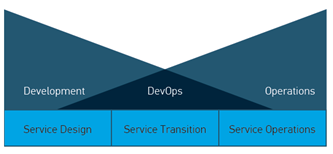
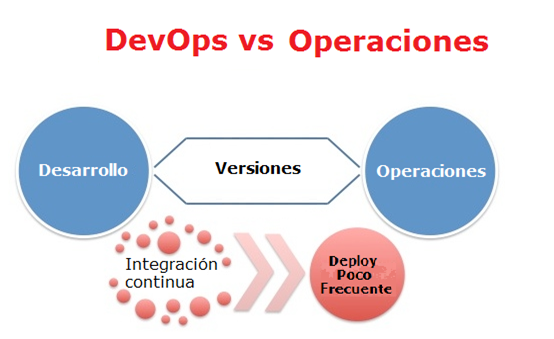
## Concepto de DevOps

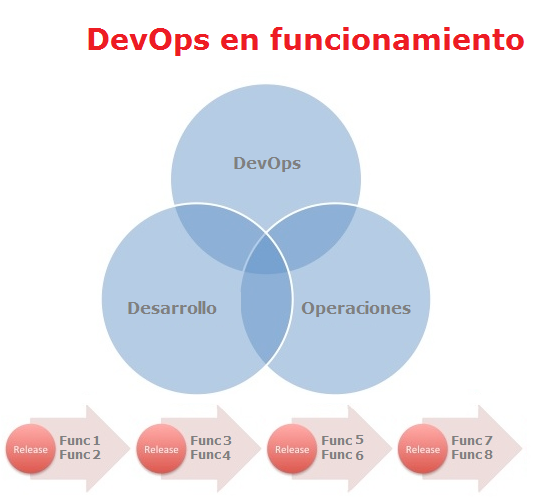
El concepto de DevOps es relativamente nuevo, pero ha sido ampliamente discutido y mejorado para el desarrollo de plataformas para Internet recientemente. Es importante destacar que actualmente los proveedores de software están creando y reforzando herramientas específicas para la implementación de DevOps, lo que indica una corriente importante hacia este concepto.



Para entender DevOps hay que comprender que se basa en el Desarrollo Ágil, donde los desarrolladores descontentos con la imposición de procesos pesados, generación de objetos innecesarios a la naturaleza del software (por lo general la documentación) describen un manifiesto ágil basado en realizar desarrollo con foco principal en el negocio y distribución de aplicaciones. A pesar de que las prácticas de desarrollo ágil han demostrado ser exitosas, el verdadero valor del desarrollo Agile (y en particular la práctica de integración continua y distribución) a menudo ha sido detenida al intentar llegar de la misma manera a producción. El siguiente diagrama muestra el enfoque que encuentran los equipos ágiles típicamente al pasar a producción el desarrollo.



Mientras Desarrollo con el fin de satisfacer las cambiantes necesidades del negocio han intentado poner en producción con mayor frecuencia las versiones generadas, mediante la entrega de cambios incrementales, Operaciones (especialmente en las grandes organizaciones) han estado tratando de reducir el riesgo al disminuir la tasa de cambio y formalización de las puertas y los traspasos a producción. Esta diferencia de enfoque, ha llevado a Desarrollo a tratar de desvanecer la barrera mediante la integración de Desarrollo y Operaciones. Una visión idealizada de DevOps se ilustra en el siguiente diagrama:



Cabe señalar que DevOps no tiene como meta evitar los objetivos de Operaciones, si no que se basa en satisfacer dichos objetivos en forma temprana y continuamente.

## Prácticas DevOps

DevOps no es una tecnología, una aplicación o un método definido como sí lo son Agile Scrum y XP. En la actualidad, no se puede fácilmente " hacer" DevOps leyendo un libro o tomando un curso y certificar en DevOps. Sin embargo, existen algunas prácticas acordadas en común que puede aportar en el enfoque DevOps:

#### Equipos multifuncionales

Considerando que el Agile defiende los beneficios de integrar el desarrollo y las funciones de control de calidad en un solo equipo, DevOps toma esto mediante la integración de las funciones operativas.

#### Habilidades funcionales cruzadas

Una vez más similar a Agile que anima las personas a tener una mezcla de habilidades, DevOps estimula el conocimiento de las habilidades sobre la operación del servicio (los ejemplos incluyen habilidades en seguridad de la aplicación, el seguimiento, mantenimiento y soporte para mantener las aplicaciones saludable y funcionando).

#### Desarrollo para producción

Creación temprana de artefactos operativos como parte del proceso de desarrollo (por ejemplo, el despliegue y scripts de actualización, scripts de migración de bases de datos automatizadas, supervisión y presentación de informes guiones) y un mayor enfoque en la entrega de los requisitos no funcionales tempranamente.

