

Gentileza de **IBM**

2ª Edición Limitada IBM

DevOps

PARA
DUMMIES®

Aprenda sobre:

- EL valor y la necesidad empresarial de DevOps
- Las capacidades y vías de adopción de DevOps
- Cómo la nube acelera DevOps
- Diez mitos sobre DevOps

Sanjeev Sharma
Bernie Coyne



DevOps

PARA

DUMMIES®

2º Edición Limitada IBM

**Sanjeev Sharma y
Bernie Coyne**

WILEY

DevOps para Dummies®, 2ª Edición Limitada IBM

Publicado por
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

Copyright © 2015 by John Wiley & Sons, Inc.

Está prohibida la reproducción, almacenamiento en un sistema de recuperación y la transmisión en cualquier forma y mediante cualquier medio, ya sea electrónico, mecánico, fotocopiado, grabación, escaneado u otro tipo de esta publicación, en todo o en parte, salvo en lo permitido por las Secciones 107 o 108 de la Ley estadounidense de Copyright de 1976, sin premiso previo por escrito de la editorial. Las solicitudes de permiso a la editorial deberán dirigirse al Departamento de permisos, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, o por Internet en <http://www.wiley.com/go/permissions>.

Marcas comerciales: Wiley, para Dummies, el logotipo del Dummies Man, The Dummies Way, Dummies. com, Making Everything Easier y las imágenes de marca relacionadas son marcas comerciales o marcas comerciales registradas de John Wiley & Sons, Inc. y/o sus filiales en Estados Unidos y otros países y no pueden utilizarse sin permiso por escrito. IBM y el logotipo de IBM son marcas comerciales registradas de International Business Machines Corporation. Todas las restantes marcas son propiedad de sus respectivos titulares. John Wiley & Sons, Inc., no tiene relación con ningún producto o proveedor mencionado en este libro.

LÍMITE DE RESPONSABILIDAD/EXENCIÓN DE GARANTÍA: LA EDITORIAL Y EL AUTOR NO OFRECEN NINGUNA ASEVERACIÓN O GARANTÍA RESPECTO A LA PRECISIÓN Y TOTALIDAD DEL CONTENIDO DE ESTA OBRA Y ESPECÍFICAMENTE RECHAZAN CUALQUIER GARANTÍA, INCLUYENDO SIN LIMITACIÓN LAS GARANTÍAS DE ADECUACIÓN PARA UN PROPÓSITO PARTICULAR. NO ES POSIBLE CREAR O AMPLIAR LA GARANTÍA MEDIANTE VENTAS O MATERIALES PROMOCIONALES. EL ASESORAMIENTO Y LAS ESTRATEGIAS CONTENIDOS EN ESTA PUBLICACIÓN PUEDEN NO SER ADECUADOS PARA TODAS LAS SITUACIONES. ESTA OBRA SE VENDE ENTENDIÉNDOSE QUE LA EDITORIAL NO SE DEDICA A OFRECER SERVICIOS JURÍDICOS, CONTABLES U OTROS SERVICIOS PROFESIONALES. SI PRECISA ASISTENCIA PROFESIONAL, DEBERÁ RECURRIR A LOS SERVICIOS DE UN PROFESIONAL COMPETENTE. NI LA EDITORIAL NI EL AUTOR SE RESPONSABILIZAN DE LOS DAÑOS QUE PUEDAN SURGIR COMO CONSECUENCIA DE ESTA PUBLICACIÓN. EL HECHO DE QUE UNA ORGANIZACIÓN O SITIO WEB A LOS QUE SE HAGA REFERENCIA EN ESTA OBRA COMO CITA Y/O POSIBLE FUENTE DE INFORMACIÓN ADICIONAL NO SIGNIFICA QUE EL AUTOR O LA EDITORIAL ACONSEJEN LA INFORMACIÓN QUE LA ORGANIZACIÓN O EL SITIO WEB PUEDAN OFRECER O LAS RECOMENDACIONES QUE PUEDAN HACER. ASIMISMO, LOS LECTORES DEBEN SER CONSCIENTES DE QUE LOS SITIOS WEB INDICADOS EN ESTA OBRA PUEDEN HABER CAMBIADO O DESAPARECIDO DESDE EL MOMENTO EN QUE SE ESCRIBIÓ ESTA OBRA Y EL MOMENTO EN EL QUE SE LEE.

Para obtener información general sobre nuestros otros productos y servicios o cómo crear un libro *For Dummies* personalizado para su empresa u organización, póngase en contacto con nuestro Departamento de desarrollo comercial en Estados Unidos llamando al 877-409-4177, por correo electrónico a info@dummies.biz o visitando www.wiley.com/go/custompub. Para obtener información para utilizar bajo licencia la marca *Para Dummies* para productos o servicios, póngase en contacto con BrandedRights&Licenses@Wiley.com.

ISBN: 978-1-119-17751-7 (pbk); ISBN: 978-1-119-17752-4 (ebk)

Fabricado en Estados Unidos de América

10 9 8 7 6 5 4 3 2 1

Reconocimientos de la editorial

Algunas de las personas que contribuyeron a crear este libro son:

Editora del proyecto: Carrie A. Johnson

Editora de adquisiciones: Katie Mohr

Director editorial: Rev Mengle

Representante de desarrollo empresarial:

Sue Blessing

Coordinadora de producción:

Melissa Cossell

Índice



| | |
|---|-----------|
| Introducción | 1 |
| Acerca de este libro..... | 1 |
| Iconos utilizados en este libro..... | 2 |
| Más allá de este libro..... | 2 |
| Capítulo 1: ¿Qué es DevOps? | 3 |
| Comprender la necesidad empresarial de DevOps..... | 3 |
| Reconocer el valor empresarial de DevOps..... | 4 |
| Mejora de la experiencia del cliente | 5 |
| Mayor capacidad para la innovación | 5 |
| Reducción del tiempo para conseguir valor | 6 |
| Ver cómo funciona DevOps | 6 |
| Desarrollar y comprobar contra sistemas similares a los de producción..... | 6 |
| Implementar mediante procesos fiables y repetibles..... | 7 |
| Monitorizar y validar la calidad operativa | 8 |
| Amplificar los bucles de feedback..... | 8 |
| Capítulo 2: Examen de las capacidades de DevOps | 9 |
| Rutas para la adopción de DevOps..... | 9 |
| Planificar | 10 |
| Desarrollar/Comprobar | 11 |
| Desarrollo colaborativo | 11 |
| Comprobación continua..... | 13 |
| Implementar | 13 |
| Operar | 14 |
| Monitorización continua | 14 |
| Feedback del cliente y optimización continuos | 14 |
| Capítulo 3: Adoptar DevOps | 15 |
| Saber por dónde empezar..... | 15 |
| Identificar los objetivos de negocio | 16 |
| Identificar cuellos de botella en el proceso de entrega..... | 16 |
| Las personas en DevOps..... | 17 |
| La cultura DevOps | 17 |
| El equipo de DevOps | 19 |
| Los procesos en DevOps..... | 19 |
| DevOps como proceso empresarial | 19 |
| Proceso de gestión del cambio | 20 |
| Técnicas de DevOps | 21 |

| | |
|--|-----------|
| La tecnología en DevOps | 24 |
| La infraestructura como código | 25 |
| Proceso de entrega | 26 |
| Automatización de la implantación y administración de versiones | 28 |
| Capítulo 4: Examinar cómo cloud acelera DevOps | 31 |
| Utilizar cloud como facilitador de DevOps | 32 |
| Implementaciones Full-Stack | 34 |
| Elegir un modelo de servicio cloud para DevOps | 35 |
| IaaS | 35 |
| PaaS | 37 |
| Comprender qué es un cloud híbrido | 38 |
| Capítulo 5: Utilizar DevOps para resolver nuevos desafíos . . . | 41 |
| Aplicaciones móviles | 42 |
| Procesos ALM | 43 |
| Escalado ágil | 43 |
| Aplicaciones en múltiples niveles | 44 |
| DevOps en la empresa | 45 |
| Cadenas de suministro | 46 |
| El Internet de las cosas | 46 |
| Capítulo 6: Poner DevOps a trabajar: La historia de IBM . . . | 49 |
| Examinar el rol del ejecutivo | 50 |
| Reunir el equipo | 51 |
| Establecer metas para DevOps | 51 |
| Aprender de la transformación a DevOps | 52 |
| Ampliar las prácticas ágiles | 52 |
| Aprovechar la automatización de las pruebas | 53 |
| Construir un proceso de entrega | 54 |
| Experimentar rápidamente | 56 |
| Mejorar continuamente | 57 |
| Examinar los resultados de DevOps | 58 |
| Capítulo 7: Diez mitos sobre DevOps. | 59 |
| DevOps solo vale para empresas “nacidas en la web” | 59 |
| DevOps es hacer que Operaciones aprenda a programar | 60 |
| DevOps solo vale para Desarrollo y Operaciones | 60 |
| DevOps no vale para ITIL | 60 |
| DevOps no vale para sectores regulados | 61 |
| DevOps no vale para desarrollo externalizado | 61 |
| Sin cloud no hay DevOps | 61 |
| DevOps no vale para sistemas grandes y complejos | 62 |
| DevOps es cuestión exclusivamente de comunicación | 62 |
| DevOps significa Implementación del Cambio Continuo | 62 |

Introducción



Como la mayoría de los nuevos enfoques de trabajo, *DevOps* (abreviatura de desarrollo y operaciones), no es para muchas personas más que una palabra de moda. Todo el mundo habla de ello, pero no todos saben qué es. En términos generales, DevOps es un enfoque basado en los principios Lean y Agile mediante los cuales los dueños de la empresa y los departamentos de desarrollo, operaciones y seguro de calidad, colaboran para entregar el software de forma continua, lo que permite a la empresa aprovechar con mayor rapidez las oportunidades del mercado y reducir el tiempo necesario para incorporar el feedback de los clientes. De hecho, las aplicaciones empresariales son tan diversas y están compuestas de tantas múltiples tecnologías, bases de datos, dispositivos de usuario final, etc., que solo un enfoque DevOps puede tener éxito para abordar estas complejidades. Sin embargo, hay diversidad de opiniones sobre cómo utilizarlo.

Algunas personas dicen que DevOps es solo para los profesionales que lo practican; otros dicen que gira en torno a cloud. IBM adopta una visión amplia y holística y considera DevOps un enfoque para la entrega de software orientado a las empresas, que hace avanzar una función nueva o mejorada fase tras fase hasta producción, ofreciendo valor empresarial para los clientes de forma eficiente y captando el feedback desde el momento que los clientes comienzan a utilizar la nueva función. Para ello, usted necesita la participación de todas las partes interesadas, no solo los equipos de desarrollo y operaciones. Un enfoque auténticamente DevOps incluye a las líneas de negocio, profesionales, ejecutivos, socios, proveedores, etc.

Acerca de este libro

Este libro adopta un enfoque de DevOps centrado en las empresas. El rápido movimiento del mercado actual hace que DevOps sea esencial en todas las empresas que precisan ser Agile y Lean para responder con rapidez a los cambios en las demandas de los clientes, las condiciones del mercado, presiones competitivas o la normativa legal.

Si está leyendo este libro, asumimos que habrá oído hablar de DevOps pero que desea comprender lo que significa y cómo su

empresa puede obtener un beneficio comercial con su aplicación. Este libro está orientado a ejecutivos, responsables y profesionales nuevos en el campo de DevOps que buscan información más completa sobre este enfoque y que quieren evitarse la propaganda e ir al grano.

Iconos utilizados en este libro

Encontrará distintos iconos en los márgenes del libro. A continuación se definen sus significados.



El icono Consejo destaca información útil sobre distintos aspectos de DevOps.



El icono Recuerde destaca cosas que es importante tener en cuenta.



El icono Advertencia destaca información de importancia crítica.



La información técnica aborda temas que van más allá del nivel básico de DevOps. Sin embargo, esta información no es esencial.

Más allá de este libro

Encontrará información adicional sobre DevOps y el enfoque y los servicios de IBM visitando las siguientes páginas web:

- ✓ **La solución DevOps de IBM:** ibm.com/devops
- ✓ **DevOps: El enfoque IBM (libro blanco):**
ibm.biz/BdEnBz
- ✓ **La ventaja del software (estudio):** ibm.co/156KdoO
- ✓ **Adoptar el enfoque DevOps de IBM (artículo):**
ibm.biz/adoptingdevops
- ✓ **Servicios DevOps para Bluemix (servicio):**
bluemix.net

Capítulo 1

¿Qué es DevOps?

En este capítulo

- ▶ Identificar la necesidad empresarial de DevOps
- ▶ Hallar valor empresarial en DevOps
- ▶ Comprender los principios de DevOps

Implementar cualquier cambio en los procedimientos habituales de trabajo es siempre complicado y suele requerir una inversión. Cuando una organización adopta una tecnología, metodología o enfoque nuevos, tiene que deberse a una necesidad empresarial. Para desarrollar un plan de negocio para la adopción de DevOps es preciso comprender su necesidad empresarial y los problemas que aborda. En este capítulo le ofrecemos los elementos básicos que le permitirán comenzar a preparar su plan de negocio.

Comprender la necesidad empresarial de DevOps

Las organizaciones quieren crear aplicaciones o servicios innovadores para resolver problemas de negocio. Posiblemente quieran abordar problemas de negocio internos (como mejorar su sistema de gestión de relaciones con los clientes) o ayudar a sus clientes o usuarios finales (por ejemplo, con una nueva app móvil).

Sin embargo, son muchas las organizaciones que tienen complicaciones con los proyectos de software, y sus fracasos suelen estar relacionados con problemas en el desarrollo y distribución del software. Aunque para la mayor parte de las empresas el desarrollo y distribución del software son elementos críticos, un reciente estudio del sector realizado por IBM comprobó que solo el 25 por ciento consideran que sus equipos son eficaces. Esta laguna en ejecución provoca pérdidas de oportunidades de negocio.

Este problema se ve aún más amplificado por un gran cambio en el tipo de aplicaciones que las empresas han de entregar, desde sistemas de registro hasta sistemas de participación:

- ✓ **Sistemas de registro:** Las aplicaciones de software tradicionales son grandes sistemas que funcionan como sistemas de registro, que contienen ingentes cantidades de datos y/o transacciones y están diseñadas con un elevado nivel de fiabilidad y estabilidad. Estas aplicaciones no precisan cambiar con frecuencia, por lo que las organizaciones pueden satisfacer las necesidades de sus clientes y las suyas propias con solo una o dos grandes revisiones al año.
- ✓ **Sistemas de participación:** Con la llegada de las comunicaciones móviles y la madurez de las aplicaciones web, los sistemas de registro se ven suplementados por sistemas de participación, a los que los clientes pueden acceder directamente y utilizarlos para interactuar con el sistema. Estas aplicaciones han de ser fáciles de utilizar, de elevado rendimiento y capaces de cambiar rápidamente para satisfacer las necesidades de los clientes y las fuerzas del mercado, en continua evolución.

