



Revista Guillermo de Ockham

ISSN: 1794-192X

Universidad de San Buenaventura

Muñoz, Diego Armando; Ordóñez, Hugo; Bucheli, Víctor
Lineamientos para la implementación del modelo CALMS de DevOps
en mipymes desarrolladoras de software en el contexto surcolombiano
Revista Guillermo de Ockham, vol. 18, núm. 1, 2020, Enero-Junio, pp. 81-91
Universidad de San Buenaventura

DOI: <https://doi.org/10.21500/22563202.4270>

Disponible en: <https://www.redalyc.org/articulo.oa?id=105368880008>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en [redalyc.org](https://www.redalyc.org)



Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

Lineamientos para la implementación del modelo CALMS de DevOps en mipymes desarrolladoras de software en el contexto surcolombiano

Diego Armando Muñoz¹
Universidad San Buenaventura (Colombia)

Hugo Ordóñez²
Universidad del Cauca

Víctor Bucheli³
Universidad del Valle (Colombia)

Recibido: junio de 2019 – Revisado: septiembre de 2019 – Aceptado: abril de 2020

Referencia norma APA: Armando-Muñoz, D., Ordóñez, H., & Bucheli, V. (2019). Lineamientos para la implementación del modelo CALMS de DevOps en mipymes desarrolladoras de software en el contexto surcolombiano. *Rev. Guillermo de Ockham*, 18(1), 81-91. doi: <https://doi.org/10.21500/22563202.4270>

Resumen

DevOps es un término relativamente nuevo que apareció por primera vez en 2009. Sin embargo, empresas como Etsy, Facebook, Amazon o Netflix son líderes en la implementación del nuevo paradigma. En este trabajo se presenta un conjunto de lineamientos para la implementación del modelo CALMS (DevOps) en mipymes de desarrollo de software en el contexto surcolombiano, así como un modelo que tiene en cuenta los aspectos técnicos y organizacionales para lograr un ambiente de desarrollo novedoso que permita la integración de DevOps al contexto del desarrollo de software colombiano. El modelo fue probado en una empresa mipyme y los resultados son alentadores. Se pasó de hacer un despliegue semanal a un despliegue diario, y de los 20 despliegues que se hicieron en total, 16 fueron puestos exitosamente en producción. Finalmente, discutimos cómo DevOps puede incrementar la productividad de las organizaciones de desarrollo de software y cómo la implementación de los lineamientos en un ambiente integrado y probado puede incrementar la competitividad de las empresas de software en un mercado globalizado.

Palabras clave: DevOps, desarrollo, operaciones, cultura, colaborativa

Guideline to implement the CALMS model (DevOps) at mipymes in the software development organizations of the South of Colombia

Abstract

The term DevOps appears at 2009, it is a new paradigm, in which the developing process and deployment are integrated. Currently enterprises such as Etsy, Facebook, Amazon or Netflix are leaders in the implementation of

1. Laboratorio de Investigación para el Desarrollo de Ingeniería de Software, Universidad de San Buenaventura, Cali, Colombia. Correo electrónico: dams9011@gmail.com
2. Universidad del Cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones. Correo electrónico: hugoordones@unicauca.edu.co
3. Escuela de Ingeniería de Sistemas y Computación, Facultad de Ingeniería, Universidad del Valle, Colombia. Correo electrónico: victor.bucheli@correounivalle.edu.co

DevOps paradigm. In this work, we present a guideline to implement the CALMS model (DevOps) at mipymes in the software development organizations of the South of Colombia. We present the technical and organizational aspects of guidelines; it integrates a newfangled development environment to develop of DevOps in the Colombian context. The purposed model was tested at mipymes from Colombia and the results are promising. The mipymes did a daily commit, when without of the guidelines they did a commit per week, in addition, 16 deployments were successful and upload to production, front 20 total deployments. Finally, we discuss how the implementation of the guidelines in an environment tested and integrated can be more productive organization, as well as how they can be competitive software companies in a globalized market.

Keywords: DevOps, develop, operations, culture, colaborative

Introducción

Con el pasar de los años el proceso de diseño de software ha ido evolucionando. Un claro ejemplo de ello es el ciclo de vida del desarrollo de software, que ha pasado de ser lineal con el modelo en cascada a iterativo e incremental con las metodologías ágiles. Esto se debe a que los clientes cada vez son más exigentes, tienen expectativas mayores con respecto al contenido de las aplicaciones, exigen aumento en la calidad del software y requieren tiempos de entrega más cortos (Brown, 2013), (Kruchten, 2013). En este contexto, las mipymes de desarrollo de software deben mantenerse a la vanguardia en cuanto a tecnologías, metodologías, tendencias, formas de trabajo, estrategias o cualquier otro factor que les dé un valor agregado con respecto a las otras empresas competidoras del mercado (Laukkarinen, Kuusinen, & Mikkonen, 2018).

Las metodologías ágiles predominan desde el 2002 (Ambler & Ambler, 2011) y los clientes de las empresas desarrolladoras de software requieren una respuesta cada vez más rápida y con mayor estabilidad, segura y predecible. Es decir, una mezcla entre estabilidad y confiabilidad. Es en este punto que nace el movimiento DevOps, el cual fomenta un ambiente de cultura organizacional basado en la colaboración y comunicación entre los equipos de desarrollo (Dev) y operaciones (Ops) en las empresas desarrolladoras de software (Dyck, Penners, & Lichter, 2015), (Hussaini, 2014).

