

Modelo de implementación de una cultura devops en organizaciones

Implementation model of a devops culture in organizations

Fernando Cutire¹, Ana Gloria Cordero^{2}*

¹Licenciatura en Ingeniería de Sistemas de Información, Facultad de Ingeniería de Sistemas Computacionales – Universidad Tecnológica de Panamá, ²Departamento de Computación y Simulación de Sistemas, Facultad de Ingeniería de Sistemas Computacionales– Universidad Tecnológica de Panamá.

(Tamaño de fuente para la afiliación de 10 puntos y en cursiva)

Resumen: Este artículo busca servir de base para ejecutivos, ingenieros y estudiantes que quieran conocer la cultura devops. El objetivo a desarrollar es informar las piezas organizacionales y el mejoramiento de procesos tecnológicos necesarios para implementar esta cultura en empresas o proyectos. También se presentan los 4 pilares básicos de una cultura devops efectiva que son: la colaboración, la afinidad, las herramientas y la escalabilidad.

Palabras clave Devops, git, cultura, docker,

Abstract This article seeks to serve as a basis for executives, engineers and students who want to know the devops culture. The objective to be developed is to inform the organizational pieces and the improvement of technological processes necessary to implement this culture in companies or projects. The 4 basic pillars of an effective devops culture are also presented, which are: collaboration, affinity, tools and scalability.

Palabras clave Devops, git, culture, docker,

* Corresponding author: fernandocutire@gmail.com

1. Introducción

El objetivo de este trabajo es conocer las partes fundamentales y necesarias para poder construir un entorno donde la cultura devops pueda existir, enfocándose en:

- Desarrollo de la cultura devops en la organización
- Adiestramiento en herramientas tecnológicas que facilitan la comunicación entre equipos de desarrollo y de operaciones
- Explicación de lo que significa devops

El propósito del trabajo es entonces identificar los pilares para crear una cultura devops efectiva en las empresas y servir como guía para ese objetivo.

1.1 La comprensión de la cultura devops

Una cultura es una colección abstracta de ideas y comportamientos.

Antes de continuar, debemos de definir qué es DevOps, una explicación sería la proporcionada por Chris Jackson donde menciona que es una cultura de compartir donde los desarrolladores y las operaciones son una unidad, que suben y bajan juntos. Son aplicaciones prácticas de “lean” y ágil. El objetivo de DevOps es ser un habilitador comercial en tiempo real al eliminar el esfuerzo desperdiciado y la burocracia que

se interponen en el camino para abordar mejor las necesidades del cliente. [3]

1.2 El surgimiento de una nueva forma de trabajar

Este concepto ha sido algo que ha ido surgiendo con el tiempo, como lo menciona Mandy Walls:

“Como proceso técnico, han surgido herramientas que permiten a los equipos trabajar de manera más rápida y eficiente. Las herramientas por sí solas no son suficientes para crear el entorno colaborativo al que muchos de nosotros nos referimos ahora como DevOps. Sin embargo, crear un entorno de este tipo es un gran desafío; requiere evaluar y alinear la forma en que las personas piensan sobre sus equipos, el negocio y los clientes.

Armar un nuevo conjunto de herramientas es simple en comparación con el cambio de cultura organizacional. DevOps se aparta de las prácticas tradicionales de desarrollo de software por este motivo. Al celebrar a los desarrolladores superestrellas antisociales, o permitir que el código pase a producción que no se probó adecuadamente, o al decidir que la única razón por la que el código no se ejecuta es porque Operaciones es ineficiente, no estamos haciendo todo lo posible para habilitar el negocio, y crear valor para nuestros clientes.” [4]

1.3 ¿Qué hace a una cultura devops?

Los detalles organizacionales que hacen a una cultura devops según Mandy Walls [4] son los siguientes:

1.3.1 Comunicación Abierta

“Una cultura devops es aquella que es creada mediante grandes discusiones y debates” [4] dice Walls y es que un equipo que toma este enfoque discute requerimientos, funcionalidades, horarios y el cronograma para llegar a mejores caminos en favor del producto.

1.3.2 Alineación de incentivos y responsabilidades

Este punto toca que el equipo es incentivado alrededor de una meta en particular, que es crear un increíble producto para los consumidores. Así que se busca alinear sus responsabilidades a ello y no a que el equipo de desarrollo escriba un montón de código o que el equipo de operaciones no haga muchos despliegues en producción.

1.3.3 El respeto

“Todos los miembros del equipo deben respetar a todos los demás miembros del equipo. En realidad, no es necesario que todos se agraden, pero todos deben reconocer las contribuciones de los demás y tratar bien a los miembros de su equipo.” [4] Esto es lo que comenta Mandy Walls en este punto.