Al ser utilizados directamente por los clientes, los sistemas de participación se centran en la experiencia del usuario, velocidad del servicio y agilidad; en otras palabras, el enfoque DevOps.



Los sistemas de participación no son islas aisladas, sino que suelen estar conectados a sistemas de registro, por lo que los cambios rápidos en los sistemas de participación provocan cambios en los sistemas de registro. De hecho, cualquier sistema que precise distribución rápida de innovaciones precisa DevOps. Estas innovaciones son impulsadas principalmente por tendencias tecnológicas emergentes, como la computación cloud, las aplicaciones móviles, Big Data y las redes sociales, que pueden afectar a todo tipo de sistemas. Hablaremos sobre estas tecnologías emergentes desde el punto de vista de DevOps en los capítulos 4 y 5.

Reconocer el valor empresarial de DevOps

DevOps aplica los principios “lean” y “ágile” a toda la cadena de suministro del software. Permite a las empresas maximizar la velocidad de entrega o prestación de un producto o servicio, desde la idea inicial hasta la versión de producción, el feedback de los clientes y las mejoras basadas en este feedback.



DevOps mejora la forma en que una empresa aumenta el valor para sus clientes, proveedores y asociados, por lo que es un proceso de negocio esencial, no una simple capacidad del departamento de IT.

DevOps proporciona un importante rendimiento de la inversión en tres áreas:

- ✓ Mejora de la experiencia del cliente
- ✓ Mayor capacidad para la innovación
- ✓ Reducción del tiempo para conseguir valor

En las siguientes secciones trataremos estas tres áreas.

Mejora de la experiencia del cliente

Mejorar la experiencia del cliente (es decir, hacerla diferente y atractiva) aumenta la fidelidad de los clientes y la cuota de mercado. Para ofrecer esta experiencia, una empresa necesita recibir y responder continuamente al feedback de los clientes, lo que precisa mecanismos para obtener rápidamente feedback de todas las partes interesadas en las aplicaciones software que se distribuyen: clientes, líneas de negocio, usuarios, proveedores, empresas asociadas, etc.

En el actual mundo de los sistemas de participación (ver “Comprender la necesidad empresarial de DevOps” en este mismo capítulo), esta capacidad para reaccionar y adaptarse con agilidad permite mejorar la experiencia y fidelidad del cliente.

Mayor capacidad para la innovación

Las organizaciones modernas utilizan enfoques “lean” para aumentar su capacidad de innovación. Pretenden reducir el trabajo desperdiciado y duplicado y utilizar los recursos en actividades de mayor valor.



Un ejemplo de práctica habitual en pensamiento “lean” es el *test A-B*, en el cual las organizaciones piden a un pequeño grupo de usuarios que prueben y puntúen dos o más conjuntos de software que tienen distintas capacidades. A continuación, el conjunto mejor valorado se distribuye a todos los usuarios, y la otra versión se retira. Estas pruebas A-B son realistas solo con mecanismos eficientes y automatizados como los que facilita DevOps.

Reducción del tiempo para conseguir valor

Reducir el tiempo para conseguir valor conlleva desarrollar una cultura, prácticas y automatización que hagan posible la distribución rápida, eficiente y fiable del software hasta la fase de producción. Cuando se adopta DevOps como capacidad empresarial, ofrece las herramientas y la cultura necesarias para facilitar una eficiente planificación, predictibilidad y éxito de las distribuciones.

La definición de *valor* varía entre distintas organizaciones, e incluso entre proyectos, pero el objetivo de DevOps es ofrecer este valor de forma más rápida y eficiente.

Ver cómo funciona DevOps

El movimiento DevOps ha producido varios principios que han evolucionado con el tiempo y que siguen evolucionando. Distintos proveedores de soluciones, entre ellos IBM, han desarrollado sus propias variantes. Sin embargo, todos estos principios tienen en común un enfoque holístico para DevOps, que pueden adoptar las organizaciones de cualquier tamaño. Estos principios son

- ✓ Desarrollar y comprobar contra sistemas similares a los de producción
- ✓ Implementar mediante procesos fiables y repetibles
- ✓ Monitorizar y validar la calidad operativa
- ✓ Amplificar los bucles de feedback

En las siguientes secciones describimos estos principios con más detalle.

Desarrollar y comprobar contra sistemas similares a los de producción

Este principio tiene sus raíces en el concepto de DevOps de *trasladar a la izquierda*, por el que los problemas de operaciones se mueven a un punto anterior en el ciclo de vida de la distribución del software, hacia la fase de desarrollo (ver Figura 1-1).

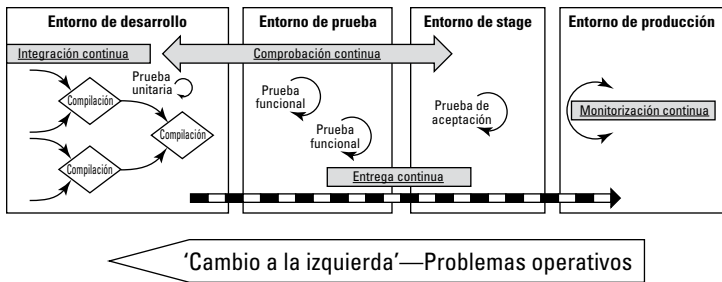


Figura 1-1: El concepto de trasladar a la izquierda mueve las operaciones a un punto anterior del ciclo de vida del desarrollo.

El objetivo es permitir a los equipos de desarrollo un seguro de calidad (QA) para desarrollar y comprobar contra sistemas que se comportan como el sistema de producción, para ver el comportamiento y rendimiento de la aplicación mucho antes de estar preparada para su distribución.



La primera exposición de la aplicación a un sistema similar al de producción deberá ser en una fase lo más temprana posible del ciclo de vida para abordar los grandes problemas potenciales. En primer lugar, permite comprobar la aplicación en un entorno similar al entorno de producción real en el que se distribuirá la aplicación; en segundo lugar, permite comprobar y validar por adelantado los propios procesos de distribución de la aplicación.

Desde la perspectiva de las operaciones, este principio ofrece también un enorme valor. Permite al equipo de operaciones examinar desde las fases iniciales del ciclo cómo se comportará su entorno al soportar la aplicación, lo que les permite crear un entorno perfectamente ajustado y sensible a la aplicación.

Implementar mediante procesos fiables y repetibles

Como sugiere el nombre, este principio permite a desarrollo y operaciones soportar un proceso de desarrollo de software ágil (o al menos interactivo) desde el inicio hasta la producción. La automatización es esencial para crear procesos iterativos, frecuentes, repetibles y fiables, que permitan a la organización crear un procedimiento de distribución que permita una implementación y comprobación automatizadas y continuadas. Volveremos a hablar sobre procedimientos de distribución en el capítulo 3.



Las implementaciones frecuentes también permiten a los equipos comprobar los propios procesos de implementación, con lo que se reduce el riesgo de fallos en la implementación en un momento posterior.

Monitorizar y validar la calidad operativa

Las organizaciones suelen ser eficaces para monitorizar aplicaciones y sistemas en producción, porque tienen herramientas que captan los indicadores de los sistemas de producción en tiempo real. Pero esta monitorización se lleva a cabo de forma aislada y desconectada. Este principio mueve la monitorización a un punto anterior del ciclo de vida y requiere que se realice una comprobación automatizada frecuente desde los primeros estadios del ciclo de vida para monitorizar características funcionales y no funcionales de la aplicación. Cada vez que se implementa y comprueba una aplicación es preciso captar y analizar los indicadores de calidad. Una monitorización frecuente permite detectar adelantadamente problemas operativos y de calidad que pueden aparecer en la producción.



Estos indicadores deben recogerse en un formato que todas las partes interesadas puedan comprender y usar.

Amplificar los bucles de feedback

Uno de los objetivos de DevOps es permitir a las organizaciones reaccionar y hacer cambios con más rapidez. En distribución de software, este objetivo requiere a la organización obtener feedback y aprender rápidamente con cada acción emprendida. Este principio exige a las organizaciones crear canales de comunicación que permitan a todas las partes interesadas acceder al feedback y actuar sobre él.

- ✓ Desarrollo puede actuar ajustando sus planes o prioridades para el proyecto.
- ✓ Producción puede actuar mejorando los entornos de producción.
- ✓ La empresa o división puede actuar modificando sus planes de distribución.

Capítulo 2

Examen de las capacidades de DevOps

En este capítulo

- Comprender la arquitectura de referencia de DevOps
- Considerar cuatro rutas para la adopción de DevOps

DevOps está constituido por un amplio conjunto de capacidades que abarcan todo el ciclo de entrega del software. El punto en el que una organización comience con DevOps depende de sus objetivos y metas de negocio: qué problemas trata de abordar y qué lagunas necesita llenar en sus capacidades de entrega de software.

En este capítulo examinará una arquitectura de referencia de DevOps y las distintas maneras en que hace posible el uso de DevOps para una empresa.

Rutas para la adopción de DevOps

Una *arquitectura de referencia* ofrece una plantilla de una solución demostrada mediante el uso de un conjunto de métodos y capacidades preferidos. Las arquitecturas de referencia de DevOps tratadas en este capítulo ayudan a los especialistas a acceder y utilizar las directrices, directivas y otros materiales que necesitan para diseñar una plataforma DevOps con personas, procesos y tecnología (ver el capítulo 3).

Una arquitectura de referencia ofrece capacidades a través de sus distintos componentes. A su vez, estas capacidades pueden ser ofrecidas por un único componente o por un grupo de componentes trabajando conjuntamente. Por tanto, podemos considerar la arquitectura de referencia de DevOps (Figura 2-1) desde la perspectiva

de las capacidades centrales que está destinada a proporcionar. A medida que la arquitectura en abstracto va tomando forma concreta, estas capacidades son ofrecidas por un conjunto de personas capacitadas, prácticas definidas y herramientas de automatización.

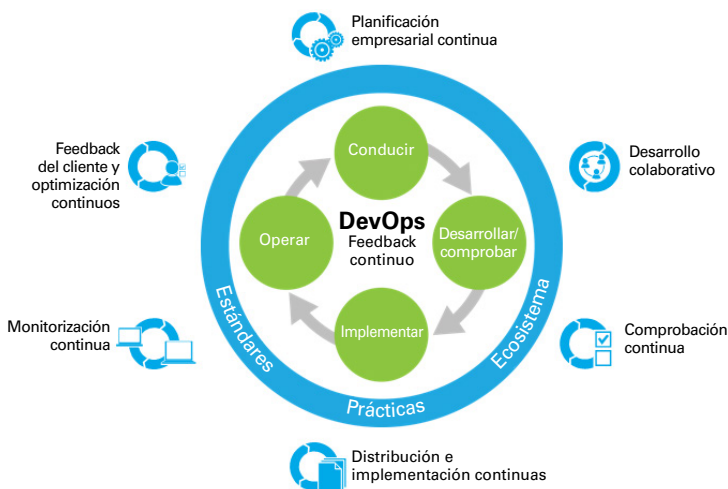


Figura 2-1: La arquitectura de referencia de DevOps.

La arquitectura de referencia de DevOps ilustrada en la figura Figure 2-1 propone los cuatro conjuntos siguientes de rutas de adopción:

- ✓ Planificar
- ✓ Desarrollar/Comprobar
- ✓ Implementar
- ✓ Operar

En las restantes secciones de este capítulo examinamos detenidamente estas rutas de adopción.

Planificar

Esta ruta de adopción se compone de una práctica centrada en establecer metas empresariales y ajustarlas según el feedback de los clientes: la *planificación empresarial continua*.

Las empresas actuales han de ser ágiles y reaccionar con rapidez al feedback de los clientes. Alcanzar esta meta se centra en la capacidad de una organización para hacer bien las cosas. Por

desgracia, los enfoques tradicionales para la entrega de productos son demasiado lentos para la velocidad actual de los negocios, en parte porque estos enfoques dependen del desarrollo a medida y procesos manuales y porque los equipos operan aisladamente. La información necesaria para planificar y volver a planificar rápidamente, manteniendo a la vez la capacidad de ofrecer valor, es fragmentada y poco uniforme. Con frecuencia el feedback adecuado no se recibe con suficiente antelación para conseguir el nivel de calidad adecuado para ofrecer un valor real.

Además, los equipos se esfuerzan por incorporar un feedback que debería informar la priorización de las inversiones y por colaborar como organización para impulsar la ejecución en un modelo de entrega continua. Algunos equipos consideran la planificación como un gasto de gobierno intrusivo y un lastre, en lugar de una actividad que les permite ofrecer valor con velocidad.

Una entrega más rápida proporciona mayor agilidad empresarial, pero usted también ha de gestionar la velocidad teniendo confianza en que lo que ofrece es lo adecuado. No es posible ofrecer software con velocidad si no tiene confianza en la exactitud de sus objetivos empresariales, sus medidas y sus plataformas.



DevOps le ayuda a reconciliar estas perspectivas enfrentadas y ayuda a los equipos a establecer colaborativamente objetivos empresariales y a cambiarlos continuamente según el feedback de los clientes, lo que mejora tanto su agilidad como los resultados empresariales. Pero las empresas también precisan administrar sus costes. Al identificar y eliminar desperdicios en el proceso de desarrollo, el equipo es más eficiente y reduce los costes. Este enfoque ayuda a los equipos a lograr un equilibrio óptimo entre todas estas consideraciones en todas las fases del ciclo de vida de DevOps al pasar a un modelo de entrega continua

Desarrollar/Comprobar

Esta ruta de adopción conlleva dos prácticas: desarrollo colaborativo y comprobación continua. Así, forma el núcleo de las capacidades de desarrollo y seguro de calidad (QA).

Desarrollo colaborativo

Los esfuerzos de entrega de software en una empresa implican a gran número de equipos multidisciplinares, como los responsables de líneas de negocio, analistas empresariales, arquitectos

empresariales y de software, desarrolladores, profesionales de QA, personal de operaciones, especialistas en seguridad, proveedores y asociados. Los miembros de estos equipos trabajan en múltiples plataformas y pueden estar situados en distintos centros. El *desarrollo colaborativo* les permite trabajar en conjunción mediante un conjunto común de prácticas y una plataforma común que pueden utilizar para crear y entregar el software.

Una capacidad clave incluida en el desarrollo colaborativo es la *integración continua* (Figura 2-2), una práctica por la que los desarrolladores de software integran de forma continua o frecuente su trabajo con el de otros miembros del equipo de desarrollo.

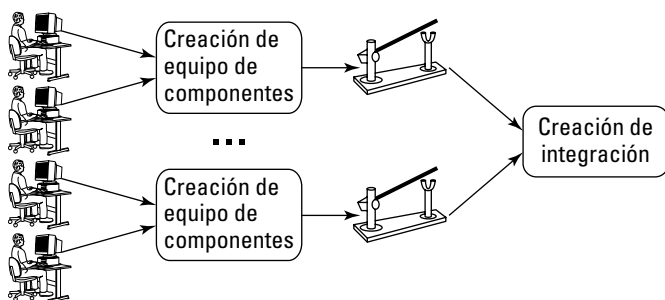


Figura 2-2: Colaboración mediante integración continua.