El fin de DevOps es reducir el tiempo que pasa desde que se sube un cambio al repositorio de control de versiones del software que se está construyendo (commit), hasta que el cambio es puesto en producción, garantizando así alta calidad en el software (Mohamed, 2015; Rajkumar, Pole, Adige, & Mahanta, 2016; Ravichandran, Taylor, & Waterhouse, 2016). DevOps, como cultura, tiene la característica principal de defender la automatización y el monitoreo en todos los pasos de la construcción del software, desde la integración, las pruebas y la liberación hasta la implementación y la administración de la infraestructura, apuntando así a ciclos de desarrollo

más cortos, mayor frecuencia de implementación, lanzamientos más confiables, en estrecha alineación con los objetivos comerciales de la empresa de desarrollo. Al modelo DevOps se le conoce también por el conjunto de actividades del modelo CALMS (por sus siglas en inglés de cultura, automatización, lean, métricas y compartir). El modelo busca cambiar la mentalidad entre los equipos de desarrollo y operaciones para aportar el valor que el negocio espera por parte del área de tecnología. Para ello, busca optimizar y automatizar al máximo las tareas que se ejecutan a lo largo del ciclo de vida de construcción de software, de forma que se pueda implantar en producción lo antes posible con las funcionalidades que el negocio ha solicitado (Kort & de Kort, 2016; Mueller, Wickett, Gaekwad, & Karayanev, 2010).

Aunque ya ha transcurrido una década desde el nacimiento de DevOps, aún sigue siendo un campo poco explorado en Latinoamérica y los casos de éxito de referencia encontrados pertenecen en su mayoría a grandes empresas de Europa y Estados Unidos. Esto conlleva poca documentación sobre métodos, técnicas, estrategias y lineamientos por seguir para adoptar DevOps en el contexto colombiano, así que la transformación a una cultura DevOps implica un aumento de esfuerzos y costos en las empresas del sector del desarrollo de software.

Basado en lo anterior, este artículo está centrado en proponer unos lineamientos culturales, técnicos y colaborativos que sirvan como guía inicial en la adopción de la cultura DevOps basado en el modelo CALMS enfocado en las mipymes desarrolladoras de software en Colombia, con el fin de que puedan responder a los desafíos cambiantes que exigen los clientes y el mercado internacional del software.

El resto del artículo se encuentra organizado de la siguiente manera. En primer lugar se presenta el contexto de motivación. En el apartado que le sigue, son descritos algunos de los trabajos de mayor relevancia en el campo. La sección siguiente presenta los lineamientos planteados para cada uno de los componentes del modelo DevOps.

Seguidamente, se describe el estudio de caso que se hizo para dar validación a los lineamientos propuestos y finalmente se presentan las conclusiones y el trabajo futuro.

Escenario de motivación

DevOps es un término relativamente nuevo que apareció por primera vez en 2009. Sin embargo, muchos de los conceptos y prácticas que propone existían desde antes. Si bien no hay documentada una especificación clara o estándar de las prácticas de DevOps, el paradigma se ha notado como una de las tendencias tecnológicas en alza en los últimos diez años (Nicolau de França, Jerónimo, & Travassos, 2016; Riungu-Kalliosaari, Mäkinen, Lwakatare, Tiihonen, & Männistö, 2016).

El informe de State of DevOps del 2018 (Velásquez, Kim, Kersten, & Humble, 2018), muestra que el 50 % de las empresas encuestadas han adoptado DevOps y están buscando mejoras y el 32 % de ellas planea adoptar DevOps en los próximos doce meses. En el mismo informe también se aclara que se obtienen muchos beneficios al adoptar DevOps en una organización, entre estos está que las organizaciones pueden reducir sus costos de desarrollo y mantenimiento en un 18 %. Además en el informe se reporta que DevOps es la razón principal detrás de los aumentos en los ingresos y la cantidad de nuevos clientes, lo cual se traduce en un aumento del 20 % en los negocios. A la luz de estas cifras, se puede ver claramente que existe un valor real y una oportunidad de mejora de las empresas si adoptan la cultura DevOps, hasta el punto de que empresas como Etsy, Facebook, Amazon y Netflix son líderes en el campo de DevOps. Hoy en día, todo tipo de negocio ha entrado en la acción de DevOps. Así, la compañía de medios Sony Pictures, la compañía de servicios financieros Barclays Bank y el fabricante de productos de construcción USG, son otros referentes del éxito de DevOps.

A pesar de todos estos casos de éxito, el informe de Gene Kim, experto en DevOps, aclara que DevOps es tanto para grandes empresas como Google, Amazon, Facebook y Netflix como también para cualquier organización tecnológica en las que se destaquen las siguientes prácticas:

- Técnicas: despliegue continuo de software, menos complejidad para administrar y rápida resolución de problemas.
- Culturales: más felicidad, equipos más productivos, mayor compromiso de los empleados y mayores oportunidades de desarrollo profesional.

- Negocio: rápida entrega de funcionalidades, entornos operativos más estables, mejora la comunicación y la colaboración y más tiempo para innovar (en lugar de arreglar/mantener).

Con base en las anteriores prácticas, es notable la necesidad de implantar un entorno DevOps en las mipymes colombianas. En este sentido, el presente trabajo expone una serie de lineamientos para la adopción de este modelo de cultura de trabajo, el cual permite mayor flexibilidad y claridad en el momento de iniciar con el proceso de implementación. Busca como efecto de su implementación, el incremento en la productividad, el reconocimiento, ingresos y la visibilidad en el mercado.

Trabajos relacionados

Adoptar e implementar cambios en los procedimientos tradicionales en las empresas siempre resulta complejo y suele requerir una inversión. De esta forma, cada vez que una organización adopta una tecnología, metodología o enfoque nuevo, dicha adopción debe ser impulsada por una necesidad empresarial. En las empresas desarrolladoras de software la adopción de DevOps es una necesidad empresarial que busca incrementar la competitividad de la organización. Con base en esto, los trabajos presentados a continuación plantean las prácticas fundamentales en la implementación de DevOps en las empresas.

