

Retour d'expérience avec Sonarqube pour le contrôle-commande de la caméra du télescope LSST

Webinaire Sonarqube, retour d'expérience

Le 12 janvier 2017

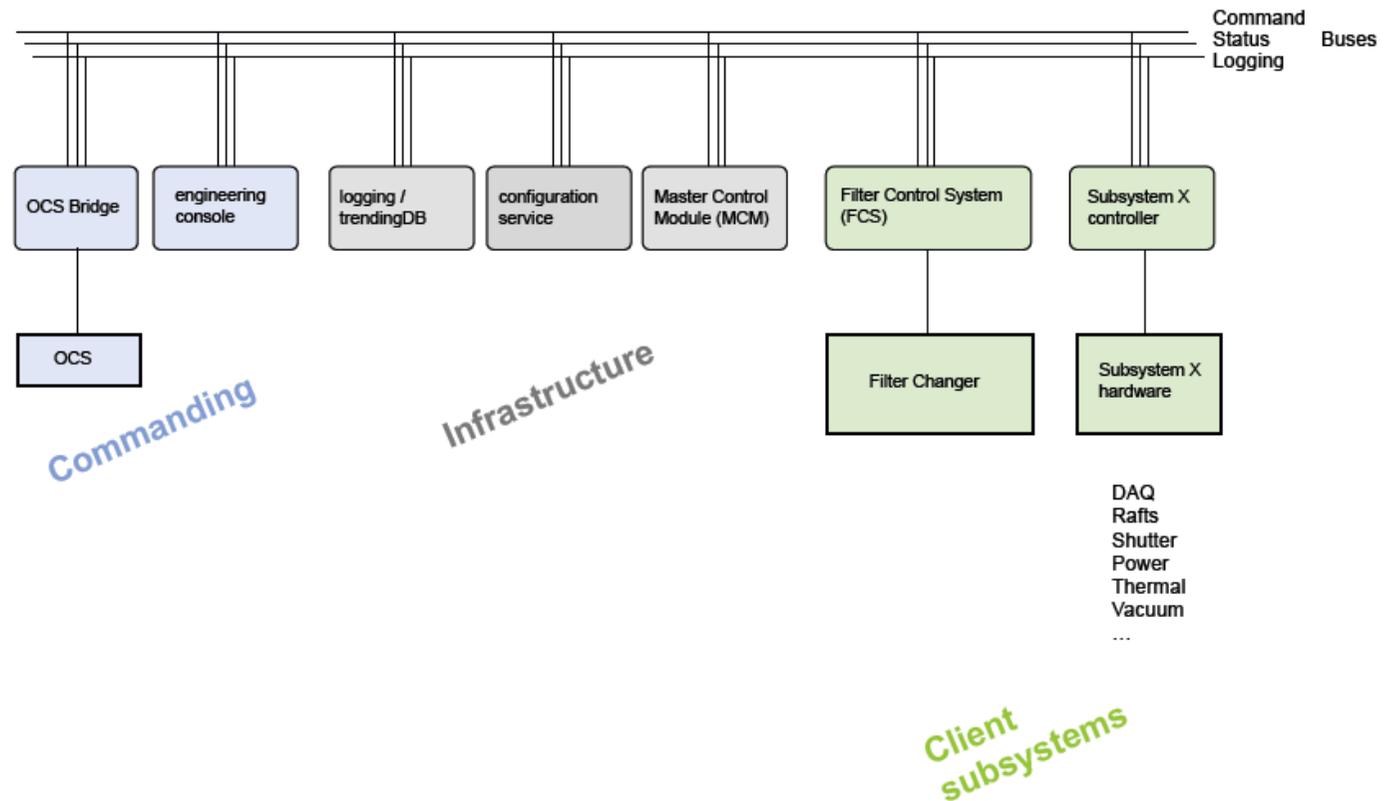
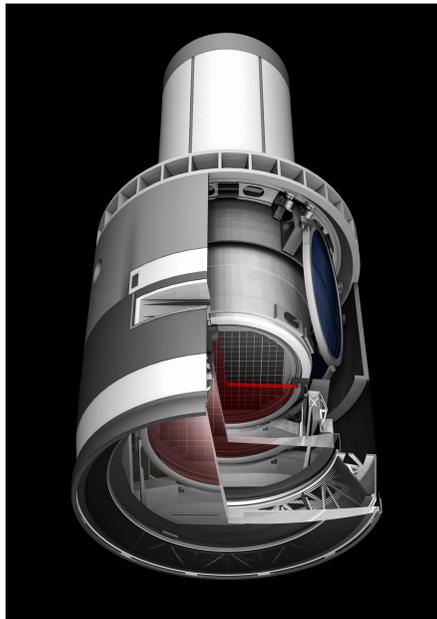
Françoise Virieux, APC