La integración continua adquirió popularidad con el movimiento Agile. La idea es que los desarrolladores integren su trabajo de forma regular con el del resto de los desarrolladores de su equipo y a continuación comprueben el trabajo integrado. En el caso de sistemas complejos compuestos de múltiples sistemas o servicios, los desarrolladores también pueden integrar su trabajo de forma regular con otros sistemas o servicios. La integración regular de los resultados permite descubrir y exponer en una fase temprana los riesgos para la integración. En sistemas complejos, también expone riesgos conocidos y desconocidos, tanto técnicos como relacionados con la temporización.

Comprobación continua

La integración continua (ver sección anterior) tiene varios objetivos:

- ✓ Posibilitar la comprobación y verificación continuas del código
- ✓ Validar que el código producido e integrado con el de otros desarrolladores y otros componentes de la aplicación funcione y tenga el rendimiento pretendido
- ✓ Comprobar continuamente la aplicación que se está desarrollando

Comprobación continua significa comprobar en una fase más temprana y de forma continua durante todo el ciclo de vida, lo que se traduce en menores costes, ciclos de comprobación más cortos y feedback continuo sobre la calidad. Este proceso es conocido también como *traslado a la izquierda de la comprobación*, lo que destaca la integración de las actividades de desarrollo y comprobación para garantizar la incorporación de la calidad desde las fases más tempranas del ciclo de vida que sea posible, no algo que se deja para más tarde. Esto es posible mediante la adopción de capacidades como la comprobación automatizada y la virtualización del servicio. La virtualización del servicio es la nueva capacidad de simular entornos semejantes a los de producción y hace viable la comprobación continua.

Implementar

La ruta de adopción Implementar es donde se originaron la mayoría de las capacidades originales de DevOps. La distribución e implementación continuas llevan el concepto de integración continua al siguiente paso. La práctica que posibilita la distribución e implementación también hace posible la creación de un proceso de implementación (Capítulo 3). Este proceso facilita la implementación continua del software hasta QA y producción, de forma eficiente y automatizada. El objetivo de la distribución e implementación continuadas es ofrecer nuevas características a los clientes y usuarios lo más rápidamente posible.

La mayoría de las herramientas y procesos que componen el núcleo de la tecnología DevOps existen para facilitar la integración, distribución e implementación continuas. En capítulos posteriores analizaremos estos temas en más detalle.



Operar

La ruta de adopción Operar incluye dos prácticas que permiten a las empresas monitorizar el rendimiento en producción de las aplicaciones publicadas y recibir feedback de los clientes. Estos datos permiten a la empresa reaccionar con agilidad y modificar sus planes de negocio según convenga.

Monitorización continua

La *monitorización continua* proporciona datos y parámetros sobre la aplicación para operaciones, QA, desarrollo, personal de líneas de negocio y otras partes interesadas, en distintos estadios del ciclo de entrega.



Estos parámetros no se limitan a la producción. Permiten a todas las partes interesadas reaccionar con mejoras o cambios en las características entregadas y/o los planes de negocio necesarios para entregarlas.

Feedback del cliente y optimización continuos

Los dos tipos más importantes de información que puede obtener un equipo de entrega de software son datos sobre el uso de la aplicación por parte de los clientes y el feedback de estos clientes al utilizar la aplicación. Las nuevas tecnologías permiten a las empresas conocer el comportamiento de los clientes y los puntos conflictivos para los clientes en el momento mismo en el que utilizan la aplicación. Este feedback permite a las distintas partes interesadas emprender las acciones apropiadas para mejorar las aplicaciones y mejorar la experiencia de los clientes. Las líneas de negocio pueden ajustar sus planes de negocio, desarrollo puede ajustar las capacidades que ofrece y operaciones puede mejorar el entorno en el que se implementa la aplicación. Este bucle de feedback continuo es un componente esencial de DevOps y permite a las empresas ser más ágiles y dar mejor respuesta a las necesidades del cliente.

Capítulo 3

Adoptar DevOps

En este capítulo

- ▶ Aumentar la eficiencia de las personas
- ▶ Optimizar los procesos
- ▶ Elegir las herramientas adecuadas

Adoptar cualquier nueva capacidad suele precisar un plan que incluye personas, procesos y tecnología. No es posible adoptar nuevas capacidades, en especial en una empresa con numerosas partes interesadas posiblemente en distintos centros, sin considerar los tres aspectos de las capacidades que vamos a adoptar.

En este capítulo trataremos los aspectos de las personas, procesos y tecnología en relación con DevOps.



Aunque el término *DevOps* sugiere capacidades basadas en el desarrollo y las operaciones, DevOps es una capacidad empresarial que abarca a todos los implicados de una organización, como los responsables de negocio, arquitectura, diseño, desarrollo, seguro de calidad (QA), operaciones, seguridad, empresas asociadas y proveedores. Si excluimos a cualquiera de estas partes interesadas, tanto internas como externas, nuestra implementación de DevOps no será completa.

Saber por dónde empezar

Esta sección ofrece orientación sobre cómo comenzar en DevOps, incluyendo la cultura adecuada, identificar los retos empresariales y localizar los cuellos de botella que es preciso eliminar.

Identificar los objetivos de negocio

La primera tarea para crear una cultura es hacer que todos se muevan en la misma dirección y avancen hacia la misma meta, lo que supone identificar objetivos empresariales comunes para el equipo y para el conjunto de la organización. Es importante incentivar al equipo entero en base a resultados empresariales en lugar de incentivos de equipo que puedan provocar conflictos. Cuando las personas conocen la meta común para la que trabajan y cómo va a medirse su avance hacia esta meta, se reducen los problemas por parte de equipos o personas que tengan sus propias prioridades.



DevOps no es la meta. Es una ayuda para alcanzar sus metas.

Los capítulos 4 y 5 destacan varios nuevos problemas empresariales que aborda DevOps. Su organización puede utilizar estos problemas como punto de partida para identificar las metas que desea alcanzar, y a continuación puede desarrollar un conjunto común de hitos que pueden ser útiles a los diferentes equipos en su camino hasta la meta.

Identificar cuellos de botella en el proceso de entrega.

Las fuentes más importantes de ineficiencias en el proceso de entrega se han categorizado como:

- ✓ Carga innecesaria (tener que comunicar repetidamente la misma información y conocimientos)
- ✓ Duplicación innecesaria de trabajos (defectos no descubiertos en la comprobación o devolución de tareas al equipo de desarrollo por parte de producción)
- ✓ Exceso de producción (desarrollo de funcionalidades que no se precisaban)

Uno de los principales cuellos de botella en el proceso de entrega es implantar la infraestructura. La adopción de un enfoque DevOps aumenta la velocidad de la entrega de aplicaciones y presiona a la infraestructura para responder más rápidamente. Es aquí donde los entornos definidos por software le permiten captar la infraestructura como un tipo de patrón programable y repetible, lo que contribuye a acelerar las implantaciones. Encontrará más información en la sección “Infraestructura como código” en este mismo capítulo.

Investigando más a fondo, es posible que desee optimizar el proceso con un flujo homogéneo de extremo a extremo. El rendimiento de cada proceso debe ser igual para evitar la acumulación de tareas. Para lograr este equilibrio, usted deberá instrumentar o medir el proceso de entrega en puntos clave para poder minimizar el tiempo de espera en colas de tareas pendientes, optimizar el trabajo en curso y ajustar la capacidad y el flujo.

Las personas en DevOps

Esta sección aborda el aspecto de las personas en la adopción de DevOps, incluyendo la creación de la cultura necesaria.

La cultura DevOps

En su origen, DevOps es un movimiento cultural, de personas. Una organización puede adoptar los procesos más eficientes o las herramientas más autorizadas posibles, pero esto puede resultar inútil sin las personas que tendrán que ejecutar estos procesos y utilizar estas herramientas. Crear una cultura DevOps es, pues, uno de los elementos básicos para la adopción de DevOps.



Una cultura DevOps se caracteriza por un elevado nivel de colaboración entre los roles, dar prioridad a los objetivos empresariales en lugar de los departamentales, confianza y conceder elevado valor a aprender a través de la experimentación.

Construir una cultura es distinto de adoptar un proceso o una herramienta Precisa, por falta de mejor término una ingeniería social de las personas, cada una de ellas con predisposiciones, experiencias y prejuicios únicos. Esta diversidad puede hacer que crear una cultura sea un desafío complejo.



En el núcleo de DevOps encontramos las prácticas de transformación Agile y Lean como el Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD) y Scrum. Si su organización ya las ha aplicado, pueden utilizarse para contribuir a la adopción de una cultura DevOps.

Medir la cultura

Medir una cultura es extremadamente difícil. ¿Cómo medir con precisión una mejora en la colaboración o en la moral del equipo? Usted puede hacer una medición directa de las actitudes y la moral del equipo mediante encuestas, pero la tasa de error estadístico puede ser elevada, ya que suele tratarse de equipos pequeños.

Por el contrario, usted puede realizar una medición indirecta comprobando

con qué frecuencia recurre un miembro del equipo de desarrollo a un miembro de un equipo de operaciones o AQ para colaborar en la resolución de un problema sin pasar por los canales oficiales o múltiples niveles directivos.

La colaboración y la comunicación entre partes implicadas: esta es la cultura de DevOps.



Construir una cultura DevOps obliga a los líderes de la organización a trabajar con sus equipos para crear un entorno y una cultura de la colaboración y el uso compartido. Los líderes han de eliminar todo tipo de barreras autoimpuestas que obstaculicen la cooperación. Las mediciones típicas recompensan a los equipos de operaciones por su tiempo de actividad y escalabilidad, y recompensan a los desarrolladores por la entrega de nuevas características, pero enfrentan a estos grupos entre sí. Operaciones sabe que la mejor protección para la producción es no aceptar ningún cambio, por ejemplo, y Desarrollo no tiene mayor incentivo para centrarse en la calidad. Sustituya estas medidas con una responsabilidad compartida para entregar nuevas capacidades de forma rápida y segura.

Los líderes de la organización deberán animar aún más la colaboración mediante una mejora de la visibilidad. Es esencial establecer un conjunto común de herramientas de colaboración, en especial con equipos separados geográficamente y que no puedan colaborar físicamente. Proporcionar a todos los interesados visibilidad en las metas y el estado de un proyecto es crucial para crear una cultura DevOps basada en la confianza y la colaboración.



En ocasiones, construir una cultura DevOps obliga a las personas a cambiar. Es posible que sea preciso cambiar la función de las personas más resistentes al cambio, es decir a adoptar la cultura DevOps,

El equipo de DevOps

Los argumentos a favor y en contra de tener un equipo independiente para DevOps son tan antiguos como el propio concepto. Algunas organizaciones, como Netflix, no tiene equipos separados de desarrollo y operaciones: en su lugar, ambos conjuntos de responsabilidades corresponden a un único equipo de “NoOps”. Otras organizaciones han alcanzado el éxito con equipos de enlace de DevOps, que resuelven los conflictos y fomentan la colaboración. Puede tratarse de un grupo ya existente de herramientas o procesos, o bien puede ser un nuevo equipo integrado por representantes de todos los equipos con responsabilidad en la aplicación que se entrega.

Si usted elige tener un equipo de DevOps, su meta más importante es asegurarse de que funcione como centro de excelencia que facilite la colaboración sin añadir un nivel adicional de burocracia ni convertirse en el equipo responsable de abordar todos los problemas relacionados con DevOps: esto podría frustrar el propósito de adoptar una cultura DevOps.

Los procesos en DevOps

En la sección anterior tratamos sobre el rol de las personas y la cultura en la adopción de DevOps. Los procesos definen lo que hacen estas personas. Su organización puede tener una estupenda cultura de colaboración, pero si las personas hacen las cosas equivocadas o hacen las cosas correctas de forma equivocada, el fracaso sigue siendo probable.

Hay un enorme número de procesos que se identifican con DevOps, demasiados para el ámbito de este libro. Esta sección trata algunos de los procesos clave a la luz de su adopción en toda la empresa.

DevOps como proceso empresarial

DevOps como capacidad afecta a toda la empresa. Hace que la empresa sea más ágil y mejora la entrega de las capacidades a los clientes. Es posible ampliar DevOps aun más considerándolo un *proceso empresarial*: un conjunto de actividades o tareas que producen un resultado específico (servicio o producto) para los clientes.

En la arquitectura de referencia introducida en el Capítulo 2, el proceso empresarial de DevOps implica tomar capacidades de la idea (generalmente identificada con los dueños de la empresa) desde el desarrollo y comprobación hasta la producción.



Aunque este proceso empresarial no es lo suficientemente maduro como para ser recogido en un conjunto de flujos de procesos sencillos, usted deberá recoger los flujos de proceso que ya utiliza su organización para ofrecer capacidades. Entonces podrá identificar áreas de mejora, tanto mejorando los procesos en sí como introduciendo automatización (ver la sección “La tecnología en DevOps” en este mismo capítulo).

Proceso de gestión del cambio

La *gestión del cambio* es un conjunto de actividades diseñadas para controlar, gestionar y hacer seguimiento del cambio identificando los productos del trabajo que probablemente cambien y los procesos utilizados para implementar este cambio. El proceso de gestión del cambio que utilice una organización forma parte esencial del flujo general de proceso de DevOps. La gestión del cambio anima la forma en que los procesos de DevOps absorben y reaccionan a las peticiones de cambio y el feedback de los clientes.



Las organizaciones que hayan aplicado la gestión del ciclo de vida de las aplicaciones (ALM) ya dispondrán de procesos de gestión del cambio bien definidos y (probablemente) automatizados.

La gestión del cambio deberá incluir procesos que posibiliten las siguientes capacidades:

- ✓ Gestión de los elementos del trabajo
- ✓ Flujos de trabajo configurables de los elementos del trabajo
- ✓ Gestión de la configuración de proyectos
- ✓ Planificación (ágil e iterativa)
- ✓ Control de acceso a los materiales basado en roles

Los enfoques tradicionales de la gestión del cambio tienden a limitarse a la solicitud de cambios o gestión de defectos, con capacidad limitada para rastrear los eventos entre las solicitudes de cambio o defectos y el código o requerimientos asociados. Estos enfoques no ofrecen una gestión integrada de los elementos de trabajo en todo el ciclo de vida ni capacidad integrada para el rastreo de todos los tipos de activos. DevOps, sin embargo, requiere que todas las partes interesadas sean capaces de ver y colaborar en todos los cambios en el ciclo de vida del desarrollo de software.

La gestión del cambio desde DevOps o ALM incluye procesos que proporcionan gestión de los elementos de trabajo para todos los proyectos, tareas y activos asociados, no solo los afectados por

solicitudes de cambio o defectos. También incluye procesos que permiten a la empresa relacionar elementos de trabajo con todos los materiales, activos de proyectos y otros elementos de trabajo creados, modificados, referenciados o eliminados por cualquier persona que trabaje en ellos. Estos procesos proporcionan a los miembros del equipo acceso basado en el rol para toda la información relacionada con el cambio y apoyan los esfuerzos de desarrollo de proyectos iterativos y ágiles.

