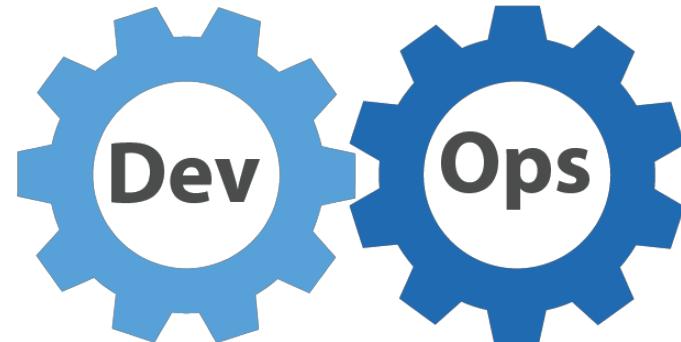


La revolución de

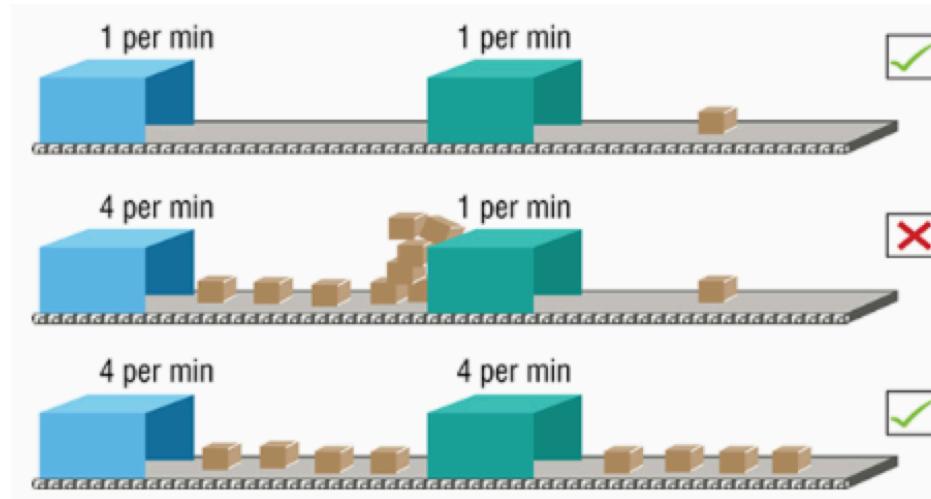


- ▶ Una empresa competitiva requiere poner en producción cosas nuevas con rapidez
- ▶ Ejemplos:
 - ▶ reaccionar a que movistar comienza a ofrecer planes familiares
 - ▶ se necesitan nuevas ofertas de roaming mas atractivas
 - ▶ un banco sacó el poder bloquear la tarjeta

Desde desarrollo ágil a devops

- ▶ Adopción de procesos ágiles en los 2000 es uno de los principales drivers del desarrollo de devops
- ▶ Código generado tenía que ser testeado rápidamente e instalado en ambientes de prueba y producción mucho más rápido
- ▶ Operaciones no estaba preparado y comenzó a generar cuellos de botella
- ▶ Weekends deployments (con sacos de dormir)
- ▶ Se requería un mejor trabajo colaborativo entre Dev y Ops

Gráficamente es fácil de entender



Desarrollo

Puesta en Producción

Minimizar el tiempo de ciclo

- ▶ Cycle time - tiempo desde que aparece el requisito (user story) hasta que está en las manos del cliente o usuario
- ▶ Se requería integración continua y entrega continua
- ▶ El ciclo ágil se extiende entonces mas allá del desarrollo involucrando a la gente de Ops

Dev vs Ops

- ▶ Tanto Dev como Ops apuntan a satisfacer al usuario pero ...
 - ▶ La tarea de Dev es crear nuevas cosas y ponerlas en las manos del usuario rápidamente
 - ▶ La tarea de Ops es asegurar que los usuarios disponen de un sistema estable, rápido y seguro
- ▶ Se espera de los desarrolladores que produzcan permanentemente cosas nuevas
- ▶ Se espera de la gente de operaciones que mantengan un sistema estable todo el tiempo

