



EVOLUCION DE LA ARQUITECTURA

Episode 525: Randy Shoup on Evolving Architecture
and Organization at eBay

Javier García González (UO276723)
Hugo Roberto Pulido Pensado (UO282823)

21-04-2023

Tabla de contenido

Randy Shoup	2
Evolución en la arquitectura de Ebay	2
Proyeto inicial	2
V2	2
V3	2
RETROSPECTIVA	3
COMPUTACION EN LA NUBE	3
ROL ACTUAL EN EBAY	3
TECNICAS PARA AUMENTAR LA EFICIENCIA	3
DIVIDIR EL CICLO DE VIDA EN ETAPAS	3
ADOPTAR CIERTAS MEDIDAS DE CALIDAD	4
REUNIONES SEMANALES	4
AUTOMATIZACION DE PRUEBAS	4
REVISION DE CODIGO	4
APLICAR Y HACER USO DEL LIBRO ACCELERATE	4

Randy Shoup

Randy Shoup es un ingeniero informático que ha trabajado en Ebay en dos ocasiones, primero de 2004 a 2009 y después desde 2020 hasta la actualidad. También ha trabajado en muchas empresas de informática muy importantes como Oracle, Google Cloud y muchas más.

Evolución en la arquitectura de Ebay

La arquitectura de Ebay ha pasado por 5 versiones, aunque aquí solo hablaremos de las 3 primeras ya que son de las que habla Randy en el Podcast. Estas versiones son el proyecto inicial, la V2 y la V3.

Proyecto inicial

El proyecto inicial fue creado por Pierre Omidyar, fundador de Ebay, en 1995. Tardó 3 días en hacerlo y no lo hizo con la intención de crear un negocio, sino solo de jugar con la Web, que en aquella época era algo bastante nuevo. Esa primera versión tenía un backend en Pearl y cada artículo de la tienda era un archivo aparte en la aplicación.

V2

La segunda versión que ya era algo más serio se hizo en C++, pero escribieron toda la aplicación en una única clase y con el paso del tiempo esa clase llegó a tener 3.4 millones de líneas de código. También tuvieron problemas con el número de métodos ya que Microsoft tenía un límite de métodos por clase de 16000 y al tenerlo todo dentro de la misma clase estuvieron muy cerca de llegar a ese límite.

En cuanto la base de datos, contaban únicamente con dos bases de datos, una de las cuales era una copia de seguridad.

V3

Para la tercera versión se introdujeron bastantes cambios.

Hicieron una reescritura completa de la aplicación pasando de C++ a Java. Este proceso de reescritura empezó en 2002 y les llevó 5 años terminarlo. Fueron reescribiendo pequeñas partes de la aplicación y cuando terminaban una parte, redirigían un poco del tráfico de la V2 a la nueva parte de la V3 y cuando comprobaban que esta funcionaba correctamente ya dirigían todo el tráfico a esta nueva versión.

Esta nueva versión en Java seguía una arquitectura de miniaplicaciones.

En cuanto a las bases de datos, pasaron de tener únicamente 2 bases de datos a tener más de 2000, siendo estas de dominios mucho más específicos.

El problema de esto es que todas las aplicaciones podían acceder a todas las bases de datos, por lo que se podía modificar y consultar una base de datos desde hasta 200 sitios diferentes. Esto hizo que encontrar los fallos y las partes que estaban ralentizando la aplicación fuese mucho más difícil y les hiciera perder mucho tiempo depurando y buscando errores. Al final lo que hicieron para resolver este problema fue crear su propio sistema de monitoreo.

RETROSPECTIVA

En el podcast se le hacen dos preguntas sobre decisiones que hubiese tomado hoy en día:

- La primera es, si estuviese construyendo la aplicación hoy en día, teniendo en cuenta todos los servicios que existen actualmente, ¿Qué servicios contrataría? A lo cual respondió que volvería a instrumentar todo con telemetría abierta (que es lo que hace Ebay actualmente), y en cuanto al backend, usaría un proveedor de rastreo distribuido, como LightStep o HoneyComb.
- La segunda es, con todos los conocimientos que posee actualmente, si hubiera hecho algo de manera distinta con las tecnologías que había por aquel entonces. A esto respondió que sí, que implementaría todo con microservicios, es decir, cada servicio con su propio almacenamiento de datos, y que solo se pudiese interactuar con ellos mediante su interfaz pública, en vez de como hizo en aquel entonces con esa relación de muchos a muchos entre servicios y bases de datos.

