

Single responsibility

Une classe est un sous-ensemble cohérents d'actions

Quand est-ce qu'une classe en fait trop ?

Questionner la classe

- Pour les attributs ou les méthodes en lecture, de type query : “Mme [classe], quelle est votre [méthode] ?”
- Pour les méthodes d'action, de type commande : “Mme [classe], que fait votre [méthode] ?”

Les indicateurs de liaison de différentes actions avec un ET, on doit vérifier si le tout constitue un ensemble d'actions cohérentes.

Exemple une classe avec différentes actions de type CRUD est un ensemble cohérent car elle gère le cycle de vie d'un objet

Les indicateurs de liaison de différentes actions avec un OU

Exemple un classe qui gère à la fois l'envoi de mail OU l'envoi de SMS ... hummm, c'est probable qu'elle en fait trop de chose différentes et on sent bien qu'il vaut mieux utiliser un interface qui va émerger avec des implémentations concrètes.

Décrire ce que fait la classe

- Si contient un “ET” : à vérifier
- Si contient un “OU” : probablement trop

Cartes CRC

-Si une classe a besoin de connaître trop de chose, on peut alors se poser des questions.

-Qu'est-ce qu'une classe sait ce qu'elle ne devrait pas savoir ->

Typiquement pour le cas des dépendances, on doit éliminer les dépendances qui n'ont pas lieu d'y être là.

-Avec qui -> Quels sont les objets qui s'interagissent et en lien avec ma classe?

Utiliser les cartes CRC

- Qu'est-ce que ma classe a besoin de savoir pour mener à bien ma mission ?
- Qu'est-ce que ma classe sait qu'elle ne devrait pas savoir ?
- Avec qui est-ce qu'elle parle ?
- Dans quel but ?

S'interroger sur le nombre de lignes et indentations au niveaux des méthodes de classes, s'il y en a trop, on doit se poser des questions

=> Décomposer (refactoring) si plus de 40 lignes de codes et plus de 2 indentations ...

Principe de fractale

- Ce qui s'applique à l'objet s'applique aussi à ses méthodes !
- Est-ce que mes méthodes font trop de choses ?