



Parte 1: Introducción

CertiProf®
Professional Knowledge

www.certiprof.com

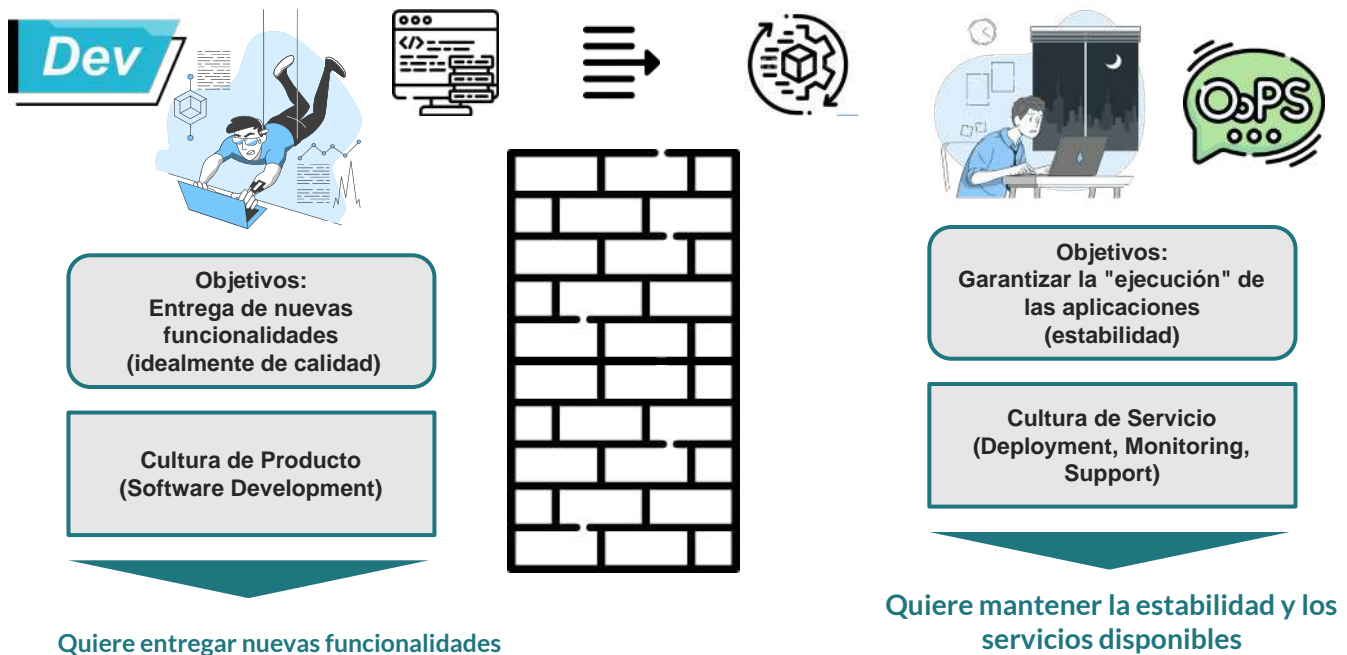
CERTIPROF® is a registered trademark of CertiProf, LLC in the United States and/or other countries.

Agenda

1. Surgimiento
2. Antecedentes
3. Conceptos Base de Devops
 - Definiciones base
 - Pilares y Principios de DevOps
 - Organización - Team Topologies
 - Integrando Ops en Dev



...Nació el Muro de la Confusión



Introducción



2007: Patrick Debois



2008: Andrew Shafer



2009: John Allspaw



2009: DevOpsDays,
Gante, Bélgica.
Patrick Debois

DevOps es un movimiento originado en 2007, cuando [Patrick Debois](#), un consultor freelance, estuvo analizando el sector IT. Su objetivo principal era ganar experiencia en todas las perspectivas en la cadena de valor IT. Tiene un [blog](#) bastante interesante.

En 2007 mientras Debois estaba en un proyecto de consultoría para el gobierno de Bélgica para la migración de un data center se frustra debido a los conflictos que se producían entre desarrolladores y administradores de sistemas. Debido a esto él decide proponer una solución.

En Agosto del 2008, en [Agile2008 conference](#), [Andrew Clay Shafer](#) y Patrick Debois se conocen. Shafer habló sobre la “infraestructura Agile” y sólo una persona atiende: Patrick Debois. Shafer, al ver que no había más que un asistente, piensa que no había interés en este tema salta esta sesión y Debois más tarde tiene con una larga conversación en el pasillo sobre el tema.

De esta y otras conversaciones surgieron todas las típicas frustraciones y conflictos que existen entre operaciones y desarrollo, así como sus motivos. Ambos crearon un grupo en Google denominado “[Agile System Administrators](#)” creado para continuar el debate.

En 2009 [John Allspaw](#) y [Paul Hammond](#) presentan “[10 deploy per Day](#)” en la O’Reilly Velocity 09 Conference. La principal premisa era enfocarse en asegurar que Dev y Ops trabajaban juntos y de forma crossfuncional a través de las herramientas y procesos ágiles.

Esto inspiró a Debois creando el evento “DevOpsDays” en Bélgica (DOD). Decide hacer exactamente lo que indica la charla y toma de aquí el concepto DevOps ([Dev and Ops Cooperation at Flickr](#)). La conferencia de puertas abiertas tiene lugar en Octubre de este año y se continua la discusión vía Twitter. El movimiento empieza a surgir.

Al mismo tiempo, la integración continua está en su momento de auge dentro del espectro Agile. Y había mucho movimiento en cuanto al continuous deployment sobre todo con la aparición del libro [“Continuous delivery”](#).

Mientras en paralelo la industria IT empiezan a tener fuerza otras metodologías como Operation Management, Lean e IT service management.

