¿Qué es DevOps? Definición y características

MIROSLAWA GOMEZ AGUIRRE
Av. Lázaro Cárdenas s/n, C.U. Zona Sur, Chilpancingo de los Bravo Guerrero.
TEL. 7341160507 C.P. 39087 MirosGA26@gmail.com

HERNANDEZ ALARCON ROGELIO FERNANDO
Av. Lázaro Cárdenas s/n, C.U. Zona Sur, Chilpancingo de los Bravo Guerrero.
TEL. 7471212159 C.P. 39087 rogeliofh@uagro.mx

SOLIS CARMONA EDGARDO
Av. Lázaro Cárdenas s/n, C.U. Zona Sur, Chilpancingo de los Bravo Guerrero.
TEL. 7471622971 C.P. 39087 esoliscr@hotmail.com

ALVAREZ HILARIO VALENTIN
Av. Lázaro Cárdenas s/n, C.U. Zona Sur, Chilpancingo de los Bravo Guerrero.
TEL. 7471622688 C.P. 39087 13701@uagro.mx

ABSTRACT

DevOps has become an approach of great relevance for software development, it arises with the objective of constituting a combination of collaborative cultures, practices and tools to increase productivity capacity in software development and services, developing and facilitating automation, integration and continuous use. This article presents the results of a research with the objective of knowing a broad definition of DevOps and its characteristics in software development.

RESUMEN

DevOps se ha convertido en un enfoque de gran relevancia para el desarrollo de software, éste surge con el objetivo de constituir una combinación de culturas colaborativas, prácticas y herramientas para incrementar la capacidad de productividad en el desarrollo y servicios de software, desarrollando y facilitando la automatización, integración y uso continuo. Este artículo presenta los resultados de una investigación con el objetivo de conocer una definición amplia de DevOps y sus características en el desarrollo de software.

KEYWORDS

DevOps, Software development, Software.

PALABRAS CLAVES
DevOps, Desarrollo de software, Software.

INTRODUCCIÓN

DevOps no es una tecnología, sino una metodología que aborda el reto de lo que a menudo se describe como una brecha que hay entre los desarrolladores y los operadores, para ello es fundamental realizar la integración y colaboración directa entre desarrolladores y administradores. Los equipos de desarrollo, sistemas e incluso QA(atributos de calidad), comienzan a colaborar y trabajar conjuntamente para cubrir todo el ciclo de desarrollo del software de manera que los procesos son mucho más ágiles y seguros, y de esta manera también garantizan un producto final de calidad. Con esta nueva metodología colaborativa se reducen considerablemente los tiempos de reacción en el desarrollo de nuevos productos, así como de las nuevas actualizaciones. Este artículo se realizó con la finalidad de dar a conocer a los estudiantes de la facultad de ingeniería, de la universidad Autónoma de Guerrero., La metodología DevOps, ¿qué es? ¿Y en qué consiste?

1.- ¿Qué es DevOps?

DevOps (Desarrollo y Operaciones), es un enfoque basado en principios ágiles en que los stakeholders y los departamentos de desarrollo, operaciones y control de calidad colaboran para entregar de manera continua, lo que permite a la empresa aprovechar rápidamente las oportunidades del mercado y reducir la carga de trabajo.

Para entender mejor el significado DevOps primero es necesario comprender los roles de Dev(desarrolladores) y de Ops (operaciones). Operadores (Ops) está compuesto en parte por los sysadmins (administradores de sistemas), los cuales tienen

la misión del mantenimiento de los sistemas y su normal funcionamiento, hacen los despliegues y las rollbacks de las versiones de aplicaciones. Mientras que desarrollo Dev, es su responsabilidad mantener integro el entorno de producción. Los sysadmins tienen que ejecutar las aplicaciones, supervisar la operación, funcionamiento, evaluar y proponer mejoras para mantener las aplicaciones con rapidez y disponibilidad.

