

Pràctica 3: **sonarqube**

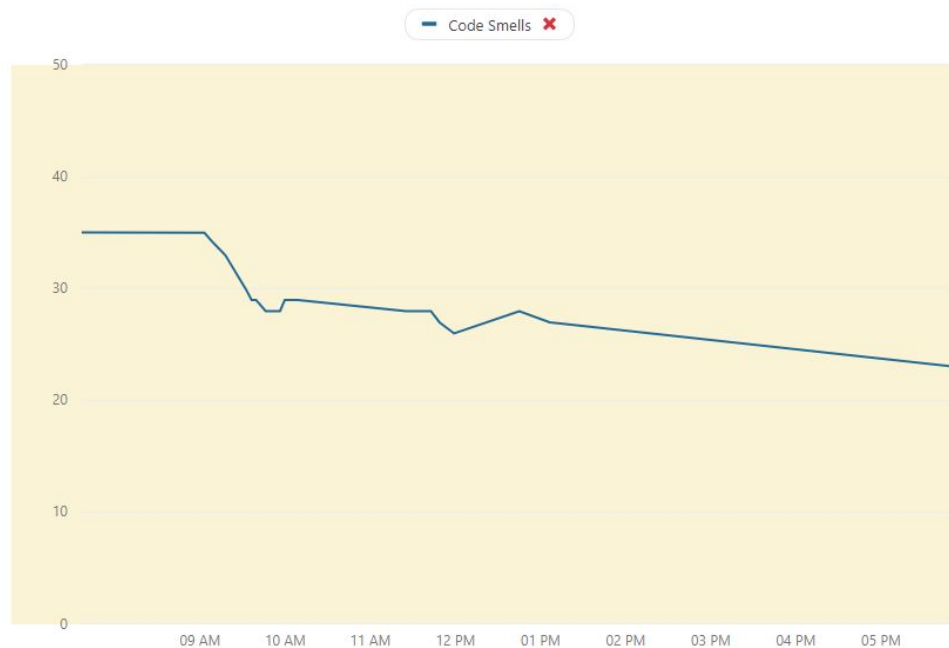
Ens vem repartit els error del projecte “*Petstore*” entre els membres del grup. Donant-li prioritats als més importants. Vem tenir alguns problemes per corregir alguns *bugs* i no els hem pogut acabar tots. Un cop corregit la majoria dels errors o mals usos del projecte, analitzarem les dades que ens proporciona **sonarQube**.

Al principi teniem **49 “Code Smells”** (mal usos), **8 Bugs** i **7 vulnerabilitats**. En conseqüència les valoracions de fiabilitat i seguretat no arriben a la A. El projecte no asoleix els llindars de qualitat per defecte el qual considera el “*Quality Gate*”. De moment és del 0%. Això és normal, ja que serà la feina a realitzar: arribar a una cobertura de com a mínim 80% per obtenir l'anomenat “*Quality Gate*”.



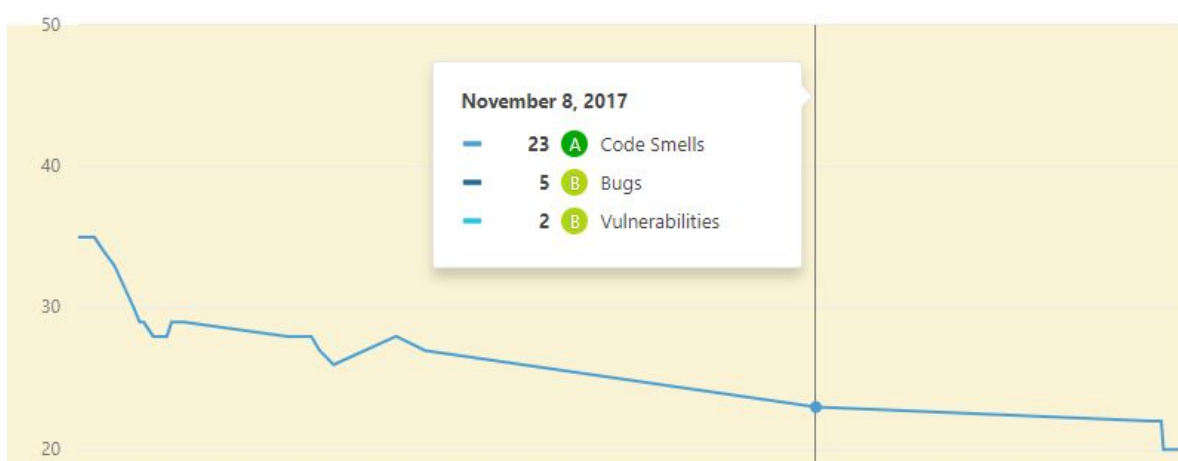
Imatge 1: Gràfic Code Smells, Bugs i Vulnerabilities 26 d'Octubre 2017