1.3.4 Confianza

“La confianza es un componente fundamental para lograr una cultura DevOps. Las operaciones deben confiar en que Desarrollo está haciendo lo que están haciendo porque es el

mejor plan para el éxito del producto. El desarrollo debe confiar en que el control de calidad no está realmente ahí para sabotear sus éxitos. El Gerente de Producto confía en que Operaciones brindará retroalimentación y métricas objetivas después de la próxima implementación. Si una parte del equipo no confía en otra parte del equipo, sus herramientas no importarán.” [4] Aquí, Walls habla de la confianza entre los equipos.

1.4 Entrega Continua como finalidad de la cultura devops

Cómo se ha ido mencionando en el artículo, también se involucra un cambio en los procesos de desarrollo y operaciones, haciendo que trabajen de forma unida. Esta sección pretende mostrar el proceso tecnológico con respuestas y herramientas en materia de tecnología.

Como mencionan Humble, Kim, Forsgren “la entrega continua es un conjunto de capacidades que nos permiten hacer que los cambios de todo tipo (funciones, cambios de configuración, corrección de errores, experimentos) entren en producción o en manos de los usuarios de forma segura, rápida y sostenible.” [1]

1.5 Prácticas de Entrega continua

En este trabajo, presentamos las siguientes capacidades clave que impulsan la entrega continua:

1.5.1 Control de Versiones

Según la investigación de Humble, Kim, Forsgren. Destacan la importancia de usar un buen sistema de control de versiones para los proyectos. Ellos mencionan: “Lo más interesante fue que mantener la configuración del sistema y la aplicación en el control de versiones estaba más relacionado con el rendimiento de la entrega de software que mantener el código de la aplicación en el control de versiones. La configuración normalmente se considera una preocupación secundaria al código de la aplicación en la gestión de la configuración, pero nuestra investigación muestra que esto es un concepto erróneo.” [1]

Para control de versiones, propongo emplear el sistema de control de versiones Git que provee de una robusta estructura para manejar este tipo de aplicaciones, algunas empresas proveedoras de repositorios para empresas se destacan GitHub y Gitlab que son líderes que proveen a empresas esta infraestructura tecnológica



Figura 1. El logo de la empresa Gitlab proveedora de servicios de infraestructura de versionamiento de control mediante git.

1.5.2 Automatización de Pruebas

Se menciona que “Tener pruebas automatizadas que son confiables: cuando las pruebas automatizadas pasan, los equipos confían en que su software es liberable. Además, confían en que los fallos de las pruebas indican un defecto real. Demasiadas suites de prueba son inestables y poco confiables, lo que produce falsos positivos y negativos; vale la pena invertir un esfuerzo continuo en una suite que sea confiable.” [1] Por lo que proveer de un número adecuado de pruebas es indispensable.

Dentro de automatización se destaca Gradle en el concepto de construcción de aplicaciones Java dotando de dependencias y facilitando la puesta en producción. Y Jenkins para toda clase de automatización de procesos, de pruebas, desarrollo, etc.

1.5.3 Automatización de Pruebas en Datos

Sobre este punto se ilustra que “Al crear pruebas automatizadas, administrar los datos de las pruebas puede resultar complicado. En nuestros datos, los equipos exitosos tenían datos de prueba adecuados para ejecutar sus conjuntos de pruebas completamente automatizados y podían adquirir datos de prueba para ejecutar pruebas automatizadas bajo demanda. Además, los datos de prueba no eran un límite para las pruebas automatizadas que podían ejecutar.” [1].

1.5.4 Desarrollo basado en Trunk (Trunk-Based Dev)

Sobre este modelo de desarrollo, se destaca “Nuestra investigación también encontró que el desarrollo fuera de troncal / maestro en lugar de en ramas de características de larga duración se correlacionó con un mayor rendimiento de entrega. Los equipos que obtuvieron buenos resultados tenían menos de tres sucursales activas en cualquier momento, sus sucursales tenían una vida útil muy corta (menos de un día) antes de fusionarse con el tronco y nunca tuvieron períodos de “congelación de código” o de estabilización. Vale la pena volver a enfatizar que estos resultados son independientes del tamaño del equipo, el tamaño de la organización o la industria.” [1]

1.5.5 Seguridad de la Información

En materia de seguridad, también contribuye cuidarla como dice “Los equipos de alto rendimiento tenían más probabilidades de incorporar la seguridad de la información en el proceso de entrega. Su personal de seguridad proporcionó comentarios en cada paso del ciclo de vida de la entrega de software, desde el diseño hasta las demostraciones y la ayuda con la automatización de pruebas. Sin embargo, lo hicieron de una manera que no ralentizó el proceso de desarrollo, integrando preocupaciones de seguridad en el trabajo diario de los equipos. De hecho, la integración de estas prácticas de seguridad contribuyó al rendimiento de la entrega de software.” [1]