1.1.- Principios DevOps

Los principios DevOps o la orientación hacia la innovación están en un estado de constante evolución. Existen múltiples versiones de los principios. Pero no existe una definición común de DevOps, Humble y Molesky (2011) definieron cuatro principios (véase la Figura 1)

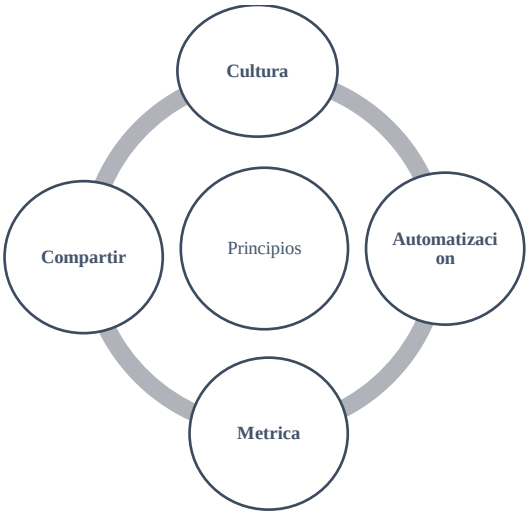

Figura 1. Principios DevOps

Cultura: Las operaciones de desarrollo requieren un cambio cultural para aceptar la responsabilidad conjunta de la entrega de software de alta calidad al usuario final.

Automatización: Las operaciones de desarrollo se basan en la automatización total de la creación, el despliegue y las pruebas para lograr tiempos de entrega cortos y, por consiguiente, una respuesta rápida de los usuarios.

Métrica: Sólo se puede comprender la capacidad de entrega actual y establecer objetivos de mejora mediante la medición, que varía desde la supervisión de las métricas empresariales habituales (por ejemplo, los ingresos) hasta la cobertura y el tiempo de implantación de la nueva versión del software.

Compartir: Se aplica a diferentes niveles, desde compartir conocimientos (por ejemplo, sobre una nueva funcionalidad en una versión), compartir herramientas e infraestructura, así como compartir la celebración de las versiones exitosas para reunir a los equipos de desarrollo y operación.

1.2.- Etapas del ciclo de vida Devops

Una cadena de herramientas es un conjunto de herramientas de programación que se pueden utilizar para llevar a cabo una tarea compleja de desarrollo, que suele ser algún otro programa informático o un conjunto de programas relacionados.

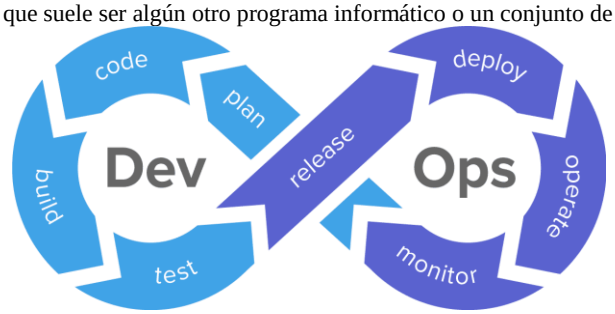

Figura 2. Ciclo de vida DevOps

Como se muestra en la figura 2 la etapa de Dev (departamento de desarrolladores) consta con las siguientes actividades:

(Plan) Planificar: Es un procedimiento cíclico y realiza mediante la planificación de pequeños lotes, la ejecución, la obtención de retroalimentación, la respuesta a la retroalimentación y ajustando el plan si es necesario, en esta etapa se tratan los requisitos como elemento principal para realizar las tareas en DevOps.

(Code) Codificar: Se compone de la construcción, codificación y configuración del proceso de desarrollo de software.

Build (Verificar): La verificación está directamente asociada a la garantía de la calidad de la versión del software; las actividades diseñadas para garantizar el mantenimiento de la calidad del código y el despliegue de la máxima calidad en la producción.

Test (Probar): Son las actividades que se llevan a cabo una vez que la versión está lista para el despliegue. También se denomina puesta en escena o preproducción.

Como se observa en la figura 2 la etapa Ops (departamento de operaciones) cuenta con las siguientes actividades:

Release (Desplegar): Las actividades relacionadas con el lanzamiento incluyen la programación, la orquestación, el aprovisionamiento y el despliegue del software en el entorno de producción y en el entorno objetivo.