En alguns punt de la gràfica els “*Code Smells*” augmenten en comptes de disminuir això es degut a que ens estem adaptant a l’entorn de treball.



Imatge 2: Gràfic Code Smells.

Un cop solucionat els problemes d’adaptació amb l’eina vem conseguir disminuir els “*Code Smells*” a 23, els *Bugs* a 5 i les vulnerabilitats a 2.

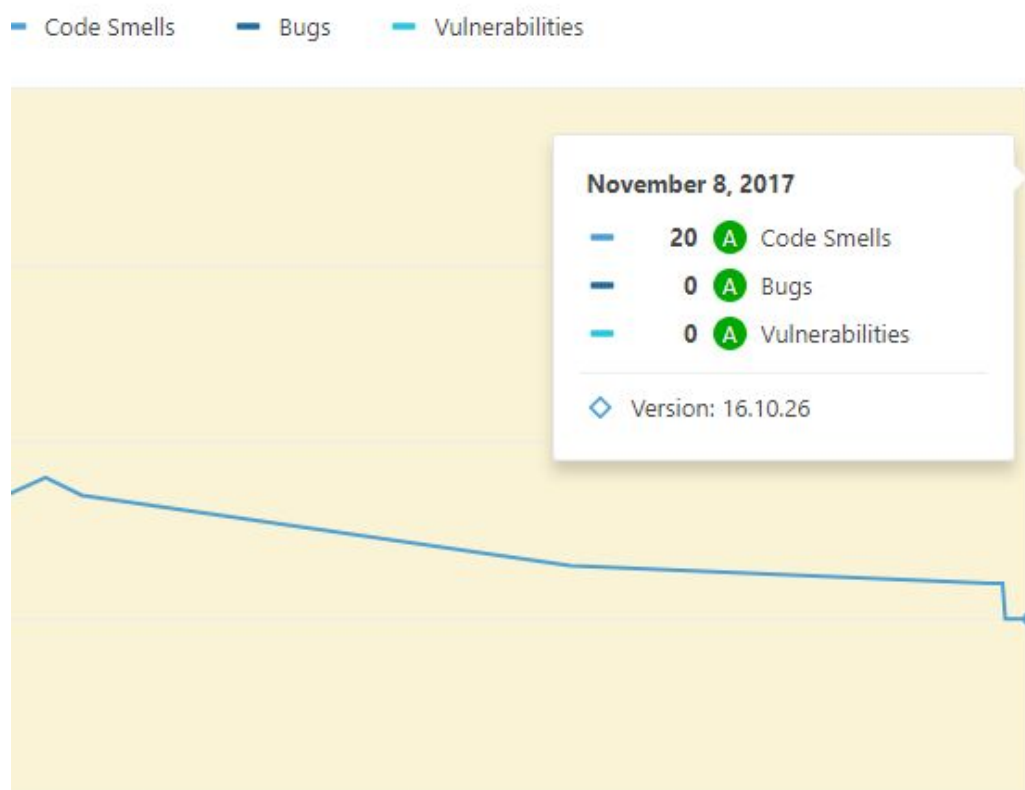


Imatge 3: Dades Code Smells, Bugs i Vulnerabilities.

