DevOps y la gestión de servicios de SI/TI

PID_00270737

Dídac López Viñas

Tiempo mínimo de dedicación recomendado: 3 horas

Universitat Oberta de Catalunya

Dídac López Viñas

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por la profesora: Isabel Guitart Hormigo (2020)

Primera edición: febrero 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Índice

Introducción

Cuando trabajamos la gestión de servicios de SI/TI en las organizaciones, siempre hemos dado por hecho un proceso de construcción y mantenimiento fundamentado en estructuras organizativas clásicas.

En estas estructuras, tenemos un equipo de desarrollo o de gestión de proyectos, organizado alrededor del concepto de oficina de proyectos o en formas parecidas, que desarrolla o incorpora productos a medida o de mercado convirtiéndolos en servicios. El equipo de desarrollo es el responsable de crear servicios y hacerlos evolucionar.

Por otro lado, tenemos un equipo, tradicionalmente llamado sistemas, que recoge estos productos, los pone en producción, asegura su disponibilidad y continuidad, cuida de su seguridad integral y atiende las incidencias y peticiones de los usuarios.

Figura 1. Puesta en producción y servicio de un sistema de información

Ejemplo

Una organización necesita una aplicación de gestión nueva, como podría ser un módulo de gestión de presencia para recursos humanos. Esta petición de la nueva aplicación se trata como un servicio nuevo. El equipo de desarrollo la construirá con metodologías tradicionales, y al finalizarla, después de las pruebas y tests correspondientes, la entregará al equipo de producción para que la ponga en servicio y active los procesos de atención al usuario que correspondan.

Este paradigma es el que hemos dado por correcto desde hace mucho tiempo en los departamentos de SI/TI y, de hecho, la gran mayoría de los modelos de organización y marcos de referencia se han construido a su alrededor. Es así como lo hemos estudiado y trabajado con los modelos más conocidos y ampliamente implantados como ITIL, por lo menos hasta la versión 3, y la ISO 20000.

ISO 20000

La ISO 20000 es un estándar de calidad para certificar un sistema de gestión de servicios de TIC.

El modelo DevOps supone, de entrada, la separación de los equipos de trabajo que construyen servicios (*building*) de los que los ponen en producción (*running*) y, con esto, la adopción de un modelo de calidad que precisa procesos cerrados con ciclos largos para la implantación de nuevos servicios y la modificación de los existentes. Aunque el resultado es y ha sido bueno, las exigencias de las organizaciones, en especial las que están fuertemente digitalizadas y que compiten proponiendo de manera continua innovaciones y cambios evolutivos, piden nuevos modelos de gestión.

Por otro lado, ya hace bastantes años que los modelos de desarrollo ágil forman parte de la realidad de los departamentos de SI/TI, y con ellos la capacidad de disponer de las soluciones de nuevos productos y de su evolución constante.

Este cambio, impulsado por el manifiesto ágil, supone la adopción de doce principios en las tareas de desarrollo:

Nota

Manifiesto ágil
https://agilemanifesto.org/

1) Nuestra principal prioridad es satisfacer el cliente mediante la entrega temprana y continua de software que aporte valor.

2) Requisitos cambiantes: aceptamos, de buen grado, cambios en los requisitos, incluso si llegan hacia el final del desarrollo. Los procesos ágiles aprovechan el cambio para dar una ventaja competitiva al cliente.

3) Entrega frecuente y funcional: entregamos con frecuencia software que funcione, desde un par de semanas hasta un par de meses, con preferencia por la escala de tiempo más corta.

4) Proximidad del cliente: la gente de negocio y los desarrolladores tienen que trabajar juntos de manera cotidiana durante todo el proyecto.

5) Equipos motivados: construimos proyectos con la ayuda de individuos motivados. Les damos el entorno y el apoyo que necesitan y confiamos en ellos para hacer el trabajo.

6) Conversaciones e interacciones: el método más eficiente y efectivo de comunicar información hacia y dentro de un equipo de desarrollo es la conversación cara a cara.

7) Software en funcionamiento como principal indicador: el software que funciona es la principal medida de progreso.

8) Desarrollo sostenible a ritmo constante: los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios tienen que ser capaces de mantener un ritmo constante de manera indefinida.

9) Excelencia técnica: la atención continua a la excelencia técnica y al buen diseño mejora la agilidad.

10) Simplicidad, minimalismo: la simplicidad, el arte de maximizar la cantidad de trabajo que no se hace, es esencial.

11) Equipos autoorganizados: las mejores arquitecturas, requisitos y diseños surgen de equipos autoorganizados.

12) Reflexión y mejora continua: en intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo, se afina y se ajusta su comportamiento de acuerdo con esto.

Como consecuencia, el modelo de gestión de servicios también pide un cambio, tanto en su gestión ágil –el llamado Agile ITSM (IT Service Management)– como en la integración y la colaboración constante entre los equipos de desarrollo y de sistemas.

En este contexto nace el concepto de DevOps (desarrollo-operaciones). En 2007, Patrick Dubois, de Bélgica, era un consultor que ejercía diferentes roles en el área de SI/TI en varias organizaciones, se dio cuenta del contraste entre las áreas de TI, entre algunos el conjunto de problemáticas entre las áreas de Desarrollo y Operaciones, las dificultades y el tiempo requerido que Operaciones, de acuerdo con los modelos y marcos de referencia clásicos, exigía a Desarrollo por el impacto en los servicios SI/TI, a los clientes y, finalmente, para el negocio.

Es así que, en la conferencia «Agile 2008» de Toronto, aparece por primera vez la palabra DevOps, que rápidamente se populariza, primero, en entornos de desarrollo ágiles y, a continuación, como una herramienta de cambio de los departamentos de SI/TI, con la necesidad de transformar digitalmente las organizaciones.

A partir de este momento, se inicia una serie de eventos que dan lugar al movimiento DevOps, de ámbito internacional, en el que practicantes con experiencia y problemáticas parecidas se unen para definir un conjunto de herramientas que buscan mejorar la situación entre desarrollo y operaciones.