COMPUTACION EN LA NUBE

En cuanto a la computación en la nube, Randy recomienda siempre a las nuevas empresas que empiecen con computación en la nube publica, en vez de crear ellas mismas su propio centro de datos, y que solamente den el salto cuando el costo de la nube publica sea mas alto que el que tendrían teniendo sus propios centros de datos. Son pocas las empresas que realizan esta transición, y siempre tienen todo muy calculado y estudiado para que realmente les salga factible.

De hecho, la transición más común suele ser a la inversa (de centro de datos propios a nube publica), ya que mantener lo primero suele ser insostenible.

ROL ACTUAL EN EBAY

Actualmente, Randy tiene un rol muy distinto al que tenía en su primera etapa, ya que ahora es el arquitecto jefe y líder de la plataforma de desarrolladores. Su función ahora es, básicamente, hacer la vida lo más fácil posible a los desarrolladores. Para ello tiene una comunicación constante con los equipos.

Para su función actual, hace mucho uso del denominado mapeo de flujo de valor, el cual consiste en mirar a un proceso de extremo a extremo, decir todos los pasos que hay para llegar de un extremo a otro, con sus correspondidos tiempos. Al final de estos pasos, suele conseguirse algo de valor. Esta técnica permite generar cierta visión del sistema, lo cual ayuda a decidir que optimizar o mejorar, y que dejar de lado para más adelante.

TECNICAS PARA AUMENTAR LA EFICIENCIA

Actualmente, tanto el cómo sus equipos están siguiendo ciertas técnicas para optimizar la eficiencia lo máximo posible. Estas técnicas son, entre otras:

DIVIDIR EL CICLO DE VIDA EN ETAPAS

Se dividiría el ciclo de producto en cuatro etapas:

- Planificación: consistiría en cómo se convierte una idea en un proyecto.
- Desarrollo de software: consiste en cómo se convierte un proyecto en algo comprometido.

- Entrega de software: consiste en cómo se convierte algo comprometido en algo que la gente realmente usa.
- Iteraciones posteriores al lanzamiento: en ellas se va activando lo desarrollado en pasos anteriores y ofreciéndolo a usuarios individuales.

ADOPTAR CIERTAS MEDIDAS DE CALIDAD

Estas medidas serían 4:

- Frecuencia de implementación: con qué frecuencia se implementa una aplicación en general.
- Tiempo de entrega para el cambio: es el tiempo que pasa desde que un desarrollador confirma su código hasta que aparece en el sitio.
- Tasa de fallos: cuando se implementa código, cada cuanto se tiene que ir hacia atrás (roll-back) o solucionar un error en caliente (hot fix).
- Tiempo de restauración: una vez se tiene uno de esos problemas, cuánto tiempo lleva arreglarlo.

REUNIONES SEMANALES

Todos los lunes por la mañana se reúnen todos los equipos para realizar una reunión, muy parecidas a las realizadas en la metodología Scrum. En estas reuniones cada equipo dice lo que realizó la semana pasada, lo que tiene estimado realizar en esa semana, y cualquier bloqueo que haya surgido con respecto a la reunión anterior. En estas reuniones se incentiva bastante a los equipos a que se ayuden entre ellos.

AUTOMATIZACION DE PRUEBAS

Cada equipo realiza la automatización de pruebas para su respectiva parte. Esta automatización permite que el proyecto sea más ágil y eficiente.

REVISION DE CODIGO

Todos los días, como primera cosa del día, los miembros de cada equipo se reúnen y revisan el código. Al darle esta importancia a esta medida, se evita que se vaya prolongando varios días, lo cual no es nada interesante.

APLICAR Y HACER USO DEL LIBRO ACCELERATE

Este libro enseña distintas técnicas de entrega continua de software, y como resolver errores en la entrega del software. También habla sobre el desarrollo basado en troncos, que consiste en que los desarrolladores fusionan pequeñas actualizaciones de forma frecuente en un tronco o rama principal. Todos los miembros de los equipos deben aplicar y seguir este libro lo máximo posible.