Este movimiento formado por debates, conferencias, twitter y lentamente tomó atención de la industria. IBM, CA Technologies, HP y BMC empiezan a aplicarlo.

En una entrevista de InfoQ en Abril del 2012, Debois admite que el nombre no fue intencional, que simplemente el título original era muy largo y escribió “DevOpsDay” para acortar. De aquí emergió el [nombre DevOps](#).

Fuente: <https://danielvillahermosa.wordpress.com/2019/11/27/la-evolucion-de-devops/>

1.2 Antecedentes

Varios movimientos importantes en gestión y tecnología convergen para preparar el escenario para el movimiento DevOps.

La aplicación de los principios del Lean al flujo de valor de la tecnología deriva en "Tres Formas":

1. Flujo
2. Feedback
3. Aprendizaje y Experimentación Continuos



DevOps es el resultado de la aplicación de los principios más confiables, desde el dominio de la manufactura, el liderazgo industrial hasta el flujo de valor de TI.

DevOps cuenta con varios conceptos provenientes de diferentes fuentes de conocimiento:

- ✓ Lean Manufacturing
- ✓ Teoría de las Restricciones
- ✓ Sistema de Producción de Toyota
- ✓ Ingeniería de resiliencia
- ✓ Movimiento de la Entrega Continua
- ✓ Organizaciones que aprenden
- ✓ Cultura de seguridad
- ✓ Factores humanos y muchos otros



1.3 Conceptos Base de Devops

Conceptos Base de Devops

- Entrega continua
- Infraestructura Ágil
- Lead time
- Kata
- Wip
- Deuda técnica



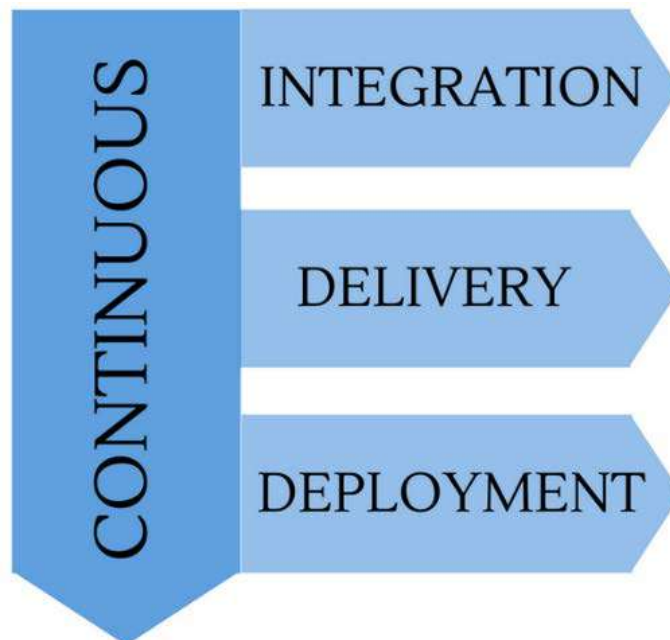
1.3 Conceptos Base: Entrega Continua

La **Entrega Continua** es una extensión de la integración continua para garantizar que se pueda liberar nuevos cambios para sus clientes rápidamente de forma sostenible.

¿Dónde está la diferencia entre Integración y Entrega Continua?

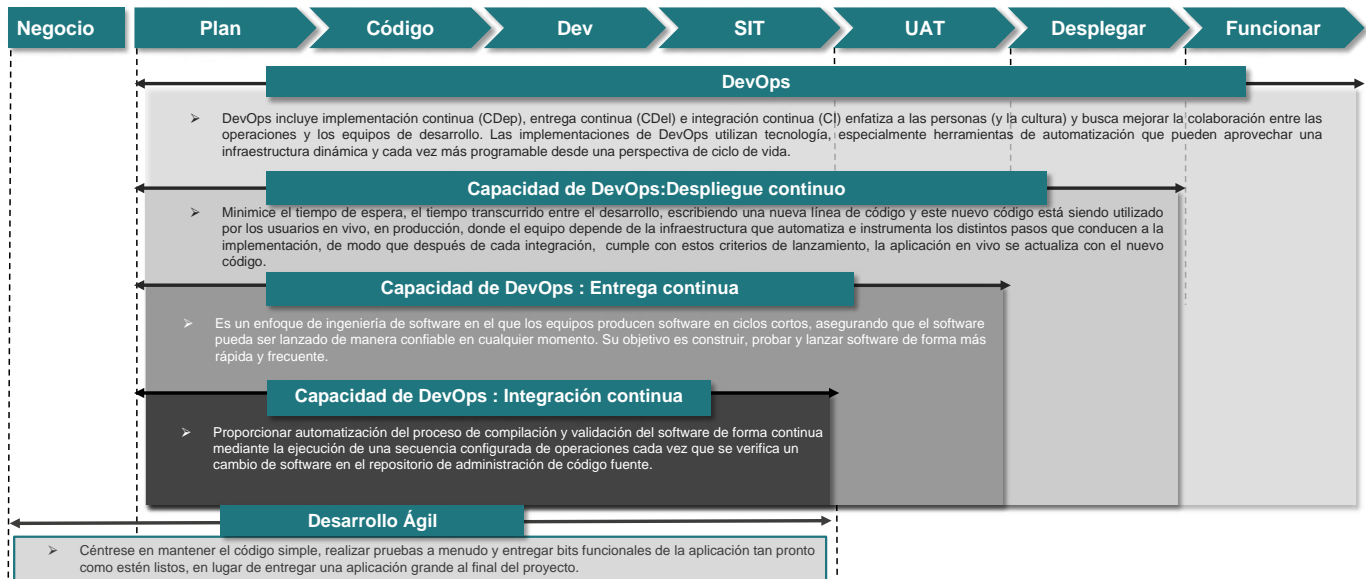
La **entrega continua** define la función de una "pipeline de despliegue" para garantizar que el código y la infraestructura estén siempre en un estado implantable y que todo el código registrado en la rama principal o master pueda ser implantado con seguridad en producción. La integración continua forma parte de la entrega continua y de el despliegue continuo.

