|  |
| --- |
| flood-dam-dt  Version not provided  Code analysis |

|  |
| --- |
| **By: Fatras Christophe**  **2024-06-25** |

# Content

[Content 1](#_Toc97156258)

[Introduction 2](#_Toc97156259)

[Configuration 2](#_Toc97156260)

[Synthesis 3](#_Toc97156261)

[Analysis Status 3](#_Toc97156262)

[Quality gate status 3](#_Toc97156263)

[Metrics 3](#_Toc97156264)

[Tests 3](#_Toc97156265)

[Detailed technical debt 3](#_Toc97156266)

[Metrics Range 5](#_Toc97156267)

[Volume 5](#_Toc97156268)

[Issues 6](#_Toc97156269)

[Charts 6](#_Toc97156270)

[Issues count by severity and type 8](#_Toc97156271)

[Issues List 8](#_Toc97156272)

[Security Hotspots 9](#_Toc97156273)

[Security hotspots count by category and priority 9](#_Toc97156274)

[Security hotspots List 9](#_Toc97156275)

# Introduction

This document contains results of the code analysis of flood-dam-dt.

# Configuration

* Quality Profiles
  + Names: Sonar way [Python];
  + Files: AWEEM9pEFeGqyuSNbABW.json;
* Quality Gate
  + Name: CLS Quality Gate 2020
  + File: CLS Quality Gate 2020.xml

# Synthesis

## Analysis Status

|  |  |  |  |
| --- | --- | --- | --- |
| Reliability | Security | Security Review | Maintainability |
| A.png | **A.png** | **A.png** | **A.png** |

## Quality gate status

|  |  |
| --- | --- |
| Quality Gate Status | **ERROR.png** |

|  |  |
| --- | --- |
| Métrique | Valeur |
| Reliability Rating on New Code | OK |
| Reliability Rating | OK |
| Security Rating on New Code | OK |
| Security Rating | OK |
| Maintainability Rating on New Code | OK |
| Maintainability Rating | OK |
| Coverage on New Code | OK |
| Duplicated Lines (%) on New Code | ERROR (11.9% is greater than 10%) |
| Security Hotspots Reviewed on New Code | OK |

## Metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Coverage | Duplication | Comment  density | Median number of lines of code per file | Adherence to coding standard |
| 0.0 % | **11.9 %** | **29.3 %** | **97.0** | **99.9 %** |

## Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Total | Success Rate | Skipped | Errors | Failures |
| 0 | **0 %** | **0** | **0** | **0** |

## Detailed technical debt

|  |  |  |  |
| --- | --- | --- | --- |
| Fiabilité | Sécurité | Maintenabilité | Total |
| - | - | 0d 3h 51min | 0d 3h 51min |

## Metrics Range

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Cyclomatic  Complexity | Cognitive  Complexity | Lines of code per file | Comment  density (%) | Coverage | Duplication (%) |
| Min | 1.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 |
| Max | 348.0 | 387.0 | 1594.0 | 77.8 | 0.0 | 69.3 |

## Volume

|  |  |
| --- | --- |
| Langage | Nombre |
| <null> | 26 |
| Python | 3441 |
| Total | 3467 |

# Issues

## Charts

## Issues count by severity and type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type / Criticité | INFO | MINOR | MAJOR | CRITICAL | BLOCKER |
| BUG | 0 | 0 | 0 | 0 | 0 |
| VULNERABILITY | 0 | 0 | 0 | 0 | 0 |
| CODE\_SMELL | 7 | 5 | 15 | 9 | 0 |

