

Plan de Gestion de Configuration CLS pour le code FloodML

CLS-ENV-NT-24-0400

Issue 1.0 – 15/04/2024

CHRONOLOGY ISSUES/HISTORIQUE DES VERSIONS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Issue/ Version | Date | Object/Objet | Written by/ Rédigé par | Checked by / Vérifié par | Approved by/ Approuvé par |
| 1.0 | 15/04/24 | Rédaction Initiale | C. Fatras | E. Lavergne | A. Andral |
|  |  |  |  |  |  |

DISTRIBUTION/LISTE DE DIFFUSION

|  |  |  |
| --- | --- | --- |
| Company/Organisme | Means of distribution/ Format de diffusion | Names/Destinataires |
| CLS | Electronique | Equipe projet |
| CS | Electronique | Equipe projet |
| CNES | Electronique | Equipe projet |

LIST OF CONTENTS/SOMMAIRE

[1 Introduction 5](#_Toc170897953)

[1.1 Objet du document 5](#_Toc170897954)

[1.2 Domaine d'application 5](#_Toc170897955)

[1.3 Terminologie 5](#_Toc170897956)

[1.4 Responsabilités et rôles 5](#_Toc170897957)

[1.4.1 Responsabilités 5](#_Toc170897958)

[1.4.2 Rôles 5](#_Toc170897959)

[2 Activités de gestion de configuration 6](#_Toc170897960)

[2.1 Identification de la configuration 6](#_Toc170897961)

[2.1.1 Description des règles applicables 6](#_Toc170897962)

[2.2 Maîtrise de la configuration : gestion des changements 7](#_Toc170897963)

[2.3 Rapport de l’état de la configuration (DDC) 7](#_Toc170897964)

[2.4 Administration de la GC 7](#_Toc170897965)

[2.4.1 Espaces utilisés 8](#_Toc170897966)

[2.4.2 Règles d’accès aux espaces 9](#_Toc170897967)

[2.4.3 Procédure de livraison 10](#_Toc170897968)

[2.4.4 Procédure d’installation 10](#_Toc170897969)

[2.4.5 Procédure de retrait de service 10](#_Toc170897970)

[2.5 Audit du système de GC 10](#_Toc170897971)

[2.6 Approvisionnements 10](#_Toc170897972)

[3 Outils de gestion de configuration 10](#_Toc170897973)

[3.1 Gestion des logiciels 10](#_Toc170897974)

[3.2 Gestion des faits techniques 10](#_Toc170897975)

[Appendix A - Glossaire 12](#_Toc170897976)

LIST OF TABLES AND FIGURES/LISTE DES TABLEAUX ET FIGURES

Liste des tableaux :

Tableau 1 : Règles d’accès aux espaces 10

Liste des figures :

Figure 3 : Maîtrise de la configuration et DDC 8

LIST OF ITEMS TO BE CONFIRMED OR DEFINED/LISTE DES AC ET AD

Liste des AC :

**Aucune entrée de table des matières n'a été trouvée.**

Liste des AD :

**Aucune entrée de table des matières n'a été trouvée.**

APPLICABLE DOCUMENTS/LISTE DES DOCUMENTS APPLICABLES

REFERENCE DOCUMENTS/LISTE DES DOCUMENTS DE REFERENCE

DR 1  CLS-ENV-NT-24-0397 - Standard de codage Python.docx

# Introduction

## Objet du document

Ce Plan de Gestion de Configuration a pour objet de décrire les principes, l'organisation, les règles et les moyens mis en place par CLS afin d'accomplir les tâches associées à la gestion de configuration du projet FloodDAM-Digital Twin (noté par la suite FloodDAM-DT).

La gestion de configuration a pour finalité de maîtriser la définition technique d'un produit, ainsi que de gérer l'ensemble des modifications apportées au cours de l'évolution du produit.

Sur le plan technique, l'objectif est de garder la :

* Maîtrise de la mise en œuvre du produit,
* Maîtrise de la composition et du périmètre du produit,
* Maîtrise des fonctionnalités du produit.