virieux@apc.in2p3.fr

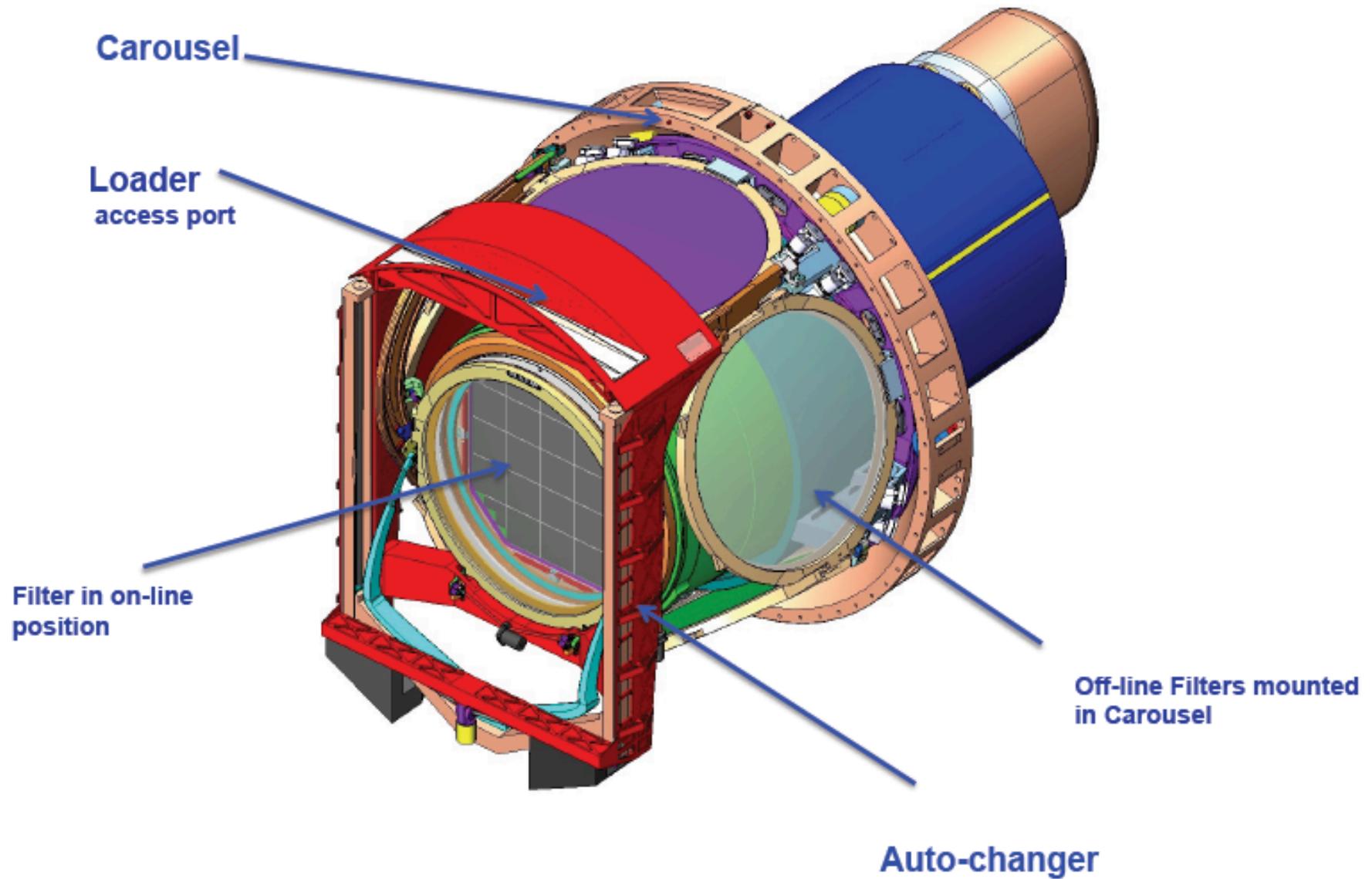


Contexte : Le CCS, Camera Control System

- Caméra de LSST est constituée de sous-systèmes : plan focal, changeur de filtres(6 filtres) , obturateur, réfrigération, cryostat, alimentation électrique, etc...
- CCS : logiciel de contrôle-commande de toute la caméra de LSST et de sa dizaine de sous-systèmes.
- Ecrit en Java et C, 100000 lignes de code, 15 développeurs en France et aux US



Le changeur de filtres de la caméra de LSST



- **Ce que Sonarqube fait**
 - Analyse statique du code par rapport à des règles de programmation, des standards, des bonnes pratiques, des règles de sécurité
 - Affiche sur une interface web les résultats de l'analyse et des indicateurs de qualité du code (en fonction des règles choisies)
- **Ce que Sonarqube ne fait pas**
 - Ne dit pas quelles règles utiliser dans un contexte particulier
 - Ne dit pas si le code fait ce qu'il doit faire
 - Dans le contexte online : ne remplacera jamais les tests sur le matériel !!!
- **Bienfaits de l'utilisation de Sonar dans un projet**