El **despliegue continuo** es una extensión de la entrega continua, excepto por el hecho de que las liberaciones ocurren automáticamente en Producción.



DevOps

DevOps complementa a Agile para aumentar la eficiencia del modelo de entrega desde la gestión de relaciones comerciales hasta el producto final en producción.

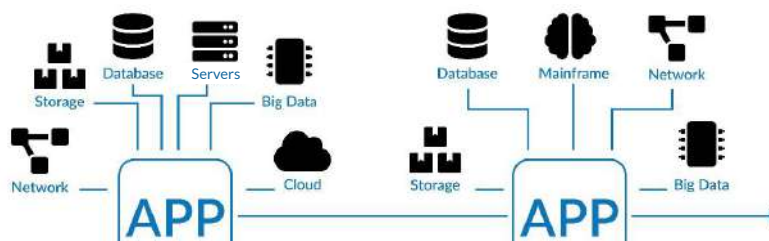


1.3.1 Conceptos Base: Infraestructura Ágil

La infraestructura como código es el enfoque para definir la infraestructura de computación y red, que se utiliza de técnicas de administración de código fuente, siendo tratado como cualquier sistema de software.

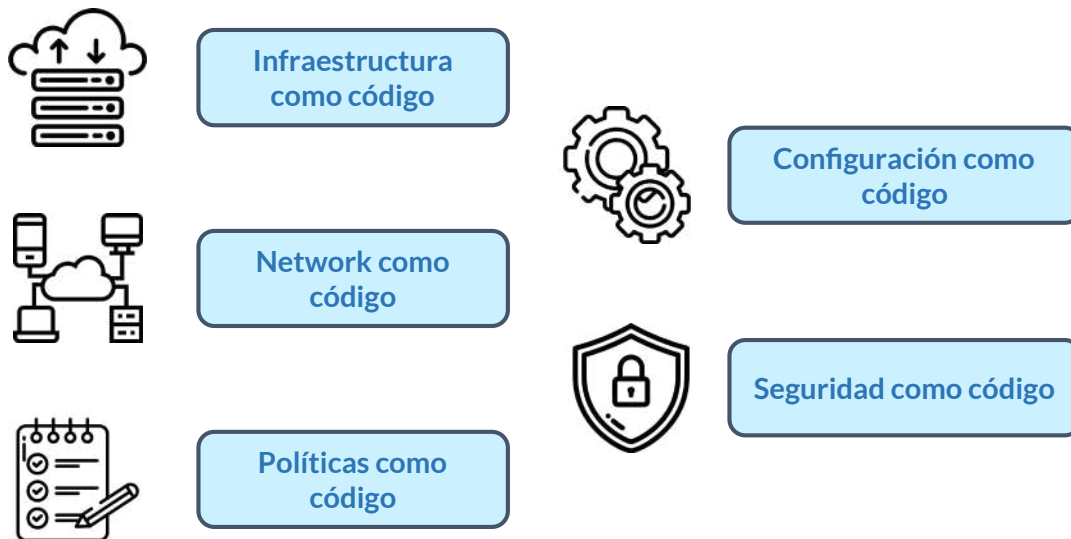
Este código puede ser mantenido en el sistema de control de versiones para permitir auditabilidad y construcción reproducible, sujeto a las prácticas de prueba y a la disciplina total de Entrega Continua. La infraestructura como código se basa en algunas prácticas:

- Utilizar archivos de definición
- Sistemas y procesos auto documentados
- Versión de todos los elementos
- Probar continuamente sistemas y procesos
- Pequeños cambios en lugar de grandes lotes
- Mantener los servicios disponibles continuamente



"Habilita la reconstrucción del negocio a partir de la nada, además de un repositorio de código fuente, copia de seguridad de datos de aplicaciones y recursos físicos crudos" - Adam Jacob, CTO de Chef.

X como código



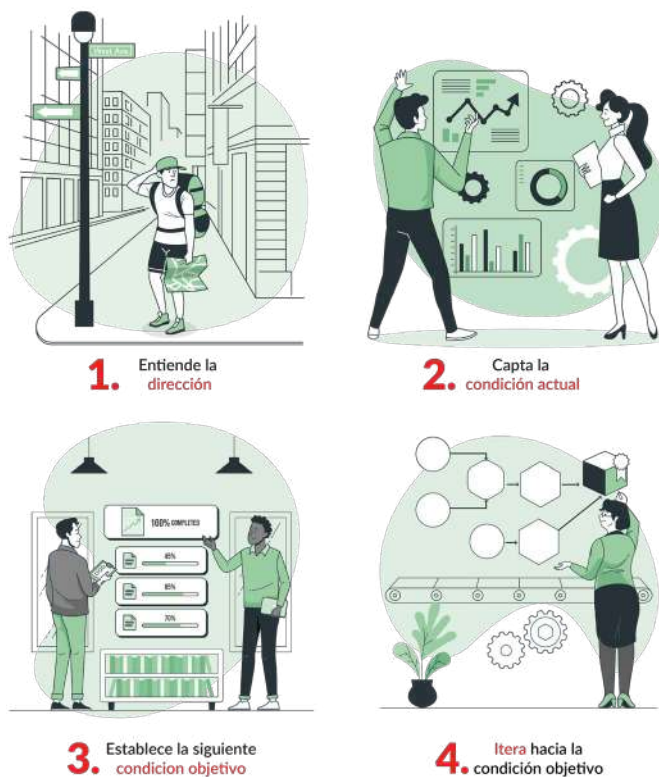
1.3.1 Conceptos Base: Kata

Los equipos generalmente no son capaces o no están dispuestos a mejorar los procesos en que operan. El resultado no es sólo que continúan sufriendo con sus problemas actuales, su sufrimiento también empeora con el tiempo.

