

Propuesta de una hoja de ruta para la adopción de DevOps como metodología de desarrollo de software.

Proposal of a roadmap for the adoption of DevOps as a software development methodology.

A. Andrade*, E. Cajar*, J. Fernández* J. Olivares* & D. Heredia**

{Antonio.andrade, Emmanuel.cajar, joinner.fernandez, jeisson.olivares} @unisimon.edu.co – {dianahv} @unisimonbolivar.edu.co

*Estudiante de Ingeniería de Sistemas **Profesor investigador del grupo ingebiocaribe

Universidad Simón Bolívar, Barranquilla-Colombia.

Resumen | En el ámbito de las tecnologías de la información se requiere una respuesta cada vez más rápida a las necesidades de los clientes y a la vez un servicio que sea estable, seguro y predecible. Esto establece un compromiso entre cambio y estabilidad. Trabajar en una solución que entregue cambios de forma frecuente (como pueden ser las metodologías ágiles) sin tener en cuenta la fiabilidad en la gestión operacional o la comunicación entre desarrollo y operaciones hace que en muchas ocasiones esa solución no sea suficiente. En este contexto, el presente proyecto busca la implementación de DevOps como una mejora en la forma en que una empresa aumenta el valor para sus clientes, proveedores y asociados.

Palabras clave: *DevOps, Ingeniería de Software, Metodologías de desarrollo de software.*

Abstract | In the field of information technology, an increasingly rapid response to customer needs is required and once a service that is stable, secure and predictable. This establishes a compromise between change and stability. Working on a solution that delivers frequent changes (such as agile methodologies) without taking into account the reliability of operational management or communication between development and operations often means that this solution is not enough. In this context, this project seeks the implementation of DevOps as an improvement in the way a company increases value for its customers, suppliers and partners.

Keywords: *DevOps, Software Engineering, Software Development Methodologies.*

I. INTRODUCCIÓN

Tras la revisión del estado de la industria del software en Colombia se puede evidenciar, que al menos el 80% del sector TI es considerado una MIPYME (Micro, pequeñas y medianas empresas), de las cuales al menos el 50% ofrece productos y/o servicios de desarrollo de software, QA, IaaS y Cloud Computing, se evidencia en esta composición empresarial la falta de inversión en I+D+i, los niveles de productividad y la calidad de los productos de software, lo cual ubica el producto de software colombiano como no competitivo en el mercado.

Entre la lista de las debilidades de la industria se destacan los productos de baja calidad, los bajos niveles de innovación, y la falta de organización empresarial; de esto sobresale que según el 64% de incidentes negativos en la percepción del producto son atribuidos a

la calidad de este. Como una posible solución a todos estos aspectos se plantea generar una estrategia para implementar DevOps en la empresa desarrolladora de software colombiana.

Con base en lo anterior en el presente proyecto de investigación se plantea la siguiente pregunta de investigación: ¿Cómo definir una estrategia para la implementación de DevOps como factor de reducción de errores y tiempos de despliegues en MIPYMES desarrolladoras en el contexto colombiano para aportar a la calidad de productos de software? Para darle respuesta a la anterior pregunta de investigación, se optó por orientar esta investigación al diseño de una estrategia DevOps como el aporte a la calidad de productos de software en MIPYMES colombianas; el cual será medido a través del uso de métricas basadas en calidad de pruebas de software.

II. ESTADO DEL ARTE

Como estado del arte para esta investigación partimos de algunos de los casos de éxito que se han presentado en algunas empresas del sector de desarrollo de software, tales empresas necesitaban realizar cambios en sus metodologías con la finalidad de ampliar sus horizontes y entregar productos no solo más rápido sino con un equipo más coordinado, reduciendo posibles errores y aumentando a su vez la satisfacción de los clientes finales.

