

DE LA RECHERCHE À L'INDUSTRIE



French Science
Ground Segment
Bilan DC-0

23 et 24 Janvier 2019

IAP - Paris



Irfu

Institut de recherche
sur les lois fondamentales
de l'Univers

www.cea.fr

Valérie PONS - NEXEYA C&F



1^{ère} PARTIE :

- **Bilan DC-0,**
- **Activités menées,**
- **Activités à venir,**
- **Quelques rappels pour l'intégration**
- **Cartographie du Code JAVA-PYTHON**

2^{ème} PARTIE :

- **Éléments complémentaires Code**

Démarrage : 17/10/2018

Clôture : 23/01/2019

Les Plans de Projet :

- **Le Plan de développement du FSGS : livré au CNES fin Novembre**
- **Le Plan Qualité et le Plan de Gestion de Configuration simplifié : livrés au CNES semaine 51.**

La SRS : FSGS System Requirements Specifications

- **Livrée au CNES semaine 51.**

La Spécification de Test du DC-0 :

- **Contient les composants intégrés,**
- **Une définition exhaustive des tests à exécuter (~ 50 tests),**
- **Contient aussi le résultat des tests.**

Sous POLARION : <https://dsm-polarion.cea.fr/polarion/#/project/Svom2015/home>

Les Plans : Plan Qualité et le Plan de Gestion de Configuration du Projet **simplifiés**

- **ont été validés globalement par le Chef de Projet**
- **et soumis au CNES pour validation semaine 51** (compatibilité avec les exigences qualité imposées : « Spécification Assurance Qualité Logiciel pour les développements avec des laboratoires » – DNO/DA/AQ-2017-0016646 – ED1, Rev0 du 07/11/2017)

La SRS :

- **Rajout dans la MOCD-Pol des exigences de l'IC FPOC**
- **Contrôle de la livraison de la SRS**
- **Proposition d'amélioration du Work Item Requirement dans Polarion** (Traceability Rules for Documents Generated with POLARION, in interface with CNES specifications – SV-PRO-AN-865-CNES- 19/03/2018)

Contrôle de Code pour le DC-0

La SRS :

- **Mise à niveau de la SRS, et génération de la matrice de traçabilité dès que le WI Requirement sera implémenté :**
 - Rajout du mode de vérification
 - Génération de la matrice de traçabilité avec les exigences de haut niveau (développement d'une macro Polarion)

Mise en place d'un workflow de « respect des bonnes pratiques » des composants candidat à l'intégration

- **Cycle de contrôle Auteur/Vérificateur (Chef de Projet et/ou SQAM) avec Jira**

Contrôle de Code pour le DC-0 :

- **À poursuivre, dès que l'usine logicielle sera disponible (Fin Février)**

Finaliser Bilan Qualité du DC-0

Ce que vous devez faire :

- Rédiger le paragraphe descriptif de vos applications dans <https://drf-gitlab.cea.fr/svom/Tools/wikis/application-directory> et y faire référence à la page d'instructions pour l'installation.
- Rédiger les instructions d'installation en suivant le modèle <https://drf-gitlab.cea.fr/svom/Tools/wikis/instructions-template> ou les instructions déjà en ligne.
- Compléter le fichier README.md de déploiement sur la page <https://drf-gitlab.cea.fr/svom/svom-deployment>
- Mettre à jour et vérifier toutes vos pages de documentation sur gitlab.
- Mettre à jour et vérifier vos WI de test dans Polarion (DC-0 Spécification)
- Nettoyer la page <https://drf-gitlab.cea.fr/svom/> en supprimant ou en mettant au grenier les dossiers obsolètes.

La méthode

- **Analyse de la volumétrie avec l'outil gratuit cloc – count line of code**
Lien : <http://cloc.sourceforge.net/#Overview>
- **Cartographie des modules et des langages**
- **Contrôle de code sur du langage C (2^{ème} Partie)**

