

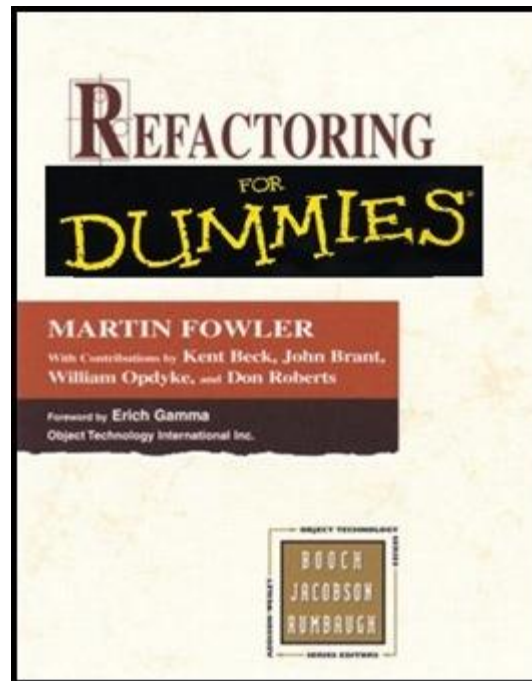
Refactoring

Dung Le Hoang

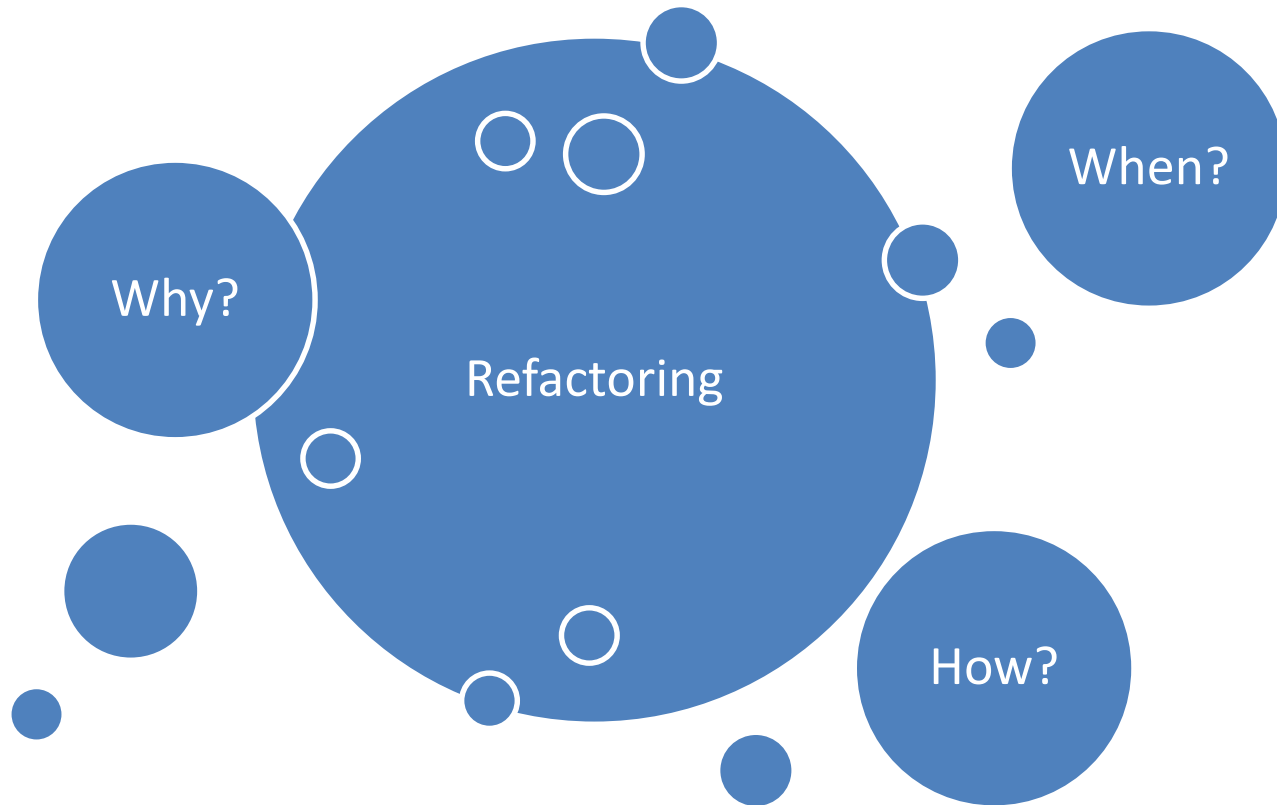


Refactoring software