En la actualidad, el concepto DevOps ya está consolidado, pero sigue sin tener un organismo oficial que lo regule, a pesar de que hay organizaciones como DevOps Institute, que patrocina un marco y lo certifica desde el punto de vista del conocimiento profesional, o DASA Forerunner, que también ha creado un esquema parecido.

1. Visión general de DevOps

DevOps es un sistema de gestión ágil (políticas, procesos, personas, herramientas) que capacita a la organización para producir, entregar y operar con software de calidad rápidamente mejorando la comunicación, la colaboración y la integración entre desarrollo, operaciones y gestión de la calidad (*Quality Assurance*, QA), automatizando la entrega y el despliegue, e institucionalizando el aprendizaje y la mejora continua.

DevOps es un compromiso con la agilidad, la repetitividad, la calidad y el gobierno en la entrega de aplicaciones mediante la colaboración entre los equipos responsables del proceso global.

Otro modo de describir DevOps es como una relación más colaborativa y productiva entre los equipos de desarrollo y operaciones. Esta relación mejorada y el aumento en la eficiencia en la colaboración reducen el riesgo de producción asociado con los cambios o las entregas frecuentes de desarrollo.

El concepto DevOps propone difuminar la línea que divide el equipo de desarrollo y operaciones, utilizando nuevas metodologías de desarrollo (como el desarrollo ágil) para adoptar un marco de colaboración donde realizar de manera integrada las tareas de desarrollo, operaciones, control de calidad e implementación.

De hecho, no hay una definición estándar de DevOps, que podemos entender como un movimiento, una tendencia, un conjunto de valores, que poco a poco se está definiendo y estructurando como un concepto consolidado.

No obstante, sí podemos decir que **no** se considera DevOps:

- Un modelo de referencia ni una metodología, a pesar de que distintos consultores y organizaciones lo quieren definir.

- Una cultura de gestión de servicios, a pesar de que implica un cambio cultural en más ámbitos, además del tecnológico, en las organizaciones que lo van adoptando.

- Una arquitectura de sistemas y desarrollo, necesariamente aplicados sobre estrategias de producción en la nube, a pesar de que los modelos que ayudan a conseguir los máximos niveles de agilidad lo pueden necesitar.

- Un paradigma aplicable solo a entornos de entrega e integración continuada, pero está presente en las organizaciones donde existe esta necesidad y por eso lo adoptaron de manera inicial.

- Un modelo de automatización de las tareas de operación, a pesar de que es imprescindible para aplicar modelos de gestión ágil de servicios de SI/TI.

1.1. Adoptar DevOps

La adopción de DevOps está siendo impulsada por factores como:

- el uso de modelos de desarrollo ágiles;

- el incremento de las versiones de producción por parte de las unidades interesadas de aplicación y de negocio, la entrega continuada;

- la amplia disponibilidad de virtualización e infraestructura en nube de proveedores internos y externos;

- el aumento de la automatización de las tareas de gestión de operaciones.

Para poder adoptar DevOps, es necesario un cambio cultural, nuevos conceptos en la gestión de servicios, y la entrega y la integración continua como un proceso básico de gestión y relación entre los equipos implicados.

Algunas claves necesarias que debemos tener en cuenta en la adopción de DevOps son:

- Difuminar la distinción entre desarrolladores y administradores de sistemas. Los profesionales tienen que tener diversidad de conocimientos y entender la arquitectura en su conjunto, así como la operatividad de los servicios con las mejores condiciones de seguridad y calidad, con independencia de que sus funciones principales o sus roles ocasionales pidan un alto grado de especialización.

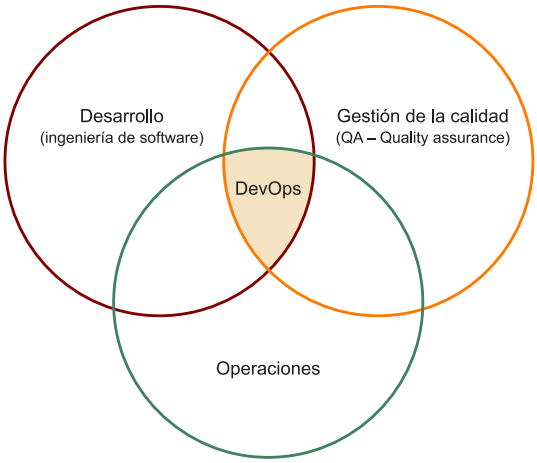
- El procedimiento de pasar de desarrollo a producción tiene que ser completamente seguro. Tiene que haber un procedimiento operativo que esté automatizado y sea 100 % seguro, con un control de todos los entornos que formen parte de él (desarrollo, test y producción). Por ejemplo, que tenga en cuenta las librerías que hay en la máquina del desarrollador, pero que no están presentes en la de producción. La gestión de la calidad en la creación, la gestión, la evolución y el mantenimiento de los servicios es un concepto integral en todos los ámbitos de su ciclo de vida.

- El código debe escribirse libre de errores. El coste económico de arreglar un error crece de manera exponencial a medida que el programa avanza

en la cadena de producción hasta el usuario final. Del mismo modo que el desarrollador tiene que asumir funciones de administrador de sistemas, también tiene que ser su propio *tester*, porque con DevOps no hay tiempo de testear cada pequeño cambio en todo el ciclo de pruebas antes de subirlo a producción. Con DevOps se deben seguir disciplinas de diseño, desarrollo y pruebas durante todo el proceso.

- Empezar por el producto viable más pequeño. Es mejor construir los servicios a partir de una versión simple y funcional que dé el servicio mínimo. De hecho, la estrategia es empezar por un núcleo mínimo de servicio para el cliente o el usuario, y establecer una estrategia de iteraciones continuas en la creación y la puesta en servicio de nuevas funcionalidades.

Figura 2. Integración de las tareas de desarrollo, operaciones y calidad de servicios



Fuente: <https://commons.wikimedia.org/wiki/File:Devops.svg#/media/Archivo:Devops.svg>.