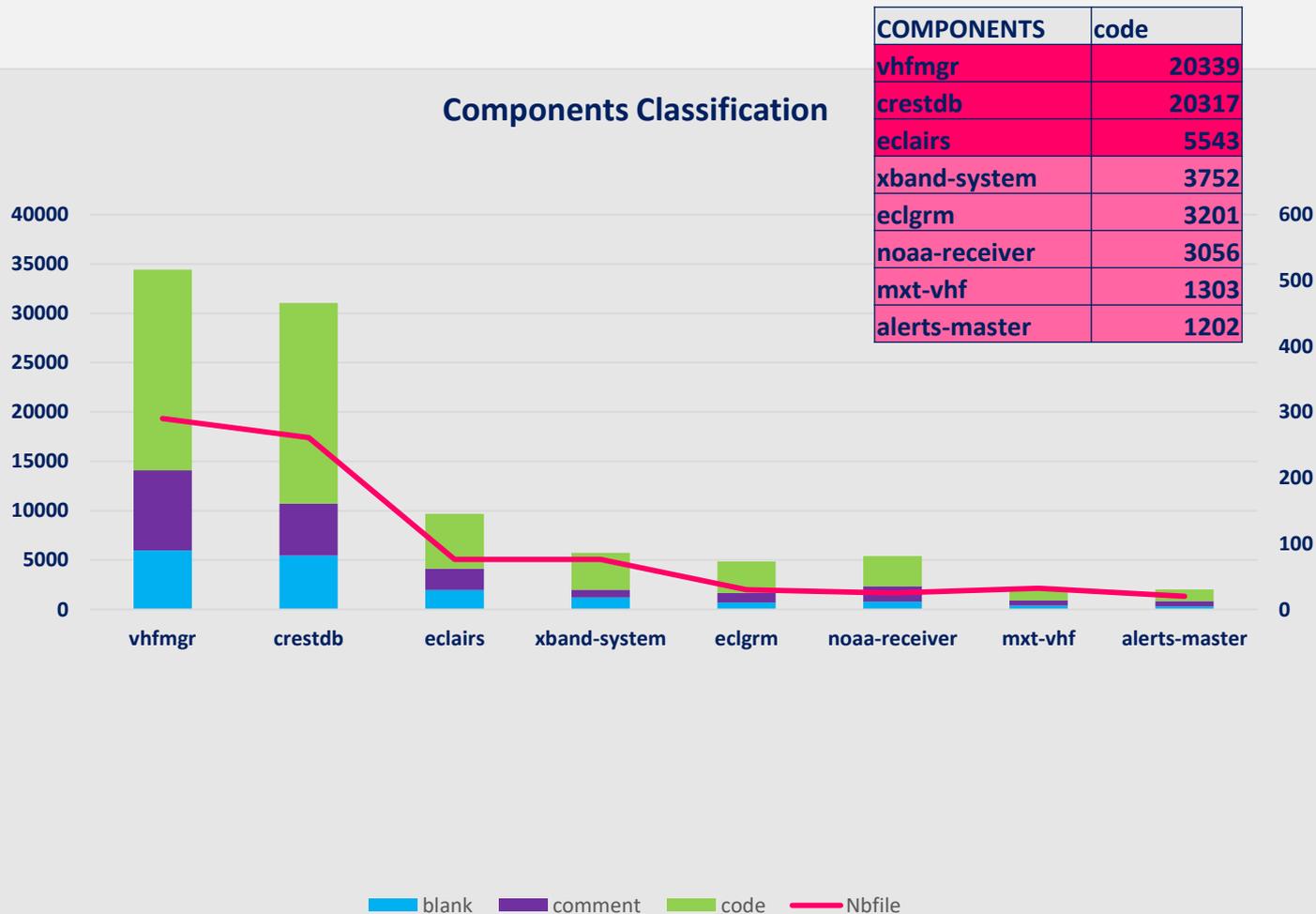
Le périmètre d'analyse

Dans la Gitlab : <https://drf-gitlab.cea.fr/svom/svom-deployment> , fichier Readme.md

- **Le périmètre de contrôle porte sur les composants ayant un répertoire « code » mentionné dans le Readme.md.**
- **La liste des composants est donnée dans le tableau ci-dessous.**

Nom	Description
alerts-master	la gestion des alertes
crestdb	le serveur de gestion de configuration
eclairs	le traitement des données Eclairs general program
eclgrm	le traitement des données Eclairs-GRM
messaging	le serveur de messagerie
monitoring	Gestion du monitoring des trois outils (Grafana, InfluxDB et Grafana)
mxt	le traitement des données MXT bande X
mxt-vhf	le traitement des données MXT VHF
noaa-receiver	l'application de réception des images NOAA
packets-decod	le décodage des paquets VHF
proxy	le serveur Proxy
vhfdb-postgre	le serveur de base de données postgresql pour la VHF
vhfmgr	la gestion des données VHF
xband-system	le serveur de données bande X