La época de Cascada (pre-ágil)

- ▶ dev y ops vivían en mundos aislados
- ▶ interacción solo en el momento del release
 - ▶ dev solo podían incluir nuevas features al momento del release (o esperar al siguiente)
 - ▶ ops sabía exactamente la fecha del release para comenzar a probar y planificar el momento de la instalación (weekends con sacos de dormir)

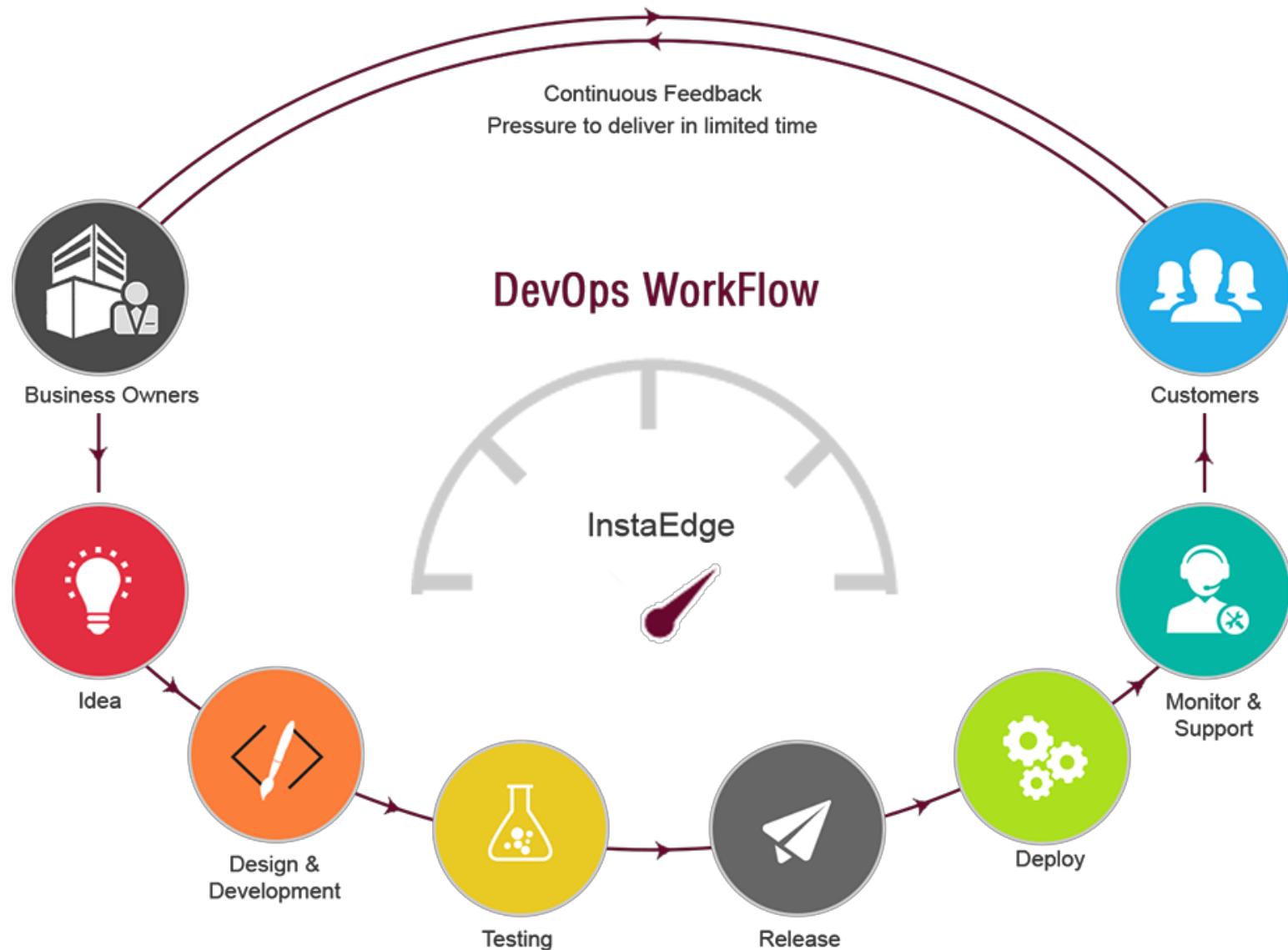
Agilidad cambió todo

- ▶ Integración continua - nuevas funcionalidades se agregan diariamente
- ▶ Devs presionan para que queden operativas cuanto antes
- ▶ En lugar de que ops espere la llegada de un tren hay una correa transportadora