Casos de Éxito con DevOps:

➤ Barclays: [1]

- Reto: Seguir el ritmo de las demandas del mercado:
- Barclays estaba sometida a grandes presiones del sector y de la competencia, y necesitaba crear un negocio con mayor capacidad de respuesta.
- Solución: Modernizar la tecnología y la cultura:
Barclays adoptó un enfoque DevOps en el cual los equipos técnicos y comerciales trabajan juntos para satisfacer rápidamente las demandas de los clientes y del mercado a través del desarrollo continuo.
- Resultados: Dar soporte durante el proceso de transición a la nube:
Con el nuevo enfoque DevOps y el entorno aPaaS, los desarrolladores de Barclays pueden trabajar más eficientemente para ofrecer actualizaciones y funciones de forma rápida.

➤ Deutsche Bank: [2]

- El desafío: optimizar la plataforma para lograr el desarrollo eficiente:
Deutsche Bank quería disminuir los ciclos de desarrollo para lanzar productos al mercado en menos tiempo.

- La solución: diseñar una plataforma nueva con tecnología open source:
Deutsche Bank optó por Red Hat para que lo ayudara a diseñar Fabric, una plataforma de desarrollo de aplicaciones basadas en microservicios y organizadas en contenedores.
- Resultados: aumento de la eficiencia y ahorro de tiempo y dinero:

Fabric ofrece un acceso más rápido a los recursos, lo cual permite que los desarrolladores trabajen de forma más eficiente y acelera la comercialización.

➤ GUPO SANTILLANA: [3]

- Reto:
Se buscaba transformar el negocio tradicional en un modelo digital, de manera que permita un crecimiento ágil, innovación continua (métodos ágiles), minimizar riesgos, y mantener una ratio de eficiencia económica muy alto.
- Solución:
Se realizó una aproximación en primera instancia de migración “as-is” hacia una arquitectura Cloud, para posteriormente, y de manera alineada con la implantación de metodologías ágiles, ir transformando los sistemas hacia arquitecturas PaaS y Serverless.
- Resultados:
 - ✓ Reducción de costes continua con un 40% el primer año, y un 15% anual mantenido.
 - ✓ Disponibilidad de los entornos del 100% durante dos años seguidos.
 - ✓ Escalabilidad sin precedentes para el negocio en 22 países diferentes.
 - ✓ Eficiencia y economía de escala en el modelo desarrollado.
 - ✓ Innovación tecnológica y referente en el sector educativo.

III. DEVOPS COMO METODOLOGÍA DE DESARROLLO DE SOFTWARE

A. ¿Qué es DevOps?

DevOps es una combinación de patrones destinados a mejorar la colaboración entre el desarrollo y las operaciones. DevOps aborda objetivos e incentivos compartidos, así como procesos y herramientas compartidas. Porque de los conflictos naturales entre diferentes grupos, metas compartidas y los incentivos no siempre se pueden lograr. Sin embargo, deberían al menos estar alineados entre sí.

Componentes básicos de DevOps

Son una serie de tecnologías o técnicas previas a la implantación de la herramienta DevOps que posibilitan el mejor resultado de dicho modelo.

- Gestión de requisitos. IV.
- Desarrollo basado en Microservicios.
- Capacidades de Testing.
- Automatización de Procesos.
- Infraestructura como Código.
- Cloud Computing.
- Medición Continua.
- Monitorización de Sistemas.
- Herramientas de Comunicación y Colaboración.

B. Ventajas [4]

Uno de los principios del proceso de desarrollo ágil es entregar software en incrementos pequeños y frecuentes en contraposición con el enfoque “big bang” del método waterfall o cascada. El objetivo es tener al final de cada sprint, típicamente cada dos semanas, un código potencialmente entregable.

“los principales motivos estratégicos por los que una organización debe considerar implementar un modelo DevOps son los siguiente:

- Reducción de costes: al implementar DevOps se mejoran los procesos del negocio y se reducen por tanto sus costos
- Mejora de calidad: al estar continuamente generando versiones y probándolas se consigue que el producto final tenga una mejor calidad y se dé un mayor valor al cliente.