Técnicas de DevOps

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

- ✓ Mejora continua
- ✓ Planificación de versiones
- ✓ Integración continua
- ✓ Entrega continua
- ✓ Comprobación continua
- ✓ Monitorización y feedback continuos

Las siguientes secciones examinan estas técnicas en detalle.

Mejora continua

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

Planificación de versiones

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

La mayoría de las empresas actuales llevan a cabo esta tarea utilizando hojas de cálculo y celebrando reuniones (con frecuencia largas) con todas las partes interesadas de la empresa para hacer seguimiento de todas las solicitudes de necesidades en desarrollo, su estado de desarrollo y sus planes de versiones. Sin embargo, la automatización y unos procesos bien definidos eliminan la necesidad de estas hojas de cálculo y reuniones y hacen posible la edición de nuevas versiones de forma optimizada y, sobre todo, predecible. Aprovechar unas prácticas Lean y Agile también resulta en versiones más pequeñas y frecuentes, que permiten centrarse en la calidad.

Integración continua

La integración continua (descrita en el capítulo 2) añade un enorme valor a DevOps al permitir a grandes equipos de desarrolladores que trabajan con componentes de múltiples tecnologías entregar el software de forma Agile. También garantiza que el trabajo de cada equipo esté integrado continuamente con el del resto de equipos de desarrollo y a continuación sea validado. Con ello, la integración continua reduce el riesgo e identifica los problemas en una etapa temprana del ciclo de vida de desarrollo del software.

Entrega continua

La integración continua conduce de forma natural a la práctica de la entrega continua: el proceso de automatizar la implantación del software en los entornos de comprobación, pruebas del sistema y producción. Aunque algunas organizaciones no llegan hasta la producción, las que adoptan DevOps suelen utilizar el mismo proceso automatizado en todos los entornos para mejorar la eficiencia y reducir el riesgo introducido por procesos poco uniformes.

En entornos de prueba, automatizar la configuración, refrescar los datos y a continuación implantar el software en el entorno de prueba, seguido de la ejecución de pruebas automatizadas acelera los ciclos de feedback a desarrollo de los resultados de las pruebas.



Adoptar la entrega continua suele ser la parte más crítica de la adopción de DevOps. Para muchos, DevOps se limita a la entrega continua, y la mayoría de las herramientas que se promocionan como DevOps solo abordan este proceso. Sin embargo, como puede ver a lo largo de este libro, DevOps tiene un ámbito mucho más amplio. La entrega continua es un componente esencial de DevOps, pero no es el único.



Basándose en las necesidades de negocio y en los problemas más acuciantes de su organización, usted puede preferir iniciar la adopción con otro de los procesos o rutas de adopción descritos en el capítulo 2.

Comprobación continua

En el capítulo 2 introdujimos el concepto de comprobación continua. Desde una perspectiva de los procesos, usted deberá adoptar procesos en tres áreas para hacer posible la comprobación continua:

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

En una organización, los equipos de QA han de determinar qué procesos adoptar para cada área. Los procesos que adopten pueden variar entre proyectos, basándose en las necesidades de comprobación concretas y en los requisitos de los acuerdos de nivel de servicio. Por ejemplo, es posible que las aplicaciones cara al cliente precisen una mayor comprobación de seguridad que las aplicaciones internas. La provisión del entorno de prueba y la administración de los datos de prueba son retos más importantes para proyectos que utilicen metodologías Ágiles y practiquen una integración continua que para proyectos que utilicen una metodología en cascada y hagan pruebas cada varios meses. De manera semejante, los requisitos de pruebas de función y rendimiento para aplicaciones complejas con componentes que tengan distintos ciclos de entrega son distintos que para apps web monolíticas y simples.

Monitorización y feedback continuos

El feedback de los clientes puede provenir de distintas fuentes, desde incidencias abiertas por los clientes, solicitudes formales de cambio, quejas informales o puntuación en los app stores. Debido especialmente a la popularidad de las redes sociales y app stores (ver Capítulo 5), las empresas precisan procesos bien definidos que absorban el feedback de muy numerosas fuentes para incorporarlos en los planes de entrega del software. Estos procesos deberán ser lo suficientemente ágiles como para adaptarse a los cambios en el mercado y en la normativa.

Medir la adopción de los procesos

Es posible medir el éxito de la adopción de un proceso comprobando si un conjunto de parámetros de la eficiencia y la calidad mejora a lo largo del tiempo. Este tipo de medición tiene dos requisitos previos:

- ✓ Identificar el conjunto adecuado de parámetros de la eficiencia y la calidad. Estos parámetros deben ser realmente importantes para la empresa.

- ✓ Establecer un punto de referencia contra el que medir la mejora.

Puede utilizar cualquiera de los distintos marcos bien definidos para medir la madurez del proceso. Para procesos específicos para DevOps, la madurez puede medirse mediante modelos como el nuevo IBM DevOps Maturity Model. Encontrará más información sobre el modelo de madurez IBM en ibm.biz/adoptingdevops.

El feedback puede proceder también de datos de monitorización. Estos datos proceden de los servidores en los que se ejecuta la aplicación, de desarrollo, QA y producción, o bien de herramientas de medición integradas en la aplicación que recoge las acciones de los usuarios.



Existe la posibilidad de que se produzca una sobrecarga de datos: las empresas precisan procesos de captura de datos y uso de datos que mejoren sus aplicaciones y los entornos en los que se ejecutan.

La tecnología en DevOps

La tecnología permite a las personas centrarse en el trabajo creativo de mayor valor y delegar las tareas rutinarias a la automatización. Además, la tecnología permite a los equipos aprovechar y escalar su tiempo y sus destrezas.

Si una organización está creando o manteniendo múltiples aplicaciones, todo cuanto haga tiene que ser repetible de forma fiable para garantizar la calidad en todas las aplicaciones. No puede partir de cero en cada nueva versión o corrección de errores para cada aplicación. Para ser eficiente y optimizar costes, la organización ha de reutilizar activos, código y prácticas.

Además, estandarizar la automatización hace que las personas sean más eficaces (ver “Las personas en DevOps,” en este mismo

capítulo). Las organizaciones pueden cambiar de empleados, contratistas o proveedores de recursos, y las personas pueden pasar de un proyecto a otro. Pero un conjunto común de herramientas permite a todos trabajar en cualquier lugar y los miembros de cada equipo solo necesitan aprender un conjunto de herramientas, y un proceso que resulta eficiente, económico, repetible y escalable.

La infraestructura como código

La *infraestructura como código* es una capacidad central de DevOps que permite a las organizaciones gestionar la escala y la velocidad de provisión y configuración de los entornos para hacer posible la entrega continua.

Evolucionar en torno al concepto de infraestructura como código es el concepto de los *entornos definidos por software*. Mientras que la infraestructura como código se ocupa de capturar definiciones y configuraciones de nodos como código, los entornos definidos por software utilizan tecnologías que definen sistemas completos formados por múltiples nodos: no solo sus configuraciones sino también sus definiciones, topologías, roles, relaciones, las cargas de trabajo y sus políticas, y el comportamiento.

Hay tres tipos de herramientas de automatización disponibles para gestionar la infraestructura como código:

- ✓ **Herramientas centradas en las aplicaciones o el middleware:** Generalmente tienen capacidad para gestionar como código tanto los servidores de aplicaciones como las aplicaciones ejecutadas en ellos. Son herramientas especializadas, combinadas con librerías de tareas de automatización típicas de las tecnologías que soportan. No pueden realizar tareas de bajo nivel, como configurar un sistema operativo (OS), pero sí pueden automatizar completamente tareas a nivel de servidor y aplicación.
- ✓ **Herramientas de entorno e implantación:** Son una nueva clase de herramientas con capacidad para implantar configuraciones de infraestructura y código de aplicaciones.
- ✓ **Herramientas genéricas:** Estas herramientas no están especializadas en ninguna tecnología y pueden adaptarse para realizar distintos tipos de tareas, desde configurar un SO en un nodo virtual o físico hasta configurar puertos de firewall. Precisan mucho más trabajo previo que las herramientas centradas en aplicaciones o middleware, pero tienen capacidad para un rango de tareas mayor.



El uso de una herramienta de administración e implantación de entornos como IBM UrbanCode Deploy with Patterns, permite a las organizaciones diseñar, implantar y reutilizar entornos rápidamente y contribuye a acelerar el proceso de entrega.

Proceso de entrega

Un *proceso de entrega* está compuesto por los estadios por los que pasa una aplicación desde el desarrollo hasta la producción. La Figura 3-1 muestra un conjunto de estadios típico. Sin embargo, estos estadios pueden variar entre organizaciones y también entre aplicaciones, dependiendo de las necesidades de la organización, el proceso de entrega del software y el nivel de madurez. También puede variar el nivel de automatización. Algunas organizaciones automatizan sus proceso de entrega, otras obligan a realizar comprobaciones manuales del software según las exigencias de la normativa o de la empresa. No es necesario abordar todos los estadios a la vez. Comience por centrarse en las partes críticas de la organización (no todo a la vez) y poco a poco vaya ampliando hasta incluir todos los estadios.

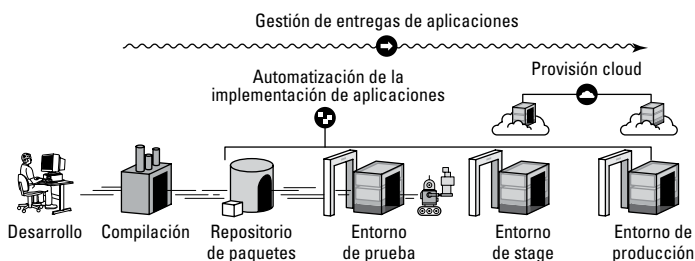


Figura 3-1: Estados de un proceso típico de entrega de DevOps.

Un proceso de entrega típico tiene los estadios descritos en las secciones siguientes.

Entorno de desarrollo

El esfuerzo de desarrollo de una aplicación tiene lugar en un *entorno de desarrollo*, que ofrece múltiples herramientas que permiten a los desarrolladores escribir y comprobar su código. Más allá de las herramientas de entorno de desarrollo integrado (IDE) que los desarrolladores utilizan para escribir código, este estadio incluye herramientas que posibilitan el desarrollo colaborativo, como por ejemplo, herramientas para gestión del control de fuentes, gestión de elementos de trabajo, colaboración comprobación de unidades y planificación de proyectos. Las herramientas de este estadio suelen ser multiplataforma y multitecnología, dependiendo del tipo de desarrollo que se realice.

Estadio de compilación

En el *estadio de compilación* se genera el código para crear y comprobar unitariamente los archivos binarios que se va a implantar. Es posible utilizar en este estadio múltiples herramientas de compilación, dependiendo de las necesidades multiplataforma y multitecnología. Las organizaciones de desarrollo suelen utilizar servidores de compilación para facilitar el gran número de compilaciones que se precisa de forma continuada para hacer posible la integración continua.

Repositorio de paquetes

Un *repositorio de paquetes* (también denominado *repositorio de activos* o *repositorio de artefactos*) es un mecanismo común de almacenamiento para los archivos binarios creados durante el estadio de compilación. Estos repositorios almacenan también los activos asociados con los binarios para facilitar su implantación, como pueden ser archivos de configuración, archivos de infraestructura como código y scripts de implantación.

Entorno de prueba

Un *entorno de prueba* es el lugar en el que los equipos de QA, aceptación por el usuario y desarrollo/comprobación realizan la comprobación. En este estadio se utilizan numerosos tipos de herramientas, según las necesidades de QA. Algunos ejemplos son:

- ✔ **Administración del entorno de prueba:** Estas herramientas facilitan la provisión y configuración de los entornos de prueba. Incluyen tecnologías de infraestructura como código y (si se trata de un entorno cloud), herramientas de provisión y gestión para cloud.
- ✔ **Administración de los datos de prueba:** Administrar los datos de prueba es una función esencial para cualquier organización que desee habilitar la comprobación continua. El número de pruebas que puedan ejecutarse y la frecuencia con que se ejecuten están limitados por la cantidad de datos disponibles para las pruebas y la velocidad con que es posible actualizar estos datos.
- ✔ **Pruebas de integración, función, rendimiento y seguridad:** Existen herramientas automatizadas para cada uno de estos tipos de prueba. Estas herramientas deben integrarse con una herramienta o repositorio común de gestión de activos de prueba donde puedan almacenarse todos los escenarios de prueba, scripts de prueba y resultados asociados y donde pueda establecerse la trazabilidad hasta el código, requisitos y defectos.

✓ **Virtualización del servicio:** Las aplicaciones modernas no son simples aplicaciones monolíticas. Se trata de sistemas complejos que dependen de otras aplicaciones, servidores de aplicaciones, bases de datos e incluso aplicaciones y fuentes de datos de terceros. Por desgracia, en el momento de la prueba estos componentes pueden no estar disponibles o resultar muy costosos. Las soluciones de virtualización del servicio simulan el comportamiento, funcionalidad y rendimiento de determinados componentes de una aplicación para posibilitar la comprobación de extremo a extremo del conjunto de la aplicación. Estas herramientas crean componentes virtuales (stubs) de las aplicaciones y servicios necesarios para ejecutar las pruebas. Es posible comprobar el comportamiento y rendimiento de la aplicación al interactuar con estos stubs. IBM Rational Test Virtualization Server ofrece estas capacidades de virtualización de las pruebas.

Entornos de preparación y de producción

Las aplicaciones se implantan en los entornos de preparación y de producción. Las herramientas utilizadas en estos estadios incluyen herramientas de gestión y provisión del entorno. Las herramientas para infraestructura como código también tienen un papel crítico en estos estadios, debido a la gran escala del entorno en estos estadios. En la actualidad, con la aparición de las tecnologías de virtualización y cloud, los entornos de preparación y producción, pueden incluir cientos o incluso miles de servidores. Las herramientas de monitorización permiten a las organizaciones monitorizar las aplicaciones implantadas en producción.

Automatización de la implantación y administración de versiones

Gestionar la automatización de la implantación de aplicaciones de un estadio al siguiente precisa herramientas especializadas, algunas de las cuales cubrimos en las siguientes secciones.

Automatización de la implantación

Las herramientas de automatización de la implantación son las herramientas centrales del espacio DevOps. Estas herramientas realizan implantaciones orquestadas y hacen seguimiento de la versión implantada en cada nodo en cualquier estadio del proceso de compilación y entrega. También pueden gestionar las configuraciones de los entornos de todas las fases en las que se implantarán los componentes de la aplicación.

Medir la adopción de la tecnología

Las herramientas y tecnología de medición del rendimiento de la inversión son relativamente sencillas. Generalmente se mide la eficiencia creada por la automatización. Además, las herramientas automatizadas le permiten mejorar la escalabilidad y

fiabilidad de las tareas, algo que no siempre es posible con herramientas manuales. Por último, utilizar un conjunto integrado de herramientas automatizadas facilita la colaboración, trazabilidad y mejora de la calidad.