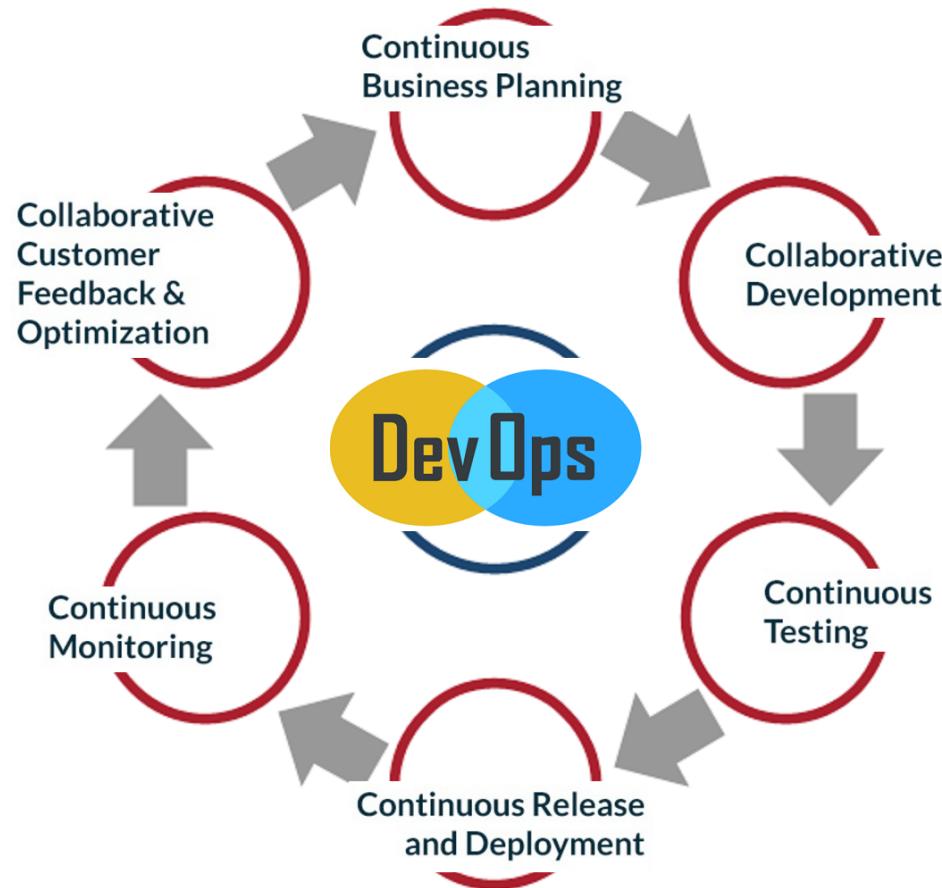
Devops: balances entre innovación y estabilidad, rapidez y calidad

- ▶ Dev tiene que trabajar con Ops para entender la naturaleza del sistema de producción: restricciones, arquitectura del sistema, etc
- ▶ Dev tiene que involucrarse mas en testing: cómo funciona en producción, trabajar mas cerca de la gente de QA
- ▶ Dev tiene que aprender a monitorear las aplicaciones deployeadas y entender las métricas que usa Ops
- ▶ Ops tiene que saber qué código viene y como impactará al sistema (entender requisitos)
- ▶ Ops tiene que automatizar su trabajo
- ▶ Ops debe monitorear toda la línea de entrega para detectar posibles inestabilidades donde ocurran
- ▶ Ops tiene que comunicarse y colaborar mejor con Dev

