Universidad de las Américas

LITI  
Bárbaro Ferro

Carlos Antonio Tapia Alvarado, 76414

Práctica Integradora de Tecnologías de Información

## Development

Facebook, al igual que la práctica total de otras empresas basadas en Internet, trabaja en el desarrollo perpetuo de modo en el que los ingenieros desarrollan continuamente nuevas características y lo ponen a disposición de los usuarios.

En consecuencia, el sistema también crece continuamente, posiblemente a una velocidad super- lineal. Estos dos atributos; el crecimiento y el despliegue rápido, son los principales retos que los ingenieros necesitan para vencer.

Los libros de texto de ingeniería de software suelen asumir un escenario donde el software se construye para alquila.

En una situación, los ingenieros, primero necesitan aprender sobre el dominio de aplicación y comprender los objetivos para el nuevo software.

En Facebook, los ingenieros son también usuarios, por lo que tienen conocimiento de primera mano de lo que el sistema hace y cuáles son los servicios que proporciona. Por otra parte, el uso interno de Facebook tiende a ser más intensivo que la mayoría de su uso por lo que existe una tensión continua entre el conocimiento y el consentimiento de primera mano derivada de examinar un amplio uso extendido. Fuera de esta tensión programadores generan ideas para mejorar la A & D.

Pero el hecho de que los ingenieros tienen conocimiento de primera mano de la aplicación es sólo un aspecto de la salida de desarrollo de software tradicional. Aún más importante es la mentalidad del desarrollo perpetuo. Los productos de software tradicionales son finitos, por definición, con un delimitado alcance y una fecha de finalización predefinida.

Esta es la base para la elaboración del contrato para producir el software, definir las pruebas de aceptación, y los problemas que surgen cuando los proyectos se retrasen o gastar su presupuesto por lo que nunca se completaron sitios como lo es Facebook. La mentalidad es que el sistema continuará siendo desarrollado de forma indefinida.

El software que sigue evolucionando durante períodos de tiempo largos en por eso es que existe en muchos dominios. Por ejemplo, el sistema operativo Linux ha evolucionado continuamente desde su primera versión oficial en

1994, creciendo 80 veces en el proceso. Sin embargo, las nuevas versiones de Linux son liberados de dos a tres meses de diferencia. Las empresas basadas en Internet como Facebook evolucionan a un ritmo mucho más rápido.

## Deployment

El despliegue continuo también tiene importantes beneficios desde el punto de vista de producción de software. Las frecuentes implementaciones implican que cada despliegue introduce sólo una cantidad limitada de nuevo código.

Esto lo reduce (pero no elimina) el riesgo de que algo va a salir mal. El despliegue frecuente aproxima al despliegue de serie, que es más fácil de depurar; por otra parte, todos los envíos se prueban individualmente para las regresiones.

Todos los nuevos empleados de Facebook se someten a un campo de entrenamiento de seis semanas en el que estos los animará a comprometerse nuevo al código tan pronto como sea posible, por parte es para superar el miedo de liberar un nuevo código. La capacidad de desplegar el código rápidamente en forma gradual y sin miedo permite una rápida innovación.

Otro de los beneficios de las implementaciones pequeñas y rápidas es que podemos identificar fácilmente el origen de los problemas y soluciones emergentes: son probablemente y el que más recientemente tiene cambios implementados en el código, y todavía está fresca en la mente de los ingenieros.

Aparentemente, el despliegue rápido está reñida con el desarrollo de funciones que requiere grandes cambios a la base del código. La solución es romper esos cambios en una secuencia desde el más pequeño y más seguro, escondidos detrás de una abstracción (una práctica acertadamente llamado " rama de la abstracción"). Por ejemplo, considere la cuestión delicada de migración de datos desde un almacén existente a uno nuevo. Esta puede desglosarse de la siguiente manera:

1. Encapsular el acceso a los datos en un tipo de datos apropiado.

2. Modifique la aplicación para almacenar datos en la antigua y nuevas tiendas.

3. La migración masiva de los datos existentes de la antigua tienda para la nueva tienda. Esto se hace en el fondo en paralelo a la escritura de nuevos datos para ambos almacenes.

4. Modifique la aplicación para leer de ambas tiendas y comparar los datos obtenidos.

5. Cuando se está convencido de que la nueva tienda está funcionando según lo previsto, pasar a utilizar la nueva tienda exclusivamente (el viejo almacén se puede mantener durante un cierto tiempo para protegerse contra imprevistos problemas).

Facebook ha utilizado este proceso para migrar de forma transparente las tablas de bases de datos que contienen cientos de miles de millones de filas a nuevos formatos de almacenamiento. Además, la implementación de nuevo código no implica necesariamente que se trate de inmediato a disposición de usuarios. Facebook utiliza una herramienta llamada "Gatekeeper " para controlar qué usuarios pueden ver qué las características del código. Así, es posible para los ingenieros puedan desplegar de forma incremental y probar implementaciones parciales de un nuevo servicio sin exponerlos a los usuarios finales.