Las herramientas de automatización de la implantación gestionan los componentes del software que se implantan, los componentes y configuraciones del middleware que es preciso actualizar, los componentes de bases de datos que es preciso cambiar y los cambios de configuración en los entornos en los que se van a implantar estos componentes. Estas herramientas captan y automatizan los procesos para llevar a cabo estas implantaciones los cambios en la configuración. IBM UrbanCode Deploy es una herramienta de automatización de la implantación de este tipo.

Gestión de versiones

Orquestar los planes de versiones y las implantaciones asociadas con cada versión precisa coordinación entre los equipos de negocio, desarrollo, QA y operaciones. Las herramientas de gestión de versiones permiten a las organizaciones planificar y ejecutar versiones, ofrecen un portal único para la colaboración entre todos los participantes en una versión y ofrecen trazabilidad para la versión y sus componentes en todos los estadios del proceso de compilación y entrega. IBM UrbanCode Release ofrece estas capacidades de gestión de versiones.

Capítulo 4

Examinar cómo cloud acelera DevOps

En este capítulo

- ▶ Utilizar cloud como elemento facilitador de DevOps
- ▶ Comprender las implementaciones “full-stack”
- ▶ Examinar distintos modelos de servicio cloud
- ▶ Hacer realidad cloud híbrido

DevOps y cloud son elementos catalizadores y facilitadores mutuos. A medida que las organizaciones adoptan sistemas cloud, se evidencia la propuesta de valor de aprovechar estos sistemas cloud para alojar una carga de trabajo DevOps. La flexibilidad, resiliencia, agilidad y servicios que aporta una plataforma cloud permiten optimizar el proceso de aplicaciones alojadas en la nube. Es posible aprovisionar y configurar los entornos desde el desarrollo hasta las pruebas y producción, según se precise. Este proceso minimiza los cuellos de botella en el proceso de entrega debidos al entorno. Las organizaciones también buscan cómo aprovechar las plataformas cloud para reducir el coste de los entornos de desarrollo y pruebas o para ofrecer una experiencia de desarrollo moderna y optimizada para los profesionales de DevOps. En conjunto, ofrecen una justificación empresarial muy atractiva para la adopción de cloud con y para DevOps.

Este capítulo explora distintos modelos de cloud para DevOps y examina la justificación comercial de DevOps como carga de trabajo en un sistema cloud.

Utilizar cloud como facilitador de DevOps

El objetivo principal de DevOps es minimizar los cuellos de botella en el proceso de entrega para hacerlo más eficiente y Lean. Uno de los principales cuellos de botella que experimentan las organizaciones es en relación con la disponibilidad y configuración del entorno. No es extraño que los profesionales de DevOps, en especial los desarrolladores y responsables de pruebas, soliciten un entorno mediante un proceso formal de ticketing, que puede tardar días o incluso semanas.

Uno de los principios básicos de DevOps es desarrollar y hacer pruebas en un entorno similar al de producción. Un problema añadido a los cuellos de botella de disponibilidad del entorno es que el entorno disponible no se corresponda con el de producción. Esta falta de correspondencia puede ser tan simple como diferencias en la configuración del entorno (en el nivel del sistema operativo o del middleware) o tan complejo como que el SO o el middleware sean totalmente diferentes en el entorno de desarrollo y en el de producción.



La falta de disponibilidad de entornos tiene como resultado unos tiempos de espera potencialmente importantes para los profesionales de DevOps. Esta falta de correspondencia entre los entornos de desarrollo y producción puede introducir importantes problemas de calidad, ya que los desarrolladores no pueden verificar cómo se comportará la aplicación que están desarrollando en el entorno de producción, o ni siquiera si puede implementarse a producción mediante los procesos utilizados para la implementación en los entornos de prueba.

Los sistemas cloud abordan estos problemas de las siguientes maneras:

- ✓ La velocidad en la provisión del entorno en plataformas cloud puede proporcionar un sistema de autoservicio con disponibilidad y acceso al entorno bajo demanda para los profesionales de DevOps.
- ✓ La posibilidad de provisionar y desprovisionar dinámicamente estos entornos según convenga permite mejorar la gestión del entorno y reducir los costes al minimizar la necesidad de entornos estáticos permanentes.
- ✓ La posibilidad de utilizar tecnologías de “pattern” que permiten a las organizaciones definir y crear versiones de los entornos

como software posibilita la disponibilidad de entornos de provisionamiento que satisfagan las necesidades de los profesionales, y, sobre todo, que sean entornos similares a los de producción.

- ✓ Desde la perspectiva de la automatización, la disponibilidad de tecnologías de implementación de aplicaciones como IBM UrbanCode Deploy puede provisionar con una sola herramienta todo el entorno cloud e implementar las versiones adecuadas de las aplicaciones en estos entornos a medida que se precise. También pueden configurar rápidamente el entorno y la aplicación para satisfacer las necesidades de los profesionales.
- ✓ La disponibilidad de una tecnología de virtualización del servicio como IBM Rational Test Virtualization Server, funcionando en conjunción con entornos cloud, hace posible la simulación de servicios necesarios para las pruebas sin necesidad de provisionar instancias reales de los servicios.

La figura 4-1 muestra cómo funcionan los entornos de cloud en conjunción con las tecnologías de automatización de la implementación y virtualización del servicio para ofrecer entornos de desarrollo/prueba de extremo a extremo.

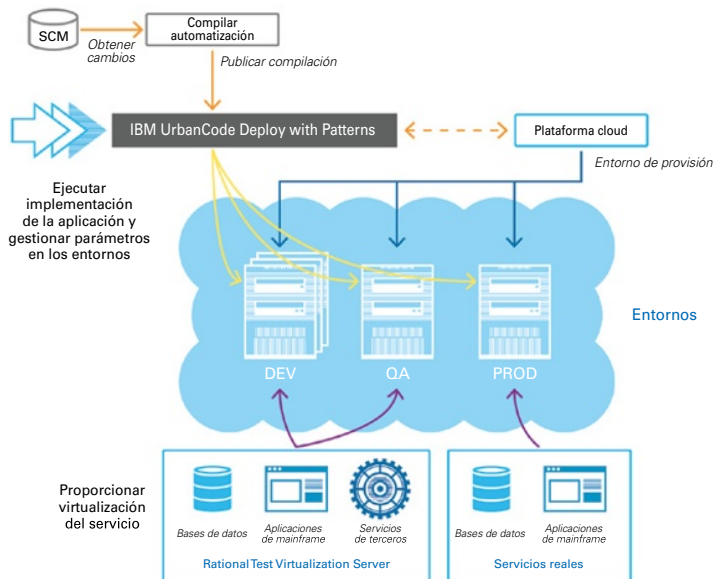


Figura 4-1: Desarrollo/prueba de extremo a extremo en cloud.



RECUERDA

Cloud sin DevOps significa no aprovechar todas las ventajas de cloud. Adoptar DevOps con entornos alojados en la nube ofrece las capacidades necesarias para ofrecer todos los beneficios de cloud a las organizaciones que ofrecen aplicaciones de software.

Implementaciones Full-Stack

Implementar una aplicación cloud se compone de implementar la aplicación y configurar el entorno cloud en el que se ejecuta. Estas dos tareas pueden realizarse por separado, pero la combinación de ambas recibe el nombre de *implementación full-stack*. A continuación examinamos estos dos enfoques más en detalle.

El primer enfoque es separar el provisionamiento del entorno cloud de la implementación de la aplicación. En este escenario no hay ningún punto único de orquestación de entornos cloud y la aplicación implementada en ellos. La herramienta de automatización de la implementación de la aplicación considera los entornos cloud como entornos estáticos. Este escenario no maximiza los beneficios de la implementación cloud.

El segundo enfoque es aprovechar la herramienta de automatización de la implementación como herramienta única de orquestación para el provisionamiento del entorno cloud y la implementación de aplicaciones en los entornos provisionados. Podemos conseguir esto creando “modelos” que capturan la definición y topología del entorno cloud y a continuación asignan los componentes y configuraciones de la aplicación a los nodos definidos en el entorno cloud.

Es posible utilizar múltiples tecnologías “pattern” como IBM Virtual System Patterns y plantillas de OpenStack HOT para definir los entornos cloud como plantillas. Herramientas de automatización de la implementación como IBM UrbanCode Deploy with Patterns pueden ofrecer provisionamiento full-stack utilizando estos modelos. Esto incluye provisionar el entorno cloud definido en el modelo e implementar la aplicación al entorno provisionado. Una vez provisionado el entorno, los cambios posteriores en la aplicación, configuración y contenido pueden implementarse de forma continua al entorno cloud como actualizaciones.

Alternativamente, las organizaciones pueden elegir tener siempre implementación full stack, en la que los entornos y las aplicaciones asociadas se provisionan siempre juntos como un mismo activo implementable. En este caso, no se hacen actualizaciones en el entorno existente.

Elegir un modelo de servicio cloud para DevOps

Al adoptar cloud, usted tiene que decidir primero el ámbito de responsabilidad que tiene previsto delegar en la plataforma cloud y qué responsabilidad quiere asumir usted. Hay dos modelos de servicio primarios para cloud: Infraestructura como servicio (IaaS) y Plataforma como servicio (PaaS).

IaaS

Cuando se adopta cloud en un modelo de servicio IaaS, la plataforma cloud gestiona la infraestructura subyacente y le ofrece las capacidades y servicios necesarios para gestionar toda la infraestructura virtualizada. La instalación, parches y gestión del SO, middleware, datos y aplicación se mantienen bajo responsabilidad del usuario.

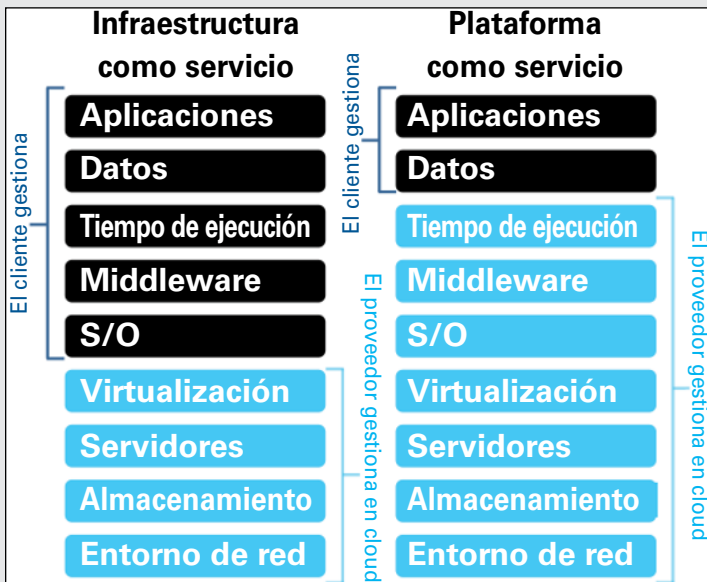
En el contexto de adopción de DevOps como carga de trabajo en cloud, la decisión de qué modelo de servicio cloud utilizar determina la forma de adoptar DevOps. Para un modelo de servicio IaaS, la organización del usuario es responsable de gestionar todo el proceso de entrega. Todas las herramientas e integraciones del proceso de entrega pasan a ser responsabilidad de la organización del usuario, incluida la adquisición del juego de herramientas adecuado y de garantizar su integración para formar el proceso de entrega. Además, deberán asegurarse de que la colaboración entre los equipos de desarrollo y operaciones siga una cultura DevOps. El que se utilice una plataforma cloud no cambia la necesidad de eliminar los silos de responsabilidades entre los desarrolladores que entregan el código y los equipos de operaciones que entregan la infraestructura, ahora en forma de servicio cloud.

Aunque la nube añade un enorme valor en términos de proporcionar IaaS a los equipos de entrega de la aplicación, siguen necesitando las capacidades DevOps adecuadas para entregar el valor deseado que aporta DevOps.

Separación de tareas

Una de las preguntas clave al hacer uso de DevOps en IaaS cloud es definir la separación de tareas entre la plataforma cloud y la herramienta de implementación de aplicaciones. ¿Qué herramienta es responsable de qué? Una forma sencilla de verlo es

desde la perspectiva de activos con movimiento lento y rápido en la pila de cloud. La figura de la barra lateral muestra las distintas capas de una pila de aplicación, desde el SO, almacenamiento y red subiendo hasta la capa de la aplicación.



Las capas de aplicación, datos y configuración del middleware son de movimiento rápido por naturaleza. Cambian con frecuencia porque la aplicación, sus datos y su uso son iterativos. Esta velocidad del cambio puede ser muy elevada para una aplicación aún en fase de desarrollo. Las capas inferiores incluyen el middleware (servidor de aplicaciones, base de datos, etc.), el SO y el almacenamiento, que no cambian con tanta frecuencia. Como actualizar y re-provisionar todas las capas para un

sencillo cambio que solo impacta en la aplicación, su contenido o su configuración no es eficiente, es lógico separar las tareas de estas capas con movimiento rápido o lento entre una herramienta de implementación de aplicaciones y una herramienta de administración de cloud. Las capas de movimiento rápido son gestionadas y automatizadas por la herramienta de implementación de aplicaciones y las capas con movimiento lento, por el software de administración de cloud proporcionado por la plataforma cloud.

PaaS

Al adoptar un modelo de cloud PaaS, su única responsabilidad como usuario pasa a ser la aplicación y los datos. Todas las restantes capacidades son ofrecidas por la plataforma y servicios cloud. El resultado es una gran mejora de la experiencia para los equipos de entrega de aplicaciones. Las herramientas de desarrollo y comprobación de aplicaciones quedan disponibles como servicios en la plataforma a la que pueden acceder los profesionales. La organización de entrega de aplicaciones ya no es responsable de gestionar el proceso de entrega. En su lugar, el proceso se integra en PaaS y permite a los profesionales centrarse exclusivamente en entregar rápidamente las aplicaciones. Las herramientas de desarrollo y comprobación y la provisión de infraestructura se abstraen los profesionales como servicios, lo que permite a los profesionales centrarse en sus tareas centrales de entregar aplicaciones.



IBM Bluemix es un PaaS. IBM y sus empresas colaboradoras gestionan la plataforma y los servicios ofrecidos en ella. La plataforma integra IBM DevOps Services, un conjunto de servicios que ofrecen todas las capacidades para que los equipos adopten DevOps y, concretamente, un proceso de entrega de aplicaciones como conjunto de servicios. Los equipos de entrega de aplicaciones pueden utilizar los servicios sin preocuparse de cómo se alojan los servicios y cómo les son entregados. Los servicios de DevOps incluyen:

- ✓ Entorno de desarrollo integrado (IDE) web como servicio
- ✓ Compilación como servicio
- ✓ Planificación y administración de tareas como servicio
- ✓ Exploración de seguridad como servicio
- ✓ Implementación como servicio
- ✓ Monitorización y análisis como servicio

La plataforma también ofrece entornos para tiempo de ejecución escalables para aplicaciones ejecutándose en distintos entornos durante el ciclo de vida de entrega, desde el desarrollo, comprobación y preparación, hasta la producción.