En Wettinger, Breitenbücher, Kopp, & Leymann, (2016), se define a DevOps como un paradigma emergente que tiene como objetivo integrar estrechamente los desarrolladores con el personal de operaciones, para lo cual enfatiza en que es una ventaja competitiva crítica responder rápidamente a los cambios culturales y organizativos e impulsar constantemente nuevos enfoques, herramientas y artefactos de código abierto para implementar aplicaciones en la nube mediante un proceso automatizado. Para ello, presenta una clasificación sistemática de artefactos DevOps para transformar diferentes tipos de artefactos hacia Tosca (acrónimo del inglés Topology and Orchestration Specification for Cloud Applications), un estándar emergente que propone determinados mecanismos para permitir la portabilidad entre diferentes nubes. El marco se implementa mediante una cadena de herramientas de código abierto de extremo a extremo. Además, valida el enfoque presentado para mostrar su viabilidad práctica mediante un estudio de caso.

Por otra parte, Luz, Pinto, & Bonifácio (2019), señalan que debido a su creciente interés y definiciones, se sabe poco sobre la comprensión de los profesionales sobre los

caminos exitosos para la adopción DevOps (Zhu, Bass, & Champlin-Scharff, 2016). Por lo tanto, se detallan escenarios reales de adopción de DevOps presentando una teoría, un modelo y un estudio de caso. Utiliza quince escenarios de adopción exitosa de DevOps en empresas de diferentes dominios y países. Presenta un modelo (es decir, un flujo de trabajo para la adopción de DevOps) evaluado mediante un estudio de caso en una institución del Gobierno brasileño, y utilización de un grupo de enfoque para recopilar las percepciones de la compañía sobre la adopción de DevOps. Del caso de estudio se detallan escenarios reales y explica el rol de cada categoría durante la adopción de DevOps. A través de esto proporciona evidencia de que la colaboración es el factor la principal en DevOps, en contraste con la automatización, y las herramientas pueden ser suficientes para lograr DevOps.

En Lwakatare *et al.*, (2019), define a DevOps como importante en la capacidad de actualizar de manera frecuente y confiable un sistema en estado operativo, para lo cual se basa en colaboración y automatización multifuncional entre el desarrollo de software y las operaciones, enfocados en los cambios requeridos en los aspectos técnicos, organizativos y culturales. El trabajo se base en el estudio exploratorio de cómo se implementa DevOps en la práctica en el desarrollo de aplicaciones y servicios web en pequeñas y medianas empresas. Se hizo un estudio de casos múltiples en cinco contextos de desarrollo diferentes con implementaciones de DevOps, ya que se lograron sus beneficios, como lanzamientos rápidos y errores mínimos de implementación. Los datos se recopilaron principalmente a través de entrevistas con 26 profesionales y observaciones realizadas en las empresas.

A partir del estudio de los trabajos relacionados, se puede identificar que la mayoría de trabajos están centrados en las herramientas, en la integración de la colaboración, en las prácticas de desarrollo y de despliegue continuo. Aunque las prácticas y herramientas son importantes en el entorno DevOps, ninguno de los trabajos se interesa por definir los lineamientos que se deben seguir antes de implementar DevOps en una empresa, ya que estos permiten identificar el estado actual de las prácticas y cuáles serían las de mayor facilidad de implementación en relación al contexto de la empresa.

Lineamientos propuestos

Según Gary Gruver, DevOps debe centrarse en principios de propiedad y flexibilidad en lugar de seguir una

hoja de ruta rígida propuesta por un modelo de madurez (Sharma, 2017; The, Of, In, & Software, 2014). Por lo tanto, los lineamientos propuestos se basan en CALMS que más que un modelo de referencia es una guía de los principios que se deben tener en cuenta en cualquier adopción de DevOps.

El entorno al que se pretende llegar con los lineamientos propuestos es una cultura de DevOps basada en la automatización de inicio a fin; es decir, desde que se sube el cambio del código al repositorio de versiones hasta que es desplegado en el ambiente de producción, mientras se crea una colaboración y comunicación efectiva entre las partes involucradas en la construcción del producto de software.

Como primera medida es recomendable realizar la encuesta de evaluación inicial propuesta,⁴ para así conocer:

- El nivel de madurez actual de colaboración y prácticas DevOps efectuadas en la organización.
- El interés y la disposición de adoptar DevOps por parte de los equipos de desarrollo y operaciones.

Una vez claros dichos puntos, se definen los lineamientos que se deben tener en cuenta en cada uno de los principios de CALMS (Figura 1).

Lineamientos culturales.

En esencia, el éxito de DevOps depende en gran medida de la colaboración entre los desarrolladores y el personal de operaciones de TI. Solo a través del intercambio diario y abierto de conocimiento y mejores prácticas, una empresa puede resolver problemas y optimizar aplicaciones de manera eficiente en DevOps. Reunir puntos de vista cuando es necesario, no solo requiere las herramientas adecuadas, sino también la cultura correcta para fomentar dicha cultura. En este sentido, se propone una serie de actividades/juegos para la construcción de una cultura DevOps.

Objetivos, recompensas y valores compartidos. La colaboración entre operaciones y desarrollo es difícil no porque ninguna de las dos áreas quiera colaborar, sino porque el enfoque es diferente. Desarrollo desea entregar requerimientos nuevos lo más rápido posible, mientras que operaciones tiene como principal medida de éxito la estabilidad. Entonces, ¿cómo lograr que dos grupos de dos mundos diferentes colaboren? Establezca objetivos compartidos, recompensas compartidas y valores compa-

4. <https://docs.google.com/forms/d/e/1FAIpQLSfIMUJcTmWfxCysHI2O1sR5cVjJwttbqJl7oKia0sen9fB8TQ/viewform>

tidos. Al conectar ambas áreas con un conjunto simple de objetivos (lo que estamos tratando de hacer), recompensas —obtenes esto por hacerlo— gamificación (Navia, 2019) —y valores (esto es lo que espero de todos) se construye un entorno que fomenta y apoya la colaboración.