En el Toyota Kata que, en ausencia de mejoras, los procesos no permanecen iguales, debido al caos y a la entropía, los procesos realmente se degradan con el tiempo.

Descomponga cada secuencia de trabajo en etapas específicas, automatícelas y tráigalas de forma estandarizada, que puede usarse "reflexivamente", cuando sea necesario.

Se refiere a la forma o estándar que se puede practicar para desarrollar habilidades personales y la mentalidad.



Kata son rutinas de enseñanza usadas para preservar y pasar el know-how.

1.3.1 Conceptos Base: Work in Progress (WIP)

WIP: Trabajo que ya está en el proceso de desarrollo, pero todavía no está terminado y disponible para un cliente o un usuario.

Se refiere a todos los activos o elementos de un producto o servicio que actualmente se están trabajando o esperando para ser trabajados.

Podemos limitar la multitarea cuando usamos un Tablero Kanban para gestionar nuestro trabajo, por ejemplo, codificando e imponiendo límites de WIP (work in progress) para cada columna o centro de trabajo. Podemos colocar un límite superior en el número de tarjetas que pueden estar en una columna.



Trabajo en proceso o progreso, que incluye el conjunto de grandes cantidades de elementos inacabados para los productos en un proceso de producción.

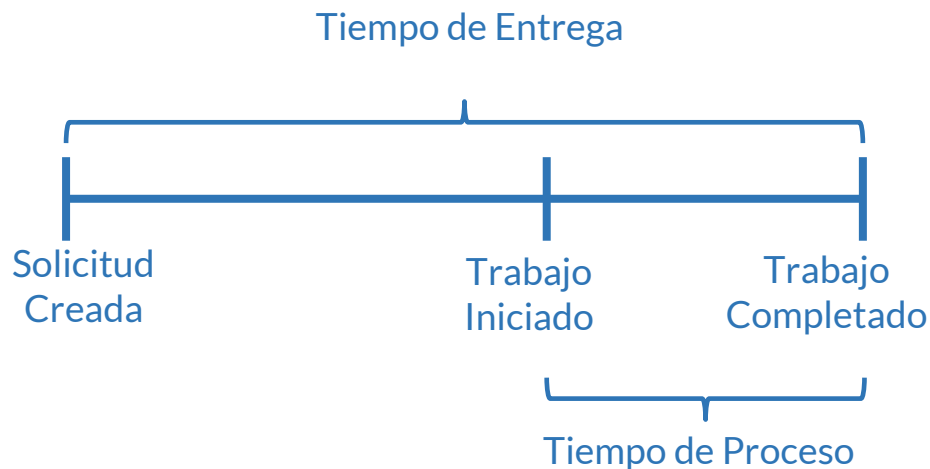
1.3.1 Conceptos Base: Lead Time

El **Tiempo de Entrega** (lead time) es una de las medidas comúnmente usadas para medir el rendimiento en los flujos de valor (Lean).

La proporción de tiempo de proceso vs el tiempo de entrega, sirve como una medida importante de eficiencia

La medición de tiempo de entrega comienzan cuando la solicitud es creada y termina cuando es entregada al cliente.

La medición del tiempo de proceso sólo comienza cuando empezamos a trabajar en la solicitud del cliente - específicamente, omite el tiempo que el trabajo está en la cola, esperando ser procesado.



Tiempo de Entrega vs. tiempo de proceso de una operación de implementación

Fuente: Manual DevOps, capítulo 1

1.3.1 Conceptos Base: Deuda Técnica

El costo implícito del retrabajo adicional causado por la elección de una solución fácil ahora en vez de usar un enfoque mejor que llevaría más tiempo.

La deuda técnica describe cómo las decisiones que tomamos conducen a problemas que se vuelven cada vez más difíciles de arreglar con el tiempo, reduciendo continuamente nuestras opciones disponibles en el futuro, incluso cuando se toman con prudencia, todavía generamos intereses sobre esa deuda.



Fuente: Vincent Dénier on Twitter: <https://www.pinterest.es/pin/798192733957711009/?d=t&mt=login>

- Definición inicial insuficiente
- Presiones de negocios
- Falta de proceso o comprensión
- Componentes fuertemente acoplados
- La falta de un conjunto de pruebas
- Falta de documentación
- Falta de colaboración
- El desarrollo paralelo
- Refactorización retrasada
- Falta de alineación con los estándares
- Falta de conocimiento
- Falta de propiedad
- Liderazgo tecnológico deficiente
- Cambios de especificación a la última hora

El costo implícito del retrabajo adicional causado por la elección de una solución fácil ahora en vez de usar un enfoque mejor que llevaría más tiempo.

1.3.1 Conceptos Base: Bimodal SoR y SoC

Sistemas de Registros

Los sistemas corporativos que giran nuestros negocios, donde la corrección de las transacciones y de los datos son primordiales

- Cambios con ritmo más lento
- Hacer correctamente

Sistemas de Contratación

Son sistemas orientados al cliente o utilizados por los empleados, sistemas de comercio y aplicaciones de productividad.

- Ritmo de cambios más rápidos
- Hacer rápido

Gartner recientemente popularizó la noción de TI bimodal, refiriéndose al amplio espectro de servicios que las empresas típicas soportan.