## Issues List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom | Description | Type | Criticité | Nombre |
| String literals should not be duplicated | Duplicated string literals make the process of refactoring error-prone, since you must be sure to update all occurrences. On the other hand, constants can be referenced from many places, but only need to be updated in a single place. Noncompliant Code Example With the default threshold of 3: def run(): prepare("this is a duplicate") # Noncompliant - "this is a duplicate" is duplicated 3 times execute("this is a duplicate") release("this is a duplicate") Compliant Solution ACTION\_1 = "action1" def run(): prepare(ACTION\_1) execute(ACTION\_1) release(ACTION\_1) Exceptions No issue will be raised on: duplicated string in decorators strings with less than 5 characters strings with only letters, numbers and underscores @app.route("/api/users/", methods=['GET', 'POST', 'PUT']) def users(): pass @app.route("/api/projects/", methods=['GET', 'POST', 'PUT']) # Compliant def projects(): pass | CODE\_SMELL | CRITICAL | 2 |
| Cognitive Complexity of functions should not be too high | Cognitive Complexity is a measure of how hard the control flow of a function is to understand. Functions with high Cognitive Complexity will be difficult to maintain. See Cognitive Complexity | CODE\_SMELL | CRITICAL | 7 |
| Track uses of "TODO" tags | TODO tags are commonly used to mark places where some more code is required, but which the developer wants to implement later. Sometimes the developer will not have the time or will simply forget to get back to that tag. This rule is meant to track those tags and to ensure that they do not go unnoticed. Noncompliant Code Example def doSomething: # TODO : Complete function See MITRE, CWE-546 - Suspicious Comment | CODE\_SMELL | INFO | 7 |
| Nested blocks of code should not be left empty | Most of the time a block of code is empty when a piece of code is really missing. So such empty block must be either filled or removed. Noncompliant Code Example for i in range(3): pass Exceptions When a block contains a comment, this block is not considered to be empty. | CODE\_SMELL | MAJOR | 7 |
| Unused function parameters should be removed | Unused parameters are misleading. Whatever the value passed to such parameters is, the behavior will be the same. Noncompliant Code Example def do\_something(a, b): # "b" is unused return compute(a) Compliant Solution def do\_something(a): return compute(a) Exceptions Overriding methods are ignored. class C(B): def do\_something(self, a, b): # no issue reported on b return self.compute(a) } Throwaway variables \_. def do\_something(a, \_): # no issue reported on \_ return compute(a) | CODE\_SMELL | MAJOR | 2 |
| Sections of code should not be commented out | Programmers should not comment out code as it bloats programs and reduces readability. Unused code should be deleted and can be retrieved from source control history if required. | CODE\_SMELL | MAJOR | 3 |
| Identity comparisons should not be used with cached typed | Identity operators is and is not check if the same object is on both sides, i.e. a is b returns True if id(a) == id(b). Integers, bytes, floats, strings, frozensets and tuples should not be compared with identity operators because the result may not be as expected. If you need to compare these types you should use instead equality operators == or !=. The CPython interpreter caches certain builtin values for integers, bytes, floats, strings, frozensets and tuples. For example, the literal 1 will create the same object as int("1"), which means that 1 is int("1") is True. However this works only by chance as other integer values are not cached, for example int("1000") is 1000 will always be False. This behavior is not part of Python language specification and could vary between interpreters. CPython 3.8 even warns about comparing literals using identity operators. The only case where using the "is" operator with a cached type is ok is with "interned" strings. Note however that interned strings don’t necessarily have the same identity as string literals. This rule raises an issue when at least one operand of an identity operator: is of type int, bytes, float, frozenset or tuple. or it is a string literal. Noncompliant Code Example def literal\_comparison(param): param is 2000 # Noncompliant literal\_comparison(2000) # will return True literal\_comparison(int("2000")) # will return False () is tuple() # Noncompliant. Always True (1,) is tuple([1]) # Noncompliant. Always False from sys import intern with open("test.txt") as f: # test.txt contains "blabla\n" text = f.read() intern(text) is "blabla\n" # Noncompliant. Always False Compliant Solution def literal\_comparison(param): param == 2000 literal\_comparison(2000) # will return True literal\_comparison(int("2000")) # will return True () == tuple() # Always True (1,) == tuple([1]) # Always True from sys import intern with open("tmp/test.txt") as f: # test.txt contains "blabla\n" text = f.read() intern(text) is intern("blabla\n") # Always True See Why does Python 3.8 log a SyntaxWarning for 'is' with literals? - Adam Johnson Equality vs identity - Trey Hunner Python documentation - sys.intern | CODE\_SMELL | MAJOR | 2 |
| Single-character alternations in regular expressions should be replaced with character classes | When an alternation contains multiple alternatives that consist of a single character, it can be rewritten as a character class. This should be preferred because it is more efficient. Noncompliant Code Example r"a|b|c" # Noncompliant Compliant Solution r"[abc]" # or r"[a-c]" | CODE\_SMELL | MAJOR | 1 |
| Local variable and function parameter names should comply with a naming convention | Shared naming conventions allow teams to collaborate effectively. This rule raises an issue when a local variable or function parameter name does not match the provided regular expression. Exceptions Loop counters are ignored by this rule. for i in range(limit): # Compliant print(i) | CODE\_SMELL | MINOR | 3 |
| "except" clauses should do more than raise the same issue | An except clause that only rethrows the caught exception has the same effect as omitting the except altogether and letting it bubble up automatically, but with more code and the additional detriment of leaving maintainers scratching their heads. Such clauses should either be eliminated or populated with the appropriate logic. Noncompliant Code Example a = {} try: a[5] except KeyError: raise # Noncompliant Compliant Solution a = {} try: a[5] except KeyError as e: logging.exception('error while accessing the dict') raise e | CODE\_SMELL | MINOR | 2 |

