Miembros:

María González Otero

Lucía García Sopeña

Sergio Murillo Álvarez

Las 4 métricas

El objetivo principal en el desarrollo de software es crear un producto o aplicación de alta calidad. Durante este proceso, los desarrolladores utilizan herramientas que les ayudan a mejorar la calidad del mismo.

Dos cosas que caracterizan a un buen equipo del software son:

1. Buscar objetivo global, de forma que los equipos no se enfrenten entre ellos.
2. En segundo lugar, centrarse en los resultados y no en la producción. Es decir, no se debe recompensar a las personas por realizar grandes cantidades de trabajo que en realidad no ayudan a alcanzar los objetivos de la organización. Se debe recompensar a aquellas personas que hayan hecho un trabajo de calidad.

Medir el rendimiento de la entrega de un equipo de software resulta una tarea complicada. Tener un medio para medir y evaluar la efectividad de la estrategia de desarrollo de software es un factor clave para lograr los objetivos, por ello hoy queremos hablar de las Four Key Metrics, desarrolladas por devops.

Métricas según su función

En un nivel alto, la frecuencia de implementación y el tiempo de entrega de los cambios miden la velocidad, mientras que la tasa de fallas en los cambios y el tiempo para restaurar el servicio miden la estabilidad. Al medir estos valores e iterarlos continuamente para mejorarlos, un equipo puede lograr resultados comerciales significativamente mejores.

DORA, por ejemplo, utiliza estas métricas para identificar equipos de desempeño elite, alto, Medio y Bajo, y descubre que los equipos Elite tienen el doble de probabilidades de alcanzar o superar sus objetivos

¿Para quién están dirigidas?

Se recomienda el uso de estas métricas a todos aquellos desarrolladores que quieran medir el rendimiento de software de su equipo, que tengan un proyecto en GitHub o GitLab o aquellos cuyo proyecto tenga despliegues.

Esto último ocurre por la forma en la que GitHub presenta los proyectos sin despliegues. Ocurre por ejemplo con las bibliotecas.

¿Cómo funciona?

- Los eventos se envían a un objetivo de webhook alojado en Cloud Run. Los eventos son cualquier ocurrencia en su entorno de desarrollo (por ejemplo, GitHub o GitLab) que se puede medir, como una solicitud de incorporación de cambios o un problema nuevo. Four Keys define eventos para medir, y puede agregar otros que sean relevantes para su proyecto.
- El objetivo de Cloud Run publica todos los eventos en Pub/Sub.
- Una instancia de Cloud Run está suscrita a los temas de Pub/Sub, realiza una ligera transformación de datos e ingresa los datos en BigQuery.
- La vista de BigQuery para completar las transformaciones de datos y alimentar el tablero.

1. Lead time

También conocida como el tiempo de espera. Mide el tiempo que un cambio tarda en llegar a producción. Estos cambios pueden ser correcciones de errores, introducir una nueva funcionalidad u otros cambios en el código.

Es un buen indicador de la eficiencia y capacidades del equipo. En el caso de tener un valor muy alto, nos estaría avisando de que existen ineficiencias o cuellos de botella. Según un estudio realizado por Google que ayudaba a encontrar las diferencias entre los equipos de mayor y menor rendimiento, clasificaba la eficiencia de los equipos en 4 grupos: élite, alto, medio y bajo.

El objetivo de una empresa debe ser reducir esta métrica, para ello contamos con dos recursos:

- Herramientas como SENTRIO, ayuda a conocer el lead time de distintos cambios como nuevas implementaciones, arreglo de bugs.... Además, también cuenta con una línea que indica la tendencia, lo óptimo para el equipo sería que descendiese.
- Buenas prácticas como la automatización de pruebas, que ayudan a reducir los tiempos de entrega mediante la integración continua y la entrega continua. Además, también se recomienda trabajar en pequeños cambios frente a grandes cambios, dado que resulta más beneficioso realizar entregas pequeñas y frecuentes para recibir un feedback más rápido y resolver errores antes. Si tardamos tiempo en entregar nuevos cambios, puede que hayan cambiado las necesidades de los usuarios.

2. Deployment frecuency

La segunda clave sería la frecuencia de despliegue, que nos indica la tasa con la que una organización pasa a producción de forma exitosa. Esta métrica es bastante importante, dado que hoy en día la mayoría de los servicios están en la nube.