Comprender qué es un cloud híbrido

Cloud híbrido se ha convertido en un término muy habitual en el sector. Probablemente se usa en exceso para describir múltiples escenarios cloud en los que coexisten múltiples tecnologías cloud o en los que coexisten una infraestructura cloud y una física. Una forma sencilla de definir cloud híbrido es comenzar a examinar esta multitud de escenarios de cloud:

- ✓ **Cloud e infraestructura física:** Es un escenario cloud híbrido extremadamente habitual. Salvo en el caso de organizaciones nacidas en el mundo cloud, es en realidad el escenario por defecto. Cualquier organización dada tiene cargas de trabajo y aplicaciones que actualmente se ejecutan en su infraestructura física existente. En muchos casos, algunas de estas aplicaciones siguen ejecutándose en la infraestructura física. Son ejemplos típicos las aplicaciones de mainframe y aplicaciones de sistema de registro con uso intensivo de datos que no van a migrarse a cloud por limitaciones tecnológicas o de costes. Incluso si una organización migra todas sus cargas de trabajo a cloud, la migración no puede hacerse de un día para otro y habrá un periodo, potencialmente prolongado, en el que las infraestructuras física y cloud coexistan.
- ✓ **Cloud on-premise y off-premise:** En este escenario, una organización puede adoptar un cloud off-premise (público o virtual-privado) para algunas aplicaciones y cargas de trabajo y un cloud on-premise (privado) para otras. Un ejemplo sería una organización que aprovecha el cloud off-premise de bajo coste para entornos de desarrollo y cloud on-premise autogestionado en su propio centro de datos para todas las cargas de trabajo de producción.
- ✓ **IaaS y PaaS:** Este escenario incluye clientes que han adoptado un modelo de servicio cloud PaaS para algunas cargas de trabajo (nuevos sistemas o aplicaciones de participación, por ejemplo) e IaaS para cargas de trabajo de sistema de registro más tradicionales.

Para la adopción de DevOps, la existencia del cloud híbrido introduce nuevos retos, ya que tiene como consecuencia procesos de entrega de aplicaciones que abarcan entornos complejos de cloud híbrido y físicos. Algunos ejemplos de estos entornos de cloud híbrido son:

- ✓ Una organización puede decidir utilizar un cloud público para el desarrollo, comprobación y otros entornos no de producción, utilizando un cloud on-premise o incluso una infraestructura física para producción.
- ✓ Una organización puede haber algún tipo de aplicaciones de sistema de participación implementadas en un entorno cloud, mientras que las aplicaciones de sistema de registro que proporcionan servicios back-end para las aplicaciones de negocio centrales pueden residir en una infraestructura física, como un mainframe.
- ✓ Las organizaciones pueden recurrir a PaaS público para experimentar con aplicaciones innovadoras, y ponerlas en un cloud privado una vez que el experimento ha resultado satisfactorio.
- ✓ Las organizaciones pueden desear tener portabilidad de cargas de trabajo de aplicaciones en múltiples plataformas de cloud para asegurarse de que no exista dependencia de un proveedor o para proporcionar capacidad para implementar cargas de trabajo críticas entre múltiples proveedores de cloud.

El elemento central para la adopción de DevOps con un enfoque de cloud híbrido es la necesidad de implementar aplicaciones en estos entornos múltiples de cloud y físicos. Algunas aplicaciones como IBM UrbanCode Deploy with Patterns utilizan modelos de aplicaciones para asignar aplicaciones y configuraciones a múltiples entornos, físicos y cloud, y permitir la implementación automatizada de aplicaciones en entornos complejos de cloud híbrido.

Capítulo 5

Utilizar DevOps para resolver nuevos desafíos

En este capítulo

- ▶ Posibilitar aplicaciones móviles
- ▶ Manejar procesos ALM
- ▶ Escalado ágil
- ▶ Gestionar aplicaciones en múltiples niveles
- ▶ Examinar DevOps en la gran empresa
- ▶ Trabajar con cadenas de suministro
- ▶ Navegar el Internet de las cosas

DevOps se originó en empresas *nacidas en la web* (empresas que se originaron con Internet), como Etsy, Flickr y Netflix. A la vez que resuelven complejos retos tecnológicos a muy gran escala, estas empresas tenían arquitecturas más bien sencillas, a diferencia de las grandes empresas que crecieron en torno a sistemas heredados y/o mediante adquisiciones y fusiones, con complejos sistemas con distintas tecnologías obligados a funcionar conjuntamente. Estos retos se ven amplificados por las exigencias a que están sometidas las empresas modernas por nuevas tecnologías, como aplicaciones móviles y modelos de entrega de aplicaciones, como las cadenas de suministro de software.

Este capítulo explora algunos de estos retos a los que se enfrentan las empresas y a los que DevOps puede contribuir.

Aplicaciones móviles

En una empresa, las apps móviles no suelen ser apps independientes. Comparten muy poca lógica de negocio con el dispositivo móvil en sí y actúan más bien como front-ends para las múltiples aplicaciones empresariales ya en uso. Estas aplicaciones empresariales back-end pueden abarcar desde sistemas de procesamiento de transacciones hasta portales para empleados o sistemas de adquisición de clientes. El desarrollo y entrega móvil es complejo y precisa un conjunto de servicios dependientes que han de entregarse de forma coordinada, fiable y eficiente.

Para apps móviles empresariales, los ciclos de distribución y la publicación de nuevas características han de estar coordinados con los de las aplicaciones y servicios empresariales con los que interactúan las apps móviles. Por tanto, la adopción de DevOps deberá incluir a los equipos de apps móviles como ciudadanos y participantes de primera clase, junto con el resto de equipos de desarrollo de software de la empresa.

DevOps y los app stores

Un aspecto único de las apps móviles es que precisan implementarse en app stores. La mayoría de las apps móviles no pueden implementarse directamente en los dispositivos, sino que tienen que pasar por un app store gestionado por un proveedor. Apple introdujo este formato de distribución con su App Store (y bloqueó sus dispositivos para evitar la instalación directa de apps desde los desarrolladores o proveedores). Fabricantes de dispositivos como Research In Motion, Google y Microsoft, que en un primer momento permitieron la instalación directa de apps, ahora siguen el modelo de Apple.

Esta situación añade un paso asíncrono al proceso de implementación. Los desarrolladores ya no pueden implementar actualizaciones de una app bajo demanda. Incluso para correcciones de errores críticas, cada nueva versión de la app tiene que someterse al proceso de envío y revisión del app store. La entrega continua pasa a ser enviar y esperar. Pero la implementación continua hasta el desarrollo y comprobación sigue estando disponible, con el entorno de pruebas provisto por simuladores de los dispositivos a los que la aplicación se implementará o grupos de dispositivos físicos.



El ochenta por ciento de los datos corporativos de todo el mundo se originan en mainframes, y el 70 por ciento de todas las transacciones pasan por un mainframe. Desbloquear una ruta móvil para acceder a estas capacidades de los mainframes puede transformar su forma de hacer negocios y de implicar a los clientes, pero conseguirlo puede resultar complicado. Posiblemente usted tendrá que enfrentarse a lagunas en las capacidades, silos organizativos y múltiples plataformas, lo que se traduce en largos ciclos de distribución, retrasos innecesarios y recursos desperdiciados. Para proporcionar acceso móvil a las aplicaciones empresariales, las empresas están adoptando DevOps, un enfoque de entrega de software que se centra en la velocidad y la eficiencia, sin sacrificar la estabilidad ni la calidad.



Ningún concepto o principio específico de DevOps se aplica exclusivamente a las apps móviles. Sin embargo, las apps móviles amplifican la necesidad de DevOps por su inherentemente corto ciclo de desarrollo y rápido cambio.

Procesos ALM

La *gestión del ciclo de vida de las aplicaciones* (ALM) es un conjunto de procesos utilizados para gestionar la vida de una aplicación en su evolución desde idea (necesidad empresarial) hasta una que es aplicación implementada y posteriormente mantenida. Por tanto, si examinamos DevOps como una capacidad de negocio de extremo a extremo, ALM es el concepto fundamental que subyace al proceso de DevOps. DevOps amplía el ámbito de ALM para incluir en el proceso a los responsables de negocio, clientes y operaciones.

La ruta de adopción de Desarrollo/comprobación de DevOps (encontrará más información en el capítulo 2) es la que más en línea está con las capacidades tradicionales de ALM de gestión de requisitos, gestión del cambio, control de versiones trazabilidad y gestión de pruebas. Sin embargo, otras capacidades de ALM como el seguimiento y la planificación se producen como parte de la ruta de adopción Planificar, y la ruta de adopción Operar incluye paneles de control e informes.

Escalado ágil

El desarrollo Lean y Agile son los puntales del enfoque DevOps: uno de sus resultados es la reducción de desechos por parte de unos equipos más eficientes. La eficiencia y repetición de mejores prácticas conducen a ciclos de desarrollo más breves, lo que permite a

los equipos ser más innovadores y receptivos, con el consiguiente aumento del valor para el cliente. Escalar los principios Lean y Agile más allá del equipo de desarrollo a un equipo de equipos y a todo el ciclo de entrega del producto y del software es un elemento central para el enfoque DevOps.

Muchos equipos ya han adoptado el concepto de ágil y quieren escalar sus procesos actuales dentro de su proceso de adopción de DevOps. Están disponibles numerosos marcos que ayudan al escalado ágil. Entre ellos se incluye Scaled Agile Framework (SAFe) y Disciplined Agile Delivery (DAD). Algunas organizaciones ya han conseguido escalar eficazmente el proceso Scrum a equipos de muy gran tamaño. La finalidad de estos marcos es proporcionar una metodología para adoptar el concepto de ágil a nivel de empresa. Esto significa tener en consideración no solo el desarrollo de código, sino también la arquitectura, la financiación del proyecto y el gobierno de los procesos y roles requeridos por la dirección, aplicando los mismos principios Lean y Agile que funcionaron a nivel de equipo. Cualquiera que sea el marco utilizado para el escalado ágil, usted adopta los principios básicos y aplica las mejores prácticas para utilizarlos para impulsar la eficiencia y efectividad en todos los niveles de la empresa.

Aplicaciones en múltiples niveles

En la mayoría de grandes centros de IT no es extraño encontrar aplicaciones en múltiples niveles que abarcan muchas plataformas, cada una con su propio proceso de desarrollo, herramientas y requisitos en cuanto a destrezas. Estos sistemas en múltiples niveles suelen integrar aplicaciones web, de sobremesa y móviles en sistemas front-end y back-end, como aplicaciones en paquete, sistemas de almacén de datos, aplicaciones ejecutadas en mainframes y sistemas de gama media. Gestionar y coordinar las distribuciones de las distintas partes de estos sistemas, muchos de los cuales pueden encontrarse en plataformas distintas, puede sobrepasar las capacidades incluso de las más disciplinadas organizaciones de TI.



Un enfoque sensato es seguir procesos automatizados y uniformes de compilación, configuración e implementación en todos los estadios del desarrollo. Este enfoque garantiza que usted construye todas las piezas que necesita, y solo aquellas piezas que necesita. También garantiza que la aplicación se mantenga íntegra a medida que se sucedan los cambios y que el proceso avance en los ciclos de pruebas, QA y producción. IBM UrbanCode Deploy tiene un modelo de aplicación que ayuda a automatizar la compleja implementación de aplicaciones en múltiples niveles.

Mantener herramientas separadas para los distintos equipos dependiendo de la plataforma es una realidad en el actual mundo de múltiples plataformas y múltiples proveedores. Es aquí donde las plataformas abiertas como IBM Jazz pueden integrar herramientas dispares y ofrecer una solución unificada. Unas prácticas de implementación uniformes pueden ayudar a garantizar que los equipos utilicen una implementación fiable y repetible entre plataformas para proporcionar un auténtico valor empresarial.

DevOps en la empresa

La empresa actual depende de la velocidad con que IT pueda entregar el software. Estas empresas suele operar aplicaciones de sistemas de registro (creadas internamente o aplicaciones comerciales) implementadas en sistemas mainframe y de gama media. Se enfrentan a numerosos desafíos:

- ✓ Obstáculos legales
- ✓ Complejidad de los procesos
- ✓ Lagunas en las destrezas
- ✓ Silos organizativos
- ✓ Plataformas y herramientas que tienen como resultado prolongados ciclos de distribución, retrasos innecesarios y desperdicio de recursos

En el nivel empresarial, DevOps permite a los responsables de planificación, desarrollo, comprobación y operaciones entregar software de forma continuada en sus organizaciones. Actualmente las empresas implementan aplicaciones que son realmente multiplataforma, desde dispositivos móviles hasta mainframes. El enfoque de desarrollo DevOps utiliza principios Lean para crear un proceso de entrega eficiente y eficaz que permita desarrollar, probar y entregar las aplicaciones a la vez que contribuye a aumentar la calidad y velocidad y reducir los costes de desarrollo.



Dada la naturaleza auténticamente multiplataforma de las actuales empresas, con presencia de aplicaciones móviles, cloud, distribuidas y de mainframe, todas las cuales tienen que crearse, integrarse, implementarse y operarse, la necesidad de las eficiencias, optimización y colaboración que ofrece DevOps está convirtiéndose en un elemento diferenciador competitivo clave.

Cadenas de suministro

Con el aumento de la externalización y de las colaboraciones estratégicas para suministrar destrezas y capacidades a las empresas, las cadenas de suministro del software están convirtiéndose en la norma. Una *cadena de suministro* es un sistema de las organizaciones, personas, tecnología, actividades, información y recursos implicados en llevar un producto o servicio desde el proveedor hasta el cliente. Los distintos proveedores de la cadena pueden ser internos o externos.

En una organización que haya adoptado un modelo de cadena de suministro para entregar el software, adoptar DevOps puede constituir un reto porque las relaciones entre los proveedores se gestionan más por contratos y acuerdos de nivel de servicio que por la colaboración y comunicación. Pero este tipo de organizaciones también puede adoptar DevOps. Los equipos centrales de proyectos mantienen la propiedad de la planificación y capacidades de medición, con otras capacidades compartidas entre los restantes proveedores. En el proceso de entrega, distintos proveedores pueden ser responsables de distintas fases del proceso. Por tanto, es esencial utilizar conjuntos de herramientas comunes y un repositorio de activos común. Por ejemplo, una herramienta de gestión de elementos de trabajo ofrece informes sobre todos los elementos en los que trabajan todos los proveedores, así como transferencia de la responsabilidad de los elementos de trabajo entre todos los proveedores. Utilizar un repositorio de activos común proporciona un mecanismo para pasar los activos por el proceso y posibilitar la entrega continua.

El Internet de las cosas

El siguiente gran paso para DevOps es la evolución en el espacio de sistemas o dispositivos integrados en lo que se suele denominar *ingeniería continua*. Cuando nació Internet, la mayor parte de sus datos eran generados por humanos. En la actualidad, innumerables dispositivos conectados a Internet (como sensores y actuadores) generan mucho más tráfico que los humanos. Esta red de dispositivos interconectados en Internet suele denominarse *el Internet de las cosas*.

