Architecture evolution

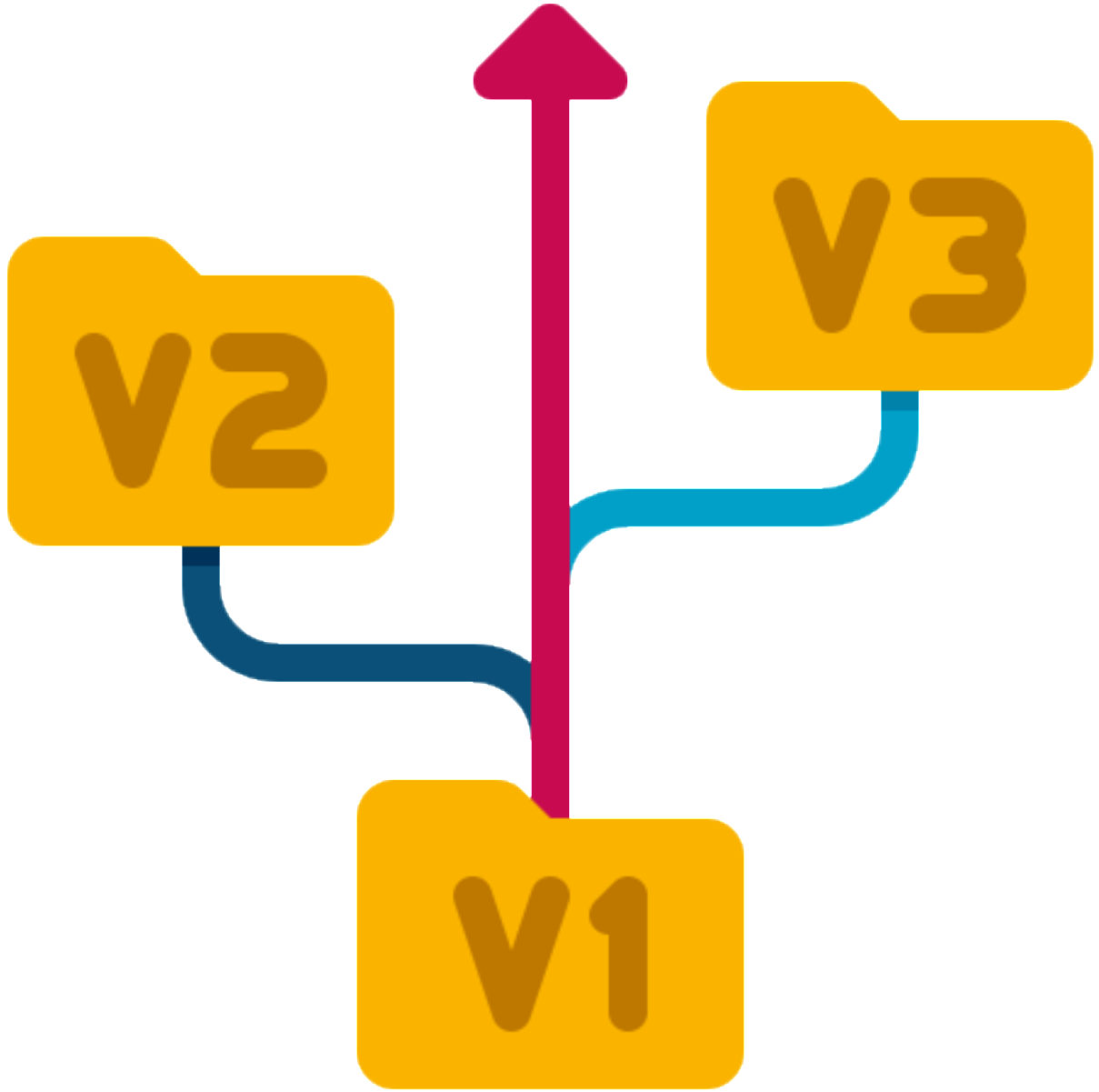
- [Episode 525: Randy Shoup on Evolving Architecture and Organization at eBay]
- Jorge Casatejada Santamarta - UO282294
- Jonathan Arias Busto - UO283586
- Alejandro Campa Martínez - UO282874

iNDiCE

- Iteraciones
- Trabajo entre versiones
- Diferencias entre monolitos y aplicaciones
- Cloud
- Pasado-presente
- Cambio BBDD
- Trabajo actual
- Libro Accelerate
- Tiempo de desarrollo
- Traffic Mirroring
- Mejoras en desarrollo
- Scrum
- Legacy services y clusters