# Security Hotspots

## Security hotspots count by category and priority

|  |  |  |  |
| --- | --- | --- | --- |
| Catégorie / Priorité | LOW | MEDIUM | HIGH |
| LDAP Injection | 0 | 0 | 0 |
| Object Injection | 0 | 0 | 0 |
| Server-Side Request Forgery (SSRF) | 0 | 0 | 0 |
| XML External Entity (XXE) | 0 | 0 | 0 |
| Insecure Configuration | 0 | 0 | 0 |
| XPath Injection | 0 | 0 | 0 |
| Authentication | 0 | 0 | 0 |
| Weak Cryptography | 0 | 0 | 0 |
| Denial of Service (DoS) | 0 | 0 | 0 |
| Log Injection | 0 | 0 | 0 |
| Cross-Site Request Forgery (CSRF) | 0 | 0 | 0 |
| Open Redirect | 0 | 0 | 0 |
| Permission | 0 | 0 | 0 |
| SQL Injection | 0 | 0 | 0 |
| Encryption of Sensitive Data | 0 | 0 | 0 |
| Traceability | 0 | 0 | 0 |
| Buffer Overflow | 0 | 0 | 0 |
| File Manipulation | 0 | 0 | 0 |
| Code Injection (RCE) | 0 | 0 | 0 |
| Cross-Site Scripting (XSS) | 0 | 0 | 0 |
| Command Injection | 0 | 0 | 0 |
| Path Traversal Injection | 0 | 0 | 0 |
| HTTP Response Splitting | 0 | 0 | 0 |
| Others | 0 | 0 | 0 |

## Security hotspots List

## Commentaire sur les erreurs détectées

36 issues ont été repérées par sonarqube lors de l’analyse du code selon les standards CLS (CLS Quality Gate 2020). Elles relèvent dans leur intégralité de mauvaises pratiques selon les standards CLS, et n’ont pas d’impact sur le fonctionnement du code en l’état.