Sur le plan humain, l'objectif est de mettre en place un service fiable et pratique (mise à disposition d'informations à jour, accessibles…).

## Domaine d'application

Ce document s'applique à toute activité de développement ou de maintenance pour des produits logiciels réalisés par CLS et/ou ses sous-traitants.

Il s'applique en particulier à tout composant du produit, logiciel, nouveau ou modifié, développé en interne ou en sous-traitance, dans le cadre de projets ou de besoins internes ou externes.

Il s'applique aussi aux approvisionnements non accompagnés de développement spécifique.

Dans le cas de systèmes développés pour un client externe, les exigences client s’appliquent en priorité, les exigences de ce plan s’appliquent s’il n’y a pas d’exigence client contradictoire.

Ce plan s'adresse à l’ensemble des intervenants d’un projet, aussi bien aux ingénieurs, qu'aux chefs de projets et gestionnaires de configuration.

Ce plan présente la gestion de configuration pour le logiciel : il décrit plus spécifiquement la gestion de configuration pour les composants logiciels.

## Terminologie

La terminologie spécifique à la gestion de configuration est explicitée dans le PGC CLS. Les abréviations utilisées dans ce document sont listées en page liminaire.

## Responsabilités et rôles

### Responsabilités

La responsabilité du code est du ressort du CNES à la suite de la livraison de FloodDAM-DT.

### Rôles

Le responsable projet coté CNES est Raquel Rodriguez-Suquet.

# Activités de gestion de configuration

## Identification de la configuration

Le code FloodML a été développé et s’utilise dans un environnement python 3. L’environnement de travail associé a été créé et modifié en utilisant Anaconda. L’environnement est décrit et peut être reproduit dans son intégralité via le fichier requirements-R02108.txt

Le code source comprend les fichiers python utilisés pour la préparation des données d’apprentissage, l’apprentissage lui-même, puis leur utilisation pour inférences des zones inondées.

### Description des règles applicables

#### Arborescence

Dossier racine :

RDF-1-prepration.py ## Préparation de l’apprentissage

RDF-2-training.py ## Apprentissage

RDF-3-inference.py ##Utilisation de la base entraînée pour inférence.

Sous-dossiers :

Chain : contient les fichiers liés à chaque capteur (S1, S2, L8, L9, TSX)

Common : contient l’ensemble des outils utilisés dans FloodML

Docs : contient le DDC-FLoodML, le Manuel de FloodML, le standard de codage utilisé, et le présent document.

#### Nommage

Pas de règle en particulier (suit les standards de codage python DR1).

#### Version

Tout article de configuration livré doit faire l'objet d'une version, et celle-ci doit être clairement identifiée.

L'identifiant de version est un numéro. Il est composé de 2 champs qui sont :

* L’indice fonctionnel : pour une évolution majeure donnant lieu à la modification des interfaces externes, ou mettant en jeu la refonte de tout ou parti de l’architecture de l’article.
* L'indice de révision : pour une évolution ou correction ne mettant en cause, ni les interfaces, ni le mécanisme de fonctionnement de l’article. C'est un nombre entier incrémenté lorsque le service rendu par l'article change ou lorsque la définition est modifiée sans que le service rendu le soit (évolutions et anomalies)

## Maîtrise de la configuration : gestion des changements

Le CNES a la charge de la gestion de changement via le github du CNES.

Le CNES définit la branche sur laquelle apporter la modification, et reste maître de sa révision et de la décision de fusionner la branche de travail vers la branche principale.

## Rapport de l’état de la configuration (DDC)

Il est nécessaire de produire un rapport traduisant l’état de la configuration (ou Dossier Descriptif de la Configuration (DDC)) :

* A chaque livraison d’un Article de configuration
* Sur demande

La livraison d’un article de configuration se fait à l’occasion :

* De sa revue de spécification
* De sa revue conception
* De chacun de ses débuts de validation (validation initiale, puis validations pour chaque livraison durant la phase de maintenance)

Le DDC permet de connaître, pour chaque version d’un article, la liste des éléments de configuration le constituant, son historique, les raisons de ses évolutions, ses anomalies, les dérogations connues, etc.