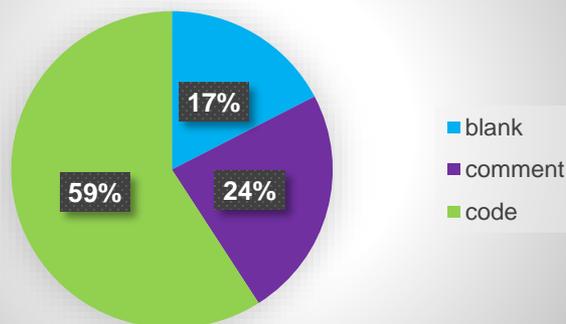
Components Classification



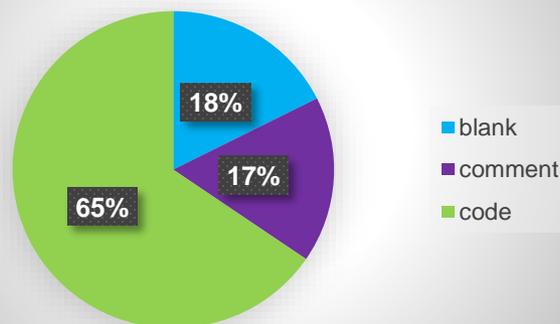
Synthèse Volumétrie

- **8 composants,**
- **810 fichiers,**
- **Lignes blanches : 20194,**
- **Lignes de commentaires : 16798,**
- **Lignes de code : 58713,**
- **Un taux de commentaires moyen de l'ordre de 20%**
- **Nblignes code/fichier : 72**