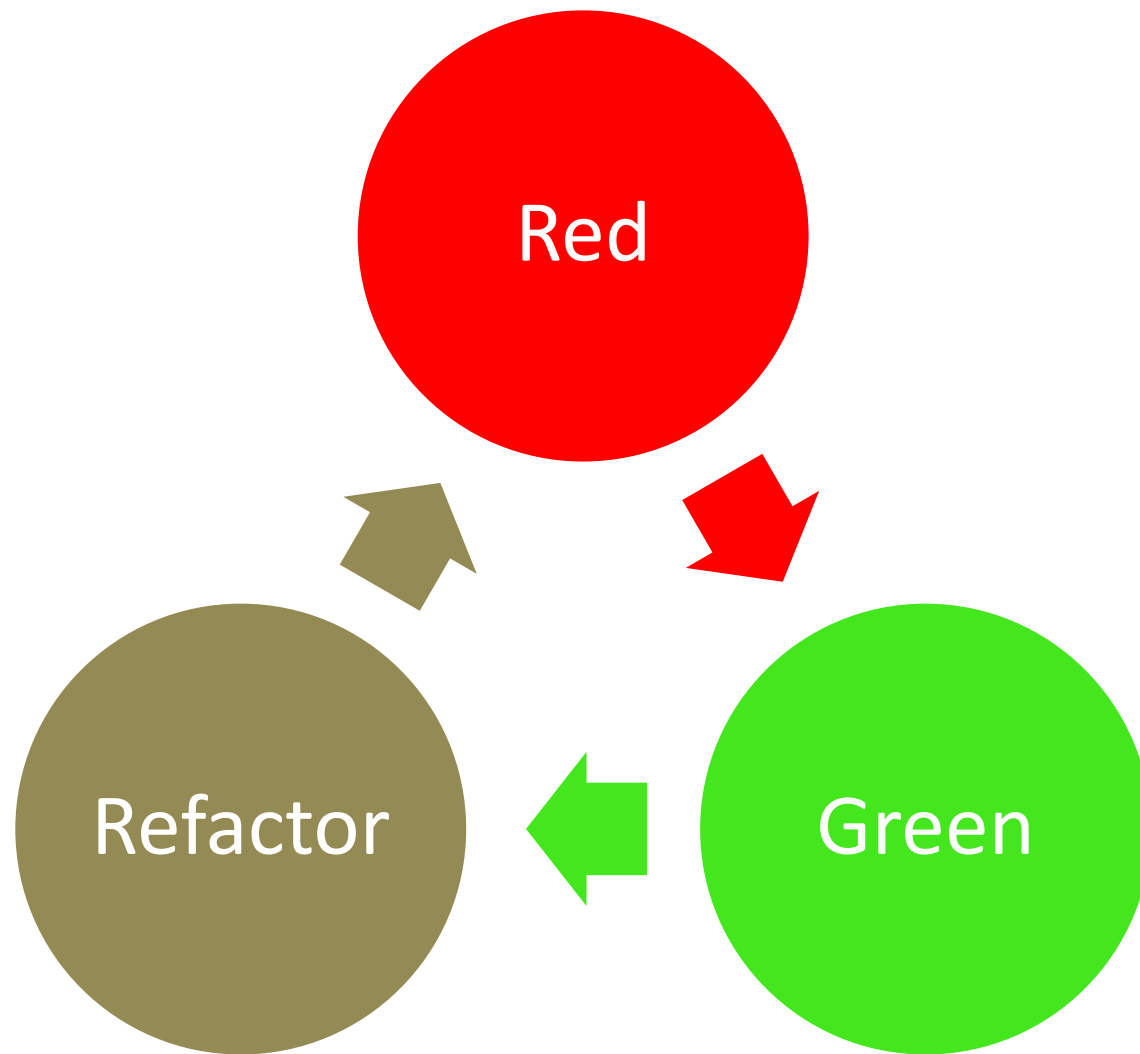
- Changing the internal implementation without changing the external functionality