Le contenu du DDC doit être le suivant :

* Identification de l’Article : Nom, Version
* Instructions détaillées de prise en compte
* Moyen requis (pour développer, ou générer, ou exploiter, ou utiliser)
* Modifications apportées par rapport à une configuration de référence (liste des FFT prisent en compte, liste des éléments de configuration ajoutés, supprimés, modifiés)
* Etat des faits techniques connus en cours
* Composition : documents, composants
* Informations complémentaires

Le DDC peut être constitué de plusieurs parties (ou fichiers). Tout ou partie du DDC peut être généré automatiquement.

## Administration de la GC

L’administration de la GC est l’activité permettant de garder la maitrise du contenu et des fonctionnalités du produit logiciel géré. Les tâches constitutives de cette administration sont :

* La collecte : il s’agit de collecter les informations produites et partagées par les différents acteurs modifiant le produit logiciel.
* Le contrôle : il s’agit de s’assurer que les informations du référentiel de GC sont le reflet des produits générés.
* La mise en référence des articles : Ajouter de nouveaux articles à la gestion de configuration.
* La sauvegarde
* La mise à disposition des différentes versions de chaque article.

### Espaces utilisés

Tout projet se rapproche de l’organisation décrite ci-dessous en simplifiant pour les petits projets.

On distingue à CLS deux familles d’espaces :

* Les espaces de développement, utilisés par les équipes de développement et de maintenance, allant jusqu’à la validation du produit côté développement
* Les espaces côté exploitation, comportant les espaces nécessaires à la préparation au passage en exploitation

L’espace de livraison fait le tampon entre ces deux familles.

Espaces côté développement :

* **Espace de référence** : sert de référence à l’ensemble du projet. Il contient la totalité des éléments de l’arborescence gérés en configuration (sources, bibliothèques, etc.). Il est utilisé pour constituer une version du système livrée au client et pour l'archiver. Il est unique pour chaque projet. Cet espace est en général géré dans un outil de GC (CVS, SVN)
* **Espace de développement :** espace de travail pour les développeurs pour effectuer les développements, les modifications et les tests unitaires. On a en général un espace de développement par développeur
* **Espace d’intégration :** espace pour testerl’intégration d’un ou plusieurs composants en cours de développement
* **Espace de Qualification Technique :** permet de générer une version du produit et d’exécuter les tests de Qualification Technique. Il s’agit d’une validation côté développement (par opposition à la Qualification Produit présentée plus loin), même si cette validation est faite par une personne externe à l’équipe de développement et connaissant le métier

En fin de phase de QT, cet espace accueil également le déroulement de scénarios métiers issus de la Qualification Produit (déroulés par l’équipe métier) pour valider la maturité de l’application et son aptitude à subir les tests de qualification sur la pré-production.

Espaces côté client / exploitation :

* **Espace de livraison :** espace de transfert qui permet de réceptionner une fourniture client, de générer les exécutables et préparer une livraison client. Cet espace de livraison peut avoir son propre référentiel (autre espace de référence) : ne contenant pas toutes les versions intermédiaires de travail, mais uniquement les versions livrées côté exploitation. Ce référentiel serait alors un référentiel côté exploitation, il n’est pas représenté sur les schémas suivants.
* **Espace de pré-production :** permet de faire la qualification du système dans des conditions opérationnelles et de préparer sa mise en exploitation. Cette qualification a toujours deux composantes : la Qualification Opérationnelle (scénarios opérationnels) et la Qualification Produit (scénarios métier). A noté que la Qualification Produit débute en général sur l’espace de QT et se poursuit sur la pré-production.
* **Espace d’exploitation** : mise à disposition des ressources stables (exécutables, scripts, etc.) pour l’exploitation par les utilisateurs finaux.

Dans le cadre des projets FLoodML puis FLoodDAM et actuellement FloodDAM-DT, l’ensemble des travaux (excepté le développement du dockerfile) ont été effectués sur le HPC du CNES (HAL puis depuis 2024 T-rex). L’ensemble des données annexes, bases d’entrainement générés et données produites clefs sont en possession du CNES dans le répertoire de projet dédié /work/FLOODML.