 - Oblige les développeurs à se mettre d'accord sur les règles à respecter
 - Aide chaque développeur à savoir ce qu'il faut faire pour améliorer son code
 - Distille une habitude de bonnes pratiques dans l'équipe
 - Incite à bien tester le code

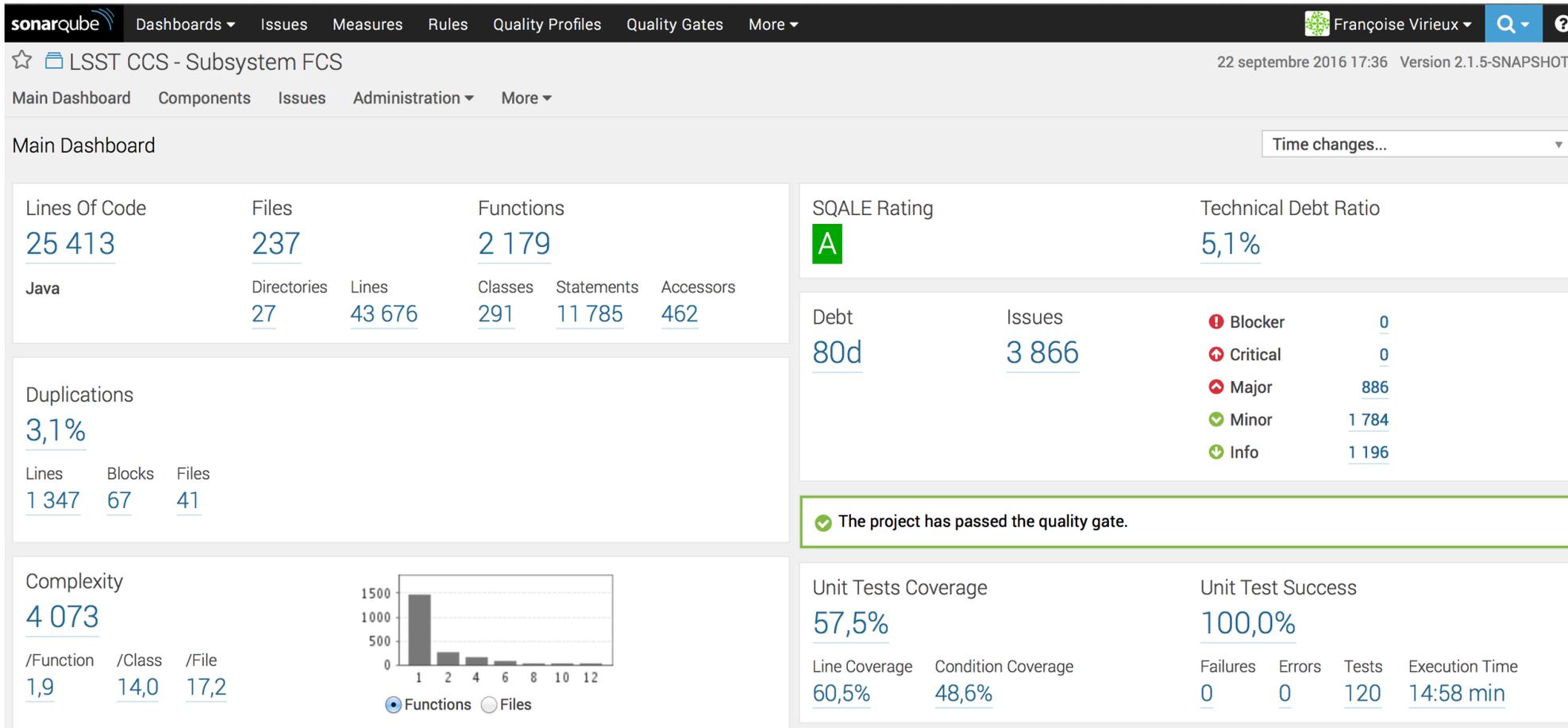
- **Risques importants sur notre projet => comment produire du code de qualité pour éviter une catastrophe ?**
 - **Un filtre de LSST : \$700000**
 - **Volume de données de l'expérience 15TO/j**
- **Nous pratiquons déjà l'analyse statique de code**
 - **Sur serveur Jenkins : analyse avec Findbugs**
 - **Mais utilisation laborieuse :**
 - **Ajout, suppression de règles,**
 - **Lecture des résultats de l'analyse**
- **Élément déclencheur : Ecole Informatique 2015 : Outils pour la mise en production du logiciel**
 - **Utilisation de Sonarqube sur un portable pendant quelques mois pour le sous-système FCS, contrôle-commande du changeur de filtres**
 - **Puis installation d'un serveur Sonar pour tout le projet CCS**
 - **Adoption par toute l'équipe CCS**