Deploy (Ejecutar): Las actividades de configuración pertenecen a la parte operativa de DevOps. Una vez que se despliega el software, puede haber actividades adicionales de aprovisionamiento y configuración de la infraestructura de TI.

Monitor (Monitorizar): La supervisión es un eslabón importante en la cadena de herramientas de DevOps. Permite a la organización de TI identificar problemas específicos de versiones concretas y comprender el impacto en los usuarios finales.

2.- Características de DevOps

Como se puede observar en la tabla 1. En este capítulo se exponen las ventajas y desventajas de DevOps. Los beneficios de DevOps son los que pertenecen a una implementación exitosa de DevOps en empresas de software. Cabe destacar que la mayoría de los beneficios identificados son los beneficios percibidos de la adopción de DevOps en las organizaciones. Las desventajas de DevOps se presentan por obstáculos/limitaciones/impedimentos para su adopción.

Tabla 1. Ventajas y Desventajas de DevOps.

Ventajas	**Desventajas**
La comunicación, colaboración y confianza entre los miembros del equipo de desarrollo y operaciones aumenta.	La falta de una estructura de gestión adecuada en una organización puede dificultar el proceso de adopción.
Mejora de la calidad de la implementación del software.	La distribución geográfica de los equipos de desarrollo y operaciones puede dificultar la adopción de DevOps.
La implementación de los requisitos del cliente a lo largo del proceso de desarrollo de software	Las diferencias entre los entornos de desarrollo, prueba y producción pueden complicar la colaboración y también la entrega y el despliegue continuos.
DevOps ayuda a brindar servicios de mayor calidad de una manera más eficiente.	Múltiples entornos de producción pueden causar complejidad y dificultar el uso de herramientas y procesos comunes, lo que puede afectar la entrega continua.

2.1.- Criterios de adopción
DevOps ofrece una oportunidad sin precedentes para que las organizaciones transformen su ciclo de vida de desarrollo de software para aumentar la eficiencia y satisfacer las expectativas cambiantes de los usuarios finales.

La cultura colaborativa (compartir conocimiento)
Es la categoría central para la adopción de DevOps. Una cultura colaborativa apunta esencialmente a eliminar los sistemas que separan a los empleados según el departamento en el que trabaja (silos), entre los equipos y actividades de desarrollo y operaciones. Como resultado, las tareas de operaciones, como implementación, aprovisionamiento de infraestructura gestión y seguimiento deben considerarse como regulares, día a día, actividades de desarrollo. Esto lleva al primer concepto relacionado con esta categoría central: los equipos de desarrollo deben realizar las tareas de operaciones de manera transparente. Sin DevOps, un escenario común es un desarrollo de software acelerado desarrollo de software sin preocuparse por las operaciones. Al final, cuando el equipo de desarrollo tiene un producto de software mínimo viable, se envía al equipo de operaciones para su publicación. Sabiendo pocas cosas sobre la naturaleza del software y cómo se ha producido, el equipo de operaciones tiene que crear y configurar un entorno y publicar el software. En este escenario, la entrega del software suele retrasarse y aparecen conflictos entre los equipos. Cuando se fomenta una cultura de colaboración, los equipos colaboran para realizar las tareas desde el primer día de desarrollo del software. Con el ejercicio constante de las prácticas de aprovisionamiento, gestión, configuración y prácticas de despliegue, la entrega de software se vuelve más natural, reduciendo los retrasos y, en consecuencia, los conflictos entre equipos.

2.2.- Herramientas
Es importante señalar que las herramientas solas no son DevOps. Lo que hace que las herramientas sean "DevOps" es la forma en que se usan, no las características fundamentales de las propias herramientas. Además, de las herramientas que se utilizan, una parte igualmente importante de esta cultura son los valores, normas y conocimientos. Examinando cómo trabaja la gente, la tecnología que utilizan, cómo la tecnología influye en la forma en que trabajan y cómo influyen las personas, se puede tomar decisiones intencionales sobre el panorama del desarrollo de software en las empresas (Davis y Daniels, 2016).