Pautas de rendimiento. El rendimiento de la aplicación (velocidad y disponibilidad) ha sido la principal fuente de fricción entre el personal de operaciones y el de desarrollo, ya que cuando el rendimiento falla los equipos suelen culparse unos a otros, por lo que se recomienda establecer un conjunto claro de pautas de rendimiento. Por ejemplo, si un producto de software no funciona a un cierto umbral, no entra en producción y todas las versiones futuras quedarán en espera indefinidamente hasta que el primer producto sea corregido.

Empezar poco a poco. A menudo, las empresas que desean cambiarse a un modelo de DevOps intentan hacer el cambio de manera general. Pero normalmente es mejor comenzar de a poco y utilizar un proceso que permita colaborar a todo el equipo. Es decir, en lugar de tratar de automatizar una compilación complicada existente que aporta mucho al negocio y es difícil de administrar por todo el equipo, es mejor comenzar con una compilación simple y trivial para establecer un nuevo proceso limpio de integración y despliegue continuo que todo el equipo pueda entender.

Elija las personas correctas y construya confianza entre los equipos. Seleccione equipos pequeños de trabajo (no más de diez personas) que se caractericen por ser buenos para el trabajo en equipo, dispuestos y capaces de aceptar cambios y nuevas maneras de trabajar. Una vez esté seguro del equipo elegido para la adopción de DevOps, fomente la confianza, la comprensión y la importancia en el trabajo de los demás, incluidos los miembros del personal de un equipo en el otro. En conclusión, DevOps no es el trabajo de un equipo sino el trabajo de todos, pues la cultura de DevOps tiene que ver con responsabilidad compartida (Grass Ramírez, Collazos Ordóñez, & González González, 2017; Ramírez, 1939). Eso significa un cambio hacia la transparencia, la comunicación y la colaboración entre el desarrollo, las operaciones de TI y el negocio.

Lineamientos de automatización

Uno de los principios básicos de DevOps es implementar la automatización en todas las etapas de entrega, desde la verificación del código hasta la implementación, que incluye: integración de código, compilaciones, pruebas, implementación y verificación de las compilaciones im-

plementadas. Esta automatización acelera todas las etapas de la entrega de software para que los desarrolladores obtengan retroalimentación e impacto en sus cambios rápidamente, lo que ayuda a acelerar el tiempo total de comercialización. DevOps depende en gran medida de la automatización y eso significa que necesita herramientas. Estas herramientas no deben estar dispersas en la organización, sino ser una cadena de herramientas compatibles que permitan automatizar gran parte del desarrollo de software de extremo a extremo y el proceso de despliegue (Jabbari, Ali, Petersen, & Tanveer, 2016).

Por las razones anteriores, se hizo un análisis comparativo de las distintas herramientas de cada una de las fases presentes en el ciclo de vida de DevOps, teniendo siempre en cuenta que la propuesta está enfocada en mipymes desarrolladoras de software, las cuales posiblemente no disponen de muchos recursos económicos para la compra de herramientas licenciadas muy costosas

Cadena de herramientas propuesta

Integración continua. La clave de la integración continua es tener una compilación sin errores con las mejores técnicas y estándares de desarrollo y control de versiones para la posterior puesta en producción del código fuente en los entornos específicos. Herramientas seleccionadas: Jenkins + Git + Maven + SonarQube.

Pruebas de código continuo. El objetivo de las pruebas continuas es proporcionar una respuesta rápida y continua sobre el nivel de riesgo empresarial, en la última compilación que se utiliza para determinar si el software está listo para avanzar a través del proceso de entrega en un momento dado. Herramientas seleccionadas: JUnit + RobotFramework.

Orquestación continua y despliegue. Los contenedores se crean para una implementación rápida y confiable de los componentes de la aplicación en cualquier infraestructura subyacente. Herramienta seleccionada: Docker+ AWS EC2.

Monitoreo continuo. La monitorización continua permite recopilar y analizar datos en entornos de prueba y producción. Herramientas como Librato, Nagios y Logstash generan los datos para el equipo que muestran si el rendimiento ha mejorado o empeorado y ayuda a tomar medidas correctivas para mejorar el rendimiento. Herramienta seleccionada: Prometheus.

Colaboración y comunicación continua. Trabajar por medio de incidentes y cultivar una cultura DevOps requiere comunicación constante y colaboración en equipo. Esto se hace mucho más fácil con la implementación de herramientas de ChatOps, que permiten a las personas comunicarse por correo electrónico, chat, texto, video y teléfono. Un entorno verdaderamente colaborativo brinda acceso a conversaciones y datos en todos los dispositivos, de múltiples formas integradas. Herramienta seleccionada: Slack.