1.5.6 Adoptando Entrega Continua

Este concepto es difícil de adoptar, ya que requiere un cambio estructural y tecnológico, como mencionan “Nuestra investigación muestra que las prácticas técnicas de entrega continua tienen un gran impacto en muchos aspectos de una organización. La entrega continua mejora tanto el rendimiento como la calidad de la entrega, y también ayuda a mejorar la cultura y a reducir el agotamiento y el dolor de la implementación. Sin embargo, implementar estas prácticas a menudo requiere repensar todo, desde cómo trabajan los equipos, cómo interactúan entre sí, hasta qué herramientas y procesos utilizan. También requiere una inversión sustancial en la automatización de pruebas e implementación, combinada con un trabajo incansable para simplificar la arquitectura de los sistemas de forma continua para garantizar que esta automatización no sea prohibitivamente costosa de crear y mantener.” [1]

Algo que agregar en este punto. Es el uso de contenedores Docker. Este software open source ha logrado solucionar problemas de dependencias de desarrollo entre máquinas, eliminando la necesidad de una máquina virtual.

Actualmente Docker es empleado por una gran cantidad de empresas y contribuye enormemente al desarrollo de otras aplicaciones open source.

2. Materiales y métodos

En este trabajo se consultaron una cantidad de fuentes de libros que exploran este concepto nuevo.

La metodología empleada es una investigación descriptiva donde analizamos la cultura y la forma de aplicación organizacional y técnica.

3. Resultados y discusión

En esta sección proponemos como resultados los 4 pilares organizacionales para una cultura devops efectiva. Estas

surgen de haber analizado libros e investigaciones sobre el tema.

Los 4 pilares a mencionar son

3.1 Colaboración

La colaboración es el proceso de construcción hacia un resultado específico a través de interacciones de apoyo y el aporte de varias personas. Un principio rector que dio forma al movimiento devops fue la cooperación de los equipos de desarrollo y operaciones de software.

3.2 Afinidad

Además del crecimiento y mantenimiento de las relaciones de colaboración entre individuos, equipos y departamentos dentro de una organización y en toda la industria en general, es necesario tener relaciones sólidas. La afinidad es el proceso de construir estas relaciones entre equipos, navegar por diferentes objetivos o métricas mientras se tienen en cuenta los objetivos organizacionales compartidos y fomentar la empatía y el aprendizaje entre diferentes grupos de personas.

3.3 Herramientas

Las herramientas son un acelerador que impulsan el cambio en función de la cultura y la dirección actuales. Las opciones de herramientas pueden percibirse como ganancias fáciles. Comprender por qué son beneficiosos y su impacto en las estructuras existentes es importante para evitar problemas ocultos en equipos y organizaciones. No examinar los problemas en valores, normas y estructura organizacional conduce a condiciones invisibles de falla a medida que se acumula la deuda cultural.

3.4 Escalabilidad

El escalamiento es un enfoque en los procesos y pivotes que las organizaciones deben adoptar a lo largo de sus ciclos de vida. Más allá de simplemente considerar lo que significa

abordar devops en grandes organizaciones empresariales, el escalado tiene en cuenta cómo los otros pilares de devops efectivos se pueden aplicar en todas las organizaciones a medida que crecen, maduran e incluso se reducen.

4. Conclusiones

En conjunto, los cuatro pilares de un devops eficaz le permitirán resolver problemas tanto culturales como técnicos que pueden afectar el desarrollo de software. En cada una de las siguientes cuatro partes de este libro, cubrimos estos pilares en profundidad. Se han extraído ejemplos de la industria, que abarcan un conjunto diverso de empresas, desde nuevas empresas web hasta grandes empresas. Si bien no es estrictamente necesario leer los capítulos en orden, se recomienda que eventualmente lea todos los capítulos, ya que es la combinación y armonía de estos cuatro pilares lo que realmente hace que DevOps sea efectivo.

AGRADECIMIENTOS

A la Universidad Tecnológica de Panamá por proveer el documento base que sirvió como plantilla para la investigación.

REFERENCIAS

Citación de un libro

- [1] Forsgren. Nicole, Humble. Jez, Kim. Gene. "Capítulo 4: Prácticas Técnicas" en "Accelerate". Portland, USA: IT Revolution Press (En Inglés), 2018 .
- [2] Davis. Jennifer, Daniels. Ryan, "Los cuatro pilares de los Devops efectivos" en "Effective DevOps". Anderson, Brian. California, USA: O'Reilly Media (En Inglés), 2016.
- [3] Jackson. Chris, Preston. Hank, Wasko. Steve. "Capítulo 2. Nube. Un nuevo modelo de operaciones para TI" en "CCNA Cloud CLDADM 210-455 Official Cert Guide". Taub. Mark. Indiana USA: Cisco Press (En Inglés), 2017.
- [4] Walls. Mandi. "Building a DevOps Culture". California USA: O'Reilly Media (En Inglés), 2013.