SoR

Los sistemas de registro típicamente tienen un ritmo de cambio más lento y muchas veces tienen requisitos de conformidad y reglamentación (por ejemplo, SOX).

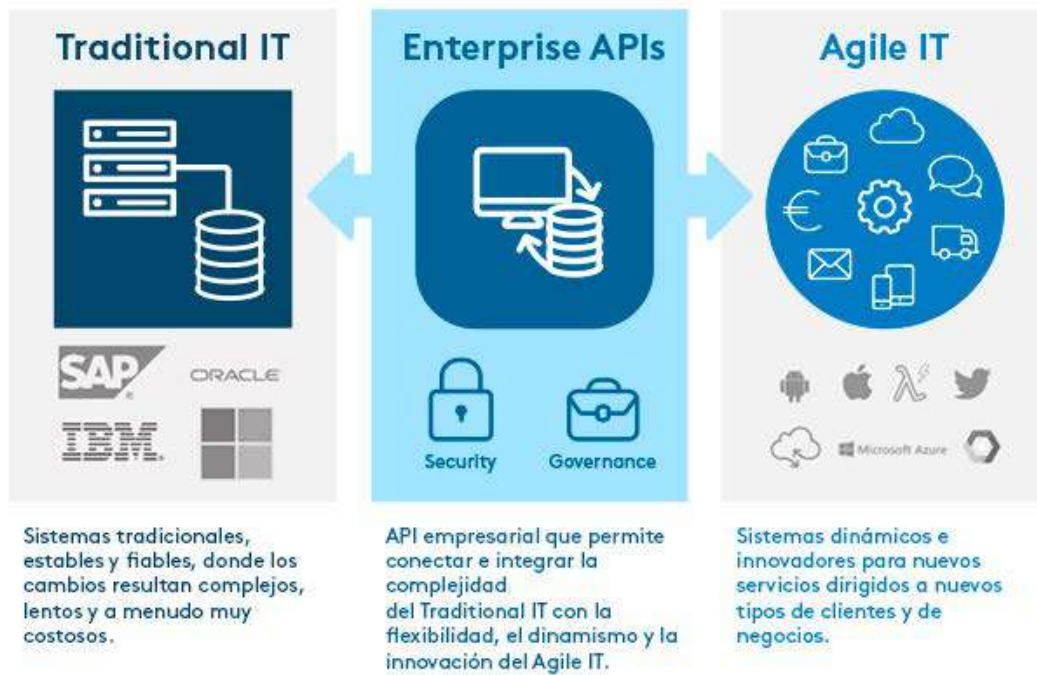
Gartner reúne estos tipos de sistemas Tipo 1, donde la organización se concentra en "hacer correctamente".

SoC

El sistema de contratación generalmente tiene un ritmo de cambio mucho mayor para soportar ciclos de feedback rápidos que le permiten realizar experimentos para descubrir cómo satisfacer mejor las necesidades del cliente.

Gartner llama a estos tipos de sistemas Tipo 2, "donde la organización se concentra en hacer rápidamente".

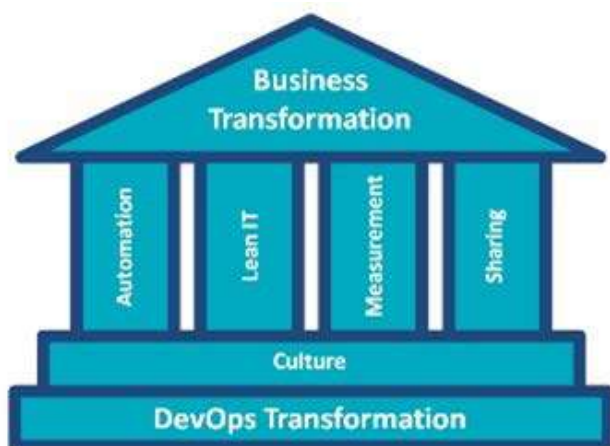
Para complementar ambos entornos tecnológicos y alcanzar la **eficiencia del Bimodal IT** en el que conviven muchas empresas, apostamos por arquitecturas que permitan la integración de datos y la integración de aplicaciones de forma adecuada, y recomienda soluciones basadas en Enterprise APIs que permiten conectar e integrar la complejidad del Traditional IT con la flexibilidad, el dinamismo y la innovación del Agile IT.



Fuente: <https://www.ithinkupc.com/es/blog/como-conseguir-la-eficiencia-en-un-entorno-bimodal-it>

1.3.2 Pilares de DevOps

Mantenga la CALMa(S)



Culture

Automation

Lean

Measurement

Sharing



Cultura

1. Sin un cambio cultural, la transformación no va a funcionar
2. Más que una metodología, es una colección de principios para romper los silos

Automatización

1. Automatización ayuda pero no es una bala de plata
2. Puede habilitar la mejora en la calidad y la entrega continua a través de la integración continua

Lean

1. Ayuda a reducir los desperdicios en el proceso
2. Ayuda a poner foco en el objetivo final que quieras lograr desde el punto de vista del negocio

Métricas

1. Debes Medir todo - repetidamente
2. Debes adaptarte y planificar áreas de mejor basado en tu lectura de esas métricas

Sharing (Colaboración)

1. Uno de los principales aspectos de DevOps
2. Ayuda a romper los silos
3. Un habilitador fantástico para ampliar los ciclos de retroalimentación

1.3.3 Organización

Arquetipos Organizativos

Hay tres tipos principales de Estructuras Organizacionales que informan sobre cómo proyectamos nuestros flujos de valor DevOps con la Ley Conway en mente:

Orientación Funcional	Orientación Matricial	Orientación del Mercado
<ul style="list-style-type: none"> Las organizaciones orientadas a funciones optimizan la experiencia, la división del trabajo o reduciendo el costo. 	<ul style="list-style-type: none"> Las organizaciones orientadas en matrices combinan orientación funcional y de mercado. 	<ul style="list-style-type: none"> Las organizaciones orientadas al mercado optimizan para responder rápidamente a las necesidades del cliente.

