for this purpose.

3. Results and discussion

The results of the present study are evidenced in terms of the Cycletime indicator developer speed and code release frequency, and quality ratios, before and after DevOps implementation.

Table 1. Statistics and parameters of DevOps components: speed: Cycletime, code release frequency and quality: success rate.

Cross table								
Dimension	Indicator		Levels			Statistics / parameters		
		Test	Low efficiency (fi%)	Promedium (fi%)	High efficiency (fi%)	$(\bar{x}, s_x)$	W / t	sig. Bilateral
Speed	Cycletime	Pre test (48)	44%	56%	0%	(18.11; 8.91)	258,000	0,017
		Pos tes (48)	28%	22%	50%	(11.61; 5.19)		
	Frequency of code deployment	Pre test (96)	50%	50%	0%	(1.63, 0.74)	51,000	0,046
		Pos test (96)	25%	25%	50%	(3.25; 1.75)		
Quality	Success rate	Pre test (47)	25%	37%	38%	(54.58; 34.87)	-1.413	0,020
		Pos test (47)	0%	25%	75%	(72.92; 19.47)		
Significance level 0.05; t $_{(7; 0.05)}$								

Source: Own elaboration.

The results of the experiment are shown in Table 1, which shows the behavior of the existing data compared to the observations after the implementation of the DevOps culture in the insurance sector. With respect to the Cycletime indicator, before the experiment, 44% had a low level of efficiency and 56% were at the average level. After the experimentation, 50% of the observations were at the high level of efficiency, and only the remaining 28% were at the low level of quality in terms of speed and the required levels of production and web applications for the quality of the process. The mean difference between the post and pre-test was 6.5 points. This difference represented the efficiency in the time it takes for the delivery process of the finished product to meet the customer's demand. The comparison of the statistics allowed affirming that the experimental implementation was significant in terms of the decrease of the Cycletime time (in days) for the process and flow of software integration and deployment in the indicated company. On the other hand, the value of Wilcoxon's W statistic presented a value of 258.0 (p value <, 05) which confirmed that the adoption of DevOps allowed to decrease the time span (in days) compared to t

With respect to the frequency of code deployment, favorable results were obtained in terms of the comparison of the frequencies before and after the implementation of the experiment; 50% of the low level in the average efficiency at the beginning was observed. However, after the implementation, 50% of the high level of efficiency was generated for the deployment, which generated an approach to the expected target. The reported statistics show the difference in metrics between the pre and post test of 1.62 points, being this data representative with Wilcoxon's W parameter value equal to 51 (p value < ,05), which led to affirm that the code deployment frequency was representative for the DevOps metric, which means that if its implementation is possible, it will allow minimizing the reporting time of any failed process and the ease of its location to be treated in a timely manner.

Finally, the success ratio that represented the quality of DevOps showed that before the experience only 38% of the records presented a high level of efficiency; however, after the implementation of the experiment, a value of 75% was generated for the high level in the samples analyzed. These results are the result of the actions developed in the experiment. The difference in means of the recorded observations represents 18.34 points in comparison. The data of this indicator, in comparison with the two previous ones, presented a normal distribution, so the Student's t-test statistic for related samples was applied, generating a value of -1.413 (p value < ,05). This comparison of the number of successful deployments compared to the number of failures was significant for the company.

From the experiences described, we can affirm that the practice and implementation of DevOps allowed the improvement of the software integration and deployment process within the company. An improvement was generated with respect to speed of up to 6.5 days less than Cycletime. As for the code release per sprint, it increased considerably; therefore, significant improvements in quality were shown by increasing the success rate of the software, deployed in considerable percentages of success. However, another study detected similar results regarding the existence of improvements in incident resolution time and improvements in deployment times thanks to the practice and implementation of the DevOps culture [30].

Regarding the results of Cycletime, thanks to the implementation of the culture in the company, it allowed the reduction of the time from the beginning of the development history of the user. Such a result was achieved thanks to the communicative practices among members and loyalty under the collaborative culture. It is worth mentioning that an equivalent study found that the implementation of DevOps allowed a decrease in the code release time and the deployment of pipeline services, based on a joint and experimental work to enhance their production [31]. On the other hand, another study showed that the adoption of the DevOps culture helped to increase the number of reports and improve the quality of the different stages of software development [26].

The acquisition of a culture different from the conventional with respect to productivity, undoubtedly allows a philosophy of adoption with common goals. Such culture was thanks to the adoption of DevOps allowing a high competitiveness in its workers and a marked productivity. Such arguments are shared with a study that showed that the adoption of DevOps, allowed the strategic generation in the productivity of software developers, detached in MSMEs. In addition, it was evidenced that the application of various techniques in different environments, allowed the release of failed code and continuous improvement of deployment [32].

On the other hand, it is worth mentioning that the considerable increase in software deployments was identified, thanks to the DevOps practice, which allowed the frequent release of codes independently of certain companies. Its 'High Performance' technological implementation allowed high performances for the release of failed code and the detection of Cycletime and Leadtime failure, as well as the rapid detection of recovery time in developers with problems [33]. This practice allowed the automation of processes, as evidenced in an experiment where the results were visualized, emphasizing the support of the Scrum framework for the optimization of visible products in the performance flow thanks to the culture that supports DevOps, which is oriented towards risk minimization, representing the fulfillment of the purposes of the Squad within the organization [2].