- Mejora de la productividad: al implementar ciclos más cortos de desarrollo con sus correspondientes pruebas en el entorno de producción se consigue aumentar la productividad y reducir los tiempos de entrega finales.
- Diferenciación: al implementar DevOps se reducen las ventajas competitivas de los competidores
- Innovación: al implementar DevOps se consigue innovar introduciendo cambios radicales sobre los procesos del negocio, reduciendo los costes y mejorando la calidad, la eficiencia y el servicio al consumidor (menor time to market).

C. Desventajas

Por supuesto, DevOps no es una panacea para desarrollar aplicaciones comerciales. Hay una serie de desventajas que deben tenerse en cuenta:

- V. La subcontratación de una infraestructura de DevOps requiere un nivel específico de experiencia en desarrollo de software, incluido un conocimiento profundo de la integración, la infraestructura y la organización del flujo de trabajo. Necesita expertos junto con las herramientas para que DevOps como servicio tenga éxito.
- VI. La seguridad es siempre una preocupación. El equipo de seguridad generalmente no forma parte de DevOps, y el equipo de DevOps tiende a elegir la velocidad sobre la seguridad al desarrollar software. El uso de servicios en la nube puede crear riesgos y exposición innecesarios, especialmente porque es la capa de transporte la que generalmente no está protegida; Los ciberdelincuentes tienden a utilizar la capa de transporte para suplantación de identidad y ataques de intermediarios.

D. Arquitectura

DevOps se utiliza para aplicaciones alojadas en plataformas en la nube y grandes aplicaciones distribuidas. El desarrollo ágil se utiliza aquí para que la integración y la entrega puedan ser continuas. Cuando el equipo de desarrollo y operaciones trabaja por

separado, el diseño, la prueba y la implementación requieren mucho tiempo. Además, si los equipos no están sincronizados entre sí, puede causar un retraso en la entrega. Por tanto, DevOps permite a los equipos corregir sus deficiencias y aumentar la productividad.

➤ Características de la arquitectura:

• Automatización:

La automatización reduce de manera más efectiva el consumo de tiempo específicamente durante la fase de prueba e implementación. Los aumentos de productividad y los lanzamientos se hacen más rápidos a través de la automatización con menos problemas a medida que las pruebas se ejecutan de manera más rigurosa.

• Colaboración:

El equipo de Desarrollo y Operaciones colabora como equipo de DevOps, lo que mejora el modelo cultural a medida que los equipos se vuelven más efectivos con su productividad, lo que fortalece la responsabilidad y la propiedad.

• Integración:

Las aplicaciones deben integrarse con otros componentes del entorno. La fase de integración es donde el código existente se integra con la nueva funcionalidad y luego se realiza la prueba. La integración y las pruebas continuas permiten un desarrollo continuo.

• Gestión de la configuración:

Esto garantiza que la aplicación solo interactúe con los recursos relacionados con el entorno en el que se ejecuta. Los archivos de configuración se crean donde la configuración externa a la aplicación se separa del código fuente.

E. Herramientas para Devops según sus fases:

A continuación, se presentan algunas de las herramienta más comunes utilizadas por los desarrolladores DevOps según cada una de las fases de la metodología

➤ Code:

✓ Jira: Es una poderosa herramienta de gestión de trabajo. [5]

✓ Git: Es un sistema de control de versiones distribuido gratuito y de código abierto. [6]

➤ Deploy:

✓ Docker: Simplifica y acelera los flujos de trabajo de desarrollo con un canal de desarrollo integrado y mediante la consolidación de los componentes de la aplicación. [7]

✓ AWS: AWS cuenta con una cantidad de servicios y de características incluidas en ellos que supera la de cualquier otro proveedor de la nube. [8]

➤ Release:

✓ CodeShip: Es un ambiente digital IC/EC para programadores en la nube. [9]

✓ Jenkins: Es un servidor de automatización de código abierto autónomo. [10]

➤ Build:

✓ Maven: Maven es una herramienta de gestión y comprensión de proyectos de software. Basado en el concepto de un modelo de objetos de proyecto. [11]

✓ Sbt: Es una herramienta de compilación para Scala y Java. [12]

➤ Operative:

✓ Chef: Se utiliza en la automatización de la infraestructura y ayuda a reducir las tareas manuales y repetitivas para la gestión de la infraestructura.