Iteraciones

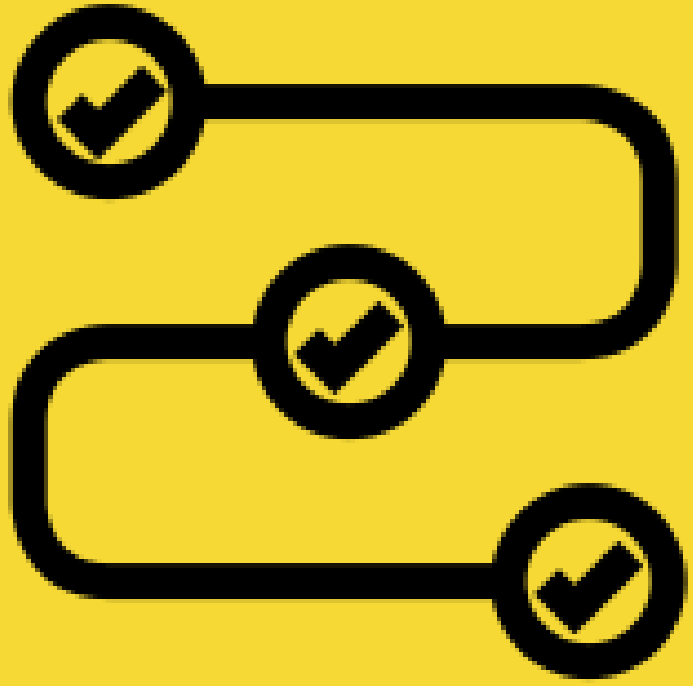
1.- Página back end Pearl
    - Poco escalable
2.- Monolito C++
    - Código espagueti imposible de mantener
    - Dos grandes bases de datos
        - Principal
        - Backup
3.- Java con archivos EAR
    - Paso a ficheros EAR y división en unas 220 aplicaciones.
    - Transición de bases de datos a unas mas específicas (dominio, entidades, etc).



Trabajo entre versiones

Entre las diferentes versiones había un periodo de adaptación, que por ejemplo en el caso de la segunda iteración a la tercera fue de 5 años aproximadamente.

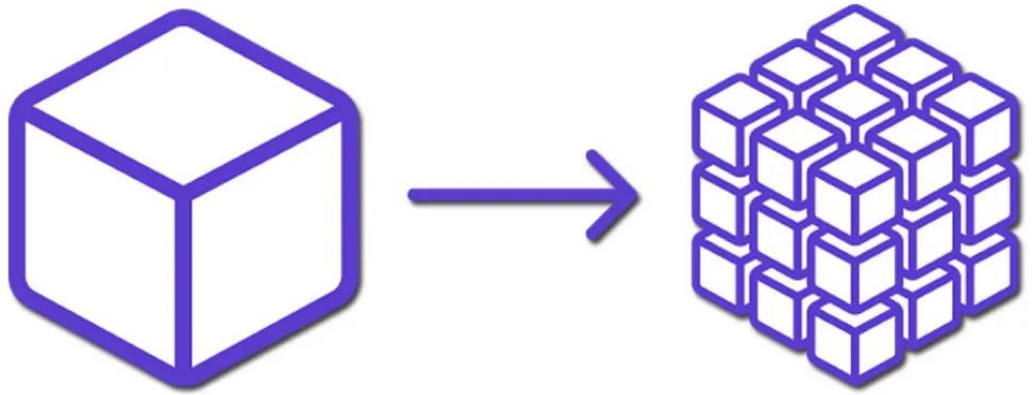
En el caso de las bases de datos lo que se hizo en eBay fue pasar de dos bases de datos enormes en la iteración 1 a otras mas específicas que serían las usadas en la V2 y V3.



Diferencias entre monolitos y aplicaciones

Una de las ventajas que supone el paso de un monolito a trabajar con la arquitectura de aplicaciones es el hecho de trabajar de forma mas eficiente y la independencia entre equipos.

Por otro lado un inconveniente es el control de los errores, ya que al existir tantas aplicaciones tener controlado donde se producía un error era algo complejo para aquella época.



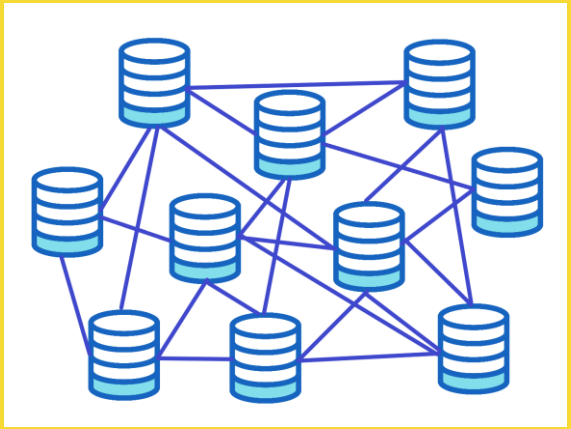
Cloud

eBay es tan eficiente que se puede permitir tener su propia nube y sus propios servidores.