Una buena práctica es hacer implementaciones más pequeñas pero que sean constantes, esto produce que sea más fácil hacer pruebas y despliegue.

Esta medida puede hacerse mediante aplicaciones como PULSE, aunque podríamos hacer nuestros propios scripts.

El objetivo sería hacer un despliegue bajo demanda, es decir, siempre que sea necesario. Debemos evitar los retrasos que van desde uno hasta los seis meses. Una buena frecuencia indicaría la cohesión del equipo y, su eficiencia y efectividad.

3. Mean time to restore

Conocido como el tiempo medio de restablecimiento o MTTR. Nos ayuda a cuantificar el tiempo de un fallo. Para ello se anotaremos el tiempo que pasa desde que se detecta un error hasta que se arregla. Se utiliza principalmente como valor para establecer en qué nivel dicho fallo está, la manera es la que está perjudicando al sistema y qué prioridad deberíamos de darle para solventarlo lo antes posible, estos fallos pueden ser bases de datos dañadas o con inconsistencias, commits que producen algún error.

Se espera que en equipos de alto rendimiento recuperen los servicios en menos de un día. El objetivo para bajar el tiempo medio de restablecimiento es aumentar la velocidad de implementación con la que arreglamos los errores. Para ello podemos tenemos:

- Herramientas como Better UpTime, que sincroniza la infraestructura con el código y en el caso de que se produzca algún error, nos notificaría el fragmento de código que debemos de arreglar.
- Buenas prácticas como simplificar procesos. Al ser más sencillos, tendremos errores que sean más fáciles de identificar, por lo que corregir errores de funcionalidad o rendimiento nos llevaría menos tiempo dado que tenemos una mejor idea de que partes debemos tocar.

4. Change failure rate

Mide con que frecuencia un cambio de código provoca un fallo en producción. Cambios que producen rollback, que el sistema falle en producción o que se detecte un bug provocado por los últimos cambios.

¿Por que es importante “Change Failure Rate”?

Esta métrica es importante porque todo el tiempo dedicado a resolver un problema es tiempo que no se está usando en liberar nuevas funcionalidades que aporten valor al negocio. Siempre es deseable reducir el número de problemas en los desarrollos.

¿Como se calcula “Change Failure Rate”?

Normalmente, esta métrica se calcula contando el numero de veces que un despliegue acaba en un error y divido por el numero total de despliegues para obtener un valor medio. Un valor medio bajo será siempre bueno.

¿Como se mejora “Change Failure Rate”?

Change Failure Rate puede verse reducido si se mejora:

- Asegurar que todo el codigo esta cubierto por test automatizados.
- Mejorando los procesos de CI para incorporar y ejecutar los test implementados.
- Realizando revisiones de código exhaustivas y completas para ayudar a evitar que se produzcan errores en producción.
- Análisis de dependencias y sistemas sensibles a ser afectados por un cambio.

Los resultados

Con todos los datos ahora agregados y procesados en BigQuery, puede visualizarlos en el panel de Four Keys. El script de configuración de Four Keys utiliza un conector de DataStudio, que le permite conectar sus datos a la plantilla del tablero de Four Keys. El tablero está diseñado para brindarle categorizaciones de alto nivel basadas en la investigación de DORA para las cuatro métricas clave, y también para mostrarle un registro continuo de su desempeño reciente. Esto permite que los equipos de desarrolladores tengan una idea de una caída en el rendimiento desde el principio para que puedan mitigarlo. Alternativamente, si el rendimiento es bajo, los equipos verán los primeros signos de progreso antes de que se actualicen los depósitos.

Bibliografía

https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance

https://itrevolution.com/measure-software-delivery-performance-four-key-metrics/

https://sentrio.io/blog/four-key-metrics/

https://sentrio.io/blog/lead-time-for-changes/

https://blog.codacy.com/how-to-measure-deployment-frequency/

https://www.pulse.codacy.com/

Métricas de DevOps Research and Assessment (DORA) para mejorar los procesos de desarrollo. | by Jose Maria Hidalgo Garcia | Medium

https://github.com/GoogleCloudPlatform/fourkeys