Lineamientos *lean*

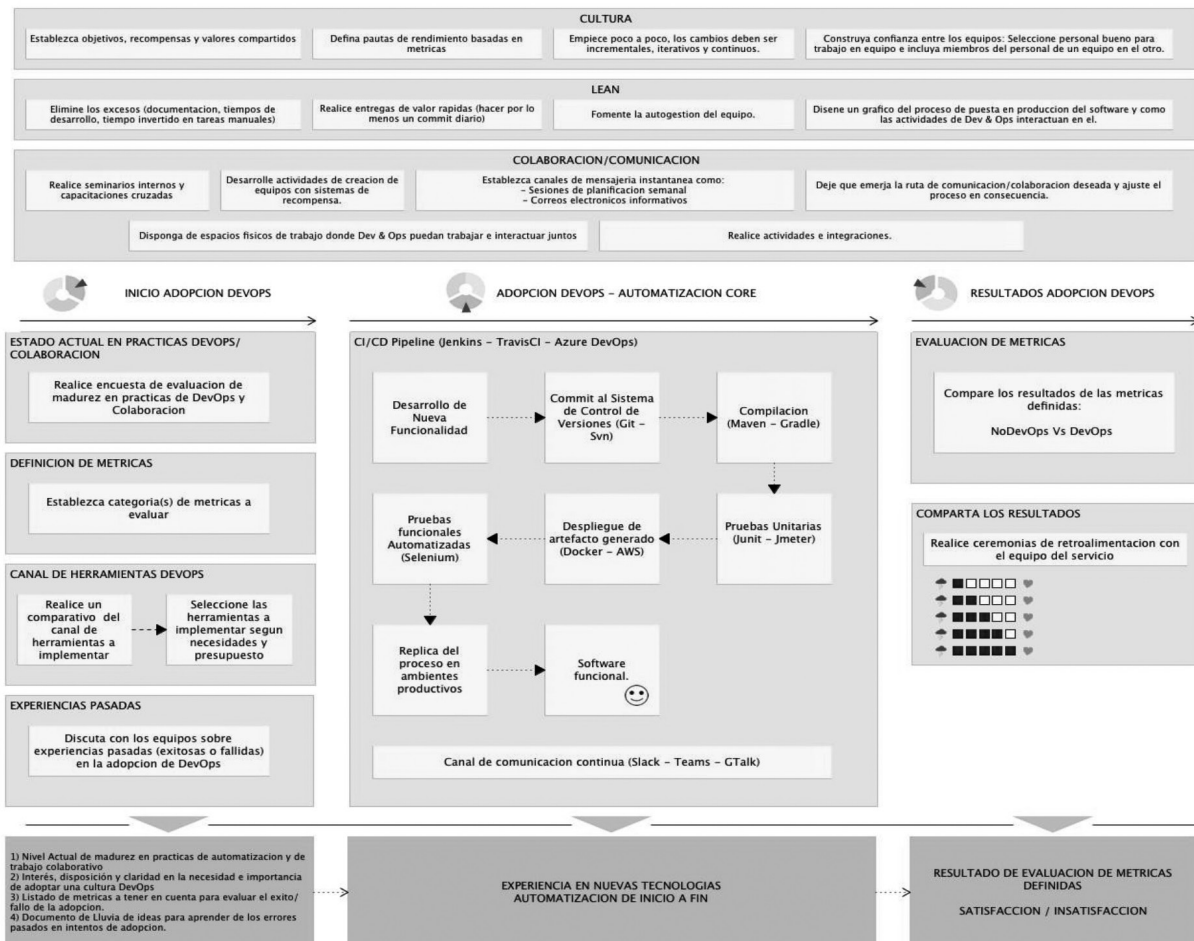
Automatizar todo puede ser un problema que complica en exceso la adopción completa de DevOps. Por lo tanto, es necesario concentrarse en mantener todo lo mínimo, pero útil. Lo que se recomienda es seguir con los siguientes principios de *lean*:

- **Eliminar los excesos.** Eliminar el exceso de documentación, exceso de funciones que no son explícitamente

formuladas por el cliente, transmisión de desarrollo de un grupo a otro, exceso de tiempo de desarrollo, etc. La clave es entregar exactamente lo que los clientes quieren.

- **Entregar lo más rápido posible.** Tradicionalmente, se ha favorecido un enfoque libre de errores sobre un desarrollo rápido. Sin embargo, Womack en Xu, David, & Kim (2018) argumenta que la velocidad no es un factor de costo. La entrega rápida tiene un ciclo de retroalimentación más rápido y está mejor equipada para mejorar el producto. Este es uno de los componentes clave de DevOps.
- **Empoderar al equipo.** Las decisiones técnicas deben involucrar a las personas que son responsables de la ejecución de los planes. Debido a que las decisiones se difieren hasta el último minuto posible, no hay tiempo para que las autoridades superiores orquesten las actividades de los trabajadores.

Figura 1
Lineamientos propuestos bajo principios CALMS



- Ver el todo. La integridad en sistemas complejos requiere una profunda experiencia en muchas áreas diversas. Desde una perspectiva de DevOps, desarrollo y operaciones tradicionalmente tienen un pensamiento dividido que los lleva a optimizar solo sus respectivos extremos del ciclo de desarrollo en lugar del rendimiento general. Esto da como resultado un producto o servicio final subóptimo.

Lineamientos de métricas y monitoreo

Los lanzamientos frecuentes proporcionan una gran flexibilidad, pero pueden poner en peligro el entorno de producción. Por ello, una aplicación desarrollada debe estar equipada con métricas útiles y ser monitoreada en tiempo real.

Estos son solo algunos ejemplos de lo que puede medir dentro de su entorno DevOps (Forsgren & Kersten, 2018; HERING, DeGrandis, & Forsgren, 2015):

Experiencia del cliente	Rendimiento de la aplicación	Velocidad	Calidad del Software
Número de visitas por usuario / por semana	Tiempo de respuesta de la aplicación	Frecuencia de liberaciones de código	Tasa de éxito de implementación
Tasa de crecimiento de usuarios	Tiempo de respuesta de la base de datos	Tiempo medio para la resolución de problemas.	Tasas de error de aplicación
Cantidad de tiempo pasado en la aplicación	Porcentaje utilización de recursos.		
Satisfacción del cliente	Tiempo de los queries a la base de datos		

Lineamientos de compartir/ colaboración / comunicación

Compartir es esencial para mejorar el flujo de comunicación y hacer que las personas trabajen juntas. Por lo tanto, es importante compartir ideas, experiencias y pensamientos dentro del equipo, entre equipos e incluso fuera de la empresa.

Mejorar la comunicación. Las empresas deben mejorar el compromiso general con el equipo de desarrolladores compartiendo lo que ambas partes (desarrolladores y operadores) hacen exactamente. Planee un seminario interno para explicar los procesos y las mejores prácticas que cada uno requiere y aprenda cómo integrar los dos. Desarrolle

actividades de creación de equipos que incluyan a todos y emplee un sistema de recompensa para la colaboración en equipo, con gamificación, por ejemplo. Haga que valga la pena que ambos equipos aprendan a trabajar juntos. La educación conjunta tanto del desarrollo como de las operaciones es clave.