✓ Ansible: Es una herramienta de gestión, implementación y organización de la configuración de TI de código abierto.

➤ Monitor

✓ Nagios: Nagios es una de las herramientas DevOps más populares en el campo de la monitorización. Esta aplicación Open Source permite monitorizar el estado de cualquier infraestructura, de modo que se pueda corregir cualquier problema.

✓ Splunk: Splunk es un software para buscar, monitorizar y analizar macrodatos generados por máquinas de aplicaciones, sistemas e infraestructura IT a través de una interfaz web.

- Test
 - ✓ Se: Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas
 - ✓ JUnit: JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java
- Plan
 - ✓ Enterprise Architect: es una herramienta de diseño y modelado visual basada en OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelado de procesos comerciales; y modelado de dominios basados en la industria.
 - ✓ Rational Software: Rational Software es actualmente conocida como una familia de software de IBM para el despliegue, diseño, construcción, pruebas y administración de proyectos en el proceso de desarrollo de software.

IV. METODOLOGÍA DE LA INVESTIGACIÓN

Esta investigación fue desarrollada por fases las cuales fueron divididas por tres y estas a su vez en pequeñas tareas, haciendo más sencillo el desarrollo de estas.

La primera fase consistió en Establecer el nivel de uso de DevOps en las empresas de Barranquilla, para conseguir esto desarrollamos un instrumento de recolección de información la cual consistió en una encuesta diseñada en Google forms. Siguiendo a esto, identificamos algunos desarrolladores de software de la ciudad y/o región a los cuales aplicar el instrumento antes mencionado, por último para completar esta fase, se tabularon los datos obtenidos y se estableció el nivel de uso de la metodología.

La segunda fase de la metodología consistió en Identificar los puntos clave de la metodología DevOps, para definir las recomendaciones para su implementación en una hoja de ruta, esta fase fue lograda realizando un estudio profundo de la metodología, con énfasis en las ventajas y desventajas de esta, así como de sus componentes; también definimos recomendaciones para las empresas a partir del estudio previamente realizado.

La tercera fase de esta metodología era uno de los pasos más importantes de la investigación, puesto que consistió en Desarrollar un medio de difusión de la hoja de ruta, para lo cual se decidió crear un blog con WordPress.

V. RESULTADOS

La investigación que realizamos sobre DevOps y los resultados de la misma se presentan en un blog que se encuentra público en la web, el cual ofrece no solo la información recopilada sobre DevOps y todo lo que a este se refiere, sino que además presentamos en el recomendaciones para la implementación de DevOps como metodología de desarrollo de software.

A continuación, se presentan algunas imágenes del resultado final del blog construido:

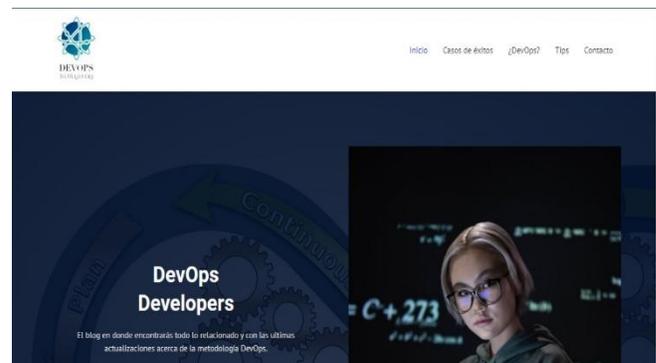


FIGURA 1. PÁGINA PRINCIPAL DEL BLOG DE DEVOPS

La figura 1, muestra la página principal del blog en la cual se presentan las opciones de consulta disponibles en un menú.