- El diseño tiene que estar pensado para poder crecer de manera incremental. Como el producto empieza siendo muy pequeño, se tendrá que ampliar muchas veces y de manera continua. El crecimiento tiene que estar bien planificado, para evitar que el código se convierta en algo ingestionable. Si bien la codificación inicial tiene que ser la mínima posible, el diseño tiene que ser muy ambicioso y, en particular, lo menos acoplado posible, de modo que se puedan introducir cambios en un subsistema sin afectar a otros. Esto implica tomar una gran cantidad de decisiones de diseño estratégicas *a priori,* eligiendo entre simplicidad frente a potencia.

- El sistema se tiene que poder auditar y corregir fácilmente.

- El sistema tiene que tener resiliencia para poder funcionar, a pesar de que se degraden las condiciones del entorno.

- Definir una política de contención de gestión de riesgos. Antes de pasarlo a producción, se tiene que determinar qué es lo peor que podría suceder,

si el nuevo cambio rompe la versión funcional final, y qué impacto puede tener sobre los servicios y los sistemas de SI/TI.

- La organización tiene que aceptar que el software es una cosa viva. En un futuro, en el software se añadirán nuevas funcionalidades y, por el contrario, alguna otra se modificará o desaparecerá.

- Definir un sistema de documentación continua y gestión de las fuentes. A la hora de añadir de manera continuada nuevas funcionalidades al software, va evolucionando hasta que llega un punto en el que nadie sabe realmente cómo se comporta o qué es lo que puede hacer en respuesta a determinados «estímulos». Para aliviar este problema, en paralelo al desarrollo, se tiene que crear un sistema de documentación continua que explique rápidamente la historia de los cambios. Nos será útil disponer de un estándar de informe diario que explique quién hizo cada cambio, cómo y por qué. La documentación tiene que ser fácil de redactar y de explotar. De nada sirve escribir cosas que nadie sabe dónde están o que nadie lee, o que, aunque se lean, no se entienden.

Como vemos, DevOps presenta un paradigma de gestión de los servicios de SI/TI muy distinto a los tradicionales procesos ITIL o ISO 20000, pero esto no significa que no puedan ser compatibles, sino que los tenemos que implementar de manera distinta y adaptarlos a una nueva realidad. Los servicios están de manera ágil e iterativa, en crecimiento constante, con una entrega continua de nuevas prestaciones sin que este cambio incesante suponga la pérdida de control, calidad y seguridad. DevOps va más allá de la gestión ágil de proyectos y supone la gestión ágil de servicios con mecanismos de comunicación y colaboración entre todos los implicados, particularmente, con los equipos de desarrollo y operaciones.

Aunque los fabricantes, los consultores y los vendedores han buscado etiquetas DevOps para sus ofertas y sus productos, realmente no existe ningún producto DevOps como tal. Solo hay software y herramientas que ayudarán al equipo de desarrollo y operaciones a trabajar conjuntamente de manera más eficiente para responder a los cambios de necesidades de manera más flexible.

2. DevOps y modelos de gestión de servicios de SI/TI

Como hemos visto, DevOps es más que una filosofía, un paradigma o un movimiento que transforma la manera como se organiza el departamento y que tiene un impacto sobre su cultura de funcionamiento y, con toda probabilidad, sobre los procesos de gestión que tenga implantados.

En relación con los modelos de gestión y certificación más conocidos, podemos valorar el impacto de DevOps según:

- la arquitectura empresarial (TOGAF). No implica un cambio grande. Facilita la adopción de DevOps a la hora de ofrecer un orden y una organización de los puntos de mejora, la eficiencia y la eficacia, y provee elementos clave respecto a los procesos que implanta, a la gestión de datos y a la información, y a la aplicación de la tecnología en general;

- el desarrollo de software (Scrum). Implica el uso de métodos de colaboración entre los miembros de los equipos, en las tareas programadas a corto plazo, que están mejor alineadas respecto a los requerimientos, y una fuerte integración en todas las fases, que permite hacer entregas continuas de manera incremental con calidad;

- la gestión de servicios SI/TI (ITIL, ISO 20000). El enfoque clásico de ITIL hasta la versión 3, ITIL v4, ya contempla los modelos de gestión ágiles y conlleva la reinterpretación de los procesos que hasta ahora hemos conocido. A pesar de que se tienen que reformular, continúan siendo útiles los procesos de gestión del cambio, configuración, entregas, transición de nuevos servicios y servicios modificados, entre otros, que sin renunciar a un modelo de gestión de servicios completo y de calidad no implique renunciar a un modelo ágil.

2.1. Ejemplo gestión de servicios SI/TI

Queremos añadir una nueva funcionalidad al sistema de información que permita hacer pedidos internos de material de oficina a los almacenes centralizados de la empresa. Lo presentaremos como un servicio que ofrece el departamento de SI/ TI basado en DevOps.

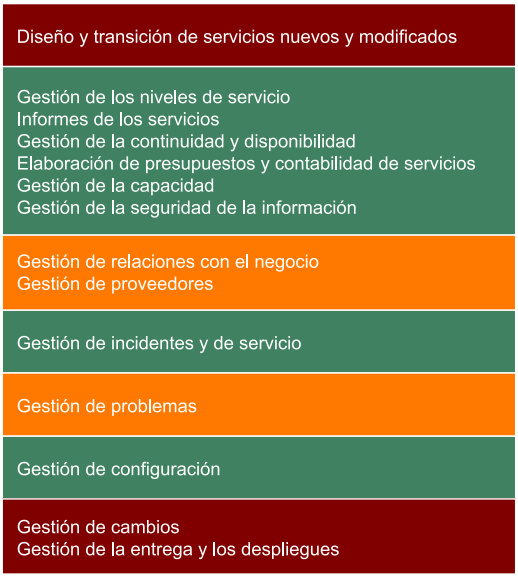
El departamento de SI/TI desarrolló el código inicial y lo subió a producción como un servicio que solo permite hacer pedidos de material que son enviados a un solo lugar de destino. La previsión era ir añadiéndole nuevas funcionalidades, como la actual, que es el envío de lotes del pedido a distintos destinos.