Etant donné que ces erreurs n’ont pas d’impact sur le bon fonctionnement du code mais relèvent plus du conseil quant aux bonnes pratiques de codage à mettre en place, il a été décidé de ne pas les corriger. En effet, la majorité de ces « erreurs » correspondent à des fonctions standard mises en place durant le projet FloodDAM pour adaptation à PEPS. Par soucis de cohérence elles sont donc laissées intactes et telles quelles. Le détail des justifications pour chaque erreur est listé dans les deux pages suivantes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Rule** | **Message** | **Type** | **Severity** | **Language** | **File** | **Line** | **Effort** | **Status** | **Justification** |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/GDal/buildvrt.py | 39 | 5min | OPEN | Code antérieur de Peter/CNES |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/GDal/rasterize.py | 43 | 5min | OPEN |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/GDal/translate.py | 40 | 5min | OPEN |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/GDal/warp.py | 43 | 5min | OPEN |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/GDalDatasetWrapper.py | 231 | 5min | OPEN |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/ImageTools.py | 186 | 5min | OPEN |
| python:S108 | Either remove or fill this block of code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/validationTools.py | 86 | 5min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Chain/TSXProduct.py | 60 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/GDalDatasetWrapper.py | 41 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/ImageApps.py | 43 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/ImageApps.py | 82 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/ImageTools.py | 54 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/ImageTools.py | 58 | 0min | OPEN |
| python:S1135 | Complete the task associated to this "TODO" comment. | CODE\_SMELL | INFO | Python | flood-dam-dt:Common/XMLTools.py | 28 | 0min | OPEN |
| python:S117 | Rename this local variable "WCmask" to match the regular expression ^[\_a-z][a-z0-9\_]\*$. | CODE\_SMELL | MINOR | Python | flood-dam-dt:RDF-3-inference.py | 354 | 2min | OPEN | Présence de majuscules dans des variables locales de fonctions. Pas d’impact sur le code. |
| python:S117 | Rename this local variable "bandIdx" to match the regular expression ^[\_a-z][a-z0-9\_]\*$. | CODE\_SMELL | MINOR | Python | flood-dam-dt:Common/ImageIO.py | 97 | 2min | OPEN |
| python:S117 | Rename this parameter "C" to match the regular expression ^[\_a-z][a-z0-9\_]\*$. | CODE\_SMELL | MINOR | Python | flood-dam-dt:Common/RDF\_tools.py | 300 | 2min | OPEN |
| python:S1172 | Remove the unused function parameter "dst". | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/ImageTools.py | 166 | 5min | OPEN | Certes inutilisé, mais initialisé pour d'éventuelles évolutions futures. Il a été choisi de ne pas supprimer ces paramètres. |
| python:S1172 | Remove the unused function parameter "options". | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/ImageTools.py | 192 | 5min | OPEN |
| python:S1192 | Define a constant instead of duplicating this literal "Cannot determine tile ID: %s" 3 times. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Chain/S2Product.py | 54 | 6min | OPEN | Demande d’utiliser une expression dans une variable plutôt que de la répéter trois fois. Jugé mineur et sans impact. |
| python:S1192 | Define a constant instead of duplicating this literal "Arrays have to be then same shape: {0} != {1}" 3 times. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Common/validationTools.py | 16 | 6min | OPEN |
| python:S125 | Remove this commented out code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:RDF-3-inference.py | 134 | 5min | OPEN | Reliquats, qui n'ont aucun impact sur le code |
| python:S125 | Remove this commented out code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Chain/TSXProduct.py | 35 | 5min | OPEN |
| python:S125 | Remove this commented out code. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Chain/TSXProduct.py | 65 | 5min | OPEN |
| python:S2737 | Add logic to this except clause or eliminate it and rethrow the exception automatically. | CODE\_SMELL | MINOR | Python | flood-dam-dt:Chain/PleiadesProduct.py | 183 | 5min | OPEN | Pas d'impact, structure mise en place pour permettre des évolutions futures. |
| python:S2737 | Add logic to this except clause or eliminate it and rethrow the exception automatically. | CODE\_SMELL | MINOR | Python | flood-dam-dt:Common/ImageTools.py | 59 | 5min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 20 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Chain/Product.py | 52 | 10min | OPEN | Correspond aux standards de CLS quant à la longueur maximale des fonctions. En effet, elle est fixée à 15 lignes, et cette longueur est ici dépassée. Sans impact. |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 23 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Common/FileSystem.py | 62 | 13min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 29 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Common/Imagery/Dataset.py | 230 | 19min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 21 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Common/RDF\_tools.py | 29 | 11min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 17 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:Common/XMLTools.py | 21 | 7min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 23 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:RDF-1-preparation.py | 25 | 13min | OPEN |
| python:S3776 | Refactor this function to reduce its Cognitive Complexity from 73 to the 15 allowed. | CODE\_SMELL | CRITICAL | Python | flood-dam-dt:RDF-3-inference.py | 28 | 1h3min | OPEN |
| python:S5795 | Replace this "is" operator with "=="; identity operator is not reliable here. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/ImageIO.py | 83 | 1min | OPEN | Reliquat d'anciennes versions mais qui n'empêche pas le code de fonctionner. |
| python:S5795 | Replace this "is not" operator with "!="; identity operator is not reliable here. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Common/Imagery/Dataset.py | 273 | 1min | OPEN |
| python:S6035 | Replace this alternation with a character class. | CODE\_SMELL | MAJOR | Python | flood-dam-dt:Chain/Product.py | 86 | 5min | OPEN |