En este espacio, DevOps resulta potencialmente aún más esencial, debido a la codependencia del hardware y el software integrado que se ejecuta en él. Los principios de DevOps se reflejan en la ingeniería continua para garantizar que el software integrado entregado a los

dispositivos sea de alta calidad, con las especificaciones de ingeniería adecuadas.

“Operaciones” en ingeniería continua es sustituido por ingenieros de hardware o sistemas que diseñan y crean hardware a medida para los dispositivos. La colaboración entre los equipos de desarrollo y comprobación y los ingenieros de sistemas es esencial para garantizar que el hardware y el software se desarrollen y entreguen de forma coordinada a pesar de que el desarrollo del hardware y el software sigan ciclos de entrega diferentes. Las necesidades de desarrollo y pruebas para la entrega y comprobación continuas se mantienen iguales. Se utilizan simuladores para comprobar el software y el hardware durante el desarrollo.

Anti-patrones

En el mundo real siempre surgen limitaciones para la adopción de los principios de DevOps. Algunas de estas limitaciones son consecuencia del sector y el entorno en el que funciona una empresa, como la normativa legal, complejos sistemas de hardware o capacidades poco maduras para entrega de software. En estos casos, es preciso adoptar DevOps a la luz de *anti-patrones* (patrones ineficaces o contraproductivos) que pueden no ser aceptables para las necesidades de negocio de una organización.

Water-SCRUM-fall

La empresa global de estudios y asesoramiento Forrester (www.forrester.com) acuñó el término *Water-SCRUM-fall* para describir el estado actual de adopción de las metodologías ágiles de desarrollo de software. Desde la perspectiva de DevOps, esto significa que, aunque

los equipos de desarrollo puedan haber adoptado prácticas ágiles, los equipos de su entorno posiblemente sigan con procesos manuales en cascada que no permiten la entrega continua. En varias empresas, esta situación es el resultado de la cultura corporativa. Una empresa que adopta DevOps ha de integrar los procesos manuales en prácticas DevOps más amplias.

NoOps

En una organización NoOps se elimina Operaciones como departamento independiente, y sus responsabilidades se fusionan con las de Desarrollo. La empresa proveedora de televisión por Internet Netflix aboga por este método. NoOps puede funcionar bien para algunas organizaciones, pero es preciso un tiempo de espera para comprobar si el atractivo práctico de este modelo organizativo se generaliza.

Capítulo 6

Poner DevOps a trabajar: La historia de IBM

En este capítulo

- Comprender las mejores prácticas para ejecutivos
- Organizar su equipo
- Identificar los objetivos de DevOps
- Tomar nota de la transformación de DevOps
- Aprender de los resultados de DevOps

DevOps se está adoptando en IBM a nivel de empresa, y sigue evolucionando de forma regular. Esta adopción es consecuencia del éxito del uso de un enfoque DevOps del que fue pionero IBM Software Group (SWG) Rational y que ahora se utiliza en Watson, Tivoli, Global Business Services y otras divisiones. Este capítulo presenta el ejemplo práctico de la adopción de las capacidades de DevOps en IBM SWG por parte del equipo de producto IBM Rational Collaborative Lifecycle Management.



Este esfuerzo de entrega de software es único en cuanto que se desarrolla de forma *abierta*: el equipo de entrega de software entrega todos sus materiales y trabajos continuados, incluidos todos los elementos detallados de trabajo, en jazz.net. Este sitio web está abierto al público y cualquier usuario registrado puede ver los trabajos planificados, los trabajos en curso y el historial de trabajos de desarrollo realizados para los productos de software.

Examinar el rol del ejecutivo

La cultura es un hilo invisible de las organizaciones. Se basa en valores y comportamientos que evolucionan a partir de la dirección y los empleados. Muchas veces no se entiende la cultura de la organización hasta que se emprende un cambio importante. Habrá escépticos que preferirán esperar a ver qué pasa antes de decidir si se trata de una moda pasajera o no. Emergerán líderes. Es esencial establecer un enfoque para comprender esta dinámica y saber quién es quién y abordar los elementos inhibidores reales.



Para abordar la dinámica de la cultura, el ejecutivo de IBM SWG recurrió a distintos enfoques:

- ✓ **Elegir el líder adecuado:** El rol del líder es aunar los distintos puntos de vista y dirigir al equipo hacia un conjunto común de objetivos, elementos inhibidores, cambios en los procesos y decisiones sobre por dónde empezar.
- ✓ **Implicar a todos los interesados:** El apoyo de estos cambios debe provenir de la alta dirección, la gerencia y de las aportaciones individuales de las distintas disciplinas de desarrollo. Deberán estar implicados representantes de la empresa, arquitectos, desarrolladores, personal de pruebas y operaciones y deberá haber líderes reconocidos de cada área que abanderarán el cambio.
- ✓ **Medir las mejoras y los resultados:** Es fundamental tener una serie de parámetros clave que incorporen las eficiencias necesarias y los resultados empresariales. Estas metas y medidas deberán establecer un listón elevado y responsabilizar a las personas, pero no deben provocar desencanto.
- ✓ **Crear un impulso con los éxitos tempranos:** Comprender las ineficiencias y medir las mejoras en cada área crea un impulso favorable al cambio.
- ✓ **Comunicar y escuchar:** Como líder, es importante comprender la dinámica real de cómo el equipo está acogiendo el cambio. Dedicar un tiempo a mantener charlas con cada uno de los implicados y celebrar reuniones presenciales regulares con los equipos técnicos, la dirección y los líderes de las divisiones ayuda a conocer la implicación del equipo y sus puntos de vista sobre los elementos inhibidores, además de ofrecer una importante oportunidad para la dirección de compartir sus perspectivas en cuanto a las prioridades y los avances realizados.

Si usted ocupa un puesto ejecutivo, deberá apoyar a los equipos y ponerse a su disposición para comprender y eliminar los obstáculos. Es necesario operar como equipo con objetivos de negocio claros para que no haya divergencias.

Reunir el equipo

El equipo de producto IBM SWG Rational Collaborative Lifecycle Management forma parte de un grupo más amplio que desarrolla un conjunto de más de 80 herramientas de desarrollo de software en las categorías de planificación de entregas de software, desarrollo de software, implementación de aplicaciones, gestión de la calidad del software y monitorización y análisis de aplicaciones.

Este equipo de producto IBM SWG es una gran organización global con cuatro equipos de producto centrales que trabajan en más de 25 centros en 10 países. Antes de adoptar un enfoque DevOps, el grupo trabajaba con un programa anual de entregas que incluía plazos adicionales de entre tres y seis meses para determinar el contenido real de la entrega del año.

Establecer metas para DevOps

El equipo de IBM SWG tenía la impresión de que tardaban demasiado en responder a los cambios en el mercado y en la demanda de los clientes. El equipo decidió acortar el ciclo de entrega, no solo en las fases de desarrollo y pruebas sino también en la colaboración e interacciones con otras partes interesadas de la compañía y los clientes. Se estableció el objetivo de pasar de un programa anual de entregas a entregas trimestrales.

Además de acelerar su desarrollo para entregar nuevas capacidades con más frecuencia, el equipo tuvo que moverse con mayor rapidez para soportar modelos de entrega cloud, desarrollo móvil, comprobación móvil y otras capacidades, para hacer frente a los cambios tecnológicos. El equipo decidió adoptar los principios y prácticas de DevOps para transformar la forma de desarrollar software por parte del grupo para ofrecer valor a sus clientes desde un momento más temprano y con mayor frecuencia.

Realizar una modificación en este ámbito precisó un cambio cultural en la organización, para lo que se establecieron cuatro grupos de trabajo compuestos por miembros del equipo directivo y responsables técnicos. Estos grupos de trabajo examinaron los procesos de entrega del software de extremo a extremo y se responsabilizaron de cambiar las formas de trabajar. Se estableció un conjunto

concreto de medidas y planes de actuación para abordar los puntos clave en el proceso de desarrollo. Se creó un equipo para abanderar la entrega continua, con un evangelizador dedicado a educar a los equipos y compartir mejores prácticas en toda la organización.

El equipo de IBM SWG comenzó su viaje hacia la adopción de DevOps con la identificación de estos objetivos:

- ✓ Optimizar el proceso e introducir nuevas metodologías
- ✓ Utilizar las herramientas para conseguir consistencia, escalabilidad a otros equipos y trazabilidad y medición de los resultados
- ✓ Evolucionar a una cultura de la mejora continua

Aprender de la transformación a DevOps

Esta sección describe los pasos emprendidos por el equipo de IBM SWG para hacer posible la transformación a DevOps.

Ampliar las prácticas ágiles

Se ampliaron las prácticas Agile existentes más allá del desarrollo y pruebas para incluir a clientes, responsables de la empresa y operaciones, con el objetivo de eliminar el trabajo aislado y mejorar los resultados. Este modelo Agile más amplio permite a los equipos trabajar conjuntamente para producir software consistente y de elevada calidad, que aporte valor a la empresa mediante la utilización de un conjunto de procesos integrados en cada paso.

Se adoptó el enfoque de “un equipo”, en el que se combinó dirección de producto, diseño y desarrollo. El equipo de desarrollo incluyó los roles tradicionales de responsables de desarrollo y jefes de equipo, pero también incluyó a la dirección de operaciones y los arquitectos para apoyar una estrategia del ciclo de vida de extremo a extremo.

Se dedicaron recursos para los equipos de apoyo y orientación sobre entrega continua y Agile en toda la organización. Pasar a centrarnos en las capacidades en lugar de los componentes de los productos contribuyó a derribar las barreras tradicionales y a posibilitar la preparación para la entrega con cada automatización y cada sprint. Además, estos equipos se vieron fortalecidos mediante la asignación de directores de desarrollo dedicados. Se celebraron reuniones Scrum de forma regular a todos los niveles de

la organización de desarrollo para identificar y resolver problemas y eliminar obstáculos, comprobar los parámetros clave, utilizar datos de control en tiempo real y comunicar la información crítica.

Buscando hacer coincidir los cambios en el mercado con prioridades en el desarrollo, se formó un comité estratégico de producto formado por directores de producto, directores de desarrollo, arquitectos y responsables de la empresa. Entre sus responsabilidades se incluyeron:

- ✓ Asignar y garantizar financiación para el éxito de la ejecución del programa
- ✓ Impulsar, colaborar y apoyar la ejecución del programa
- ✓ Establecer una visión y dirección a largo plazo para la empresa
- ✓ Priorizar las necesidades de los usuarios para entregas anuales en línea con la visión a largo plazo

Aprovechar la automatización de las pruebas

Para eliminar los tradicionales largos ciclos de pruebas back-end y mejorar la calidad de las entregas, se adoptó un enfoque de comprobación continua utilizando automatización y virtualización. Se estableció un ritmo con iteraciones de cuatro semanas finalizadas con una demo e hitos de cuatro semanas finalizados con una entrega utilizable por el cliente. Las retrospectivas tras cada hito y la comprensión de la deuda técnica contribuyeron a eliminar desperdicios en futuras iteraciones. El lema del equipo de IBM SWG fue “hacer pruebas pronto y con frecuencia”.

El equipo adoptó las siguientes mejores prácticas para la automatización de las pruebas:

- ✓ Automatizar las pruebas repetitivas y con uso intensivo del personal.
- ✓ Automatizar en las áreas en las que se encuentran errores con más frecuencia.
- ✓ Ejecutar la automatización en cada compilación: pronto y con frecuencia.
- ✓ Crear una automatización que sea resistente a cambios en la interfaz de usuario (UI): utilizar un marco que separe la UI de las pruebas.

- ✓ Facilitar la creación, entrega y mantenimiento de la automatización estableciendo una fuerte responsabilidad del equipo de características.
- ✓ Planificar el trabajo de desarrollo de la automatización en sus estimaciones y garantizar que los desarrolladores dispongan de para trabajar en él.
- ✓ Desarrollar parámetros que le permitan evaluar la utilidad de la automatización (no se puede mejorar lo que no se puede medir).
- ✓ Reevaluar constantemente si su automatización descubre errores y rehacerla en caso negativo.

Para apoyar la automatización de las pruebas, el equipo implementó IBM Rational Test Workbench para pruebas funcionales y de rendimiento, y para posibilitar una mayor frecuencia de las pruebas fue fundamental automatizar la implementación de las compilaciones. El uso de IBM UrbanCode Deploy permitió al equipo reducir los costes de implementación de las pruebas en un 90 por ciento mediante implementación automatizada de compilaciones, incluyendo automatizar las aplicaciones necesarias y los parámetros de configuración del servidor de bases de datos.

Construir un proceso de entrega

El equipo de IBM SWG decidió crear un proceso de entrega que hiciera uso de “herramientas como servicio” y permitiera a los desarrolladores presentar el código, hacer pruebas e implementarlo en un entorno de producción en alrededor de 60 minutos, en lugar de dos o tres días con el sistema anterior. Este proceso redujo la necesidad de rehacer trabajo y maximizó la productividad.

En la implementación del equipo, se reconoce que un proceso de entrega continua tenía que adoptar las siguientes mejores prácticas:

- ✓ Trasladar la comprobación hacia la izquierda y automatizarla al máximo posible
- ✓ Utilizar siempre los mismos mecanismos de implementación
- ✓ Tratar de mantener un estado constante de preparación para la entrega
- ✓ Tratar la infraestructura como código

La Figura 6-1 presenta los productos y funciones provistas como parte del proceso de entrega continua adoptado por el equipo de IBM SWG.

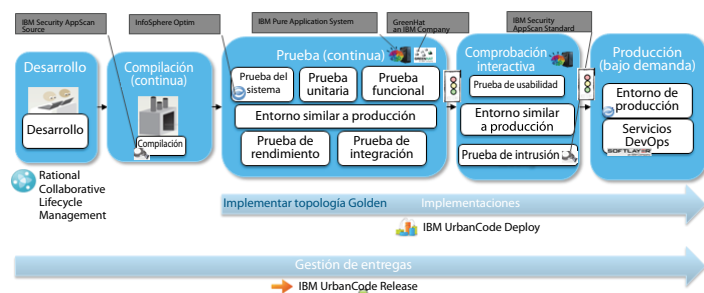


Figura 6-1: El proceso de entrega continua.



Una mejor práctica clave, esencial para implementar un proceso de entrega continua, es “tratar la infraestructura como código”. Esto significa que el desarrollador puede escribir scripts para configurar la infraestructura requerida para su aplicación como parte del código de la aplicación. En el pasado, esto era responsabilidad de un administrador de sistemas o empleado de operaciones, pero ahora el propio desarrollador puede hacerse cargo de este control y eficiencias. Puppet, Chef e IBM UrbanCode Deploy with Patterns son ejemplos de la nueva categoría de herramientas de automatización de infraestructuras que hacen que la infraestructura como código sea una realidad práctica.