Dejar que emerja la ruta deseada. Cada organización se comunica de manera diferente, razón por la cual cuando implemente medidas para alentar la colaboración, establezca una red amplia y luego deje que surja la ruta deseada. No obligue a los equipos a participar en rituales y ceremonias no deseados. Por ejemplo, si las personas dejan de aparecer en las reuniones y comienzan a comunicarse a través de un canal de *Slack* o páginas wiki en su lugar, ajuste el proceso en consecuencia.

Crear espacio. Debe crear espacio para que los desarrolladores y los equipos de operaciones trabajen. Eso significa compartir espacio físico y emocional si es posible (dejar de interactuar mediante tickets y conocerse como seres humanos).

Herramientas de comunicación. Chat y video. No se trata de centrarse en las herramientas, pero sí contar con herramientas que hagan más fluida y fácil la comunicación y colaboración. Por lo general, esto consiste en un sistema de chat basado en texto.

Comunicación continua. La comunicación continua es clave. Los mensajes de correo electrónico semanales de “qué está sucediendo en producción” (qué se está implementando, nueva infraestructura, etc.), canales de mensajes instantáneos, sesiones de planificación semanal y herramientas comunes de emisión de boletos, mantienen a todos sincronizados.

Caso de estudio

Nombre de la empresa: en adelante EMP

Tamaño de la empresa: 18 personas distribuidas así:

Desarrollo: 12 personas.

Operaciones: 3 personas.

Administrativo: 3 personas.

Descripción de la empresa

EMP es una mipyme proveedora de desarrollo ágil de software, que cuenta con tres líneas de negocio principales:

Servicios de Desarrollo de Software.

Aplicaciones Web
Aplicaciones Móviles
Outsourcing de recursos

Consultoría y capacitaciones.
Procesos de Desarrollo de Software.
Gestión de la configuración.
Arquitectura de Software.
Cloud
Gestión de Reglas de Negocio.

Productos de Software Propios.

Producto a automatizar

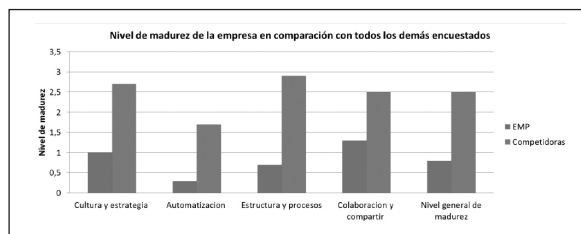
La automatización de inicio a fin se trabajó sobre uno de los productos de software propios de la empresa llamado EMPTeam, en el cual el personal debe registrar las horas y actividades trabajadas durante el día.

Estado actual DevOps-cultura. Personal de desarrollo y de operaciones abiertos al cambio, a aprender nuevas tecnologías, a iniciar cambios culturales y lo más importante de todo, conscientes de la necesidad de empezar a adoptar DevOps como parte de la transformación tecnológica del mercado y de la empresa.

Estado actual DevOps-automatización. No hay proceso automático de compilación, pruebas unitarias y funcionales, revisión de código estático y despliegue; es decir, todo el proceso de puesta en producción se hace manualmente.

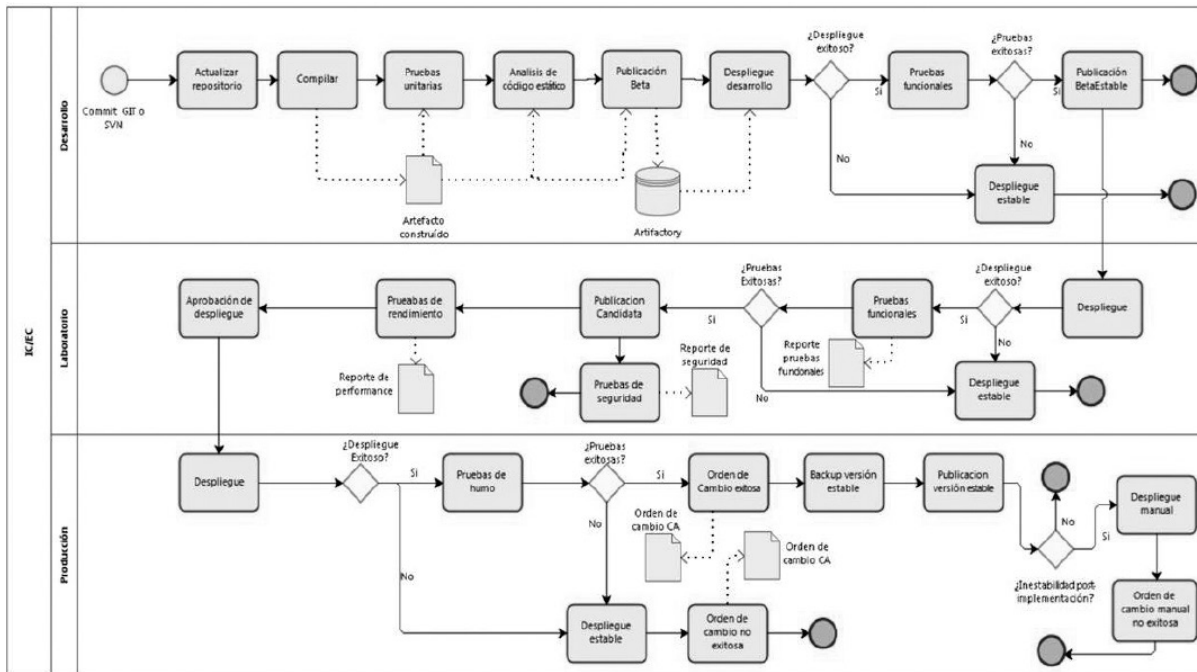
El diagnóstico del estado actual en cuanto a los dos pilares centrales de DevOps (cultura y automatización) surge como resultado de la encuesta.⁵ Por esta razón, el caso estudio se centrará en atacar el pilar mas débil de EMP (Figura 2); es decir, la automatización en todo el ciclo de construcción del software.