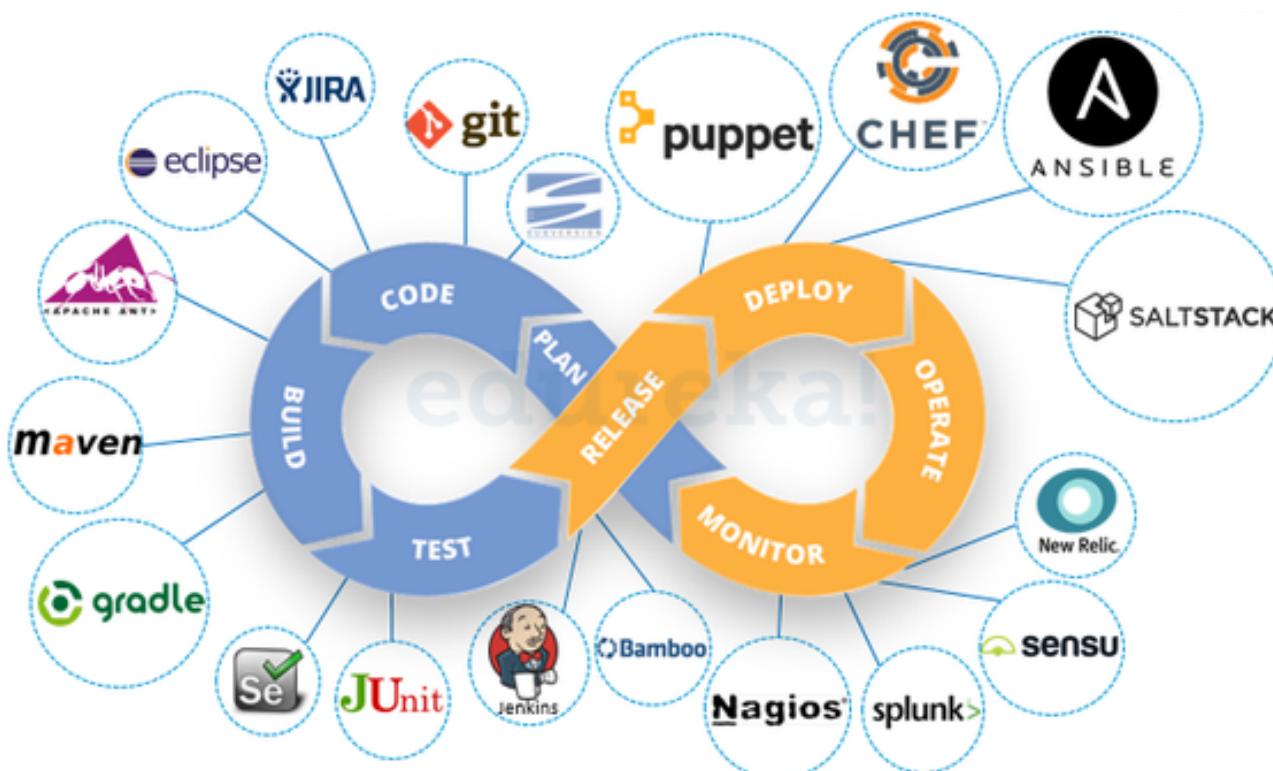
Reaccionando Rápidamente



El ciclo DevOps



DevOps requiere fuerte Automatización



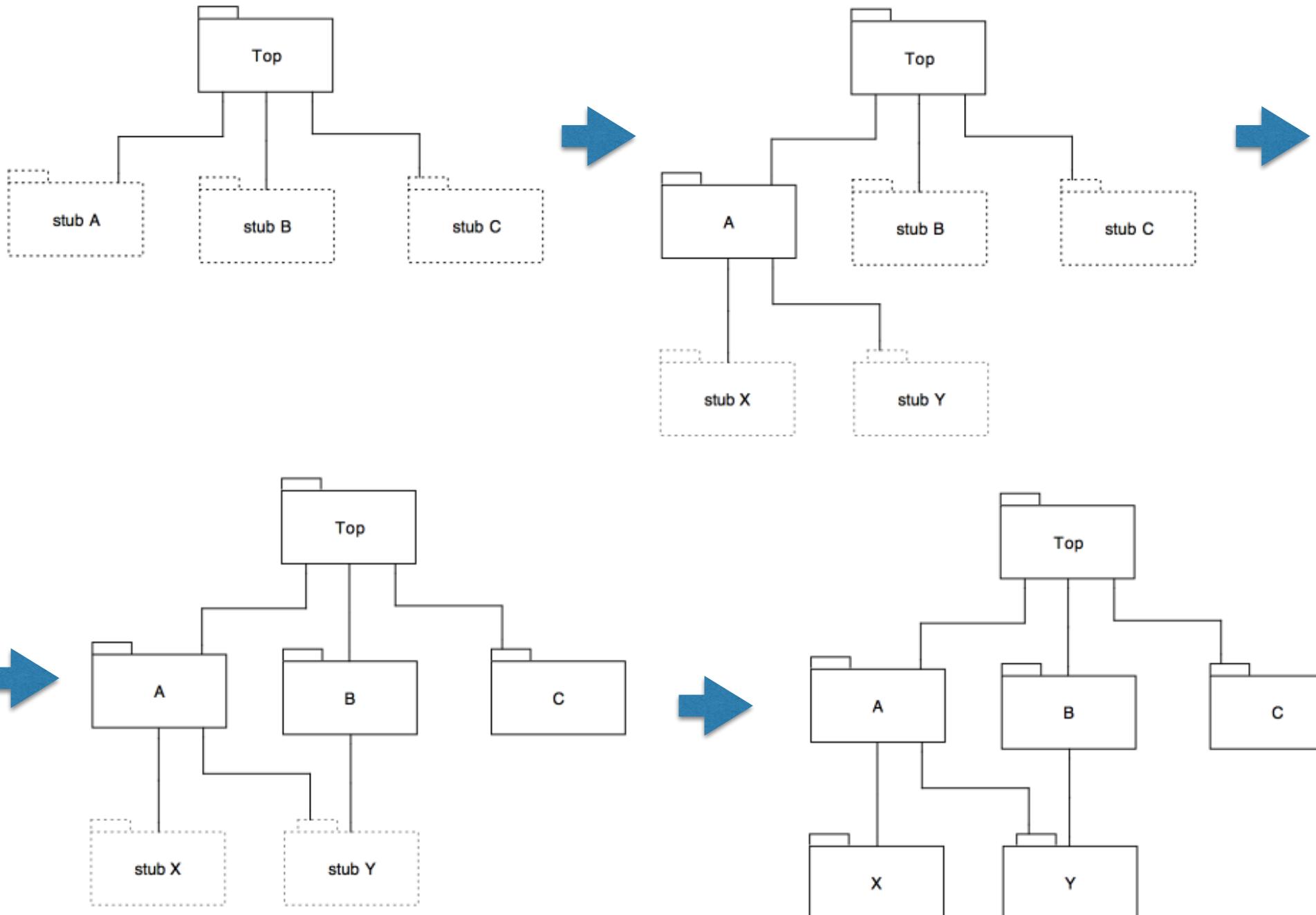
Aspectos claves para implementar Devops

- ▶ Sin ellas no podemos hablar de Devops
 - ▶ Integración continua (continuous integration)
 - ▶ Entrega continua (continuous delivery)
- ▶ Ambas se enfocan en minimizar el tiempo de ciclo