En la actualidad, el equipo de IBM SWG trata la infraestructura como código y sigue las siguientes mejores prácticas:

- ✓ Tratar las definiciones de patrones, paquetes de scripts y servicios como código.
- ✓ Hacer versiones de todo.
- ✓ Automatizar la implementación de patrones de topología en cloud.
- ✓ Gestionar versiones de patrones en múltiples entornos cloud.
- ✓ Automatizar la comprobación de patrones.
- ✓ Hacer limpieza de los recursos del catálogo para evitar dispersión.

Experimentar rápidamente

El concepto de entrega continua no solo incluye actividades de desarrollo de software, como integración continua e implementación continua, sino también la actividad más fundamental del aprendizaje, cuya mejor manera de lograrse es mediante experimentos frecuentes y medición de los resultados.

Cuando se añaden características y funciones a una aplicación, nunca se tiene la certeza de que el cliente recibirá los beneficios esperados o previstos. Por este motivo es muy importante para IBM experimentar pronto y con frecuencia, recibir feedback de los clientes sobre las cosas que realmente les sirven y descartar las características que ofrezcan escasos beneficios o incluso puedan ser un obstáculo. Esta estrategia se ilustra en el dibujo de la Figura 6-2.

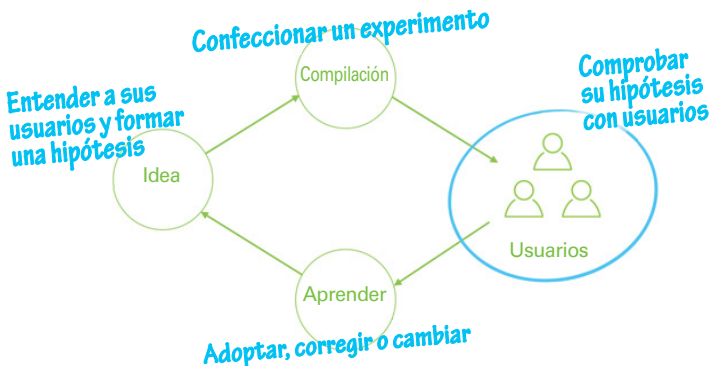


Figura 6-2: Una impresión del desarrollo impulsado por hipótesis.

El equipo de IBM SWG aprendió mucho acerca de la experimentación frecuente y desarrolló las siguientes mejores prácticas:

- ✓ Establecer parámetros y criterios del éxito/fracaso.
- ✓ Descubrir qué funciona mediante experimentos: pequeñas pruebas para un pequeño subconjunto de usuarios que ayudan a determinar la utilidad de una característica.
- ✓ Ejecutar múltiples experimentos continuamente.
- ✓ Tomar rápidamente decisiones basadas en hechos.
- ✓ Entregar más rápidamente para poder experimentar más rápidamente.

- ✓ Establecer un mecanismo que permita la experimentación a nivel de sistema (Google Analytics, IBM Digital Analytics, etc.).
- ✓ Considerar distintos modelos para experimentar (comprobación A/B clásica, máquinas de monedas múltiples, etc.).
- ✓ Seguir dos rutas simultáneamente para proyectos relacionados: experimentar en un proyecto cloud y utilizar los datos de los experimentos no solo para impulsar la dirección de este proyecto, sino también para proyectos on-premise relacionados.

Mejorar continuamente

El equipo de IBM SWG quería crear una cultura de la mejora continua y utilizar medidas de eficacia y eficiencia para asegurarse de estar mejorando realmente. Los equipos gestionan sus esfuerzos de mejora continua como un proyecto Agile. Apoyan la mejora continua haciendo seguimiento de las metas de madurez, puntos de dolor y acciones de mejora asociadas para abordar los problemas. Hacen seguimiento del trabajo en mejora continua como si se tratase de otros trabajos de desarrollo para asegurarse de que la inversión sea comprendida por todos. Los objetivos de madurez (por ejemplo, las capacidades) pueden ocupar uno o varios trimestres para desarrollarse y ser realmente adoptados. Los grandes puntos de dolor pueden tardar muchos meses en reducirse o eliminarse. En cualquier caso, las acciones concretas de mejora deberán adaptarse para ofrecer resultados en el plazo de un mes.

El equipo de IBM SWG utiliza retrospectivas para institucionalizar la mejora continua. Una *retrospectiva* es una revisión regular de lo que ha funcionado, lo que no ha funcionado tan bien y las actuaciones que es preciso emprender para mejorar. Si no está haciendo retrospectivas, implica un nivel de perfección en el desarrollo de software que aún no se ha alcanzado. En un equipo de gran tamaño puede haber una jerarquía de retrospectivas. Para el equipo de IBM SWG, cada equipo de componente hace una retrospectiva, y estas se utilizan como entradas para realizar retrospectivas a nivel de aplicación, que a su vez se utilizan como entradas para retrospectivas a nivel más elevado de solución. Las actuaciones de las retrospectivas se documentan como puntos de dolor con las correspondientes acciones de mejora para reducir o aliviar el dolor.

Y, para asegurarse de que los equipos mejoren, el equipo de IBM SWG estableció parámetros empresariales y operativos para medir la eficacia de la transformación a DevOps. Los parámetros empresariales se componen de mejoras medidas en términos de

- ✓ Mayor velocidad en la entrega
- ✓ Mejora de la satisfacción del cliente
- ✓ Reducción del gasto en mantenimiento y aumento de la inversión en innovación
- ✓ Mayor tasa de adopción por parte de los clientes

Los parámetros operativos influyen en la eficiencia del equipo a lo largo del tiempo y miden lo siguiente:

- ✓ Tiempo para iniciar un nuevo proyecto
- ✓ Tiempo de construcción
- ✓ Tiempo de comprobación de iteraciones

Examinar los resultados de DevOps

El enfoque DevOps ha ayudado al equipo de IBM SWG a obtener mejoras en satisfacción del cliente, tasa de adopción por parte de los clientes y un aumento de dos dígitos en los ingresos. La reducción de los plazos ha revitalizado a los equipos de entrega de IBM, lo que se traduce en una entrega rápida de soluciones actualizadas on-premise y nuevos servicios cloud como Bluemix, DevOp Services for Bluemix y Collaborative Lifecycle Management as a Managed Service (CLM aaMS).

Como ejemplo concreto del éxito del enfoque DevOps en IBM, la Figura 6-3 muestra los resultados medidos logrados por el equipo de producto IBM SWG Rational Collaborative Lifecycle Management.

| Medidas durante el ciclo de vida | 2008 | 2010 | 2012-2014 | Total de mejoras |
|----------------------------------|----------|----------|----------------------|------------------|
| Iniciación del proyecto | 30 días | 10 días | 2 días | 28 días |
| Groomed Backlog | 90 días | 45 días | En curso | 89 días |
| Tiempo total de desarrollo | 120 días | 55 días | 3 días | 117 días |
| Tiempo de compilación compuesto | 36 horas | 12 horas | 5 horas | 700 % |
| Disponibilidad BVT | N/D | 18 horas | <1 hora | 17 horas |
| Tiempo de prueba de iteración | 5 días | 2 días | 14 horas | 4 días |
| Tiempo total de implementación | 2 días | 8 horas | 4 horas-> 20 minutos | 2 días |
| Tiempo total hasta producción | 9 días | 3 días | 2 días | 7 días |
| Tiempo entre entregas | 12 meses | 12 meses | 3 meses | 9 meses |

Figura 6-3: Mejoras medidas del equipo de IBM SWG.

Capítulo 7

Diez mitos sobre DevOps

En este capítulo

- Comprender para qué sirve DevOps
- Saber para qué no sirve DevOps

El movimiento DevOps es joven y aún está emergiendo, sobre todo entre las grandes empresas. Como con cualquier nuevo movimiento o tendencia, han aparecido mitos y falacias. Algunos de estos mitos se originaron en empresas o proyectos que intentaron adoptar DevOps sin éxito. Sin embargo, lo que es cierto en una situación no necesariamente lo es en otras. A continuación veremos algunos mitos muy extendidos sobre DevOps, junto con los hechos reales.

DevOps solo vale para empresas “nacidas en la web”

Lo que solemos denominar DevOps se originó en empresas “nacidas en la web” (empresas que se originaron en Internet) como Etsy, Netflix y Flickr. Sin embargo, hay grandes empresas que llevan décadas utilizando principios y prácticas de entrega de software en línea con DevOps. Además, los actuales principios de DevOps, tal y como se describen en este libro, tiene un nivel de madurez que los hace aplicables a grandes empresas con múltiples plataformas tecnológicas y equipos distribuidos.

DevOps es hacer que Operaciones aprenda a programar

Los equipos de operaciones siempre han creado scripts para gestionar entornos y tareas repetitivas, pero con la evolución de la infraestructura como código, los equipos de operaciones vieron la necesidad de gestionar estas grandes cantidades de código con prácticas de ingeniería de software, como versiones del código, protección y desprotección, bifurcación y fusión. En la actualidad un miembro del equipo de operaciones crea una nueva versión de un entorno creando una nueva versión del código que lo define. Pero esto no significa que los equipos de operaciones tengan que aprender a programar en Java o C#. La mayoría de las tecnologías de infraestructura como código utilizan lenguajes como Ruby, que son relativamente fáciles de aprender, en especial para personas que tengan experiencia en scripts.

DevOps solo vale para Desarrollo y Operaciones

Aunque el nombre parece sugerir que proviene de la unión de desarrollo y operaciones, DevOps es para todo el equipo. Todos los participantes en la entrega de software (líneas de negocio, profesionales, ejecutivos, socios, proveedores, etc.) también tienen su lugar en DevOps.

DevOps no vale para ITIL

Algunas personas temen que las capacidades de DevOps, como la entrega continua, sean incompatibles con las comprobaciones y procesos prescritos por la Librería de infraestructuras de tecnologías de la información (ITIL), un conjunto de mejores prácticas documentadas para administración de servicios de IT. En realidad, el modelo de ciclo de vida de ITIL es compatible con DevOps. La mayoría de los principios definidos por ITIL están muy en línea con los de DevOps. Sin embargo, ITIL se ha ganado mala reputación en algunas organizaciones debido a que generalmente se ha implementado con procesos lentos en cascada que no permiten cambios y mejoras rápidos. Alinear este tipo de prácticas entre desarrollo y operaciones es la esencia de DevOps.

DevOps no vale para sectores regulados

Los sectores regulados tienen una fuerte necesidad de comprobaciones y balances y de aprobación por parte de los responsables de conformidad y auditabilidad. En realidad, adoptar DevOps mejora la conformidad, si se lleva a cabo adecuadamente. Automatizar el flujo de los procesos y utilizar herramientas que incorporen capacidad para incorporar pistas de auditoría.



Las organizaciones en sectores regulados siempre tendrán puertas o puntos comprobación manuales, pero estos elementos no son incompatibles con DevOps.

DevOps no vale para desarrollo externalizado

Los equipos externos deben considerarse como proveedores o suministradores de capacidades en el proceso de entrega DevOps. Sin embargo, las organizaciones han de asegurarse de que las prácticas y procesos de los equipos de proveedores sean compatibles con los de sus equipos de proyecto internos.



Usar planificación de distribuciones, gestión de elementos de trabajo y herramientas de repositorio de activos comunes mejora significativamente la comunicación y colaboración entre las líneas de negocio y los equipos de proveedores y proyectos, lo que hace posibles las prácticas DevOps. Utilizar herramientas de gestión de distribuciones de las aplicaciones puede mejorar en gran medida la capacidad de una organización para definir y coordinar el proceso completo de distribuciones entre todos los participantes.

Sin cloud no hay DevOps

Al pensar en DevOps, muchas veces pensamos en cloud por su capacidad para provisionar dinámicamente recursos de infraestructura para que desarrolladores y responsables de pruebas puedan obtener rápidamente entornos de prueba sin tener que esperar días o semanas hasta que se apruebe su petición. Sin embargo, no es necesario trabajar en cloud para adoptar prácticas DevOps, siempre

que la organización tenga procesos eficientes para obtener recursos para implementar y comprobar cambios en las aplicaciones.



La virtualización en sí es opcional. Es posible la entrega continuada a servidores físicos si los servidores pueden configurarse e implementarse a la velocidad necesaria.

DevOps no vale para sistemas grandes y complejos

Los sistemas complejos precisan la disciplina y colaboración que ofrece DevOps. Estos sistemas suelen tener múltiples componentes software y/o hardware, cada uno de los cuales tiene sus propios ciclos y plazos de entrega. DevOps facilita la coordinación de estos ciclos de entrega y la planificación de entregas a nivel de sistema.

DevOps es cuestión exclusivamente de comunicación

Algunos miembros de la comunidad de DevOps han acuñado términos humorísticos como *ChatOps* (los equipos realizan todas sus comunicaciones mediante herramientas de comunicación similares a Internet Relay Chat) y *HugOps* (DevOps es cuestión exclusiva de colaboración y comunicación). Estos términos surgen de la idea errónea de que la comunicación y la colaboración solucionan todos los problemas.



DevOps depende de la comunicación, pero las mejoras en la comunicación si van acompañadas de procesos poco optimizados, no conducen a mejores implementaciones.

DevOps significa Implementación del Cambio Continuo

Esta equivocada idea es fruto de las organizaciones que implementan solo aplicaciones web. Algunas de estas empresas dicen con orgullo en sus sitios web que realizan implementación a producción diariamente. Sin embargo, la implementación diaria no solo es poco práctica en grandes organizaciones que implementan aplicaciones complejas, sino que puede ser imposible por restricciones legales o empresariales. DevOps no se queda en la implementación, y, desde

luego, no se limita a implementar continuamente a producción. Adoptar DevOps permite a las organizaciones entregar a producción cuando lo deseen, no según una fecha concreta marcada en un calendario.

Notas

[illegible]

[illegible]

[illegible]

El mundo funciona con software

El mundo actual en constante cambio convierte a DevOps en el enfoque esencial para cualquier empresa que aspire a ser ágil y flexible. Para responder con rapidez a las cambiantes demandas de los clientes y el mercado, DevOps le ayuda a lograr innovación continua mediante el software. Este libro le ayudará a entender DevOps y a conocer cómo puede ofrecer beneficios comerciales reales para su organización. También descubrirá cómo una visión holística de DevOps que abarque todo el ciclo de vida de distribución del software, desde la fase de ideación y concepción de nuevas capacidades comerciales hasta la implementación en la producción, puede aportar una ventaja competitiva en un mundo de distribución continuada.

- *Supere las expectativas de los clientes: aumente la calidad de la experiencia ofrecida*
- *Satisfaga las necesidades de la empresa: de respuesta a cualquier aumento en la frecuencia y volumen de distribución del software*
- *Aproveche las nuevas tendencias de la tecnología: utilice tecnología móvil, cloud, big data y redes sociales*
- *Establezca un mejor sistema de colaboración: implique a las personas interesadas en todo el ciclo de vida de distribución del software*



En este libro encontrará:

- **Capacidades de DevOps**
- **Cómo trabajar con tecnologías de móvil, cloud y otras tecnologías avanzadas**
- **Cómo DevOps está en línea con los procesos de gestión del ciclo de vida (ALM)**
- **Cómo gestionar aplicaciones en múltiples niveles**
- **Experiencia de IBM en el uso de DevOps**

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.