Figura 2
EMP frente a competidoras



Para esto se plantea un pipeline de CI/CD al que se espera llegar con los lineamientos de automatización propuestos (Figura 3).

Figura 3
Diagrama de CI/CD propuesto



5. <https://ibb.co/ft56Gw5>

Paso a paso del CI/CD propuesto

Descripción del Paso a Paso

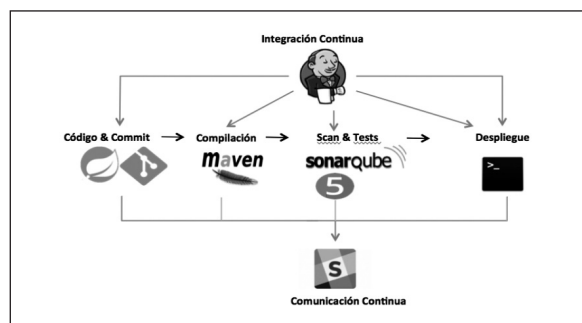
- El *commit/pull request* dispara el inicio del proceso de CI/CD.
- Se compila el código fuente.
- Se ejecutan las pruebas unitarias.
- Se realiza el análisis de código estático.
- Se despliega el artefacto generado en el ambiente de Desarrollo.
- Una vez se ejecutan y pasan las pruebas funcionales automatizadas despliega el artefacto en el ambiente de Pruebas.
- Una vez se ejecutan y pasan las pruebas funcionales automatizadas se ejecutan las pruebas de seguridad y rendimiento.
- Se realiza el despliegue del artefacto generado en el ambiente de producción.
- Una vez el despliegue pase las pruebas de humo, se guarda un *backup* como versión estable de producción.
- Se verifica la existencia de inestabilidades operacionales post implementación.
- Si se encuentran inestabilidades, se hace un despliegue manual de una versión estable.
- Si no se encuentran inestabilidades, el proceso finaliza.

Nota: si cualquier paso del *pipeline* propuesto falla, se notifica al equipo de desarrollo y el *pipeline* no avanza al siguiente paso.

Sin embargo, para llegar hasta este nivel de automatización es necesario hacer un recorrido previo en el que gradualmente se vayan incrementando las actividades por hacer y las tecnologías por utilizar en el pipeline de CI/CD, ya que como se sugirió en el lineamiento de “empezar poco a poco” la adopción de DevOps debe ser gradual, iterativa e incremental por lo que para el caso estudio propuesto no habrá una complejidad inicial tan alta y se enfocará principalmente en una configuración estándar de CI con Jenkins, compilación y ejecución de pruebas unitarias con Maven, revisión de código estático con SonarQube y despliegue por comandos Shell en un servidor controlado (Figura 4) para así obtener una vic-

toria rápida e ir creciendo exponencialmente hasta lograr la madurez esperada.

Figura 4
Pipeline de CI/CD - Caso Estudio

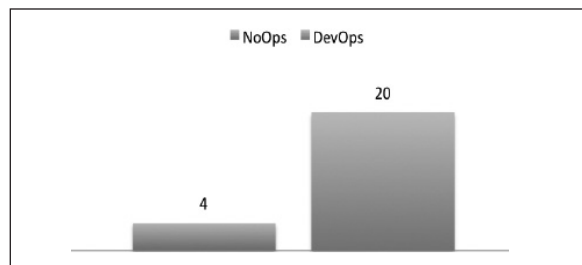


Resultados del caso de estudio

Con el *pipeline* de CI y despliegue *Shell* trabajado en el caso estudio, se obtuvieron resultados exitosos durante el primer mes de uso en cuanto a las dos métricas principales por evaluar, de la siguiente manera:

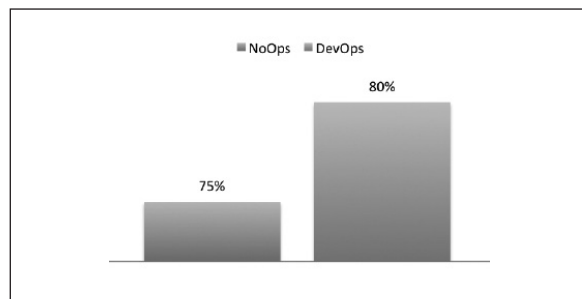
Frecuencia de despliegues. Se pasó de hacer un despliegue semanal a un despliegue diario, incrementando así en 5x la cantidad, como se observa en la Figura 5a.

Figura 5a
Frecuencia de Despliegues mensuales No DevOps Vs DevOps.



Tasa de éxito de implementación. Se pasó de un porcentaje del 75 % de despliegues exitosos al mes (uno de cada cuatro fallaba) a un 80 %. Es decir, que de los 20 despliegues que se hicieron 16 fueron puestos exitosamente en producción (Figura 5b)

Figura 5b
Tasa de despliegues exitoso



Estos resultados se vieron reflejados en una alta satisfacción del cliente, quienes informaban de manera informal que ya no percibían cuando se desplegaban cambios nuevos y finalmente creció la confianza y la felicidad del personal de desarrollo y de operaciones con las prácticas y tecnologías implementadas, pues veían cómo los frutos de sus desarrollos ya no eran retenidos y no se perdía tiempo en tareas manuales, tiempo que empezó a ser invertido en estudio y capacitaciones en hora laboral enriqueciendo así el salario emocional de todos en la empresa.