Como se observa en la figura 3, existen diferentes tipos de herramientas en primer lugar, las herramientas DevOps que permiten la interacción entre el personal de desarrollo y el de operaciones, ejemplo son los sistemas de gestión de proyectos como Jira y los rastreadores de errores. En segundo lugar, las herramientas DevOps que unen las disciplinas de desarrollo y operaciones, ejemplo de esto son los sistemas CI como Jenkins, especialmente si se utilizan conductos más avanzados para encadenar diferentes pasos en el despliegue y entrega. En tercer lugar, las herramientas DevOps de automatización de lanzamientos como Ansible, y los motores de orquestación de la nube, como Kubernetes.

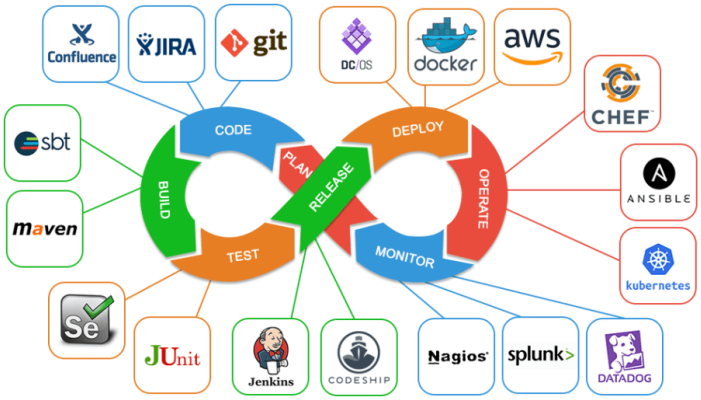

Figura 3 – Distribución de Herramientas DevOps por fases

Los sistemas de control de código fuente o SCM (Source Code Management) se utilizan para llevar el control sobre los cambios realizados en el código fuente por los distintos usuarios: modificaciones, creación y merge de ramas, resolución de conflictos entre versiones de ficheros, etc.

Integración Continua (CI)
Es una práctica de desarrollo que se basa en la integración frecuente del código fuente en un mismo repositorio compartido. Cada cambio introducido es verificado por una construcción automatizada, permitiendo a los equipos detectar los problemas más rápidamente.
Jenkins: Es un servidor de automatización de código abierto autónomo que se puede utilizar para automatizar todo tipo de tareas relacionadas con la creación, prueba y entrega o implementación de software.

Figura 4 - Logotipo característico de Jenkins.

Al usar Jenkins, las compañías de software pueden acelerar su proceso de desarrollo del código, ya que Jenkins puede automatizar, agilizar y aumentar el ritmo de toda la compilación y las pruebas de los proyectos. Además, Jenkins puede ser implementado a lo largo de todo el ciclo de vida completo del desarrollo, desde la fase de construcción inicial, la fase de pruebas, en la documentación del software, en su implementación y en todas las demás etapas existentes dentro del ciclo de vida del software.

Prueba
Las herramientas de prueba son imprescindibles para validar el código antes y después de integrarlo. Uno de los pilares de DevOps es la ejecución automatizada de las pruebas para detectar errores lo antes posible.

Entrega Continua (CD) y Despliegue Continuo
La entrega y el despliegue continuo tienen como objetivo hacer llegar al cliente final lo antes posible y con calidad, las modificaciones que se han implementado en desarrollo.

Monitorización
La monitorización es esencial para saber el estado de los sistemas y servicios en todo momento. Es necesario recopilar esa información, mostrarla de una manera adecuada y que todos los usuarios tengan acceso a ella.

Seguridad
La seguridad en DevOps es fundamental, sin embargo, al buscar desarrollos ágiles se puede correr el riesgo de dar menos importancia a la seguridad. DevSecOps se ocupa de gestionar la seguridad dentro de todo el proceso DevOps con una serie de herramientas especializadas para esa labor.