#### Automatización de release

Automatizar tanto como sea posible incluyendo construcción y publicación, el aprovisionamiento de servidores, la implementación y la integración automatizada y pruebas de aceptación.

#### Herramientas consistentes

El uso de los mismos conjuntos de herramientas entre Desarrollo y Operaciones, por ejemplo, un único repositorio para almacenar el código fuente y los scripts operacionales (que se entreguen conjuntamente), y la integración de las herramientas que apoyan el desarrollo y la gestión de los cambios.

#### Pipeline de despliegue

Construcción de un sistema automatizado de Pipeline que modele el ciclo de vida de la aplicación incluyendo el pasaje a producción, es decir, Desarrollo -> TEST -> UAT -> Producción y la provisión del servicio al usuario final por auto gestión.

#### Lanzamientos por fases

Para el despliegue a en un entorno con gran cantidad de usuarios se promueve el despliegue en un conjunto de servidores, antes de ser desplegado a conjunto grande y luego más grande (por ejemplo, regional, país, entonces global). Esto permite que se eliminen posibles problemas localmente antes de afectar a toda la comunidad.

DevOps deben ser realmente un "equipo virtual" con los individuos que participan en proyectos específicos. Sin embargo, lo mismo podría decirse de la metodología Agile, donde el desarrollo y control de calidad deben ser realmente parte de un mismo equipo, aunque muchas organizaciones hay equipos de prueba independientes del desarrollo, que han puesto en práctica metodologías ágiles.

## Beneficios de DevOps

Hay una serie de beneficios potenciales, que se intentan obtener con la aplicación de las prácticas DevOps:

#### Disminución del riesgo

En la práctica de DevOps se busca una previa consideración de los aspectos operativos del requerimiento, con esto se busca minimizar los inconvenientes cuando se realiza el Deploy de la solución en el ambiente productivo.

#### Disminución del tiempo de la entrega

Mediante la incorporación de las necesidades operacionales en el comienzo del desarrollo, se busca acortar los tiempos del pasaje de la aplicación de Desarrollo a Test luego a UAT y posterior pasaje a Producción. Dado que la principal premisa de DevOps es entrega continúa e incremental, el tiempo promedio (un Deploy semanal) de entrega debe ser mucho más corto que un enfoque tradicional. Con un enfoque DevOps hay más cantidad de Deploy en contraposición de un desarrollado con un enfoque "big bang" donde se busca una implementación definitiva desde la primera vez.

#### Mayor visibilidad

Dado que el personal operativo y de soporte está involucrado de manera efectiva en el proceso de desarrollo y test, tendrán una mejor visibilidad de los próximos cambios y las necesidades propias de la infraestructura. Por lo tanto, serán capaces de responder mejor a las necesidades del negocio y por la ampliación de provisión de la infraestructura para satisfacer la demanda.

## Métricas

Actualmente las métricas y la evidencia reunida para demostrar este valor teórico de implementar DevOps es algo limitada (especialmente en comparación con las métricas de implementaciones Agile). Una práctica recomendada es la de generar las métricas para el entorno existente, donde no se utiliza DevOps, como:

Número de funcionalidades / release pasados a producción

Tiempo de entrega de la solución requerida

Número de incidentes / problemas en la producción, propios del release

Y luego comparar estos contra las posibles mejoras de la implementación de DevOps, para determinar la viabilidad de implementar DevOps.

## Casos recomendables para DevOps

Hay una serie de áreas donde DevOps ya ha sido adoptado y funciona bien:

#### Aplicaciones basadas en Web

La arquitectura de aplicaciones web (con hilos de implementación bien definidos y la habilidad para entregar los cambios incrementales) se adapta bien a DevOps

#### Nube o infraestructura virtual

Aplicaciones DevOps necesitan una gran cantidad de infraestructura y por lo general requieren de ambientes que puedan ser configurados a demanda. Proveer servidores físicos para soportar las pruebas y soportar las diferentes release en forma continua requiere mayor cantidad de tiempo y ralentizar considerablemente el Deploy. DevOps funcionan mejor cuando la infraestructura está basada en la nube o servidores virtuales.

#### Adopción de metodologías Agile

DevOps está estrechamente alineado con prácticas ágiles, por lo tanto las organizaciones que tengan metodologías ágiles fuertemente arraigadas y optimizadas, contaran con una mejor chance de implementar DevOps exitosamente.

#### Aplicaciones independientes (Desacopladas)

DevOps funciona mejor cuando la aplicación es independiente y no depende de otras aplicaciones que estén pasando por el mismo proceso (sin importar el tamaño de la misma). Hay mecanismos para relacionar dichas aplicaciones en los diferentes release dentro de DevOps. De todas maneras este es un aspecto al que se le debe prestar atención antes de la adopción de DevOps en un proyecto.