Tienen 3 data centers propios desde los que controlan todo.

Esto les facilita el trabajo de forma que no dependen de ninguna otra empresa.

Pasado-Presente



1.-Incorporación de microservicios:

- Cada servicio posee sus datos.
- Interacción a través de interfaces.

2.-Amazon fueron los ‘pioneros’ de estos microservicios.

3.-Objetivo:

- Cada servicio trata una cosa en particular.
- No se interactúa directamente con los datos.

Cambio BBDD

1.-Se dividió la BBDD:

-Ventajas:

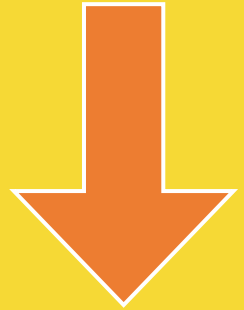
- Avance para esta época (acercamiento a microservicios)
- Cambio transparente al usuario

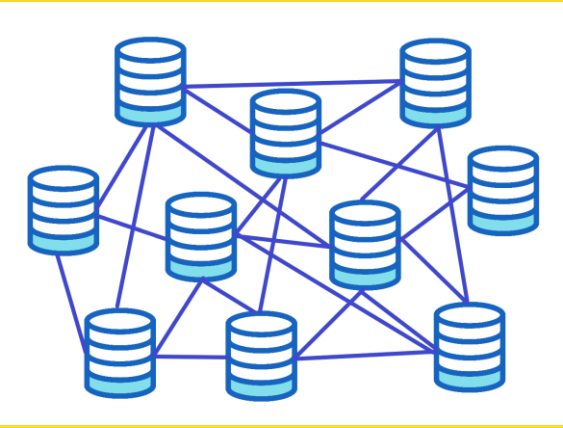
-Desventajas:

- Muchas relaciones entre las BBDD

2.-2007 grandes empresas comenzar a hacer esta separación (más ordenada)

3.-Surgió debido al crecimiento de la industria





Trabajo actual en eBay

Arquitecto jefe

-Meta:

- Hacer más productivos y exitosos a los ingenieros.

-Objetivo:

- Agilizar entregas de nuevas funcionalidades y corrección de errores.

Como:

- Uso de lean software y lean manufacturing .
- Trabajo con algunos equipos, no todos.
- Estudiar las 4 etapas.

Objetivos conseguidos:

- Trabajo de las 4 etapas de 10 a 2 días.

Libro Accelerate

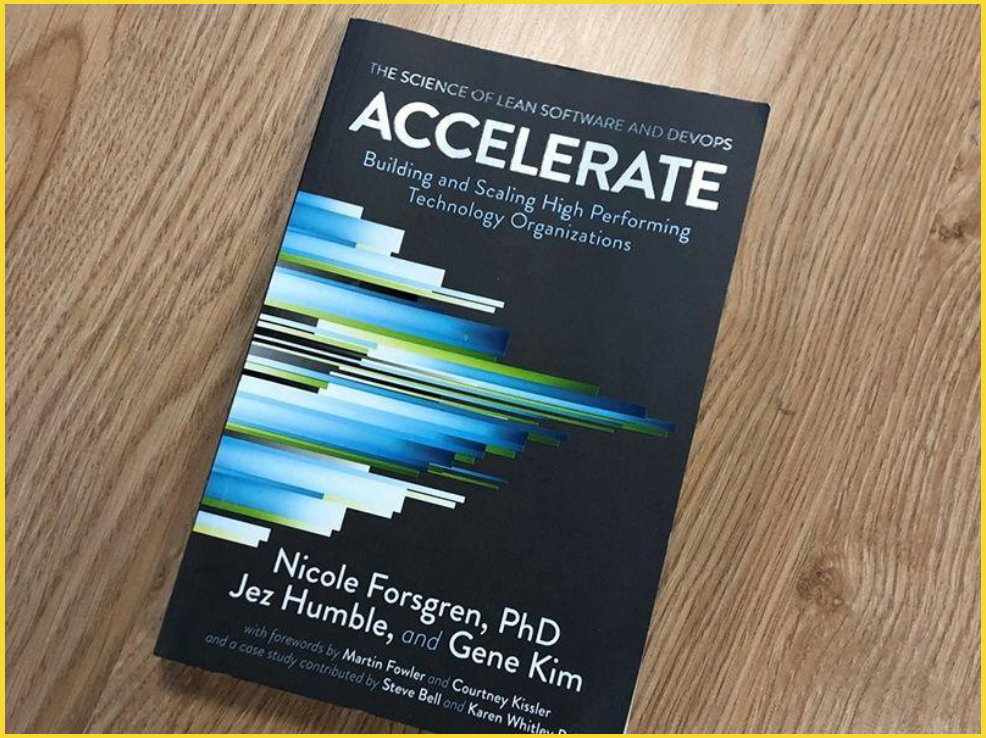
1.-Técnicas de entrega continua para resolver los problemas de entrega de software.