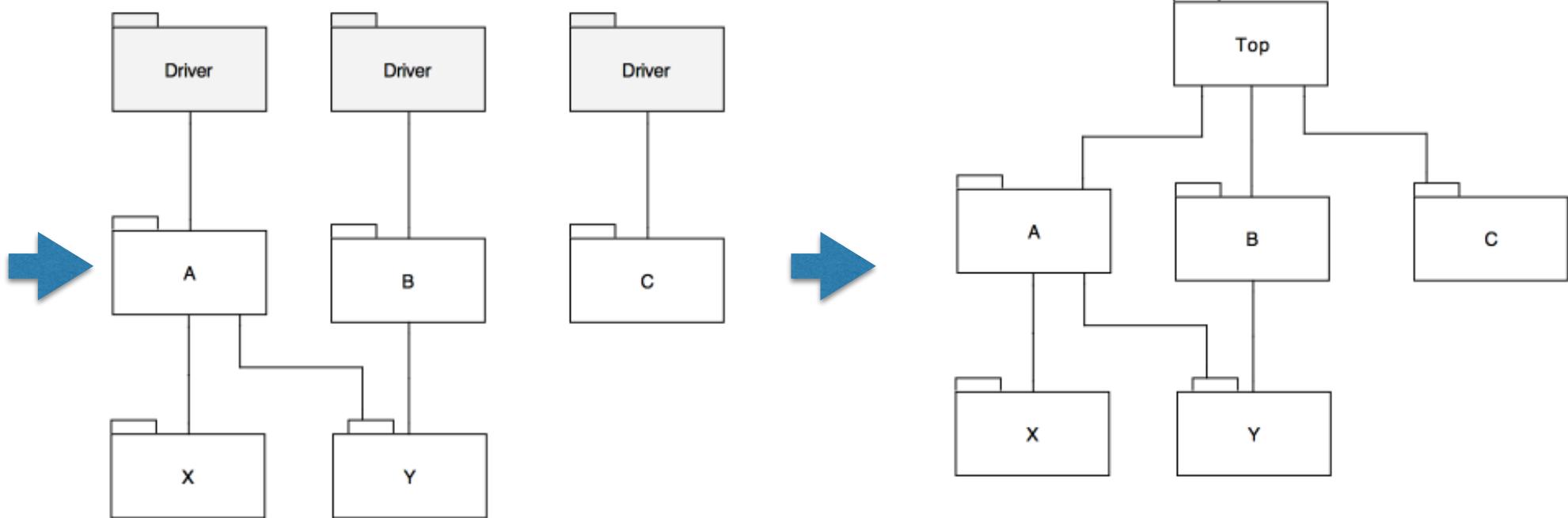
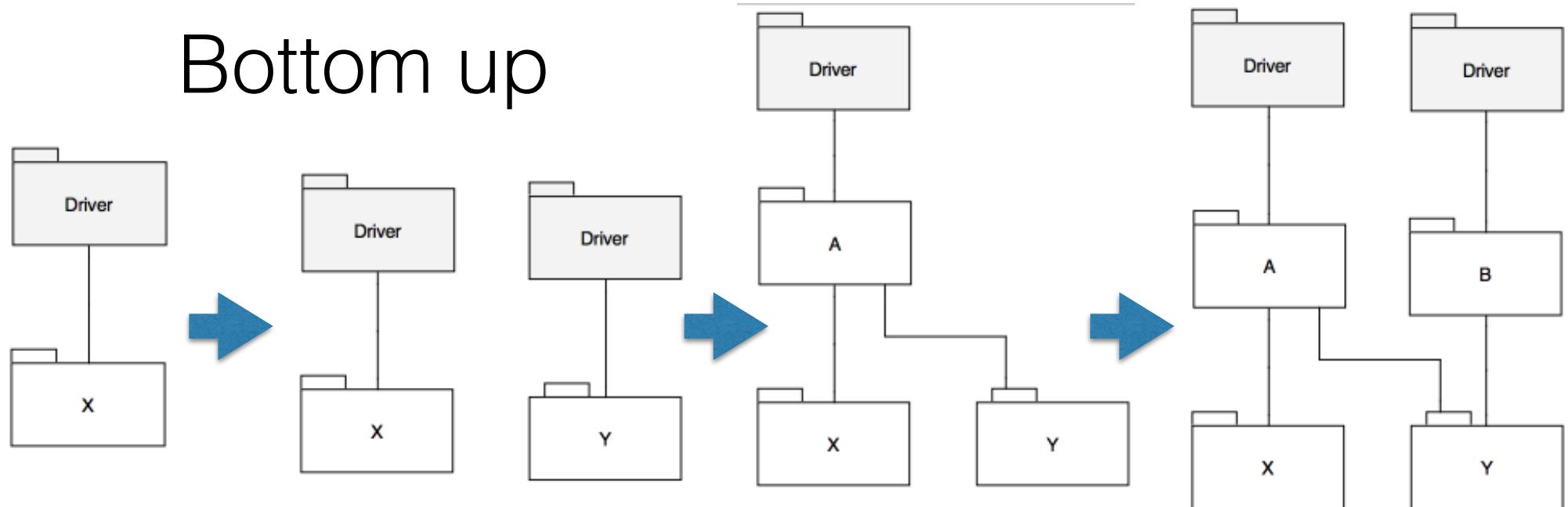
Integración Contínua