Regla del “2 pizzas Team”

- Garantiza que el equipo tenga una comprensión clara y compartida del sistema en el que están trabajando
- Limita la tasa de crecimiento del producto o servicio en el que trabajó
- Descentraliza el poder y permite la autonomía
- Liderar un 2PT es una manera para que los empleados ganen alguna experiencia de liderazgo en un entorno donde el fracaso no tiene consecuencias catastróficas

Ley de Conway

Melvin Conway observó que la forma en que las organizaciones estaban estructuradas tenía un fuerte impacto en los sistemas que creaban.

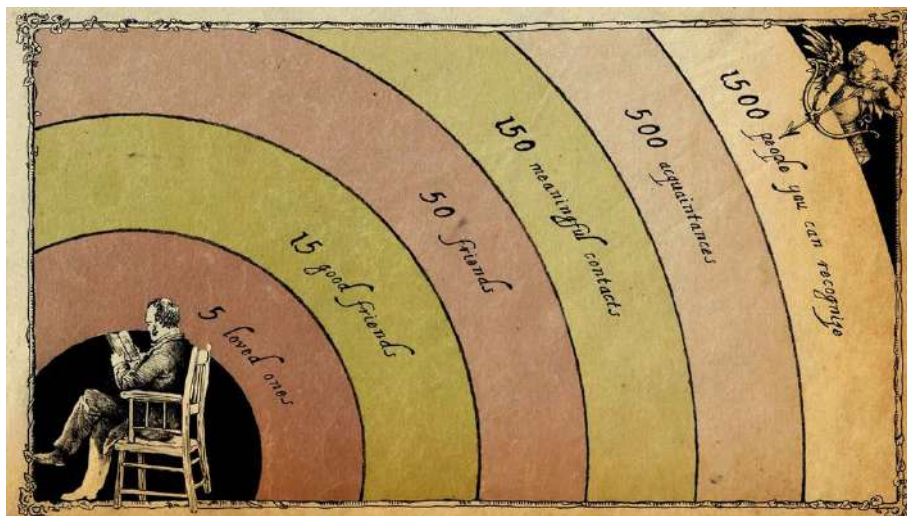
Estas observaciones llevaron a lo que ahora se conoce como Ley de Conway, que afirma que "las organizaciones que proyectan sistemas... son constreñidas para producir dibujos que son copias de las estructuras de comunicación de esas organizaciones... Cuanto mayor es la organización, menor es la flexibilidad y más pronunciado el fenómeno".



"Cualquier empresa que proyecta un sistema (definición más amplia aquí que sólo sistemas de información), inevitablemente produce un proyecto cuya estructura es una copia de la estructura de comunicación de la organización".

El número de Dunbar

"Según el antropólogo británico Robin Dunbar, su teoría de que los humanos solo pueden mantener 150 amistades ha resistido 30 años de escrutinio. Dunbar se convenció de que había una relación entre el tamaño del cerebro y el tamaño de los grupos a través de sus estudios de primates no humanos. Esta proporción se trazó utilizando neuroimágenes y observación del tiempo dedicado al aseo, un comportamiento social importante de los primates. Dunbar concluyó que el tamaño, en relación con el cuerpo, del neocórtex, la parte del cerebro asociada con la cognición y el lenguaje, está relacionado con el tamaño de un grupo social cohesionado. Esta relación limita la complejidad que puede manejar un sistema social".

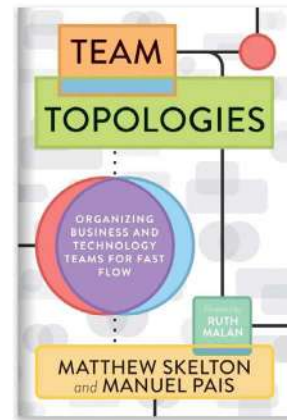


El círculo más íntimo son solo cinco seres queridos, llegando a un máximo de 1500 personas que puedes reconocer (Crédito: Emmanuel Lafont).

Team Topologies

Team Topologies – Organizing business and technology teams for fast flow

Autores:
Matthew Skelton
Manuel Pais



1.3.3 Organización

Cuatro topologías fundamentales

A continuación detallaremos las cuatro topologías fundamentales del equipo, incluido el comportamiento y las capacidades esperados. Estas son:

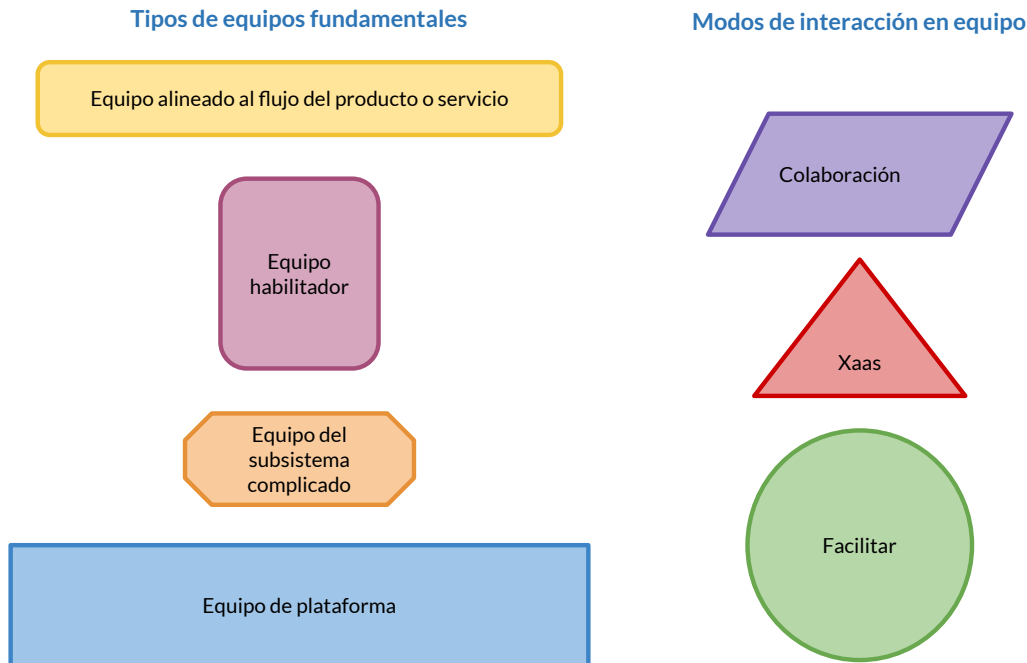
- Equipo alineado al flujo del producto o servicio
 - Equipo de plataforma
 - Equipo habilitador
 - Equipo de subsistema complicado
- **Equipo alineado al flujo del producto o servicio:** un equipo alineado con el flujo principal del producto o servicio, con una combinación de habilidades multifuncionales y la capacidad de ofrecer incrementos significativos sin esperar a otro equipo (algunos llamarían a estos equipos "equipos de productos o funciones" pero hablar de flujos tiene más sentido)
 - **Equipo de la plataforma:** un equipo que trabaja en la plataforma subyacente y da soporte a los equipos alineados con el flujo en la entrega. La plataforma simplifica la tecnología que de otro modo sería compleja y reduce la carga cognitiva para los equipos que la utilizan (una buena plataforma es "lo suficientemente grande")
 - **Equipo habilitador:** un equipo que ayuda a otros equipos a adoptar y modificar software como parte de un período de transición o aprendizaje
 - **Equipo de subsistemas complicados:** un equipo con un mandato especial para un subsistema que es demasiado complicado para ser tratado por un equipo normal alineado con el flujo o un equipo de plataforma. Opcional y solo se usa cuando es realmente necesario



Tres modos de interacción

Modos de interacción en equipo. Los modos de interacción principales para las 4 topologías de equipo fundamentales son:

- Colaboración: trabajar en estrecha colaboración con otro equipo
- X-as-a Service: consumir o proporcionar algo con una colaboración mínima
- Facilitar: ayudar (o ser ayudado por) otro equipo para eliminar los impedimentos



1.3.3 Organización: Equipos Auto Gestionados Forma I, Forma T y Forma E

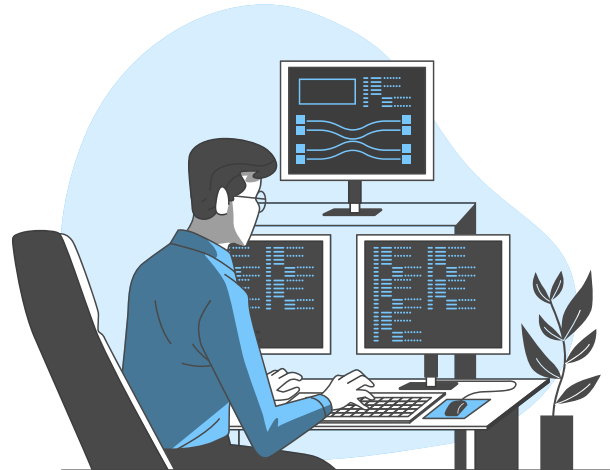
Cuando los departamentos se especializan demasiado, causan trabajos orientados en silos. Cualquier actividad operativa compleja, a continuación, requiere múltiples transferencias y fugas entre las diferentes áreas de la infraestructura, llevando a plazos más largos.

A través de la capacitación cruzada y de las crecientes habilidades de ingeniería, los generalistas pueden tener una mayor magnitud de órdenes de trabajo que sus contrapartes especializadas, y también mejora el flujo general de trabajo, removiendo colas y tiempo de espera.

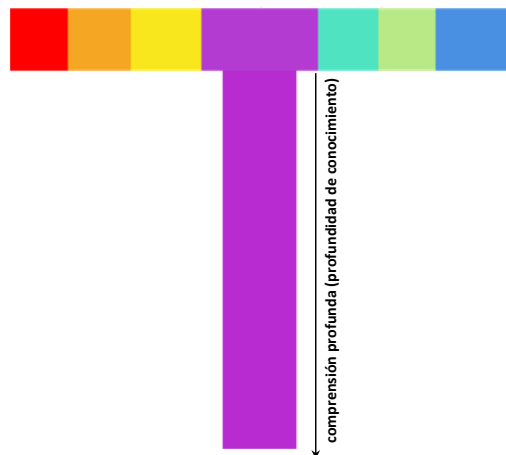


Forma I: Expertos

- Especialización profunda en un área
- Pocas habilidades o experiencia en otras áreas
- Crea cuellos de botellas rápidamente
- Insensible a los residuos y al impacto en el flujo
- Evita la planificación de flexibilidad o absorción de variabilidad



Una amplia gama de conocimientos básicos a través de muchas habilidades superpuestas



Forma T: Generalistas

- Especialización profunda en un área
- Excelentes habilidades en muchas áreas
- Puede intensificar para quitar los cuellos de botella
- Sensible a los residuos y al impacto del flujo
- Ayuda a tornar la planificación flexible y absorbe la variabilidad

Forma E

- Especialización profunda en pocas áreas
- Experiencia cruzada en varias áreas
- Presentan habilidades de ejecución
- Siempre está innovando
- Potencial casi sin límites

Cuanto mayor sea el conocimiento general y las habilidades especiales más necesarias dentro de una persona, mejor.