Virtualización y containerización
La virtualización es una tecnología que permite que un equipo anfitrión (host) con un sistema operativo propio pueda ejecutar una o más máquinas virtuales con sistemas operativos clientes. Para ello un programa crea un entorno virtual (máquina virtual) donde se sintetiza un entorno de computación real (NIC virtual, BIOS, tarjeta de sonido, vídeo, etc.,). Los contenedores son también una tecnología de virtualización, pero no necesitan un programa hypervisor ni máquinas virtuales para ser ejecutados, sino que se ejecutan directamente sobre el kernel del sistema operativo anfitrión (OS-level virtualization), que se encarga de ejecutar los distintos contenedores y de aislarlos unos de otros.

Orquestación
Actualmente las aplicaciones suelen ser complejas y por regla general no basta con desplegar un solo contenedor en producción, lo habitual es tener que desplegar varios, por ejemplo, un contenedor para el Front-End, otro para la base de datos, otro para determinados servicios, etc. Para solucionar esta necesidad surgió la orquestación de contenedores esto es herramientas que automatizan el despliegue, la gestión, el escalado, la interconexión y la disponibilidad de las aplicaciones basadas en contenedores.

Comunicación y colaboración
No son estrictamente herramientas de DevOps, sino más bien de gestión de proyectos y de trabajo en equipo, sin embargo, son imprescindibles para que la información fluya y se comparta por todos los miembros de la organización involucrados en los procesos DevOps.

3.- Recomendaciones de DevOps

3.1.- Practicas DevOps

A continuación, se detallan algunas técnicas específicas que deberá incluir al adoptar DevOps:

Mejora continua

En la auténtica filosofía Lean, la adopción de procesos no es una tarea inmediata: es un proceso continuado. Las organizaciones deberán tener procesos internos que identifiquen las áreas de mejora a medida que la organización vaya madurando y aprendiendo de los procesos que ha adoptado. Muchas empresas tienen equipos de mejora de procesos que trabajan en la mejora de procesos basándose en las lecciones aprendidas; otras permiten a los equipos que adoptan el proceso hacer una autoevaluación y determinar sus propias rutas de mejora de procesos. Independientemente del método utilizado, el objetivo es posibilitar una mejora continua

Planificación de versiones

La planificación de versiones es una función empresarial crítica, impulsada por las necesidades de ofrecer nuevas capacidades a los clientes y por la oportunidad de estas necesidades. Por tanto, las empresas precisan procesos de gestión y planificación de versiones bien definidos que promuevan hojas de ruta, planes de proyecto y calendarios de entrega, así como trazabilidad de extremo a extremo en estos procesos.

Integración continua

La integración continua, añade un enorme valor a DevOps al permitir a grandes equipos de desarrolladores que trabajan con componentes de múltiples tecnologías entregar el software de forma Agile. También garantiza que el trabajo de cada equipo esté integrado continuamente con el del resto de equipos de desarrollo y a continuación sea validado. Con ello, la integración continua reduce el riesgo e identifica los problemas en una etapa temprana del ciclo de vida de desarrollo del software.

Entrega continua

La integración continua conduce de forma natural a la práctica de la entrega continua: el proceso de automatizar la implantación del software en los entornos de comprobación, pruebas del sistema y producción. Aunque algunas organizaciones no llegan hasta la producción, las que adoptan DevOps suelen utilizar el mismo proceso automatizado en todos los entornos para mejorar la eficiencia y reducir el riesgo introducido por procesos poco uniformes. En entornos de prueba, automatizar la configuración, refrescar los datos y a continuación implantar el software en el entorno de prueba, seguido de la ejecución de pruebas automatizadas acelera los ciclos de feedback a desarrollo de los resultados de las pruebas.

Comprobación continua

Desde una perspectiva de los procesos, usted deberá adoptar procesos en tres áreas para hacer posible la comprobación continua:

- Provisión y configuración del entorno de prueba
- Administración de los datos de prueba
- Pruebas de integración, función, rendimiento y seguridad

En una organización, los equipos de QA han de determinar qué procesos adoptar para cada área. Los procesos que adopten pueden variar entre proyectos, basándose en las necesidades de comprobación concretas y en los requisitos de los acuerdos de nivel de servicio.

