

Test statique et révision de code

Claudéric DeRoy

Au menu

- Révision de code
- Test statique
- Pylint
- Exercice avec Pylint
- Réflexion/conclusion

Révision de code

Révision de code («*code review*»)

- Un peu comme le test statique mais exécutée par un humain
- C'est important parce qu'on va probablement en être le sujet ou nous même en faire un jour.

Comment ne pas faire de la bonne révision de code^[1]

- **Donner aucune explication**
 - *La fonction est mal écrite*
- **Ne pas référer à du code précisément**
 - *Plusieurs truc sont mal implémenté*
- **Ne pas offrir de possible piste de solution**
 - *Ça ne fonctionnera pas, on a déjà un problème similaire*

Test statique

Ce que je pensais des tests statiques

conversation between a
dynamic tester and a static tester



C'est quoi un test statique^[2]

- Test automatique fait par ordinateur
- Aucune exécution du code à tester

Test statique^[3]

- Surtout pour trouver les “codes puants” («*code smell*»)
- Exemples donnés par Martin Fowler et Kent Beck
 1. «*Duplicated code*»
 2. «*Long method*»
 3. «*Conditional complexity*»
 4. «*Combinatorial explosion*»
 5. «*Oddball solution*»

Pylint

- Un exemple de test statique est **Pylint** qui est l'implémentation d'un «*lint*», mais pour Python.

[Read the doc](#)[Install it](#)[Contribute](#)[Get support](#)

Pylint

- Pylint cherche les “codes puants” comme Fowler et Beck les définissent, mais il en cherche plusieurs autres

invalid-class-object (E0243):
Invalid assignment to ' __class__ '. Should be a class definition but got a '%s' Used when an invalid object is assigned to a `__class__` property. Only a class is permitted.

invalid-slots-object (E0236):
Invalid object %r in __slots__, must contain only non empty strings Used when an invalid (non-string) object occurs in `__slots__`.

no-method-argument (E0211):
Method %r has no argument Used when a method which should have the bound instance as first argument has no argument defined.

no-self-argument (E0213):
Method %r should have "self" as first argument Used when a method has an attribute different the "self" as first argument. This is considered as an error since this is a so common convention that you shouldn't break it!

declare-non-slot (E0245):
No such name %r in __slots__ Raised when a type annotation on a class is absent from the list of names in `__slots__`, and `__slots__` does not contain a `__dict__` entry.

unexpected-special-method-signature (E0302):
The special method %r expects %s param(s), %d %s given Emitted when a special method was defined with an invalid number of parameters. If it has too few or too many, it might not work at all.

class-variable-slots-conflict (E0242):
Value %r in slots conflicts with class variable Used when a value in `__slots__` conflicts with a class variable, property or method.

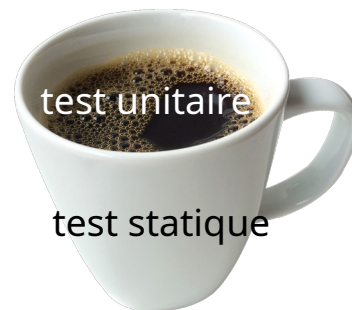
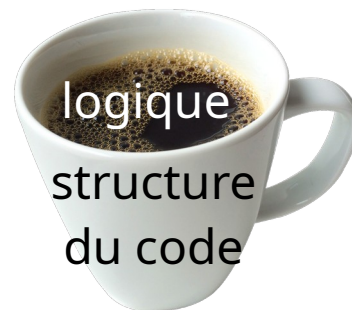
invalid-bool-returned (E0304):
__bool__ does not return bool Used when a `__bool__` method returns something which is not a bool

invalid-bytes-returned (E0308):
__bytes__ does not return bytes Used when a `__bytes__` method returns something which is not bytes

Exercices

Ce que je pense des tests statiques

- J'aime bien l'analogie du contenant et du contenu pour distinguer les tests unitaires des tests statiques.



Conclusion

- **Pour continuer avec l'analogie de la tasse à café :**
 - C'est bien beau d'avoir le meilleur café au monde, mais si on le boit avec une tasse qui rend son appréciation compliquée, c'est moins bon finalement.





Questions?

Références

- [1] S. Vessels, “How to review code effectively: A GitHub staff engineer’s philosophy,” The GitHub Blog. Accessed: Oct. 16, 2024. [Online]. Available: <https://github.blog/developer-skills/github/how-to-review-code-effectively-a-github-staff-engineers-philosophy/>
- [2] P. Louridas, “Static code analysis,” *IEEE Softw.*, vol. 23, no. 4, pp. 58–61, Jul. 2006, doi: 10.1109/MS.2006.114.
- [3] J. Kerievsky, *Refactoring to patterns*. in Addison-Wesley signature series. Boston: Addison-Wesley, 2005.
- [4] “Pylint features - Pylint 4.0.0-dev0 documentation.” Accessed: Oct. 16, 2024. [Online]. Available: https://pylint.readthedocs.io/en/latest/user_guide/checkers/features.html