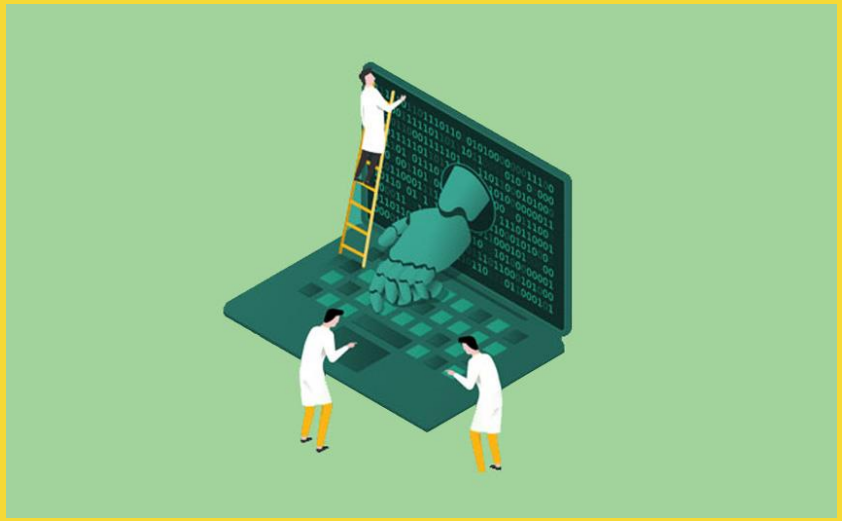
2.-Cuatro técnicas para medir eficacia:
- Frecuencia de despliegue.
- Tiempo de espera para el cambio.
- Tasa de fracaso del cambio.
- Tiempo medio de restauración.

3.-Clasificación por rendimiento.

4.-Empresas buenas en frecuencia:
- Buen tiempo medio de recuperación.
- Buena tasa de fracaso del cambio.

5.-En eBay -> consiguió avanzar de rendimiento medio a alto





Tiempo de desarrollo

- Tenían que mejorarlo
  - Comunicación con el equipo
    - Bajar build time
    - Hacer code reviews a primera hora
- Inversión en técnicas para mejorarlo
  - Traffic mirroring

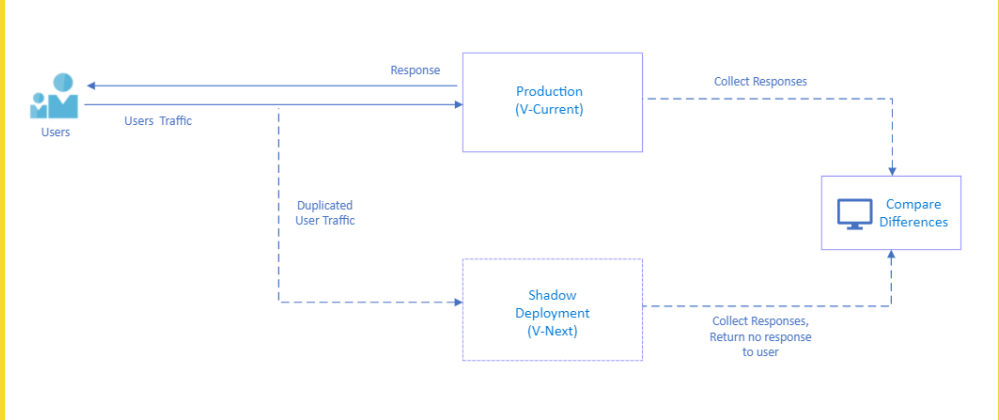
Traffic Mirroring

Al hacer cambios que no afecten al comportamiento

Se reproduce el mismo comportamiento con:

- Código actual
- Código modificado

Se comparan los resultados para comprobar que se puede hacer deploy



Mejoras del desarrollo

- Lo que no saben que no saben
- Equipo de ayuda
- Mayor frecuencia de Pull Requests y deploys
- Los principios de Desarrollo se aplican en cualquier cosa

Reunión Scrum

- Reunión de 1 hora
- Todos los equipos involucrados
- Cada semana
- Recibir y ofrecer ayuda
- Solucionar problemas

Legacy services and clusters

- Se deberían migrar

- Es complicado migrarlo por ser:
  - Importante
  - Parte del core
  - Da miedo