Aunque se desarrolle de manera ágil y se pase a producción, desde un enfoque clásico de gestión de servicios, lo trataríamos como una transición de servicios con la documentación del nuevo código general, la modificación de la base de datos de configuración, la activación de una gestión de cambios, el aseguramiento de la capacidad y la calidad, la activación del proceso de gestión de la entrega, la modificación del catálogo de servicios, etc. Si queremos orientarlo a DevOps sin perder los modelos de gestión de la calidad de servicios y obtener el máximo beneficio de los modelos ágiles, se deben modificar estos servicios y automatizar su seguimiento.

Si lo aplicamos al ejemplo que hemos mencionado, lo que correspondería sería disponer de una herramienta de gestión de versiones de código. Con esta herramienta tendríamos que implementar procesos para versionar el código creado y realizar el seguimiento de los test correspondientes para asegurar la calidad. Igualmente, con estas herramientas tenemos que mecanizar la comprobación de la capacidad de nuestros sistemas y hacer una gestión de los cambios similar a la que se hace en modelos tradicionales. Estos, al estar automatizados, los consideraríamos como los cambios preaprobados de modelos tradicionales (al ser un sistema automatizado, no hace falta un comité cambios porque en este modelo no tiene sentido). Para asegurar la coherencia del modelo de gestión de servicios hace falta que, ya en fase de diseño de la nueva funcionalidad de un servicio, esta se documente como parte de la información que forma parte de su entrada en el catálogo. De este modo se justifica la necesidad no solo de replanificar el Departamento y de hacerlo con modelos ágiles, sino también de automatizar al máximo todos los procesos para que estos se puedan llevar a cabo dentro de un sistema de gestión de servicios con criterios de calidad.

Si tomamos la ISO 20000 como modelo de gestión de la calidad de un sistema de gestión de servicios de SI/TI, normalmente implementado con procesos definidos con ITIL, veremos que hay procesos que tienen más impacto y que tendrán que volverse a describir de manera total o parcial para certificarlos, cuando una organización decida adoptar DevOps.

Tabla 1. Impacto de los procesos del sistema de gestión de servicios de SI/TI en la adopción de DevOps



Algunos procesos probablemente no recibirán ninguna modificación, o será mínima, puesto que no les afecta el seguimiento del modelo, pero esto no significa que no se puedan mejorar aprovechando sus ventajas (véase, de la tabla 1, las filas que están marcadas en verde). Hay que comentar que, en el caso de la gestión de la configuración, ya se da por hecho que se utilizan herramientas como las de descubrimiento –que permiten automatizar el seguimiento–, y que la adopción de DevOps no debería implicar un cambio significativo.

Hay otros, como es el caso de relaciones con el negocio, que habría que revisarlos, ya que la adopción de modelos ágiles suele ser una decisión de la dirección y tiene impacto sobre toda la organización y su estrategia (véase, de la tabla, 1 las filas marcadas en naranja). La relación con los proveedores puede depender del nivel de subcontratación de las tareas de SI/TI y se tendrán que alinear con el nuevo modelo de gestión interna. La gestión de problemas merece un comentario aparte. Si hasta ahora era un proceso integrado dentro de las funciones de operaciones, ahora, por el hecho de compartirlo también con el equipo de desarrollo, seguramente habrá que redefinirlo.

Los que sí se tienen que redefinir de manera completa, como vemos en el ejemplo, son los correspondientes a la transición de nuevos servicios y de servicios modificados, la gestión del cambio y la de entrega y despliegue (véase, de la tabla 1, las filas marcadas en rojo). De hecho, la gestión del cambio es el proceso clave a definir o modificar para conseguir el éxito en la adopción de DevOps.

DevOps e ITIL, a pesar de que parten de filosofías contrapuestas, se pueden compatibilizar, pero hay que hacer una adaptación que dependerá de cómo se ha organizado cada departamento y de cuáles son sus prioridades.

Como conclusión, las organizaciones que han adoptado DevOps mejoran significativamente la calidad en la gestión de sus servicios cuando implementan procesos ITIL/ISO 20000 sin mucho esfuerzo, gracias a los procedimientos y a la automatización implementados. En sí mismo no implica un gran cambio cultural. En cambio, las organizaciones que gestionan sus servicios con un modelo ITIL/ISO 20000 de acuerdo con un modelo organizativo clásico tienen que hacer un esfuerzo superior de cambio cultural.

3. Adopción de DevOps

Algunas organizaciones, sobre todo las nacidas como negocios nativos digitales, han creado y organizado sus departamentos de SI/TI de manera nativa en DevOps, mientras que la gran mayoría lo han hecho desde planteamientos organizativos clásicos que les han supuesto un cambio cultural interno en muchos ámbitos.

Las primeras organizaciones en realizar esta transformación también lo han hecho por la necesidad de orientarse a un mercado digital, con cambios y novedades sucesivas de los modelos de negocio. Sin embargo, las organizaciones que no tienen esta necesidad de manera tan intensiva lo están considerando con el objetivo de optimizar sus recursos, flexibilizarlos y mejorar la calidad de sus servicios.

3.1. Desmentir mitos

En la adopción de DevOps, hay que desmentir algunos mitos:

- Solo es para organizaciones con modelos de negocio digital. No es cierto, a pesar de que por el hecho de ser de nueva creación, muchas crearon sus departamentos de SI/TI directamente con DevOps, orientándose al desarrollo de servicios de manera iterativa con entregas continuas. Algunos ejemplos son las compañías creadas en el entorno digital, como Amazon, Netflix y Flickr, u otras que han orientado su negocio de gestión clásica en los entornos digitales, por ejemplo, la mayoría de los bancos lo han hecho o están en proceso de transformación.

- El personal de los equipos de operaciones tiene que programar. Ni tradicionalmente ni en DevOps los equipos de operaciones tienen que crear código como tal y, en realidad, DevOps tampoco implica que el personal de desarrollo tenga que realizar tareas de operaciones, aunque aproxima sus funciones y genera algunas de mixtas que comparten.