Conclusiones y trabajo futuro

De acuerdo con el concepto de adopción de DevOps y la información relevante resumida durante la revisión de la literatura, se identificaron los puntos principales que sirven como pilares en todo proceso de adopción de DevOps. Estos pilares y encuestas determinaron la necesidad, la motivación y el estado actual y nivel de madurez de DevOps.

Los lineamientos propuestos e dividen principalmente en dos grandes grupos: lineamientos culturales (actividades, reuniones, equipos de trabajo, motivaciones, salario emocional, etc.) y lineamientos técnicos (integración continua, compilaciones automatizadas, pruebas automatizadas, revisión de código estático y entrega/despliegue continuos).

La validación del método de adopción de DevOps fue realizada con la ayuda del equipo de TI de la empresa, la cual demostró que con tan solo una pequeña implementación técnica de CI/CD los resultados fueron notorios y se evidenciaron principalmente en el ahorro en tiempo invertido en hacer tareas repetitivas de forma manual.

Después de elegir el objeto de adopción de DevOps y establecer métricas relevantes, la empresa puede continuar con el proceso de adopción de DevOps completo, con la ayuda de los lineamientos de DevOps propuestos y la lista de herramientas de soporte seleccionadas.

Como trabajo futuro se pretende la aplicación de los lineamientos en más empresas, implementación de una cadena de herramientas más robusta y madura y finalmente, la generación sistémica de documentación para asegurar que todos los componentes de la propuesta estén continuamente actualizados y sean consistentes.

Referencias

Ambler, S. W., & Ambler, S. W. (2011). Agile Modeling. In *The Elements of UML™ 2.0 Style*. <https://doi.org/10.1017/cbo9780511817533.018>

Brown, D. (2013). Agile User Experience Design. In *Agile User Experience Design*. <https://doi.org/10.1016/C2011-0-06229-4>

Cesar navia, J. luis J. (2019). Estrategia mejora en el proceso de atracción y mantenimiento de clientes potenciales , mediante el uso de contenidos basados en experiencias de gamificación Cesar Navia Open Systems S . A . S (Colombia) José Luis Jurado Universidad de San Buenaventura (. *Guillermo de Ockham*, 17(1), 1–17.

de Kort, W., & de Kort, W. (2016). What Is DevOps? In *DevOps on the Microsoft Stack*. https://doi.org/10.1007/978-1-4842-1446-6_1

Dyck, A., Penners, R., & Lichter, H. (2015). Towards definitions for release engineering and DevOps. *Proceedings - 3rd International Workshop on Release Engineering, RELENG 2015*. <https://doi.org/10.1109/RELENG.2015.10>

Forsgren, N., & Kersten, M. (2018). DevOps metrics. *Communications of the ACM*. <https://doi.org/10.1145/3159169>

Grass Ramírez, B., Collazos Ordóñez, C., & González González, C. (2017). Propuesta de incorporación de competencias de formación en ingeniería. *Guillermo de Ockham: Revista Científica*, 15(1), 13. <https://doi.org/10.21500/22563202.3188>

HERING, M., DeGrandis, D., & Forsgren, N. (2015). Measure Efficiency Effectiveness, and Culture to Optimize DevOps Transformation. *DevOps Enterprise Forum*. <https://doi.org/10.1017/CBO9781107415324.004>

Hussaini, S. W. (2014). Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*. <https://doi.org/10.1109/ITSC.2014.6957687>

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*. <https://doi.org/10.1145/2962695.2962707>

Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process*. <https://doi.org/10.1002/smr.572>

Laukkarinen, T., Kuusinen, K., & Mikkonen, T. (2018). Regulated software meets DevOps. *Information and Software Technology*, 97(January), 176–178. <https://doi.org/10.1016/j.infsof.2018.01.011>

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>

Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itonen, J., ... Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information*

- and *Software Technology*, 114(June), 217–230. <https://doi.org/10.1016/j.infsof.2019.06.010>
- Mohamed, S. I. (2015). DevOps Shifting Software Engineering Strategy Value Based Perspective. *IOSR Journal of Computer Engineering Ver. IV*. <https://doi.org/10.9790/0661-17245157>
- Mueller, E., Wickett, J., Gaekwad, K., & Karayanev, P. (2010). What Is DevOps? | the agile admin.
- Nicolau de França, B. B., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by hearing multiple voices. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2973839.2973845>
- Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016). DevOps culture and its impact on cloud delivery and software development. *Proceedings - 2016 International Conference on Advances in Computing, Communication and Automation, ICACCA 2016*. <https://doi.org/10.1109/ICACCA.2016.7578902>
- Ramírez, L. A. (1939). La responsabilidad de los intelectuales. *Guillermo de Ockham: Revista Científica*, 3(10), 331–338.
- Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). DevOps for Digital Leaders. In *DevOps for Digital Leaders*. <https://doi.org/10.1007/978-1-4842-1842-6>
- Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-49094-6_44
- Sharma, S. (2017). Scaling DevOps for the Enterprise. In *The DevOps Adoption Playbook*. <https://doi.org/10.1002/9781119310778.ch6>
- The, F., Of, S., In, I., & Software, E. (2014). Introducing DevOps to the traditional enterprise. *InfoQ.Com*.
- Velasquez, N. F., Kim, G., Kersten, N., & Humble, J. (2018). State of DevOps Report 2018. In *Puppetlabs*. [https://doi.org/10.1016/S0022-3913\(12\)00047-9](https://doi.org/10.1016/S0022-3913(12)00047-9)
- Wettinger, J., Breitenbücher, U., Kopp, O., & Leymann, F. (2016). Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel. *Future Generation Computer Systems*, 56, 317–332. <https://doi.org/10.1016/j.future.2015.07.017>
- Xu, M., David, J. M., & Kim, S. H. (2018). The fourth industrial revolution: Opportunities and challenges. *International Journal of Financial Research*. <https://doi.org/10.5430/ijfr.v9n2p90>
- Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and Its Practices. *IEEE Software*. <https://doi.org/10.1109/MS.2016.81>