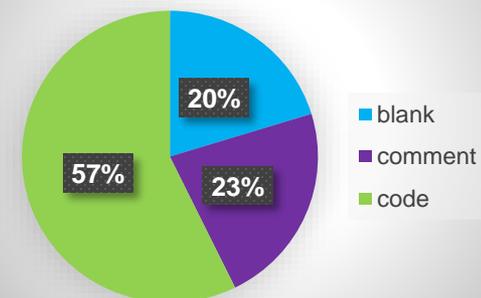
vhfmgr



crestdb

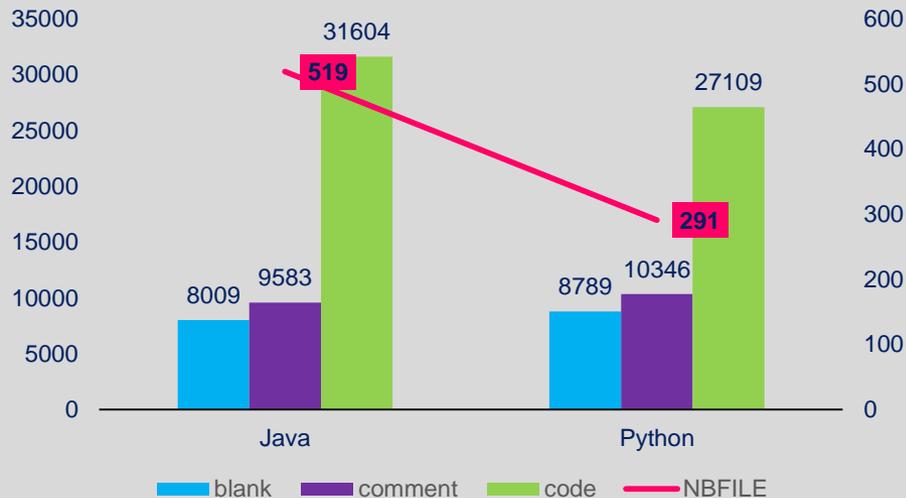


eclairs

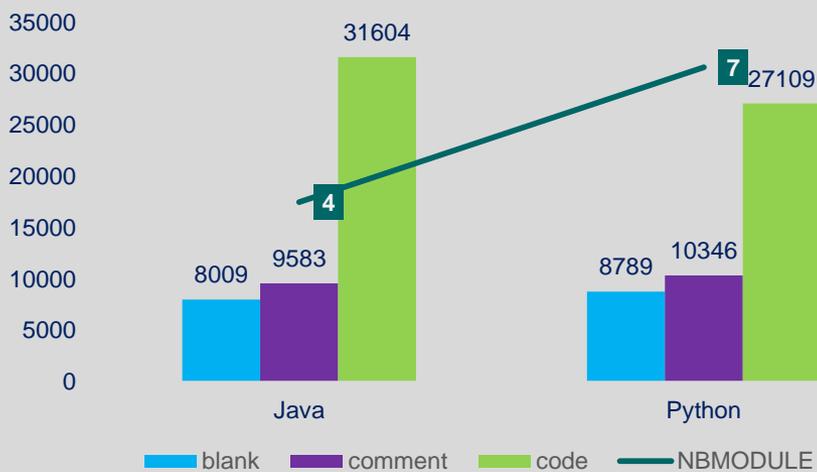


Synthèse JAVA, PYTHON

Language distribution



Language distribution



Composant	Java	Python
alerts-master	NON	OUI
crestdb	OUI	OUI
eclairs	NON	OUI
ecigrm	NON	OUI
mxt-vhf	NON	OUI
noaa-receiver	OUI	OUI
vhfmgr	OUI	OUI
xband-system	OUI	NON

- **Une volumétrie importante**
- **Un taux de commentaires respecté.**

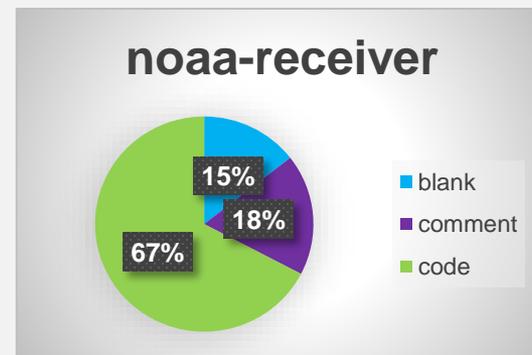
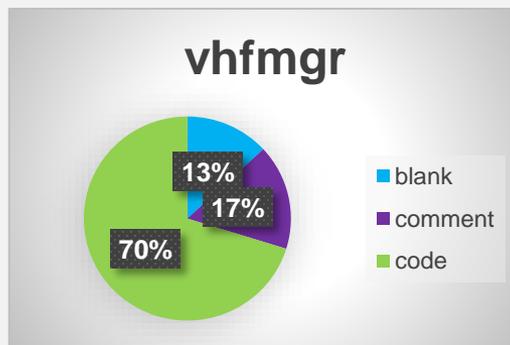
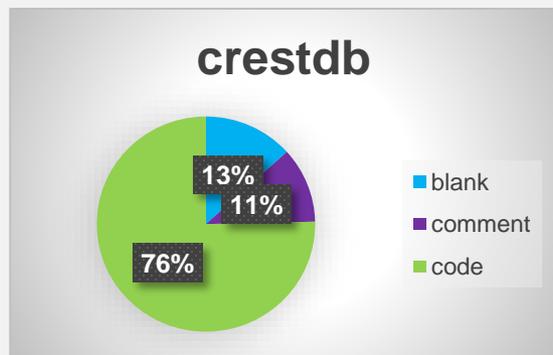


2ème PARTIE:

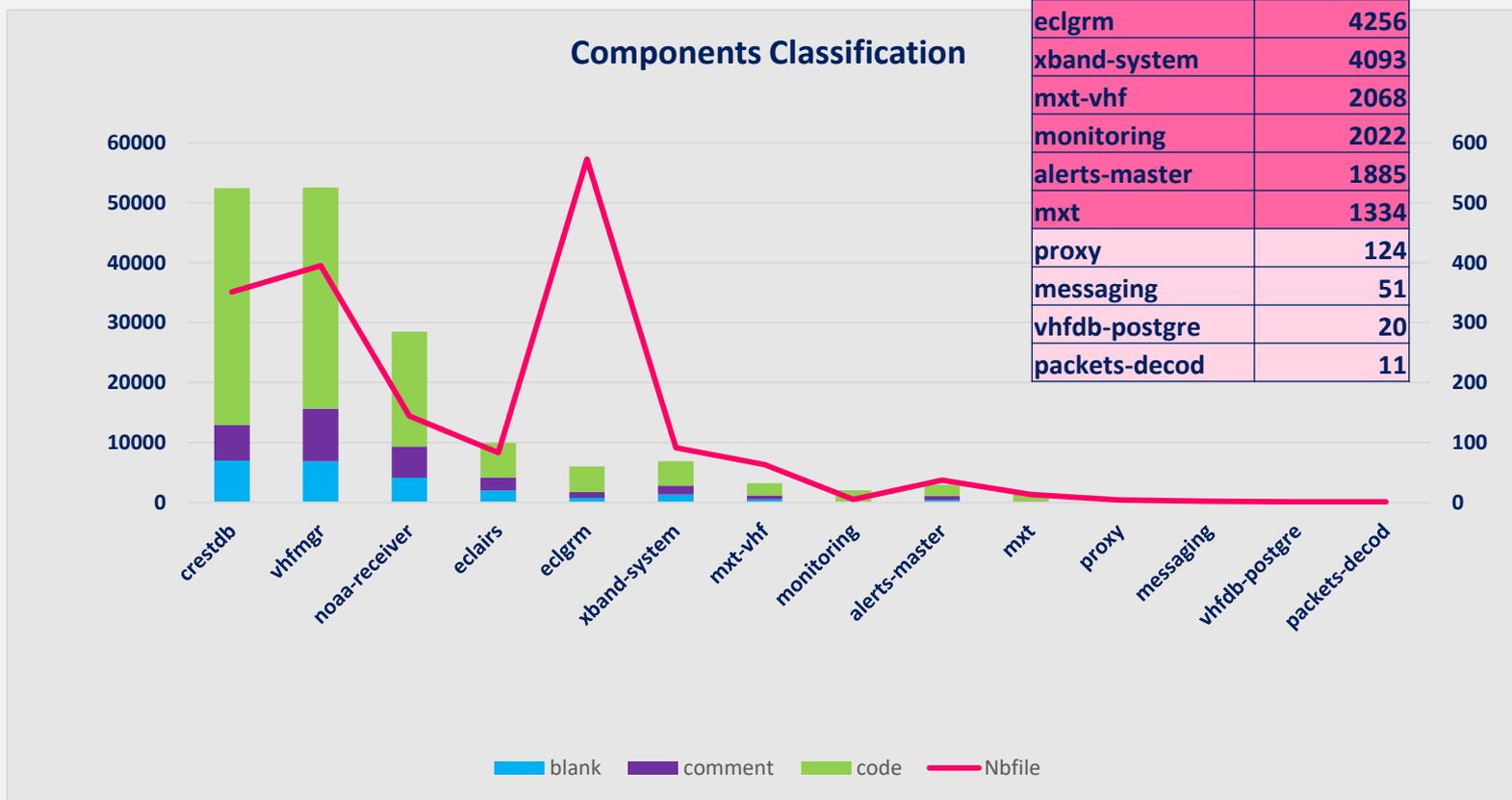
- **Éléments complémentaires Code**
 - **Volumétrie tout langage**
 - **Une analyse qualimétrie avec Logiscope**