D'aquesta manera les valoracions de fiabilitat són de B. Ens marcarem l'objectiu d'arribar a la valoració d'A en els *Bugs* i vulnerabilitats sense olvidar els 23 *Code Smells*.

Hem tingut bastants problemes amb el projecte. Al principi de tot no ens compilava, això ens va portar molta feina solucionar-ho. Hem tingut algun *marge conflict* que això ens ha fet que augmentessin els *Code Smells* degut a que hem hagut d'anar a versions anteriors del projecte.

Entre els errors que hem anat trobant els més habituals han sigut el de *Double Brace initialization*, variable que no s'utilitzaven, declaració d'atributs i no utilització del *try* i del *catch*.



Imatge 4: Dades finals Code Smells, Bugs i Vulnerabilities.

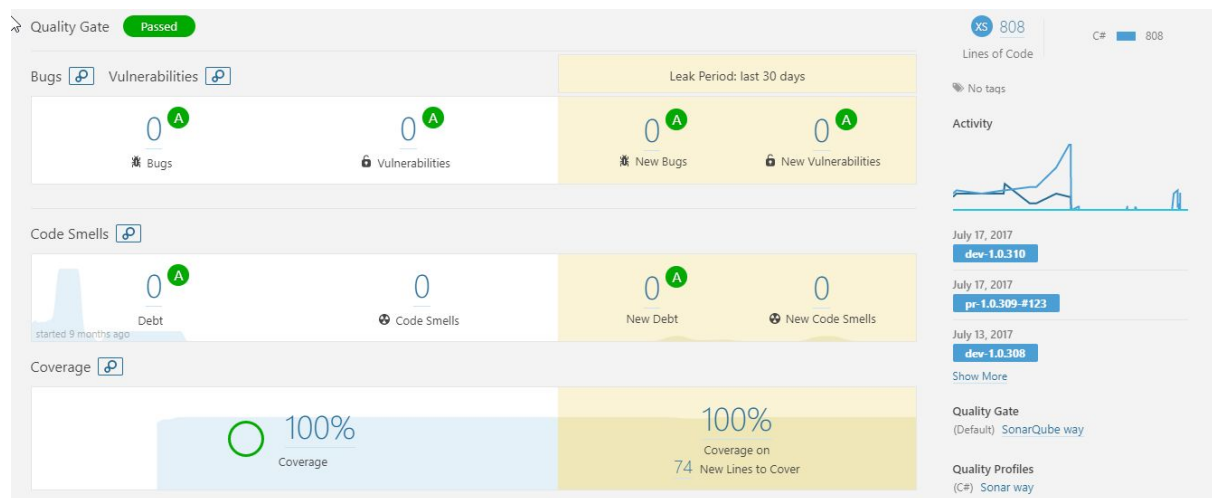
Després de solucionar tots els problemes amb l'entorn i posar-nos a treballar de forma eficient hem aconseguit reduir els "*Code Smells*" a 20, els *Bugs* a 0 i les vulnerabilitats a 0. Ens hem fixat de que un "*Code Smells*" no es pot treure i l'hem posat com a "*Won't Fix*". Hem aconseguit una valoració d'A i reduir el deute tècnic a un 12'5%, ara es de 6 hores. Hem aconseguit que la valoració de la sostenibilitat sigui A.

En conclusió hem aconseguit totes les valoracions d'A que era el nostre objectiu inicial, tant en seguretat com sostenibilitat i fiabilitat. Per contra ens han quedat masses "*Code Smells*" sense solucionar per falta de temps i ho hem de millorar. El nostre següent objectiu serà reduir els "*Code Smells*" i aconseguir el 80% per obtenir el "*Quality Gate*".

sonarcloud

En este breve documento se analizará el proyecto colgado en Sonarcloud.io:

https://sonarcloud.io/dashboard?id=04_KataFahrplanauskunft



En este proyecto desarrollado en C# vemos que cualquier error ha sido solucionado de un total de 70 problemas que había en un principio. También uno de los problemas ha sido marcado como que no se va a reparar ya que suponemos es imposible de arreglar.