## Casos NO recomendables para DevOps

Hay una serie de áreas en las DevOps potencialmente puede ser un reto:

#### Aplicaciones dependientes (Acopladas)

Cuando existen aplicaciones con dependencia explícita y se encuentran en proceso de mejora simultáneamente, las dificultades de implementación se incrementan para DevOps. Implementar DevOps en este escenario no es imposible, pero requiere una planificación y ejecución de los cambios dependientes de manera específica. Esto hace que la facilidad y automatización deseada en DevOps se encuentre forzada y genere tiempos muertos no deseables en la entrega continua de las mismas.

#### Entornos compartidos

Si una organización por ejemplo comparte entornos de prueba múltiple (mismo servidores para pruebas de diferentes Releases) potencialmente se puede generar conflictos al implementar DevOps. La rápida entrega de cambios potencialmente puede causar inestabilidad si un entorno está siendo utilizado para múltiples propósitos. Idealmente los diferentes entornos DevOps deben poseer exclusivamente a sus propios servidores.

#### Las aplicaciones multi-plataforma

Si se desarrolla y se pretende realizar el Deploy de los componentes de una aplicación, a través de múltiples plataformas (Windows / Unix y mainframe) la dificultad es similar a las aplicaciones dependientes. En estos casos es probable que el conocimiento de desarrollo y de Deploy, esté distribuido por varias áreas dentro de la organización, generando esto un grupo DevOps muy grande o con faltantes de conocimiento.

#### Sectores regulados

Si las aplicaciones se están entregando en las industrias altamente regulada (Servicios financieros o de salud) pueden existir normas específicas que impiden DevOps. Un posible ejemplo de ello es Sarbanes Oxley (Sarbox) que tiene un requisito específico en pos de la separación de tareas. Sarbox establece que quien desarrolla un software no debería tener ningún tipo de acceso a los sistemas de producción, pues pueden afectar a la información financiera de la empresa. Cualquier aplicación de DevOps en tal ambiente tendría que dejar en claro las funciones y responsabilidades de todas las partes.

## DevOps y la Gestión de Versiones

Gestión de Versiones tradicional, se encuentra mucho más difundida, entendida y definida formalmente.

Se definen políticas para diferentes tipos de release (mayor, menor, de emergencia).

Planificación y publicación del contenido de cada release. Fechas, tiempos, alcance, criterios de éxito.

Planificación y gestión de la dependencia entre las distintas versiones de varias aplicaciones.

Definición de flujo de trabajo y aprobaciones para mover release a través de los diferentes ambientes.

En organizaciones maduras, la administración de Versiones se realiza por un equipo distinto a la gestión del proyecto. Gestión de Versiones tiende a tener una visión más empresarial de la entrega de Releases y de ofrecer un servicio a los proyectos de aplicación. En organizaciones que no tienen implementada la Gestión de Versiones, dicha gestión se encuentra soportada por Desarrollo o por la Gestión de Cambios.

No hay razones expresas donde la Gestión de Versiones impida implementar DevOps, pero pueden existir políticas o procedimientos que impidan potenciar los beneficios, por ejemplo

Disponer de una política tendiente a minimizar la cantidad de cambios menores en pos de generar menor carga en los pasajes de entorno, en contraposición, DevOps fomenta las liberaciones menores.

Planificar, coordinar e implementar colas de versiones dependientes entre sí. DevOps necesitará de aplicaciones flexibles y con mínimo acople para aportar beneficios.

La imposibilidad de implementar aprobaciones laxas o de flujos de trabajo por autogestión, tienen a repercutir en la implantación de DevOps.

## Resumen

Ya sea que una organización adopte o no DevOps, siempre perseguirá el mismo fin, la entrega de versiones en producción, a tiempo y de calidad.

Si el negocio se encuentra satisfecho con los tiempos de respuesta y la calidad de las versiones entregadas por el sector IT, no es buen momento para intentar aplicar este tipo de acercamientos entre desarrollo y operaciones. No debe abordarse el esquema DevOps con el fin de potenciar, por parte de Desarrollo el control operativo de la solución en producción.

Actualmente se considera a DevOps como una tendencia, pero aún no está consolidada, como otras buenas prácticas del mercado. Siempre existe la posibilidad de mejorar sobre el esquema actual de la organización, antes de realizar cambios en el enfoque. Teniendo fuertemente arraigado el concepto Agile, es posible, bajo supervisión, abordar un proyecto con DevOps, recolectar evidencias, establecer métricas y ganar en experiencia en su implantación.

DevOps se presenta como la solución a todos los males actuales, sin expresar todas las falencias en las que se puede caer al implementarlo. Empresas maduras demuestran que con mejora continua, se pueden reducir los tiempos y minimizar los cuellos de botella actuales sin necesidad de alejarse de las buenas prácticas. DevOps es un concepto válidamente aplicable para ciertos casos, con el correspondiente análisis de riesgos y beneficios.