The implementation of the experiment allowed the systematization of processes, generating models. Such action is reflected in the projects and practices with ITIL philosophy, which resulted in the reduction of time and customer service ratio [32]. However, since the adoption of the culture in the company in reference, there were favorable results, since the reports showed effectiveness in terms of speed and quality components, associated with the fall of Cycletime from the beginning of the development of the actions to its practice exposed in productivity. In addition, the release of failed code and the reduction of times for the generation of user response, thanks to the implementation of DevOps, were appreciated.

The acquisition of speed and quality of reports thanks to the implementation, allowed the improvement of software deployments is a consequence of the developed application. The continuous improvement and adoption of the workflow, through the practice of Scrum, evidenced in the development and work through kanban boards, ensure continuous improvements and adoption of new practices and timely correction through feedback for the decrease of batch size (work package size) [1],

The automation practices within the company, allowed to respond effectively to the high demand, since time was optimized in looping tasks acquiring improvements in continuous attention, detecting crashes through testing as is the case of the SonarQube used for static code testing or Selenium as well as Yield for vulnerability testing automation, Gitlab in branch management, and pipeline testing in Jenkins, which allowed development improvements in timely times of code release in productivity for software deployment. Similar studies made known that the adoption of DevOps enabled early code deployment in quite reduced times, from commits to reporting, acquiring a significant impact on high-performance companies for demanding audiences [3].

4. Abstract

The implementation of DevOps as a culture of integration and deployment of software developers is based on collaborative work, focused on communication and automation of processes in the recording of data in real time in collaboration with communicative work teams in permanent monitoring of development and operations metrics. This culture allowed to increase the speed and quality of the integration and deployment process, detecting a decrease in Cycletime and an increase in the frequency of code release from 1.62 to 3.25 releases per sprint and, with respect to the percentage ratio, there was an increase in the attention of the software integration and deployment process in a company of the insurance sector. It is recommended to adopt and implement DevOps in order to improve the speed and quality of the software integration and deployment process, associated to a communicative and collaborative work culture, which will allow generating competitiveness, especially in the delivery time and quality of its applications with respect to users.

References

[1] Bass L. El arquitecto de software y DevOps. IEEE Software 2018; 35 (1): 8-10. doi: 10.1109 / MS.2017.4541051

[2] Babar Z, Lapouchnian A, Yu E. Modelado de opciones de implementación de DevOps utilizando dimensiones de diseño de arquitectura de procesos. En Lecture Notes in Business Information Processing Springer 2015; 235. doi: 10.1007/978-3-319-25897-3_21

[3] Rodríguez M, De Araújo LJP, Mazzara M. Good practices for the adoption of DataOps in the software industry. Journal of Physics: Conference Series 2020; 1694 (1). doi: 10.1088/1742-6596/1694/1/012032

[4] He X, Tu Z, Xu X, Wang Z. Programming framework and infrastructure for self-adaptation and optimized evolution method for microservice systems in cloud–edge environments. Future Generation Computer Systems 2021; 118: 263-281. doi: 10.1016/j.future.2021.01.008

[5] Sen A. DevOps, DevSecOps, AIOPS- Paradigms to IT Operations. Lecture Notes in Electrical Engineering 2021; 694: 211-221. doi: 10.1007/978-981-15-7804-5_16

[6] Mishra A, Otaiwi Z. DevOps and software quality: A systematic mapping. Computer Science Review 2020; 38: 100308. doi: 10.1016/j.cosrev.2020.100308

[7] Ordóñez HA, Buchelli V. Lineamientos para la implementación del modelo CALMS de DevOps en mipymes desarrolladoras de software en el contexto sur colombiano. Revista Guillermo de Ockham 2020; 18 (1): 81-91. doi: 10.21500/22563202.4270

[8] Choudhry A, Premchand A. Microservices and DevOps for Optimal Benefits from IoT in Manufacturing. Advances in Intelligent Systems and Computing 2021; 1245: 375-384. doi: 10.1007/978-981-15-7234-0_33

[9] El Khalyly B, Belangour A, Erraissi A, Banane M. Meta-model approach of applied devops on internet of things ecosystem. 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS; 2020. doi: 10.1109/ICECOCS50124.2020.9314552

[10] De Lange P, Nicolaescu P, Klamma R, Koren I. DevOpsUse for rapid training of agile practices within undergraduate and startup communities. Lecture Notes in Computer Science 2016; 9891 LNCS: 570-574. doi: 10.1007/978-3-319-45153-4_65

[11] Carrizo D, Alfaro A. Quality assurance method in a software development methodology: A practice approach. Ingeniare 2018; 26 (1): 114-129. doi: 10.4067/S0718-33052018000100114

[12] Bisht S, Shukla P. Edge Computing Approach to DEVOPS. Lecture Notes in Electrical Engineering 2021; 698: 23-30. doi: 10.1007/978-981-15-7961-5_3