Monitorización y feedback continuos

El feedback de los clientes puede provenir de distintas fuentes, desde incidencias abiertas por los clientes, solicitudes formales de cambio, quejas informales o puntuación en la app stores. Debido especialmente a la popularidad de las redes sociales y app stores, las empresas precisan procesos bien definidos que absorban el feedback de muy numerosas fuentes para incorporarlos en los planes de entrega del software. Estos procesos deberán ser lo suficientemente ágiles como para adaptarse a los cambios en el mercado y en la normativa.

3.2.- ¿Qué no es DevOps?

A) No es Ops: No es "¡nos están quitando el empleo!"

Algunas personas creen que DevOps implica que los diseñadores están asumiendo las tareas de control y haciéndolo sin la ayuda de nadie más. Una parte de eso es válida y otra no. Es un juicio equivocado que DevOps se origina desde el lado de avance de la casa para acabar con las tareas - DevOps, y sus precursores en las actividades hábiles, se están iniciando fuera de los grupos de actividades como regla.

B) No es (simplemente) aparatos

DevOps es, además, no sólo la actualización de una disposición de los instrumentos. Una de las razones por las que creo

que se necesita un significado más reconocido de DevOps es que tener diferentes definiciones confusas e inadecuadamente organizadas construye el riesgo de que los individuos vayan por la "hipótesis" y actualicen los procedimientos o aparatos de DevOps sin los estándares como una prioridad principal, lo que es ciertamente un anti-patrón.
La robotización puede hacer tanto daño como la mecanización inteligente puede aportar ventajas. En consecuencia, los profesionales coordinados le revelarían que empezar simplemente a trabajar en énfasis o abrazar otras prácticas particulares sin iniciar un esfuerzo conjunto importante probablemente no va a funcionar genuinamente bien. Hay algunos grupos de organizaciones para los que he trabajado que recibieron una parte de las técnicas, así como los aparatos de destreza, pero no sus estándares y los resultados fueron problemáticos.

C) No es (simplemente) cultura
Numerosas personas exigen que DevOps "es simplemente cultura" y que no se puede importar la palabra a una regla o práctica determinada, sin embargo, siento que esto es exagerado y equivocado. Coordinado no ha ayudado a un gran número de tiendas de DEV a la luz del hecho de que el trabajo en él cesó en la "cultura", con consejos para abrazar a los colegas y los especialistas principales que distinguieron los procedimientos prescritos básicamente pronunciando que todo era claramente obvio y declinando ser más prescriptivo. (A pesar de que hay algo de eso).

D) No es (simplemente) Dev y Ops
Es más, por fin no es excluyente. Algunas personas se han quejado: "¡No debería decirse algo sobre los individuos de seguridad! ¡Y también a los administradores de los arreglos! ¿Por qué abandonarnos?" El hecho del asunto es que cada uno de los miembros al hacer un elemento o marco debe formar equipo desde el principio: empresarios, ingenieros y personal de actividades, esto incorpora la seguridad y el sistema. Hay varios tipos de socios empresariales e ingenieros también; a la luz del hecho de que todo el mundo no obtiene una salida particular ("¡Tengan en cuenta a los planificadores de símbolos!") no implica que estén excluidos. La primera gente de avance de pies ligeros estaba, en su mayor parte, contemplando el esfuerzo coordinado de "negocio + dev", y DevOps está sacando a relucir cuestiones y acuerdos en torno a la cooperación "DEV + Operaciones", sin embargo, la consecuencia de desarrollo de esto es "todo el mundo haciendo equipo". En ese sentido, DevOps es sólo un avance notable para que un maestro participe en la cultura general de esfuerzo coordinado de pies ligeros que debería incluir todos los órdenes en una asociación. Por lo tanto, cualquiera que se interese por la transmisión del producto o la administración es una pieza de DevOps.