Synthèse Volumétrie

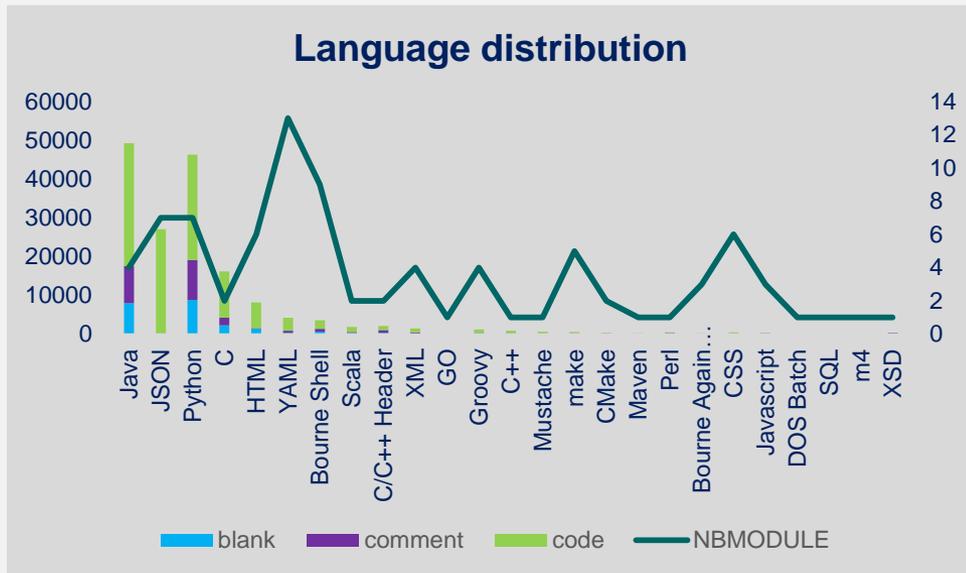
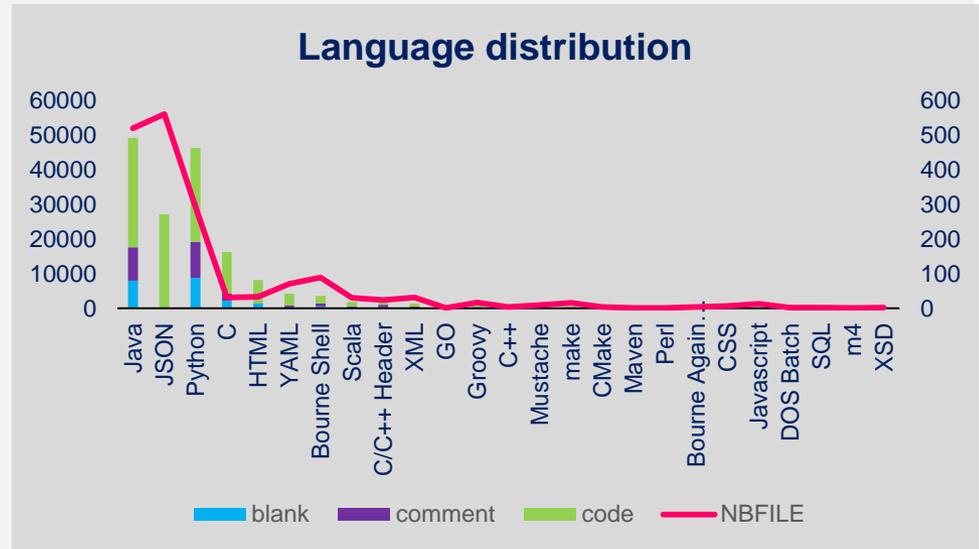
- **14 composants analysés,**
- **1763 fichiers,**
- **Lignes blanches : 23044,**
- **Lignes de commentaires : 26060,**
- **Lignes de code : 117251,**
- **Un taux de commentaires moyen de l'ordre de 16%**



COMPONENTS	code
crestdb	39529
vhfmgr	36891
noaa-receiver	19174
eclairs	5793
edlgrm	4256
xband-system	4093
mxt-vhf	2068
monitoring	2022
alerts-master	1885
mxt	1334
proxy	124
messaging	51
vhfdb-postgre	20
packets-decod	11



Synthèse
24 langages
Les plus utilisés :
Java, JSON,
Python, C



Langages de Programmation : Java et Python

Configurations: JSON, YML....

Langage C, code réutilisé

- **Le code : C de nooa-receiver - Code Réutilisé**
 - Nombre de fichiers C 29,
 - Nombre de fichiers H 19,
 - Nombre de ligne de code : >12000
- **L'outil : Logiscope**

Métrique de code	Seuil	Logiscope	HORS SEUIL
Nombre cyclomatique McCabe V(g)	<=25	OUI	4%
Nombre cyclomatique simplifié	<=20		
Nombre de lignes de code	<=100	OUI	4%
Nombre d'imbrication	<=7	OUI	0,66 %
Taux de commentaires	>20%	OUI	82%
Le taux de copier/coller	<15%		
L'état de la dette technique	<5%		

- **Une forte volumétrie**
- **Disparité des langages. Certains langages incontournables pour l'intégration (fichiers de configuration) méritent une attention particulière.**
 - Des normes de programmation devraient être définies en vue de faciliter l'étape d'intégration.
- **Code C : Code réutilisé**
 - peu complexe au regard des métriques imposées
 - En revanche, le taux de commentaires est bas