Un peu de vocabulaire

- **Dette technique :**
 - temps qu'il faudrait pour résoudre tous les problèmes détectés (s'exprime en jours/homme)
- **Ratio technique :**
 - dette technique rapportée au nombre total de lignes
- **Nombre de problèmes détectés**
- **Quality Profile**
 - ensemble de règles utilisées pour analyser le code
- **Quality Gate**
 - défini la hauteur de la barre que l'on veut passer
- **Code Coverage**
 - indicateur du nombre de lignes de code couvertes par les tests.

Résultats de l'analyse du CCS par Sonarqube

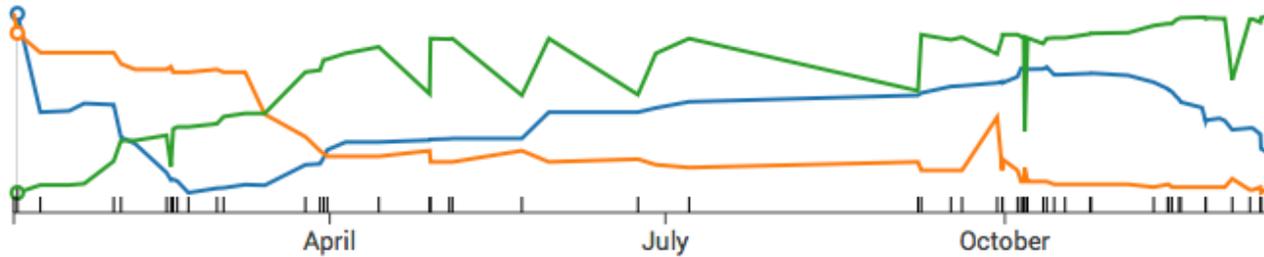
- Démo si Internet marche sinon copies d'écran ☹



Démo sans Internet (FCS timeline)

