Deuda Técnica

Tabla de Contenidos

El cálculo de la deuda técnica está basada en la metodología SQALE (Software Quality Assessment based on Lifecycle Expectations).

SQALE es una metodología que fue desarrollada por inspearit y fue cedida como código abierto. Si usted lee la documentación en sqale.org, podrá observar se trata de "la organización de los requisitos no funcionales que están relacionados con la calidad del código".

En la implementación del método SQALE de SonarQube, los requisitos no funcionales son las reglas de codificación en su perfil de calidad.

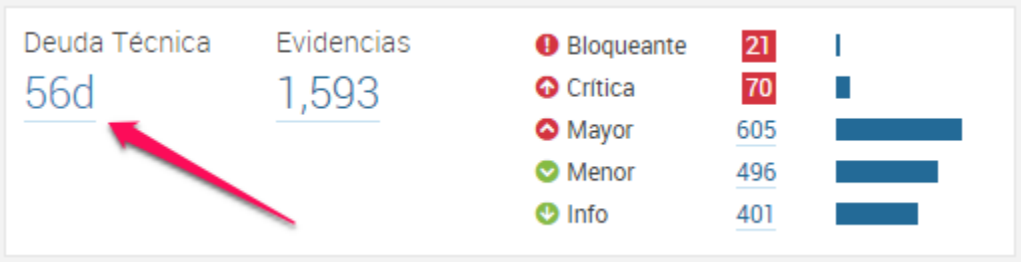
Efectivamente, la implementación de SQALE de SonarQube se basa únicamente en reglas y evidencias. Esto significa que si se desea gestionar toda la deuda técnica con SQALE, primeramente se necesitará habilitar las reglas en el repositorio común de SonarQube que etiquetan:

- Los bloques duplicados
- Las pruebas unitarias falladas
- Ramas cubiertas por las pruebas unitarias insuficientes
- Densidad de comentarios insuficientes
- Cobertura de líneas cubierta por pruebas unitarias insuficientes
- Pruebas unitarias omitidas

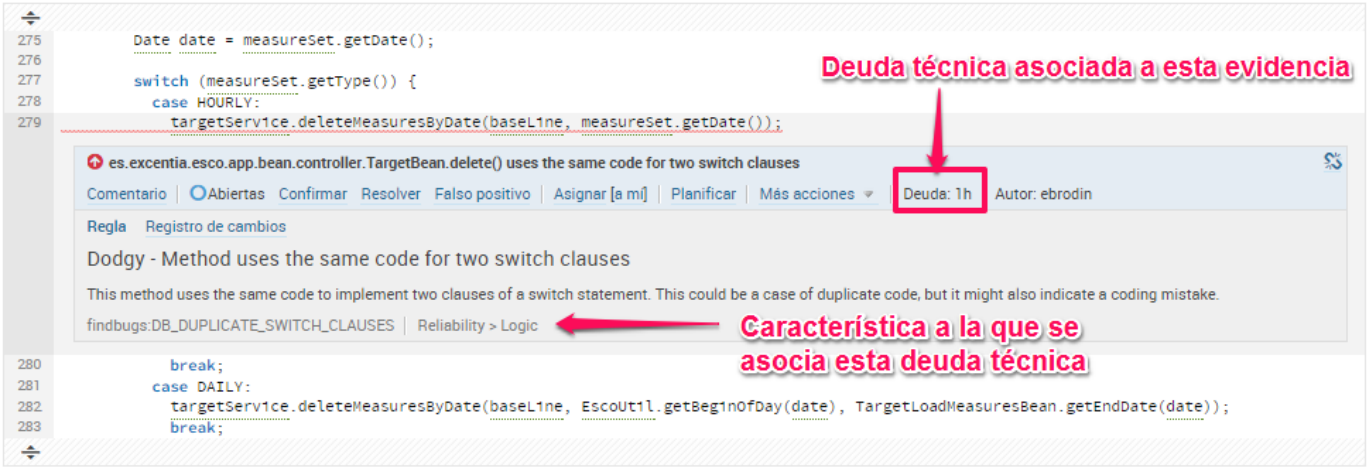
Estas reglas están en el repositorio común de SonarQube porque son comunes a todos los lenguajes. Una vez habilitadas se puede realizar seguimiento de cada fallo como una evidencia y realizar seguimiento de la deuda técnica, que el modelo SQALE mide en días.

Cálculo de la deuda técnica

En el cuadro de mando por defecto, se puede ver un resumen de la deuda técnica del proyecto con el widget "Evidencias y Deuda Técnica":



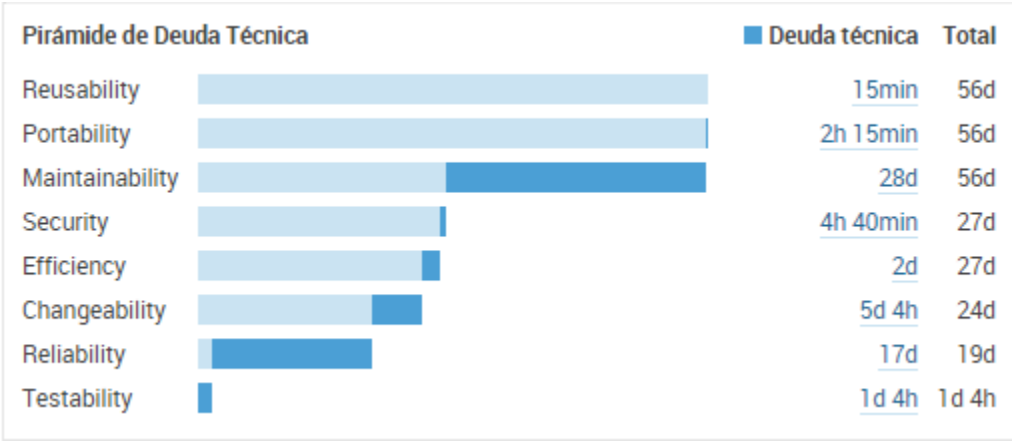
Esa medida en días se realiza sumando la deuda técnica asociada a cada evidencia, que puede verse en el bloque de evidencias cuando se abre un fichero de código:



La deuda técnica de cada evidencia se establece a nivel de la regla. Si se tiene el plugin comercial SQALE, se puede ajustar la estimación de cada regla (al hacer esto, se está editando el Modelo de Análisis SQALE - el coste asociado a la resolución de cada regla). Sin embargo, estas estimaciones están realizadas por profesionales expertos de forma que no se requiera modificarlas.

¿Cuando empezar a pagar la deuda técnica?

Se conoce el coste asociado a resolver la deuda técnica, pero ¿como priorizar este trabajo? Existe un widget para ello, denominado la pirámide de deuda técnica, que no se parece a una pirámide como comúnmente la entendemos:



No se confunda por el hecho de que no se parezca a una pirámide como las del antiguo Egipto; se trata de una pirámide figurativa. La forma de interpretarla es desde la base hacia arriba. La linea base siempre tiene la barra mas pequeña en el gráfico, pero el coste asociado más alto ya que es fundacional. Las filas en este widget representan una "característica" y cada característica se constituye a partir de las que tiene debajo. La testeabilidad está en la base porque es más importante: primero hay que asegurarse de que la aplicación es testeable, entonces se puede asegurar la fiabilidad, después la adaptabilidad, eficiencia...etc.

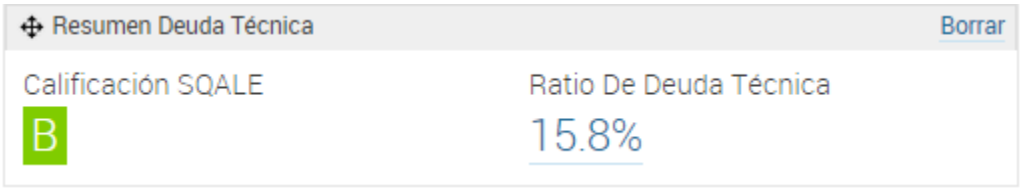
Las barras en el grafo muestran el tiempo de resolución por característica. El linea azul oscura muestra el tiempo para solucionar esta característica, y la porción azul claro el tiempo acumulado desde la base. Como es usual, cada porción del widget es pulsable y nos lleva a una ventana de servicio de inspección para poder ver exactamente donde está la deuda técnica de una característica:

Comparando proyectos con el Ratio de Deuda Técnica y el Ratio SQALE

Es cierto que es muy interesante ver que el equipo debe invertir 95 días para poder resolver la aplicación. ¿Pero cómo comparo este proyecto con otro que tiene aproximadamente la misma deuda técnica pero es mucho mas pequeño?

En este momento es cuando la métrica Ratio de Deuda Técnica es interesante. Esta métrica proporciona la ratio entre la deuda técnica actual y el esfuerzo que se debería de invertir para reescribir todo el código desde el principio. Esto último se estima basándose en el número de lineas de código o la complejidad global. Esto significa que el valor de esta métrica depende del tamaño del proyecto. Dado un valor de esta métrica, el proyecto recibe una calificación SQALE desde A (mejor cualificación) hasta E (peor calificación).

A continuación se puede ver el widget "Resumen Deuda Técnica":



Estas dos métricas se computan a nivel de fichero (y de esta forma visible en el visor de componentes) y ase agregan a nivel de proyecto.

Ejemplo

Considérese dos proyectos con la misma deuda técnica:

- Proyecto *Foo*: 95 días y 80.000 líneas de código
- Proyecto *Bar*: 95 días y 7.000 líneas de código

La fórmula de ratio de deuda técnica es: "deuda_técnica" / "coste_estimado de desarrollo". Y por defecto (puede cambiarse en las herramientas de SonarQube), el "coste_estimado de desarrollo" se computa como "LC x 30 minutos". Esto devuelve (la deuda técnica debe convertirse a minutos y el número de horas por día se puede personalizar):

- Proyecto *Foo*: (95 x 8 x 60) / (80 000 x 30) = **1,9%**
- Proyecto *Bar*: (95 x 8 x 60) / (13 000 x 30) = **21,7%**

Es obvio que el proyecto *Foo* presenta mejores resultados que el proyecto *Bar*.

De esta forma, usando la tabla de clasificación por defecto (esto también se puede personalizar), el proyecto *Foo* tiene una "**A**" mientras que el proyecto *Bar* recibe una "**C**"

Yendo más allá

El plugin SQALE de SonarSource extiende la característica de deuda técnica de SonaQube. Entre otras características, permite ajustar el modelo SQALE (ajustar las estimaciones de resolución para cada regla, establecer la lista y el orden de las características, cambiar a que características pertenece una regla, y más) provee widgets adicionales, etc.