- ▶ Un producto de software se compone de muchas partes que son desarrolladas en forma independiente y que eventualmente deben ser ensambladas
- ▶ La integración es una etapa fundamental en el desarrollo
- ▶ En enfoque tradicional ello se hace cuando la mayoría de las componentes están ya desarrolladas
- ▶ Muchos defectos e incompatibilidades se descubrían muy tardíamente

Top Down



Bottom up



Integración Contínua

- ▶ Introducida por el movimiento ágil
- ▶ desarrolladores integran su trabajo con el resto en forma regular (diariamente) y se prueba de inmediato
- ▶ si hay que integrar con otros sistemas se hace tan pronto sea posible

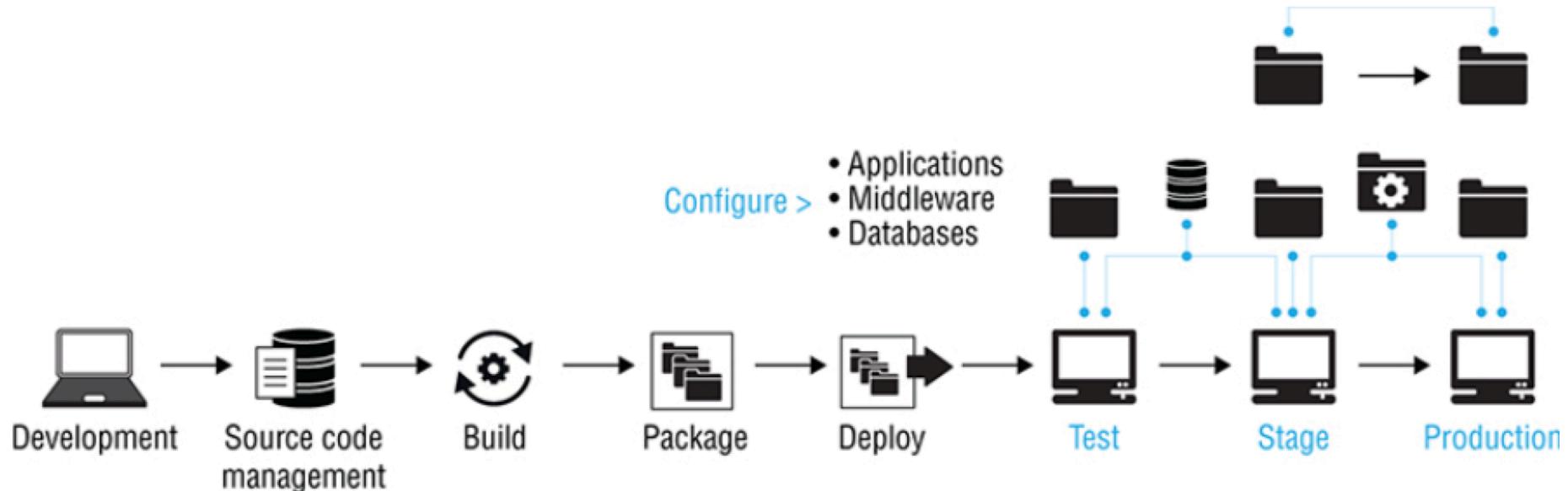
Buenas Prácticas para Integración Contínua