- Las organizaciones sometidas a regulaciones y certificaciones no pueden adoptar DevOps. No solo no es cierto y lo hemos mencionado en las organizaciones que han apostado por ITIL o ISO 20000, sino que tenemos el ejemplo de los entornos de banca y seguros, sometidos a regulaciones y auditorías, que necesitan adaptarse a un mercado muy competitivo que ahora se desarrolla de manera digital, con constantes innovaciones y cambios. Es necesario, eso sí, adaptar los procesos que tradicionalmente se han seguido y aprovechar la disponibilidad de herramientas y oportunidades tecnológicas existentes. De hecho, se considera que DevOps no solo apor-

ta el concepto de agilidad, sino una mejora de los niveles de calidad y certificación.

- DevOps no aplica en la externalización del desarrollo. Es cierto que presenta un reto con dificultades añadidas a lo que supone la contratación en sí, pero se puede hacer si se reinterpreta el rol de proveedor externo y se asegura que este se adapta a los procedimientos. Aquí es clave la disponibilidad de herramientas de automatización, de gestión de versiones, de entrega y de documentación.

- No se puede adoptar DevOps sin la orientación en la nube y/o tecnologías de virtualización. Se puede hacer, pero las facilidades que ofrece la nube combinadas con tecnologías de virtualización permiten optimizar todo el ciclo de la entrega continuada por lo que, en la actualidad, es difícil separar estos conceptos.

- DevOps no se puede aplicar a sistemas grandes y complejos. Es uno de los argumentos negativos que se dan en las organizaciones con una estructura tradicional. Lo cierto es que la gestión del cambio es más difícil, pero no solo se pueden aplicar modelos evolutivos ágiles a partir de un determinado punto, sino que dan buenos resultados. Debemos comentar que algunas organizaciones aplican modelos mixtos que separan la gestión de sus servicios o crean servicios de manera tradicional y, posteriormente, hacen la evolución con modelos ágiles y en entornos de tipo DevOps.

- DevOps solo se aplica en entornos de entrega continuada. No es cierto, aunque sí que es verdad que los primeros que lo hicieron son los que tenían esta necesidad.

- DevOps solo implica los equipos de desarrollo y operaciones. Lejos de ser verdad, es más bien lo contrario. La aplicación de un modelo ágil de gestión de servicios viene dado por la necesidad de digitalizar de manera rápida procesos de negocio. Por lo tanto, la dirección de la organización tiene que ser la impulsora y, poco o mucho, tiene que impactar en toda la organización.

3.2. Aspectos a tener en cuenta

En la adopción de DevOps es importante la implicación de la dirección de la organización:

- DevOps es un movimiento cultural y profesional.

- Fomenta y mejora la comunicación interna y la transparencia entre equipos.

- Crea una cultura orientada al cambio.
- La adopción de DevOps no es un proyecto únicamente de automatización.
- Busca y aprovecha la integración de marcos de referencia para la gestión de servicios de SI/TI para obtener mejores resultados.
- La complicidad entre todos los implicados permite conseguir sinergias de mejora continua.
- Es necesaria la formación previa del personal.
- Es mejor empezar en pequeño y avanzar de manera incremental.
- Se tiene que adoptar una actitud crítica, orientada a la solución.

Una organización con un departamento de SI/TI tradicional que quiere plantear la adopción de DevOps debe tener en cuenta:

- No existe un DevOps estándar. Cada organización lo tiene que interpretar en su contexto.
- Aceptar el punto de partida en el que se encuentra la organización y, en particular, el departamento de SI/TI. Hay que evaluar no solo cuáles son los procesos actuales en la creación y gestión de los servicios, sino también la arquitectura tecnológica y las herramientas que se utilizan. También es importante hacer un análisis de la cultura en la organización, en aspectos clave como la comunicación y la colaboración, particularmente entre los equipos de desarrollo y de operaciones.
- Tener claro cuáles son los objetivos y las expectativas de la organización. Es un proyecto que redefine la estrategia del departamento, pero lo tiene que hacer a partir de la estrategia empresarial. DevOps no es un objetivo en sí mismo, ni siquiera para el departamento de SI/TI. Es un mecanismo para lograr objetivos empresariales.
- Hacer un plan de formación. Afrontar el proyecto de transformación del departamento como un proyecto de cambio cultural, centrado en las personas involucradas y fundamentado en nuevos valores que están basados en la colaboración, el compartimiento de información, la automatización y el concepto de agilidad.
- No se trata de crear un equipo nuevo. Aunque existen organizaciones que mantienen una doble organización, una parte de manera tradicional y otra que funciona con DevOps, no es lo habitual, y solo sucede en grandes organizaciones que pueden separar servicios dentro de su catálogo. Suelen

ser organizaciones que han apostado por modelos como el Bimodal que propone Gartner.

- Acelerar el cambio cultural. Utilizando herramientas que permitan la colaboración, el compartimiento de información y la automatización, es necesario definir cuál será el entorno de compartimiento, colaboración y automatización.

- Definir nuevos procedimientos y flujos de trabajo. Si la organización previamente ha definido procesos, de tipo ITIL, habrá que hacer la modificación al respecto para adaptarlos al nuevo modelo.

- Cambio cultural sobre personas. Hay que determinar cuáles serán los líderes de este cambio e identificar las personas que pueden ser resistentes. ¡Cuidado! Es posible que haya conflictos y, de hecho, es habitual que se produzcan.

Un modelo de cambio que ha dado buenos resultados en muchas organizaciones es empezar por proyectos de nuevos productos, creando equipos modulares para cada uno con el personal tanto de desarrollo como de operaciones. Con estos proyectos, se va aprendiendo como grupo, se ajustan las herramientas elegidas y se generan las nuevas dinámicas. Seguidamente, hay que extender el modelo de trabajo a la gestión del resto de servicios. Suelen ser organizaciones que ya han adoptado modelos ágiles de tipo Scrum a sus equipos de desarrollo y que lo viven como una extensión del cambio cultural ya iniciado.