Topics



The cycle

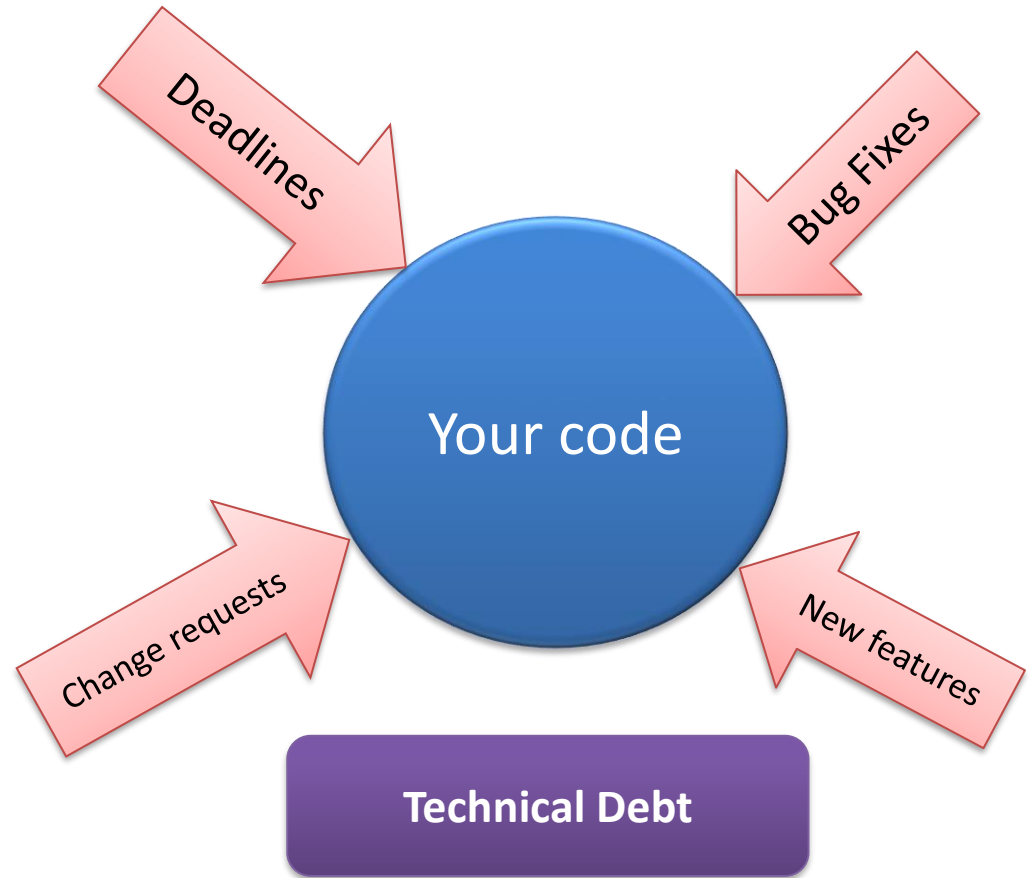


When refactoring done?