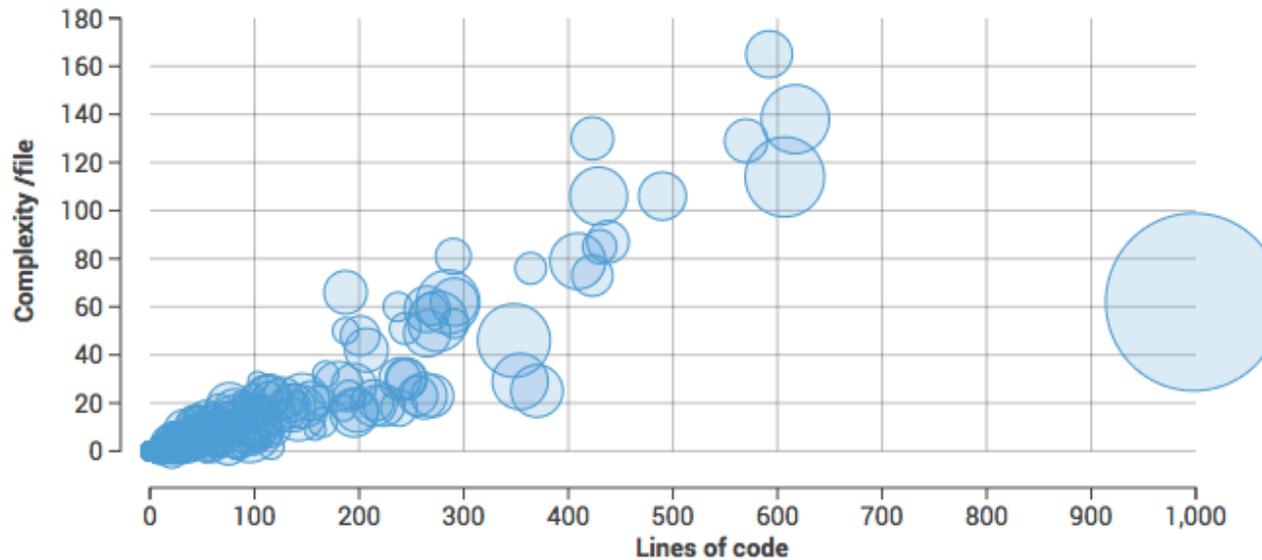
LSST CCS - SUBSYSTEM FCS

7 janvier 2016 ● Complexity: 4 283 ● Technical Debt Ratio: 10,0% ● Overall line coverage: 36,5%
Green (was Red), Changes in 'CCSProfile' (Java)



LSST CCS - SUBSYSTEM FCS

Complexité



Démo sans Internet – ma vue préférée

Issue Tag Cloud

bad-practice **brain-overload** bug cert confusing
convention cwe design error-handling java8 junit misra
multi-threading obsolete owasp-a9 performance pitfall sans-top25-risky
security unused

Demo sans Internet - Issues

LSST CCS - Subsystem FCS LSST CCS - Subsystem FCS Main src/main/java/org/lsst/ccs/subsystems/fcs/AutoChangerModule.java

The Cyclomatic Complexity of this method "checkConditionsForOpeningLatches" is 11 which is greater than 10 authorized. ... il y a 6 jours ▼ L308 ⚙️ ▼
🔴 Major ▼ 🟡 Open ▼ Not assigned ▼ Not planned ▼ 11min debt Comment 🧠 brain-overload ▼ >

LSST CCS - Subsystem FCS LSST CCS - Subsystem FCS Main src/main/java/org/lsst/ccs/subsystems/fcs/AutochangerTwoLatches.java