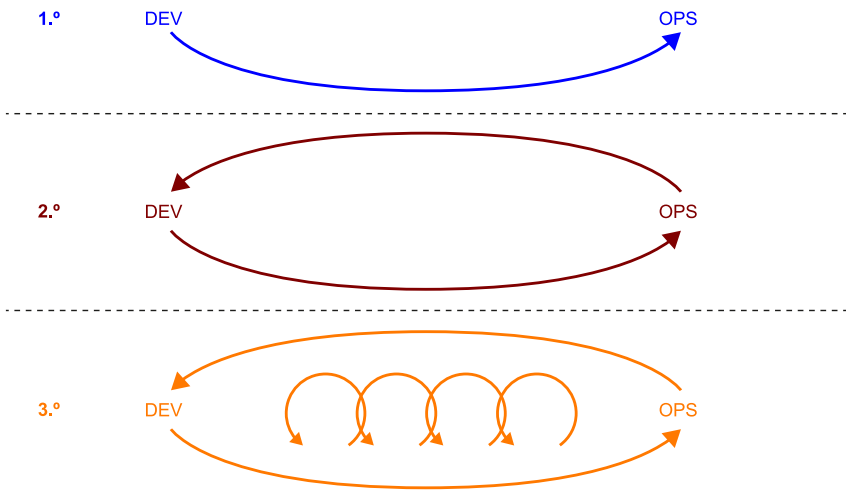
Otro modelo es el denominado incremental, que implica los dos equipos y se aplica en tres fases:

a) Iniciar con el equipo de desarrollo, para que trabajen pensando en el modelo de operaciones existente.

b) Retroalimentación entre los equipos de desarrollo y operaciones, aproximando ambos equipos uno al trabajo del otro.

c) Apoyar la dinámica de retroalimentación con procesos iterativos de colaboración y mejora continua.

Figura 3. Modelo incremental



Fuente: CertiProf.

CertiProf

https://www.certiprof.com/

4. Modelo CALMS

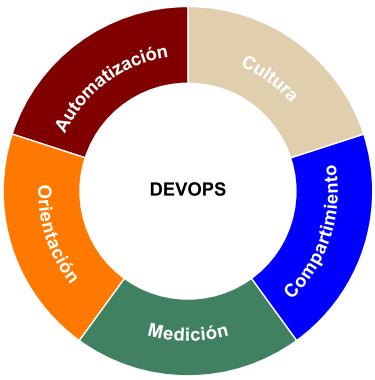
El hecho de que no exista un modelo de referencia formal correspondiente a DevOps hace que, a veces, sea complicado para una persona no iniciada comprender cuál es el objetivo de la transformación del departamento.

No hay un modelo de referencia formal correspondiente a DevOps, pero sí que se han definido algunos marcos de trabajo para su adopción, como por ejemplo DevOps Implementation Framework, DevOps Roadmap, DevOps Journey y DevOps Process, y también muchas de las grandes empresas del mundo de la consultoría han creado su propio modelo para ayudar a los departamentos de SI/TI. Entre los marcos más conocidos existe CALMS.

CALMS es el resultado del acrónimo de los conceptos clave que forman parte de DevOps (véase figura 4):

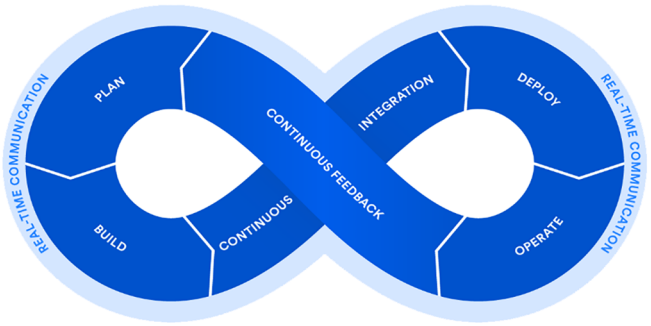
- *Culture*: cultura.
- *Automation*: automatización en todo el ciclo y el uso de arquitecturas y herramientas que lo facilitan.
- *Lean*: orientación al cliente, eliminando derroches. Simplicidad.
- *Measurement*: medición vinculada a la mejora continuada.
- *Share*: compartimiento y transparencia con herramientas que permiten este trabajo en equipo.

Figura 4. CALMS



El objetivo de CALMS es iniciar, mantener y mejorar el ciclo de vida de gestión de servicios que implica DevOps. Este ciclo de vida lo podemos dibujar como la superposición combinada de dos ciclos, el correspondiente al desarrollo y a operaciones, con la combinación cruzada de los procesos de integración continuada y feedback continuado del funcionamiento del producto resultante. En la figura 5 tenemos una representación gráfica muy habitual.

Figura 5. Ciclo DevOps



Fuente: Atlassian.

Atlassian

https://www.atlassian.com/es/devops

CALMS es un marco de referencia conceptual para la integración de equipos, funciones y sistemas de DevOps dentro de una organización.

CALMS se utiliza como un modelo de madurez que ayuda a los directivos comprometidos a evaluar en qué grado su organización está preparada para DevOps y, si no lo está, qué se tiene que cambiar y, en todo caso y en todo momento, qué habría que mejorar como parte de un ciclo de mejora continua.

El *framework* que nos presenta CALMS puede llegar a considerarse como una alternativa a los modelos clásicos de gestión de servicios de SI/TI, ya que presenta un enfoque estratégico para diseñar, proporcionar, gestionar y mejorar la manera de utilizar las TI en una organización. En realidad, CALMS nos ofrece un mecanismo para aproximar las diferencias entre ambos enfoques y hacerlos compatibles. De hecho, el modelo CALMS podría aplicarse, en mayor o menor medida, en otras áreas funcionales de la organización, como son los sistemas de gestión de servicios fundamentados en ITIL.