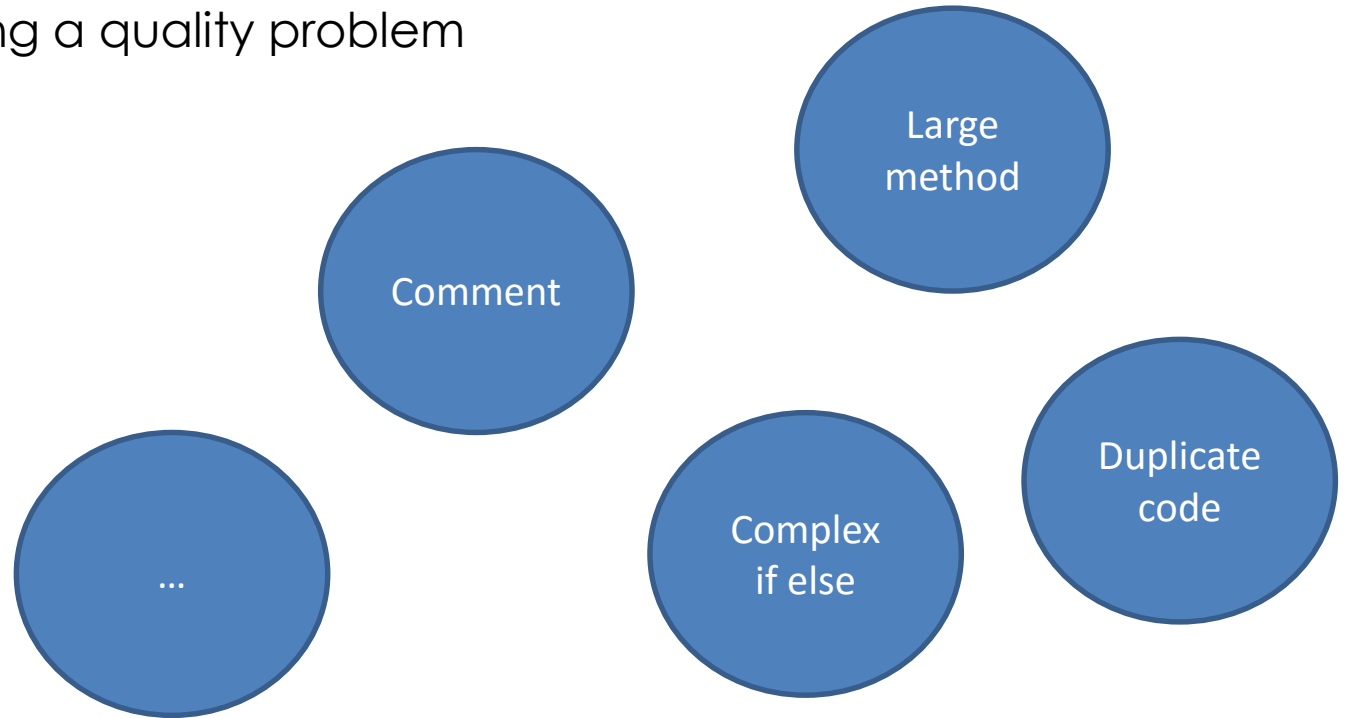
Why refactor?

- To improve the quality of the code
 - Readability
 - Maintainability
 - Scalability
 - Extensibility



When to refactor?

- After fixing a failing test
- Before adding a new feature
- After identifying a quality problem



When not to refactor?