- ▶ mantener un solo repositorio de código fuente (incluso si se usan varias plataformas)
- ▶ automatizar el "build" (armado de la aplicación)
- ▶ hacer que el build se testee solo
- ▶ asegurar que todos hagan commit de su código a la linea base cada día
- ▶ asegurar que el build no tome demasiado tiempo
- ▶ testear en un clon del ambiente de producción (production-like)
- ▶ facilitar a cualquiera el acceso al último ejecutable (validación temprana)
- ▶ asegurar que todos sepan lo que está sucediendo (facilita la comunicación y colaboración)

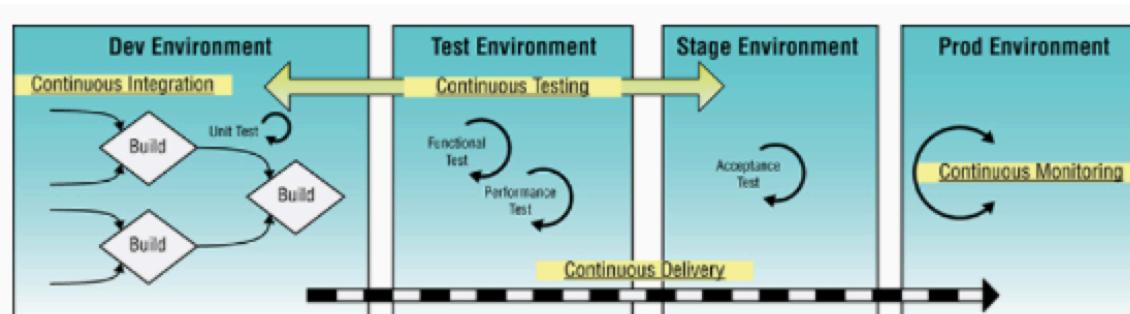
Entrega Contínua

- Lleva el concepto de integración continua a la siguiente etapa
- Build es tomado por QA y de ahí a operaciones para ponerlo en producción
- No todos los builds tienen que ir, solo los que se considera pueden ir a QA (los buenos)
- No todos los que pasan por QA deben ir a producción, solo los que están listos en términos de estabilidad, etc
- Para ver si los builds son production-ready son puestos en un "staging area"

El pipeline de entrega continua



- requiere cuidadosa orquestación entre los deployments de código, contenidos, middleware, etc.



Continuous Delivery or Continuous Deployment

- ▶ Hay empresas que tienen lo segundo y por ejemplo hacen deploy a producción hasta 100 veces a la semana
- ▶ Cada cambio de cada desarrollador es llevado a producción
- ▶ Ello puede alejar a muchos de las prácticas de DevOps y no es necesario

"Continuous delivery doesn't mean every change is deployed to production ASAP. It means every change is proven to be deployable at any time. —Carl Caum (Caum, 2013)"

Devops > Dev + Ops

- Achicar la brecha entre Dev y Ops no es lo único para lograr un ciclo de entrega rápido
- Se debe incorporar perspectiva del negocio
- El negocio necesita ser visible en lo que está siendo producido por IT
- El negocio debe proveer feedback sobre como los clientes están usando los nuevos servicios
- El negocio debe proveer feedback sobre el valor de los nuevos servicios

Devops como cambio de cultura

- ▶ cambia la forma en que los individuos piensan sobre su trabajo
- ▶ valoriza la diversidad del trabajo realizado
- ▶ incorpora procesos que aceleran la tasa en que el negocio agrega valor
- ▶ mide efectos de cambios sociales y técnicos
- ▶ todos son responsables de que el producto o servicio llegue finalmente a las manos del cliente

¿Que clase de cultura se requiere para adoptar Devops ?

- ▶ Una que se base en
 - ▶ confianza
 - ▶ comunicación
 - ▶ colaboración
- ▶ La adopción de Devops no traerá como efecto esas cosas
- ▶ Deben cultivarse en la organización
- ▶ Problema de huevo y la gallina