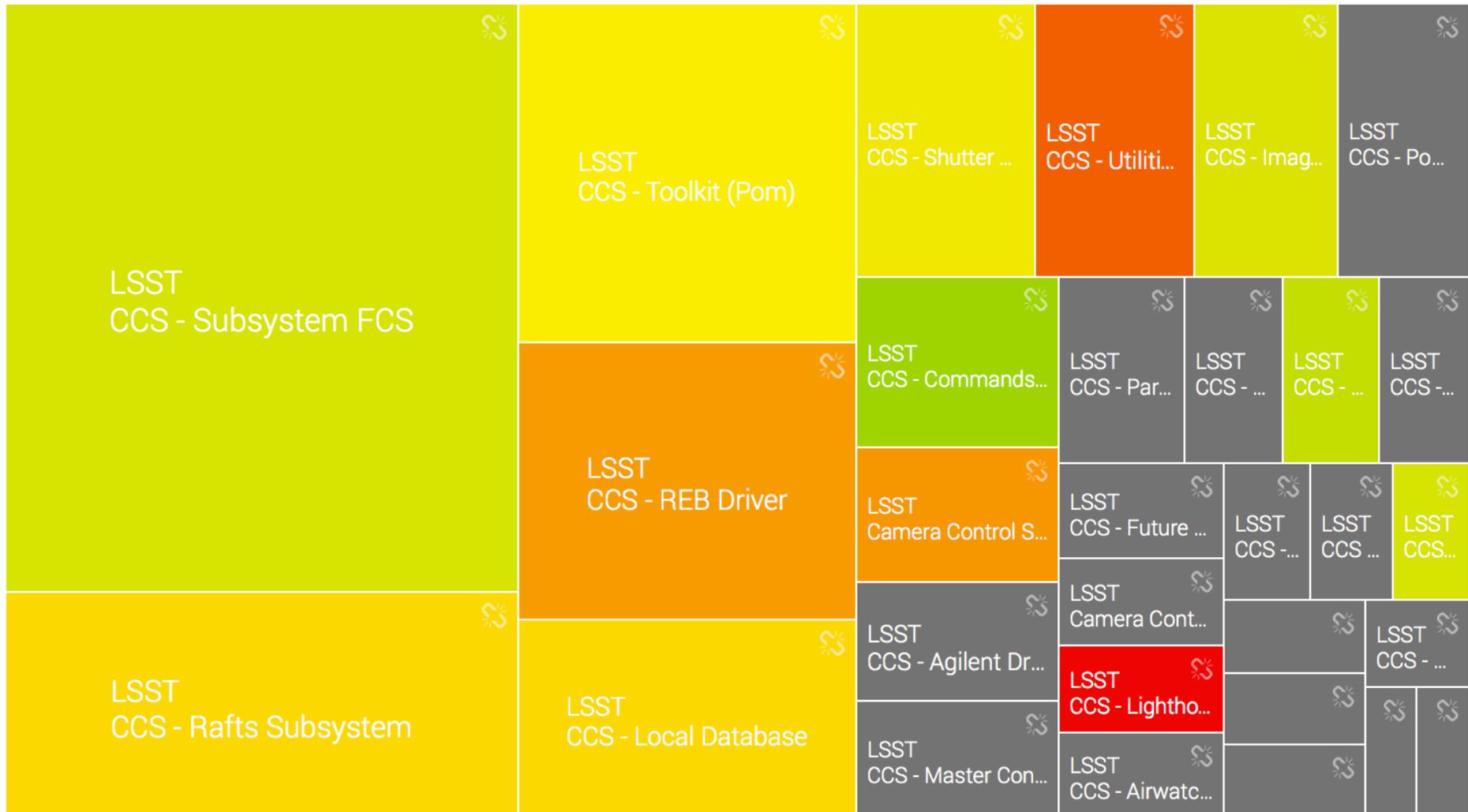
Reduce this switch case number of lines from 6 to at most 5, for example by extracting code into methods. ... il y a un an ▼ L320 ⚙️ ▼
🔴 Major ▼ 🟡 Open ▼ Not assigned ▼ Not planned ▼ 5min debt Comment 🧠 brain-overload ▼ >

Reduce this switch case number of lines from 6 to at most 5, for example by extracting code into methods. ... il y a un an ▼ L327 ⚙️ ▼
🔴 Major ▼ 🟡 Open ▼ Not assigned ▼ Not planned ▼ 5min debt Comment 🧠 brain-overload ▼ >