Una de las dificultades puede venir dada en la selección e implementación de las herramientas que nos tienen que facilitar la adopción de DevOps, ya sea o no, en el contexto de CALMS. Existen diferentes opciones y cada organización tiene que considerarlas de acuerdo con sus puntos de partida y sus objetivos. Algunos ejemplos que tenemos pueden ser:

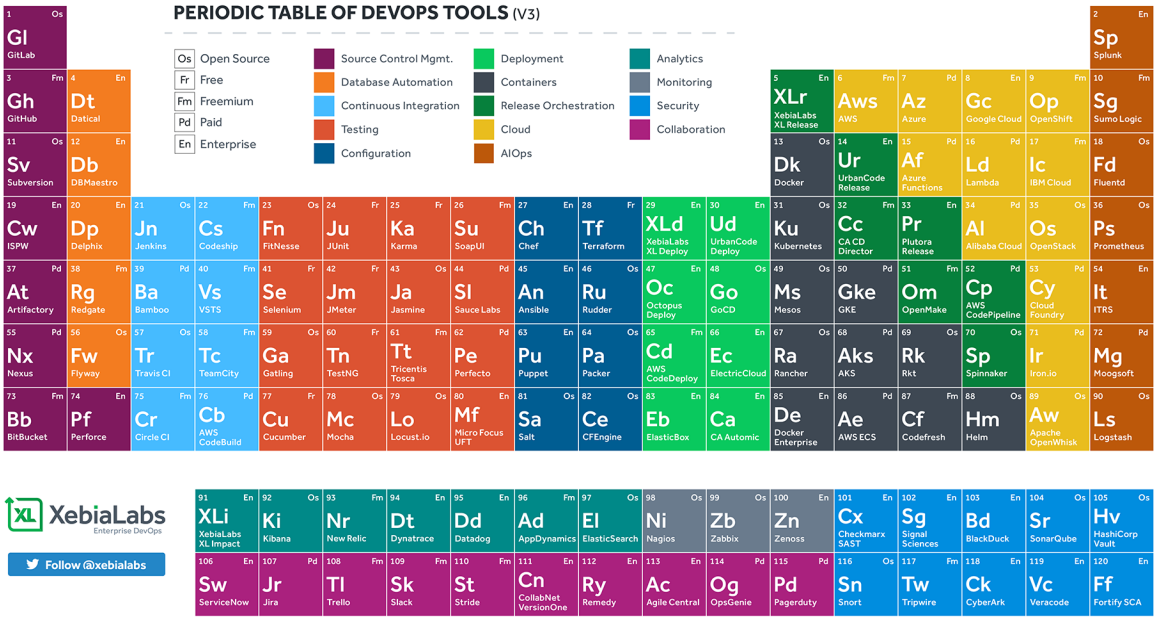
- Atlassian. *Suite* de productos que cumplen con la gestión de todo el ciclo de vida de DevOps de manera completa. Es muy popular entre los equipos de desarrollo y de operaciones, bastante completa para cada uno, y ha construido un buen vínculo que les permite colaborar.

- Chef Automate. Como complemento de la *suite* de Atlassian, para automatizar flujos de trabajo de DevOps.

- Puppet. Herramienta de gestión de la configuración de código abierto, que permite la orquestación y automatización de infraestructuras.

- XebiaLabs. Entorno completo de orquestación de diferentes herramientas, propias y de terceros, para integrar todo el ciclo de DevOps.

Figura 6. Tabla periódica de XebiaLabs de las herramientas de apoyo a DevOps



Fuente: XebiaLabs.

5. Tendencias

Como hemos visto, DevOps es un movimiento relativamente reciente que parte del concepto de la gestión ágil de servicios y que todavía está en evolución. Así es cómo podemos distinguir algunas tendencias, entre las que destacamos:

- Infraestructura como código (IaC). Es una alternativa de arquitectura tecnológica del departamento para las herramientas de gestión del mecanismo de virtualización automatizado. Lo podemos entender como la automatización en la nube. Básicamente, lo podemos resumir como la práctica de utilizar programas y *scripts* para configurar máquinas virtuales, ya sea en servidores propios o en la nube, y como consecuencia de herramientas de integración y despliegue automático. La denominada «infraestructura programable» es una oportunidad para optimizar la automatización del ciclo de DevOps. Muchos entornos donde se aplica esta tendencia utilizan tecnologías de contenerización, como Kubernetes y Dockers, que permiten a los desarrolladores disponer de entornos controlados en las distintas fases del ciclo de vida y automatizar al máximo el despliegue y la integración, garantizando la calidad del funcionamiento.

- DevSecOps. Aunque DevOps considera la seguridad como parte de la calidad, aparece esta tendencia con el objetivo de integrar las prácticas de seguridad dentro del ciclo de DevOps. DevSecOps implica crear una cultura de tipo «Security as Code» con una colaboración permanente y flexible entre ingenieros de versiones y equipos de seguridad. El movimiento DevSecOps no deja de ser una integración de la gestión de la seguridad dentro del movimiento de la gestión ágil de servicios. El objetivo es solucionar las barreras tradicionales entre desarrolladores y operaciones con los equipos de seguridad, a la vez que se garantiza una entrega rápida y segura del código.

- NOPS (NoOps, sin operaciones). Es una tendencia basada en el concepto en el que el sistema que permite la puesta en producción de los servicios se puede convertir en algo tan automatizado y abstraído de la infraestructura subyacente, local o en la nube, que no es necesario que haya un equipo dedicado para gestionarlo. El objetivo es automatizar completamente el despliegue, el control y la gestión de las aplicaciones y la infraestructura donde se ejecutan. Los principales son tanto la automatización como las tecnologías de virtualización y la nube (en realidad, Amazon AWS lo ofrece como un nombre de producto). Hay quién lo considera un destino lógico, resultado de la maduración de DevOps.

- Low-Code. Es el uso de plataformas que permiten crear aplicaciones con herramientas gráficas o con parametrizaciones específicas, con una baja

necesidad de tener que programar directamente. No es una idea nueva, pero ha vuelto con más fuerza, y más aún en el contexto de la gestión ágil de servicios. En cierto modo, es una tendencia a partir de DevOps. Esta da lugar a otra tendencia en el mercado, que es ZeroCode.

Ejercicios de autoevaluación

1. Explica cuál es el principal objetivo de DevOps.

2. ¿DevOps solo se puede utilizar en organizaciones nacidas en el mundo de los negocios digitales con procesos de entrega continua?

3. Lo más importante para empezar a aplicar DevOps es disponer de herramientas específicas y orientar las operaciones a la nube.

4. ¿Es CALMS un modelo de madurez para evaluar el grado de adopción de DevOps?

