SOFTWARE ENGENEERING AT GOOGLE

Jorge Blanco Sánchez, Adrian Dumitru, Alejandro Vega García

Contenido

Ley de Hyrum

Con un volumen de clientes lo suficientemente grande, cualquier cambio en la API por mínimo que parezca podría afectar a alguien.

Managers tradicionales vs grandes managers

Los managers tradicionales se centran en cómo hacer las cosas, de manera más imperativa, estableciendo unas pautas muy fijas sobre cómo abordar soluciones.

Los buenos managers, en cambio, son más declarativos, se centran en qué hacer y confían en el equipo de desarrollo para ejecutar la mejor solución, generando confianza y fomentando la autonomía y el aprendizaje.

El mito del genio programador

El hecho de centralizar una obra maestra de software en una única figura reconocida (Linus Torvalds, Bill Gates, etc.) es un error, ya que el trabajo es de un equipo con talento, esfuerzo y buena sintonía. El trabajo en grupo supera al individual

Para evitar caer en él, Google promueve la colaboración y la retroalimentación constante para acelerar el proceso y evitar errores tempranos.

Escalabilidad

Los principales desafíos de Google son de escalabilidad, no de innovación funcional.

Problemas como el indexado de código son clave para mantener la calidad y facilitar la navegación y el mantenimiento a gran escala.

Conceptos de medición de productividad en Google

Objetivo: Resultado esperado, aunque no siempre sea fácil de definir con precisión.
Señal: Indicador de progreso hacia el objetivo, aunque a veces sea difícil de medir directamente.
Métrica: Un dato concreto que actúa como un proxy de la señal y sí que puede medirse.

Testing

Boilerplate

Boilerplate se define en la presentación como un excesivo código repetitivo en las pruebas, siendo el objetivo de este, el testing a librerías externas y no al sistema en si

Flaky test

Test de código que bajo las misma condiciones de ejecución da errores.

Test double

Test cases que usan mocks y fakes para similar comportamientos

Code coverage

Herramienta que nos ayuda a detectar que líneas de códigos son testeadas, aunque es muy útil estas herramientas no aseguran que los test son efectivos, una forma de saberlo es haciendo uso de los mutation test, si estos no fallan, el code coverage es ineficiente.

Cambios a gran escala en el código

Modificar miles de líneas de código de golpe no es viable. Es mejor dividir los cambios en tareas pequeñas y, si no pueden automatizarse, asignarlas a un equipo especializado o gestionarlas con plataformas como **Busy Beavers**.

Creación del libro

Cada capítulo fue escrito por un experto en la materia, lo que hizo el proceso más largo, pero aseguró un resultado de mayor calidad.

Limitaciones y restricciones

Las empresas deben establecer restricciones en herramientas y sistemas para garantizar escalabilidad y orden, evitando problemas de integración.

El tiempo en el software

Si un software tendrá una vida útil larga, debe diseñarse para ser mantenible y adaptable. Si es temporal, no requiere tanta planificación.

Consejos para ingenieros de software

- **Junior**: Más allá de programar, deben aprender sobre diseño, toma de decisiones y testing.
- **Senior**: Evaluar su ecosistema laboral y mejorar su entorno o considerar un cambio si es necesario.