Cuando entramos en el apartado de los problemas solucionados vemos que en su caso hay de dos tipos, del tipo Bug que significa que ahí hay un posible causante de un bug para la plataforma sonar cloud y luego hay Code Smells que son líneas de código a mejorar que tienen un patrón coincidente con la base de datos de sonarqube y que son potencialmente mejorables; estos errores van desde poner el

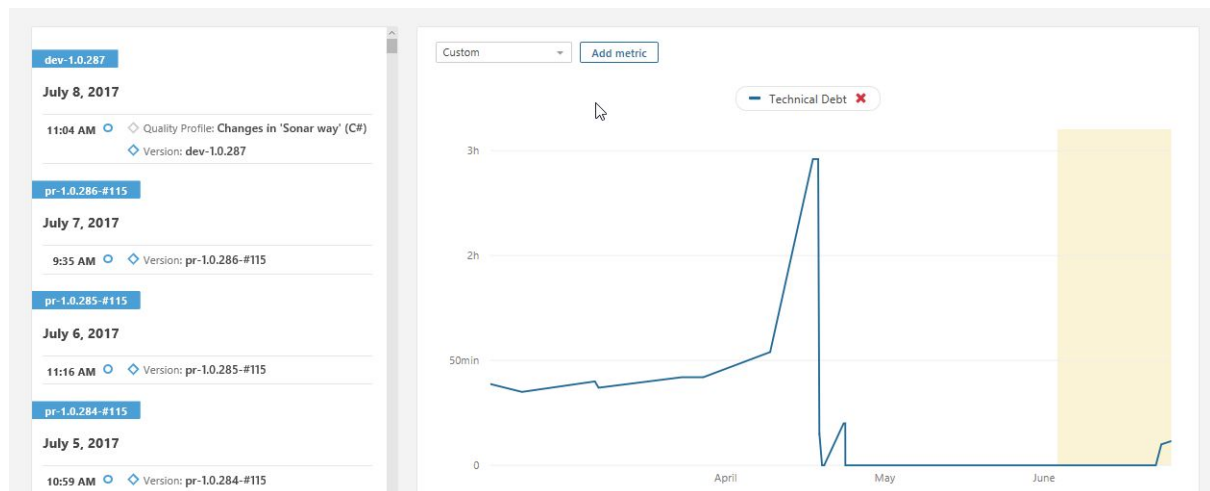
tratamiento de una excepción hasta errores de nomenclatura de variables y funciones.

En cada error se ve el estado del mismo (en este caso están tratados todos menos el que se ha decidido obviar), el tipo de error (comentado anteriormente), el grado de criticidad (de Blocking a Minor en orden descendente), quien ha sido asignado para arreglar el error y el tiempo potencial que estima la plataforma en el que se va a reparar esa incidencia.

En el proyecto, la plataforma marca que no hay ninguna repetición de código, cosa que es un buen indicador ya que si existe repetición de código suele bajar la cambiabilidad del sistema que se está desarrollando.

Nunca ha habido errores de seguridad lo que significa que el código nunca ha tenido vulnerabilidades respecto a posibles entradas maliciosas ajenas. Si las hubiese habido, probablemente la plataforma las habría marcado como error de vulnerabilidad.

En cuanto a deuda técnica marca que hay 0 pero en un principio era de 35 minutos, es decir que se ha arreglado todo lo que antes producía errores o estaba mal implementado. La deuda técnica mide los días que se requerirían para arreglar el sistema en base a los diferentes parámetros establecidos en sonarqube.



El último aspecto a tratar es el de la cobertura, en este proyecto sonar cloud afirma que todas las pruebas se han pasado y que no hay ninguna línea sin dejar de probar, lo que indica que el código se ha comprobado de manera correcta. Los informes sobre la cobertura son ofrecidos mediante un plugin de sonar cloud.