Solucionario

1. El objetivo de DevOps es presentar un sistema ágil, que implique políticas, procesos, personas y herramientas, que permita a la organización producir, entregar y operar software de calidad rápidamente, mejorando la comunicación, colaboración e integración entre desarrolladores, operaciones y gestión de calidad, automatizando la entrega y el despliegue, e institucionalizando el aprendizaje y la mejora continua.

2. No. Cualquier organización que quiera adoptar un modelo de gestión ágil de servicios de SI/TI puede adoptar DevOps. Aunque las organizaciones que han nacido dentro del mundo de los negocios digitales con procesos de entrega continua ya han creado el departamento de SI/TI en DevOps o lo han adoptado con más facilidad.

3. Lo más importante para adoptar DevOps es empezar por considerar que es un proyecto de cambio cultural. Es cierto que la disponibilidad de herramientas ayudará, así como adoptar estrategias de TI fundamentadas en tecnologías de virtualización y computación en la nube. DevOps es una cultura y, para adoptarla, es necesario que las personas que forman parte de los equipos implicados la adopten.

4. Aunque CALMS se usa como un modelo de madurez para evaluar el grado de adopción de DevOps, en realidad no lo es. CALMS es un marco de referencia para la integración de equipos, funciones y sistemas de DevOps dentro de una organización que responde al acrónimo de cultura, automatización, *Lean* (orientación al cliente), mejora continuada y compartimiento (*sharing*).

Glosario

Agile *m* Es un término que hace referencia a los modelos de gestión ágil de proyectos que podemos considerar grupos de metodologías de desarrollo de software fundamentadas en el desarrollo iterativo, donde los requisitos y las soluciones evolucionan a través de la colaboración entre equipos autoorganizados.

Bimodal *m* Es un modelo de gestión de la cartera de proyectos y la práctica de servicios, que supone funcionar con dos estilos de trabajo separados, pero coherentes dentro de un mismo portafolio: uno centrado en la predicción y el otro en la exploración. El modelo 1 está optimizado para zonas más previsibles y bien entendidas, y que se suele gestionar con modelos tradicionales. El modelo 2 es explorador, innovador para resolver nuevos problemas y optimizado para zonas de incertidumbre, y que se enfrenta con modelos ágiles.

CD *f* Véase **entrega continua**.

CI *f* Véase **integración continua**.

entrega continua *f* Es el proceso definido para iterar en ciclos relativamente cortos la creación, prueba, configuración y despliegue desde un entorno de desarrollo hasta un entorno de producción. Varios entornos de desarrollo, prueba o test crean un *pipeline*de lanzamiento para automatizar la creación de infraestructuras y el despliegue de una nueva versión.
sigla **CD**

integración continua *f* Es el proceso de automatización de la creación y la prueba del código, cada vez que un miembro del equipo hace cambios en el control de versiones, aportando una de nueva.
sigla **CD**

lean *m* Es un modelo que orienta a la organización para gestionar, focalizado en el concepto de mejora continua, con una visión de largo plazo de las tareas a realizar, que busca sistemáticamente conseguir pequeños cambios incrementales en los procesos para mejorar la eficiencia y la calidad.

Scrum *m* Es una metodología de desarrollo ágil y de carácter colaborativo que desglosa grandes procesos en piezas pequeñas, que pone el foco de las tareas a realizar en el equipo, la rendición de cuentas y el progreso iterativo hacia un objetivo muy definido. El modelo se presenta con una premisa sencilla, consistente en empezar con lo que se sabe o se puede conocer.

Bibliografía

Brown, A. (2015, julio). «What's the Best Team Structure for DevOps Success?». *Puppet* [en línea]. <https://puppet.com/blog/what%E2%80%99s-best-team-structure-for-devops-success>.

Byte TI (2015, junio). «DevOps: Rompiendo las barreras entre desarrollo y operaciones. *Revista Byte TI* [en línea]. <https://revistabyte.es/tendencias-byte-ti/devops-rompiendo-las-barreras-entre-desarrollo-y-operaciones-bajo-la-el-marco-de-la-agilidad-la-iso-20000-y-la-iso-1550412207/>.

Chawla, H. (2019). «DevOps: cómo romper la barrera entre Desarrollo y Operaciones». *Atlassian* [en línea]. <https://www.atlassian.com/es/devops>.

Hewlett Packard (2019). «¿Qué es la infraestructura como código?». *Hewlett Packard Enterprise* [en línea]. <https://www.hpe.com/es/es/what-is/infrastructure-as-code.html>.

ISACA (s/f). «DevOps Overview». *ISACA* [en línea]. <www.isaca.org/dev-ops>.

Quijano, J. (2018, abril). «El ciclo de DevOps, una guía para iniciarse en las fases que lo componen». *Genbeta* [en línea]. <https://www.genbeta.com/desarrollo/el-ciclo-de-devops-una-guia-para-iniciarse-en-las-fases-que-lo-componen>.

Rapid7 (2014, octubre). «How to Keep CALMS and Release More!». *Rapid7* [en línea]. <https://blog.rapid7.com/2014/10/24/how-to-keep-calms-and-release-more/>.

Rodriguez, C. (s/f). «DevOps, una breve introducción». *Github* [en línea]. <https://chrodriguez.github.io/devops-short-intro/>.

Silverthorne, V. (s/f). «DevOps Advice and Tips for Beginners». *TechTarget* [en línea]. <https://searchsoftwarequality.techtarget.com/360guide/Advice-and-tips-about-DevOps-for-beginners>.

Splunk (s/f). «The 5 Foundational DEVOPS practices». *Splunk* [en línea]. <https://www.splunk.com/en_us/form/5-foundational-devops-practices.html>.

Treadway, J. (2013, octubre). «Reshaping IT organizations to fulfill a DevOps strategy». *TechTarget* [en línea]. <https://searchitoperations.techtarget.com/tip/Reshaping-IT-organizations-to-fulfill-a-DevOps-strategy>.

UpGuard (2019). «DevOps Lessons for CIOs?». *UpGuard* [en línea]. <https://www.upguard.com/ebooks/devops-lessons-for-cios>.

XebiaLabs (2019). *Enterprise DevOps is Hard. We Make it Simple* [en línea]. <http://xebialabs.com>.