4.- CONCLUSIONES

Actualmente las empresas que se dedican al desarrollo de aplicaciones están evolucionando rápidamente y de una forma compleja, lo cual imposibilita realizar un análisis cuidadoso para adoptar una plataforma tecnológica que le permita responder rápidamente las necesidades y exigencias de los clientes. DevOps se ha convertido en una cultura que abarca buenas prácticas y herramientas útiles que tienen como objetivo unificar el desarrollo de software (Dev) y la operación de software (Ops), ya que amplía el uso de las prácticas ágiles a las operaciones para fomentar la colaboración y agilizar todo el proceso de entrega de software de forma holística. El objetivo de DevOps es la unificación y la automatización de procesos, y los ingenieros de DevOps son fundamentales para las tareas relacionadas con la combinación de código y el mantenimiento y la gestión de aplicaciones. Para todas estas tareas, no solo es necesario comprender los ciclos de vida del desarrollo, sino también la cultura de DevOps, su filosofía, prácticas y herramientas.

REFERENCIAS

[1] Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., & Mikkonen, T. (Eds.). (2016). *Product-Focused Software Process Improvement* (Vol. 10027). Springer International Publishing. https://doi.org/10.1007/978-3-319-49094-6

[2] Erich, F. (2019). Devops is simply interaction between development and operations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11350 LNCS*. Springer International Publishing. https://doi.org/10.1007/978-3-030-06019-0_7

[3] Fitzgerald, B., & Stol, K. J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, *123*, 176–189. https://doi.org/10.1016/j.jss.2015.06.063

[4] Ghantous, G. B., & Gill, A. Q. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *Proceedings Ot the 21st Pacific Asia Conference on Information Systems: "'Societal Transformation Through IS/IT'", PACIS 2017*.

[5] Jabbari, R., Ali, N. Bin, Petersen, K., & Tanveer, B. (2016). What is DevOps? A systematic mapping

study on definitions and practices. *ACM International Conference Proceeding Series*, *24-May-201*. https://doi.org/10.1145/2962695.2962707

[6] Jha, P., & Khan, R. (2018). A Review Paper on DevOps: Beginning and More To Know. *International Journal of Computer Applications*, *180*(48), 16–20. https://doi.org/10.5120/ijca2018917253

[7] Krishna Kaiser, A. (2018). Introduction to DevOps. In *Reinventing ITIL® in the Age of DevOps* (pp. 1–35). Apress. https://doi.org/10.1007/978-1-4842-3976-6_1

[8] Lassenius, C., Dings yr, T., & Paasivaara, M. (2015). DevOps: A Definition and Perceived Adoption Impediments. *Lecture Notes in Business Information Processing*, *212*, 166–177. https://doi.org/10.1007/978-3-319-18612-2

[9] Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, *157*, 1–16. https://doi.org/10.1016/j.jss.2019.07.083

[10] Maroukian, K., & Gulliver, S. R. (n.d.). *LEADING DEVOPS PRACTICE AND PRINCIPLE ADOPTION*.

[11] Martin, L., & Affairs, V. (2015). *Architectural Implications of DevOps Stephany Bellomo Senior Member of Technical Staff*.

[12] Melys, I. D., & Delgado, B. (n.d.). *SOFTWARE THE DEVOPS PARADIGM AND ITS IMLEMENTATION IN THE SOFTWARE DEVELOPMENT*. http://revistas.unica.cu/uciencia

[13] NOCERA, D., NOIA, T. DI, & GALLITELLI, D. (2016). *Innovative techniques for agile development: DevOps methodology to improve software production and delivery cycle*. *October*. https://doi.org/10.13140/RG.2.2.33234.96961

[14] P P, J. (2019). Exploring Devops: Challenges and Benefits. *Journal of Information Technology and Digital World*, *01*(01), 27–37. https://doi.org/10.36548/jitdw.2019.1.004

[15] Pérez Hoyos, L. (2018). DevOps: IT Development in the Era of Digitalization. *Universidad de Valladolid. Escuela de Ingenierías Industriales*. https://uvadoc.uva.es/handle/10324/31469

[16] Stahl, D., Mårtensson, T., & Bosch, J. (2017). Continuous practices and Devops: Beyond the buzz, what does it all mean? *Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017*, *2017-Janua*, 440–448. https://doi.org/10.1109/SEAA.2017.78

[17] Villamarín, A. É. (n.d.). *Grado de Tecnologías de Telecomunicación Área del Proyecto: Administración de redes y sistemas operativos Trabajo Fin de Grado Introducción a DevOps para la mejora de los procesos de desarrollo con herramientas Open Source*