

BEHAVIORAL CODE ANALYSIS (análisis de comportamiento del código)

Abordar la idea del código desde el lado de las personas.

¿QUÉ PROBLEMAS AYUDA A RESOLVER?

Añade mucha información adicional. Por ejemplo, si miramos solo un código en si o una captura estática del mismo, nunca podremos saber el comportamiento la dinámica de lo que sucede en el sistema, por tanto, no podremos diferenciar un código estable, con una deuda técnica baja, de un código volátil que pueda afectar a la organización.

DE DÓNDE SALE LA INFORMACIÓN PARA HACER ESTE ANÁLISIS Y QUÉ HERRAMIENTAS UTILIZA

Lo único que necesitamos son herramientas de control de versiones, como Git. Mediante estos sistemas podemos obtener toda la información de como los desarrolladores han interactuado con el código.

SEGUIMIENTO DE ERRORES CON JIRA

Con este tipo de herramientas podemos mostrar exactamente cuántos errores hubo, por ejemplo, en los últimos 6 meses en un código concreto.

DIFERENCIA ENTRE EL ANÁLISIS DE COMPORTAMIENTO DE CÓDIGO Y EL ANÁLISIS DE CÓDIGO ESTÁTICO

La principal diferencia, es que, aunque el análisis estático sin duda es una herramienta realmente buena, su principal función es identificar mal código, pero nunca te dirá el impacto que tiene dicho código, ni si es una deuda técnica o un desperdicio, mientras que el análisis de comportamiento sí.

OPTIMIZAR LA ORGANIZACIÓN DEL EQUIPO Y LA ARQUITECTURA

El análisis de comportamiento del código permite ver cuando arquitectura y equipo no están alineados, por ejemplo, si hay cuellos de botella en la comunicación entre personas o problemas de fusión de ramas.

CÓDIGO HEREDADO

El análisis del comportamiento del código puede identificar que una parte del código fue hecho por otro equipo diferente.

PROYECTOS GREENFIELD

El análisis del comportamiento del código funciona en proyectos Green Field a partir de pocas semanas cuando ya se tienen suficientes datos. Permite ver a tiempo malas conductas que pueden provocar problemas futuros en el proyecto.

INTEGRACIÓN EN EL CICLO DE VIDA DEL DESARROLLO SOFTWARE

Para integrar el análisis del comportamiento del código en el desarrollo de un sistema hay varias maneras. Una es hacer un análisis en cada pull request para asegurarse que no baja la calidad y no se añade deuda técnica.

FORMA DE EVALUAR

Este método es bueno para poder saber el rendimiento del equipo, pero éticamente sería incorrecto debido a que los desarrolladores van a empezar a jugar con las métricas en vez de trabajar en común hacia un objetivo perdiendo además la dinámica de equipo, ya que se centraran más en hacer sus respectivos commits en vez de apoyar a los compañeros.

CÓMO SABEMOS QUE FUNCIONA

Es efectivo porque identifica patrones, detecta anomalías y mejora la calidad del código.

CÓMO LO USAMOS

Si observamos el código y ves lo que realmente está sucediendo con el código a medida que evoluciona el sistema se puede ver como hay unas partes que cambian más que otras.

Las partes que más cambian es en la que más trabajo se realiza y es muy relevante para nuestra hoja de ruta y es donde se debe dejar el código más limpio que la manera inicial.

IA

Priorización de pull requests: Los algoritmos de machine learning son capaces de analizar datos de comportamiento del código para predecir el riesgo asociado con una solicitud de incorporación de cambios específica. Esto permite marcar las solicitudes de incorporación de cambios con más riesgo para una revisión más detallada, y aquella con un riesgo menor marcarlas para simplemente ser ojeadas.

CÓMO EMPEZAR EN EL ANÁLISIS DE COMPORTAMIENTO DEL CÓDIGO

El entrevistado sugiere que cualquier equipo que empiece a realizar un análisis de comportamiento del código debe hacerlo principalmente por la parte más simple, siendo esta el análisis de puntos críticos o hotspots. Puesto que tener estos puntos controlados son la base para continuar con cualquier otra cosa.

CODEMAAT HERRAMIENTA PARA REALIZAR UN ANÁLISIS DE COMPORTAMIENTO DEL CÓDIGO

Es una herramienta open-source de línea de comandos creada por el entrevistado (Adam Tornhill) con la que se puede llevar a cabo un análisis de puntos críticos, de acoplamiento de cambios y algún otro relacionado con la distribución del conocimiento en la base del código.

REFERENCIAS

- <https://codescene.com/blog/validation-for-behavioral-code-analysis/>
- <https://codescene.com/blog/code-analysis-tool/>
- https://en.wikipedia.org/wiki/Legacy_system