1.3.4 Integrando Ops en Dev

Integrando Ops en Dev

El objetivo es tener resultados orientados al mercado, donde muchos pequeños equipos pueden proporcionar un valor para el cliente de manera rápida e inmediata.

Podemos crear resultados más orientados al mercado, integrando mejor las capacidades Ops en equipos Dev, haciéndolas más eficientes y productivas.

Ops puede mejorar significativamente la productividad de los equipos de Dev en toda la organización, además de permitir una mayor colaboración y resultados organizativos.

Podemos usar tres estrategias:

- Creando habilidades de autoservicio que permita su activación por los desarrolladores
- Incorporando a los ingenieros de Ops en los equipos de servicio
- Desarrollando vínculos entre Ops y los equipos de servicio



Cree capacidades de autoservicio para uso de los desarrolladores

Habilitamos a los equipos de Dev a pasar más tiempo creando funcionalidades para sus clientes... Al hacer esto, habilitamos a los equipos de productos a obtener lo que necesitan, cuando lo necesitan, además de reducir la necesidad de comunicación y coordinación.

Una manera de proporcionar resultados orientados al mercado es que las Operaciones creen un conjunto de plataformas centralizadas y herramientas de servicios, las que cualquier equipo de Dev pueda usar para volverse más productivo:

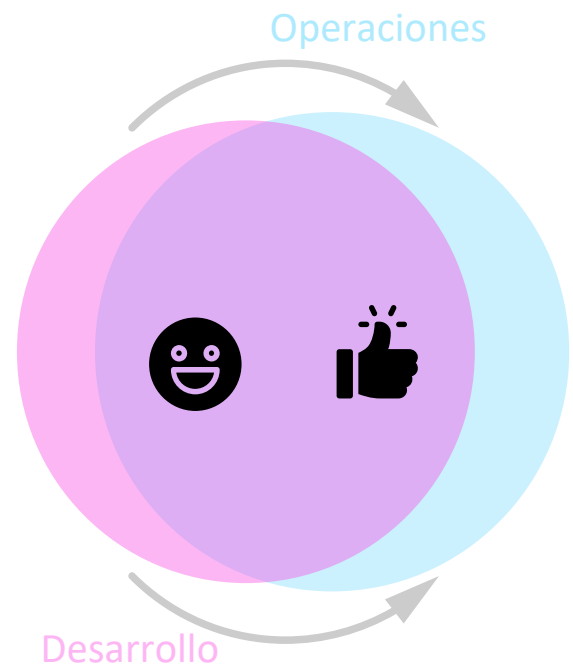
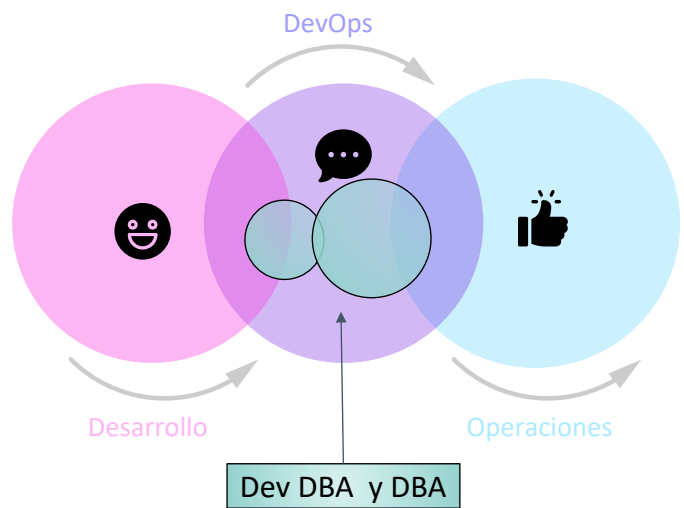
- Obtener entornos de producción
- Pipelines de despliegue
- Herramientas de prueba automatizadas
- Paneles de producción de telemetría
- Entre otros

Incorpore los ingenieros de Ops en los equipos de servicio

Habilitando equipos de productos para llegar a ser más autosuficientes incorporando ingenieros de operaciones dentro de ellos, reduciendo así su dependencia de operaciones centralizadas.

Las prioridades de los ingenieros de operaciones se dirigen casi enteramente a los objetivos de los equipos de productos, cuando se integran en los equipos del Dev.

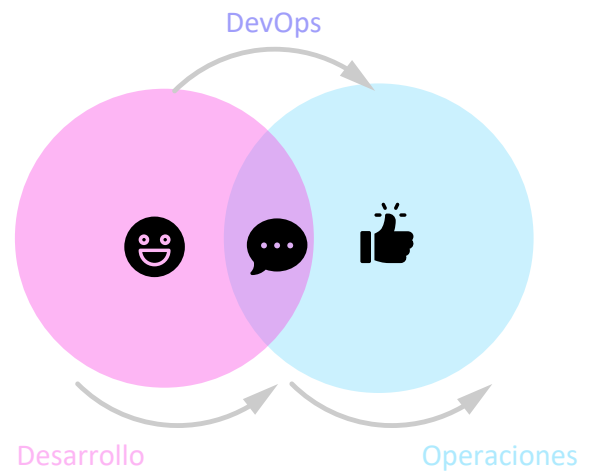
Como resultado, los ingenieros de Ops están más conectados a sus clientes internos y externos.



<https://web.devopstopologies.com/>

Desarrolle conexiones entre Ops y otros equipos cuando la incorporación de Ops no sea posible

Por una variedad de razones, como costo y falta de recursos, tal vez no podamos integrar ingenieros OPS en cada equipo de productos.



<https://web.devopstopologies.com/>