Démo sans Internet – Taux de couverture de tests 1

PROJECTS

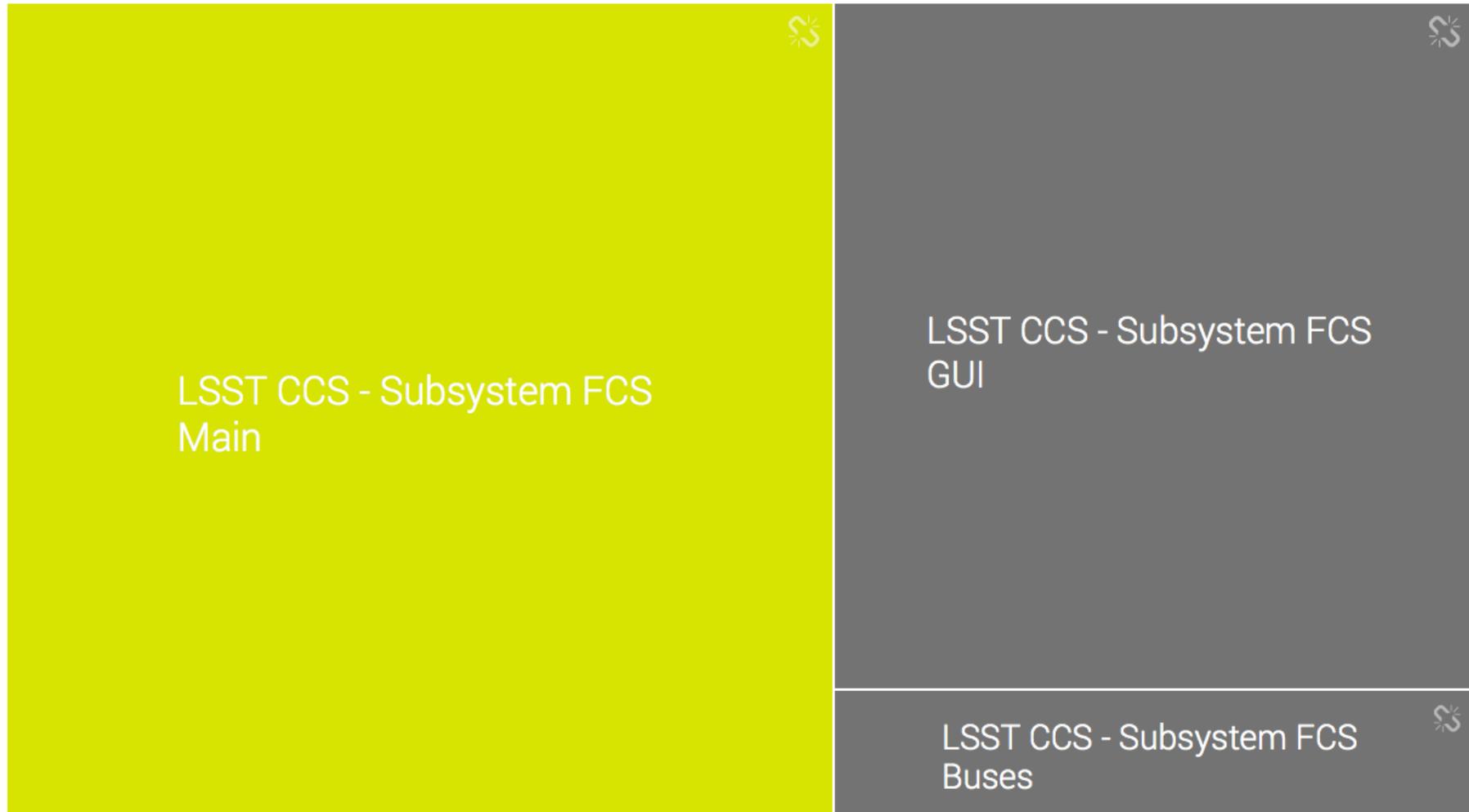
Size: Lines of code Color: Coverage



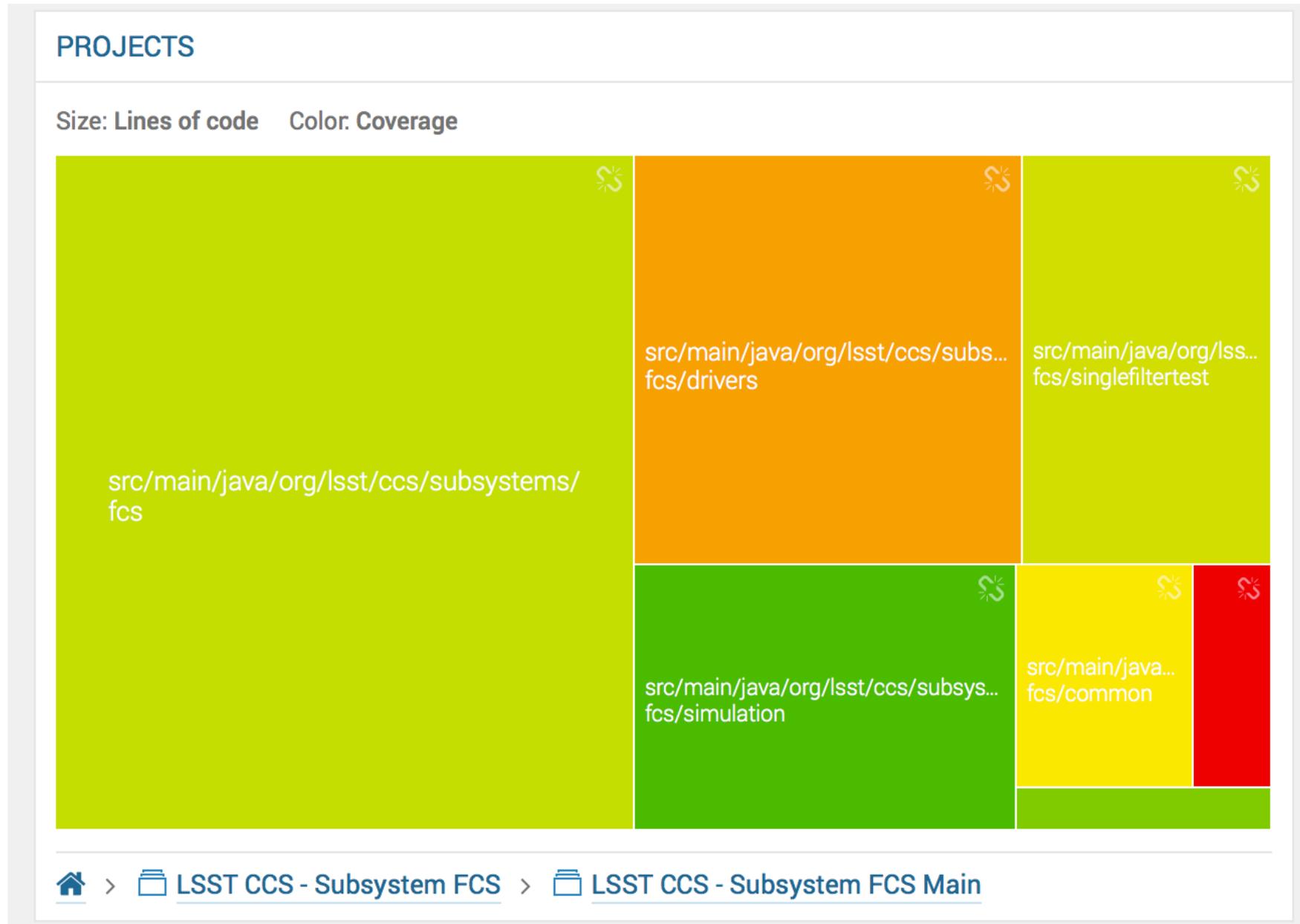
Démo sans Internet – Taux de couverture de tests 2

PROJECTS

Size: Lines of code Color: Coverage



Démo sans Internet – Taux de couverture de tests 3



Conclusion

- **Sonarqube c'est un outil**
 - **Il faut apprendre à s'en servir.**
 - **Il ne faut pas lui demander plus que ce qu'il peut donner.**
- **Son apport sur le logiciel CCS pour LSST**
 - **Discussion sur les règles à utiliser**
 - **Augmentation du taux de couverture de code**
 - **Code beaucoup plus documenté**
 - **Diminution importante du nombres de lignes de code (-10%)**
 - **Apporte une culture des bonnes pratiques**