### Règles d’accès aux espaces

Chaque acteur selon son rôle a accès en lecture et/ou écriture aux différents espaces. Le tableau suivant donne les règles générales applicables sur un projet.

Le lien entre côté développement et côté exploitation se fait via l’espace de livraison. Les responsables de gestion de configuration (développement et exploitation) sont responsables de la livraison : à affiner en fonction de l’activité.

| **Espace / profil** | **Responsable d’activité** | **RGC** | **Equipe Développement-Maintenance** | **Responsable exploitation** |
| --- | --- | --- | --- | --- |
| Espace de référence (côté dév) | Lecture | Lecture /  Ecriture | Lecture / Ecriture par « commit » [[1]](#footnote-2) |  |
| Espace de développement[[2]](#footnote-3) | Lecture | Lecture | Lecture / Ecriture |  |
| Espace de validation dév. | Lecture | Lecture /  Ecriture | Lecture |  |
| Espace de livraison | Lecture | Lecture /  Ecriture | Lecture | Lecture |
| Espace de pré-production |  |  |  | Lecture /  Ecriture |
| Espace d’exploitation |  |  |  | Lecture /  Ecriture |

**Tableau 1 : Règles d’accès aux espaces**

### Procédure de livraison

Chaque livraison de nouvelle version suit une procédure standard.

Cette procédure prévoit :

* L’emplacement, l’arborescence et le mode de livraison
* Le contenu d’une livraison avec au minimum
  + Les articles de configuration concernés par la livraison
  + La documentation pour l’installation, la configuration et l’utilisation du produit

Le code dans le cadre du projet FloodML est livré via la branche flooddamdt du projet ‘wp3-rapid-mapping’ du gitlab du CNES.

### Procédure d’installation

Le code s’exécute en python. L’environnement de travail associé est résumé dans le fichier de spécification requirements-R02108.txt, à la racine du projet cloné au préalable depuis le gitlab du CNES via la commande suivante :

git clone -b flooddamdt https://gitlab.cnes.fr/flooddam/wp3-rapid-mapping.git

Ce fichier permet de recréer l’environnement de travail idoine. En utilisant Anaconda, la commande de création de l’environnement (ici nommé rapids-0.21.08) est alors :

conda create --name rapids-0.21.08 --file requirements-R02108.txt

### Procédure de retrait de service

NA

## Audit du système de GC

Aucun audit n’est mis en place actuellement pour cette gestion de configuration.

## Approvisionnements

NA

# Outils de gestion de configuration

## Gestion des logiciels

Quelque soit le langage de développement utilisé, Il est recommandé d’utiliser l’outil de GC officiel à CLS : SVN pour tout produit logiciel.

## Gestion des faits techniques

Aucun outil ni process n’est actuellement en cours pour la gestion des faits technique.

1. Glossaire

|  |  |
| --- | --- |
| AC | A Confirmer |
| AD | A déterminer |
| AQM | Assurance Qualité Méthodes |
| DA | Document Applicable |
| DD | Demande de Dérogation |
| DG | Direction Générale |
| DM | Demande de Modification |
| DR | Document de Référence |
| FA | Fiche d'Anomalie |
| FFT | Fiche de Fait Technique |
| PAP | Plan d'Assurance Produit |
| PGC | Plan de gestion de configuration |
| PGCM | Plan de gestion de configuration matériel |
| PGCL | Plan de gestion de configuration logiciel |
| RDQ | Représentant de la Direction pour la Qualité |
| GC | Gestion de Configuration |
| RGC | Responsable Gestion de Configuration |
| CGC | Commission de Gestion de Configuration |

1. Un développeur a accès à l’espace de référence en lecture/écriture lorsqu’il extrait ou remet en référence un composant par une commande de l’outil de gestion de configuration. [↑](#footnote-ref-2)
2. Chacun a les droits en lecture/écriture dans son propre espace de développement [↑](#footnote-ref-3)