FIGURA 2. CASOS DE ÉXITO DE DEVOPS

La figura 2 muestra algunos casos de éxito de empresas que han implementado DevOps como su metodología de desarrollo de software.



FIGURA 3. PRESENTACIÓN DE LOS CONCEPTOS DE DEVOPS

La figura 3 muestra toda la información recopilada durante el proceso de investigación referente a DevOps de manera organizada.



FIGURA 4. PRESENTACIÓN DE LAS RECOMENDACIONES PARA LA ADOPCIÓN DE DEVOPS

La figura 4 muestra tips y/o recomendaciones para adoptar DevOps como metodología de desarrollo de software



FIGURA 5. PÁGINA DE CONTACTO

En la figura 5 se encuentra un formulario de contacto para comunicarse con el equipo de desarrollo del sitio y hacer preguntas, sugerencias o aportes.

VI. CONCLUSIONES

Como resultado de la investigación se presenta un blog como instrumento de difusión de los aspectos más relevantes de DevOps como metodología de desarrollo de software para que las micro, pequeñas y medianas empresas de la ciudad de Barranquilla-Colombia obtengan de manera fácil y rápida la información que necesitan si están pensando en utilizar esta metodología de desarrollo, comparando ventajas y desventajas, así mismo como información sobre las herramientas que se pueden utilizar en ella.

En suma, esta investigación nos permite concluir que DevOps como metodología de desarrollo de software no es solo una buena opción por permitir desarrollo en la nube y la fácil integración del equipo en los proyectos, sino que las relaciones costo-beneficio y el tiempo de entrega de los proyectos son realmente buenos, además DevOps mitiga los errores en el proyecto pues continuamente se generan versiones que a su vez son probadas, esto implica una mejor calidad del producto final.

Referencias

- [1] Red Hat, «Red Hat,» [En línea]. Available: <https://www.redhat.com/es/success-stories/barclays>. [Último acceso: 15 10 2020].
- [2] Red Hat, «Red Hat,» [En línea]. Available: <https://www.redhat.com/es/success-stories/deutsche-bank>. [Último acceso: 12 10 2020].
- [3] everis, «everis,» [En línea]. Available: <https://everis.cloud/caso-de-exito-santillana-negocios-digitales/>. [Último acceso: 12 10 2020].
- [4] A. É. Villamarín, Introducción a DevOps para la mejora de los procesos de desarrollo con herramientas Open Source, España, 2019.
- [5] Atlassian, «Jira Software,» [En línea]. Available: <https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for#Jira-for-requirements-&-test-case-management>. [Último acceso: 20 10 2020].
- [6] Git, «Git,» [En línea]. Available: <https://git-scm.com/>.

[Último acceso: 20 10 2020].

- [7] Docker, «Docker,» [En línea]. Available: <https://www.docker.com/company>. [Último acceso: 20 10 2020].

- [8] Amazon Web Services, «Aws,» [En línea]. Available: https://aws.amazon.com/es/what-is-aws/?nc1=f_cc. [Último acceso: 20 10 2020].

- [9] CouldBees, «CouldBees,» [En línea]. Available: <https://www.cloudbees.com/products/codeship>. [Último acceso: 20 10 2020].

- [10] Jenkins, «Jenkins,» [En línea]. Available: <https://www.jenkins.io/>. [Último acceso: 20 10 2020].

- [11] Apache Software Foundation, «Apache Marven,» [En línea]. Available: <https://maven.apache.org/>. [Último acceso: 20 10 2020].

- [12] Sbt, «Sbt,» [En línea]. Available: <https://www.scala-sbt.org/>. [Último acceso: 20 10 2020].