





**SOLID**

SINGLE-RESPONSIBLE PRINCIPLE

MAGNUM2-QUALITÉ D'SI-THO MAS HAE SLÉ & DUE NINBURG

**« There should never be more than one reason for a `class` to change. »,**

*Robert C. Martin a.k.a Uncle Bob*

**Peut être modifiée en**

**« There should never be more than one reason for an `abstraction` to change. »**

**Cela se traduit dans les logiciels par:**

- **Créer des abstraction qui ont un faible couplage entre elles**
- **Séparer les données et les comportements: *Visitor pattern, Iterators, Functional programming, Modular programming***

INDÉPENDANTES

# SOLID

## SINGLE-RESPONSIBILITY PRINCIPLE

« There should never be more than one reason for a `class` to change. »,  
*Robert C. Martin a.k.a Uncle Bob*

Peut être modifiée en

« There should never be more than one reason for an `abstraction` to change. »

Cela se traduit dans les logiciels par:

- Créer des abstraction qui ont un faible couplage entre elles
- Séparer les données et les comportements: *Visitor pattern, Iterators, Functional programming, Modular programming*

INDÉPENDANT DES IDIOMES



# SOLID

## OPEN-CLOSED PRINCIPLE

« Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification », *Bertrand Meyer*

« You should be able to extend the behavior of a system without having to modify that system. », *Uncle Bob*

Indissociable du S, cela se traduit dans les logiciels par:

- La définition de contrats publics / implémentations privées
- L'extension grâce au polymorphisme : sous-typage, paramétrique (a.k.a générique) ou ad-hoc