[13] John W, Meirosu C, Pechenot B, Sköldström P, Kreuger P, Steinert R. Scalable Software Defined Monitoring for Service Provider DevOps. Proceedings - European Workshop on Software Defined Networks, EWSDN 2015; 61-66. doi: 10.1109/EWSDN.2015.62

[14] Castellanos C, Varela CA, Correal D. ACCORDANT: A domain specific-model and DevOps approach for big data analytics architectures. Journal of Systems and Software 2021; 172: 110869. doi: 10.1016/j.jss.2020.110869

[15] Fayollas C, Bonnin H, Flebus O. SafeOps: A Concept of Continuous Safety. Proceedings - 16th European Dependable Computing Conference, EDCC 2020: 65-68. doi: 10.1109/EDCC51268.2020.00020

[16] Lie MF, Sánchez-Gordon M, Colomo-Palacios R. DevOps in an ISO 13485 regulated environment: A multivocal literature review. International Symposium on Empirical Software Engineering and Measurement 2020; 3410679. doi: 10.1145/3382494.3410679

[17] Stillwell M, Coutinho JGF. A DevOps approach to integration of software components in an EU research project. 1st International Workshop on Quality-Aware DevOps, QUDOS, 2015. Proceedings 2015; 1: 1-6. doi: 10.1145/2804371.2804372

[18] Maroukian K, Gulliver SR. The Link between Transformational and Servant Leadership in DevOps-Oriented Organizations. ACM International Conference Proceeding Series; 2020: 21-29. doi: 10.1145/3393822.3432340

[19] Olszewska M, Waldén M. DevOps meets formal modelling in high-criticality complex systems. 1st International Workshop on Quality-Aware DevOps. Proceedings 2015: 7-12. doi: 10.1145/2804371.2804373

[20] Wang Y, Lee C. A Role-Based Orchestration Approach for Cloud Applications. In 2021 International Conference on Information Networking (ICOIN) 2021. doi: 10.1109/icoin50884.2021.9333919

[21] Ashenden D, Ollis G. Putting the Sec in DevSecOps: Using Social Practice Theory to Improve Secure Software Development. ACM International Conference Proceeding Series 2020: 34-44. doi: 10.1145/3442167.3442178

[22] Karapiperis C, Chasapi A, Angelis L, Scouras ZG, Mastroberardino PG, Tapio S, Atkinson MJ, Ouzounis CA. The Coming of Age for Big Data in Systems Radiobiology, an Engineering Perspective. Big Data 2021; 9 (1): 63-71. doi: 10.1089/big.2019.0144

[23] Cois CA, Yankel J, Connell A. Modern DevOps: Optimizing software development through effective system interactions. IEEE International Professional Communication Conference; 2015. doi: 10.1109/IPCC.2014.7020388

[24] Díaz O, Muñoz M. Implementación de un enfoque DevSecOps + Risk Management en un Centro de Datos de una organización mexicana. RISTI, Revista Iberica de Sistemas e Tecnologias de Informaçao 2018; 26: 43-53. doi: 10.17013/risti.26.43

[25] West SN, Cribb M, McCarthy TR. Applied design structure matrices to quantify impact of proposed model based engineering solutions. AIAA Scitech 2021 Forum 2021. doi: 10.2514/6.2021-0238

[26] Mao R, Zhang H, Dai Q, Huang H, Rong G, Shen H, Chen L, Lu K. Preliminary Findings about DevSecOps from Grey Literature. Proceedings-2020. IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS 2020: 450-457. doi: 10.1109/QRS51102.2020.00064

[27] Rodríguez-Figueroa JJ. Diseño Prospectivo de Escenarios para la Ciencia, Tecnología e Innovación al 2040. Industrial Data 2016; 16 (2): 92. doi: 10.15381/idata.v16i2.11927

[28] Amorós-Pons, A, García-González A. Cambios tecnológicos en el contexto publicitario: Dinámicas y tendencias en investigación Presentación. Icono14, Revista científica de Comunicación y Tecnologías emergentes 2015; 13 (1): 1-8. doi: 10.7195/ri14.v13i1.772

[29] Batanero C. Los retos de la cultura estadística. Yupana 2014; 1 (4): 27-36.

[30] Stillwell M, Coutinho JGF. A DevOps approach to integration of software components in an EU research project. 1st International Workshop on Quality-Aware DevOps. Proceedings 2015: 1-6. doi: 10.1145/2804371.2804372

[31] Ganeshan M, Vigneshwaran P. A Survey on DevOps Techniques Used in Cloud-Based IOT Mashups. Advances in Intelligent Systems and Computing 2021; 1270: 383-393. doi: 10.1007/978-981-15-8289-9_37

[32] Baron C, Louis V. Towards a continuous certification of safety-critical avionics software. Computers in Industry 2021; 125. doi: 10.1016/j.compind.2020.103382

[33] Gimenez PJA, Santos G. DevOps maturity diagnosis - A case study in two public organizations. ACM International Conference Proceeding Series; 2020. doi: 10.1145/3411564.3411608