- You have a fear of breaking the software



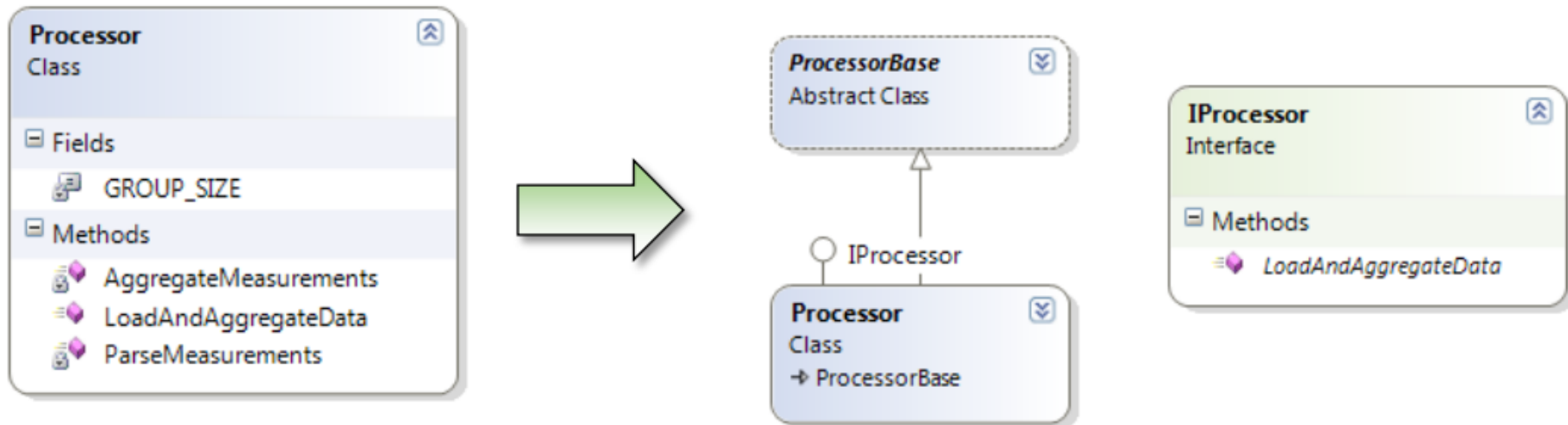
There are no unit
test for this code

Simple Refactoring

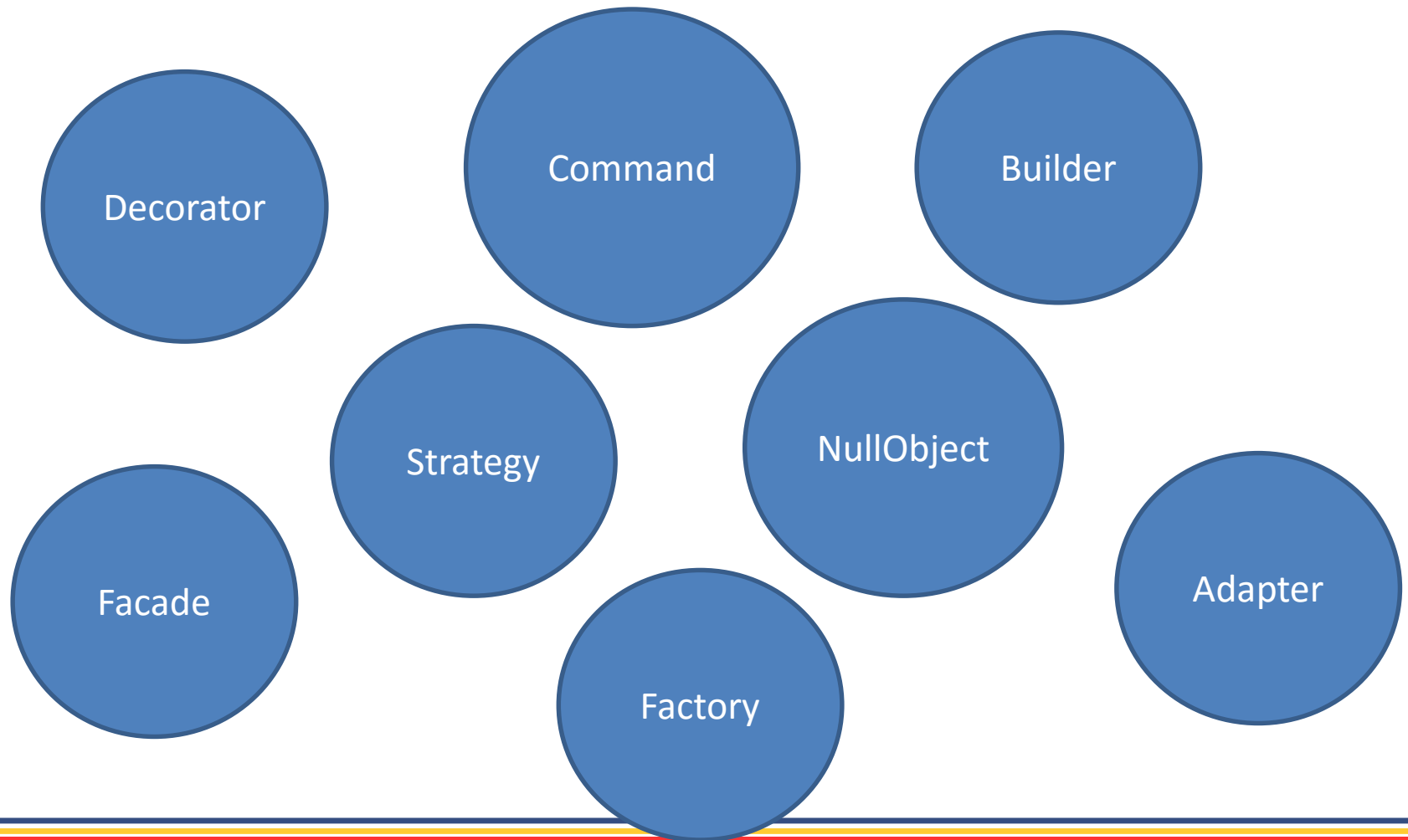
- Rename
- Introduce Parameter
- Extract method

Refactoring to abstraction

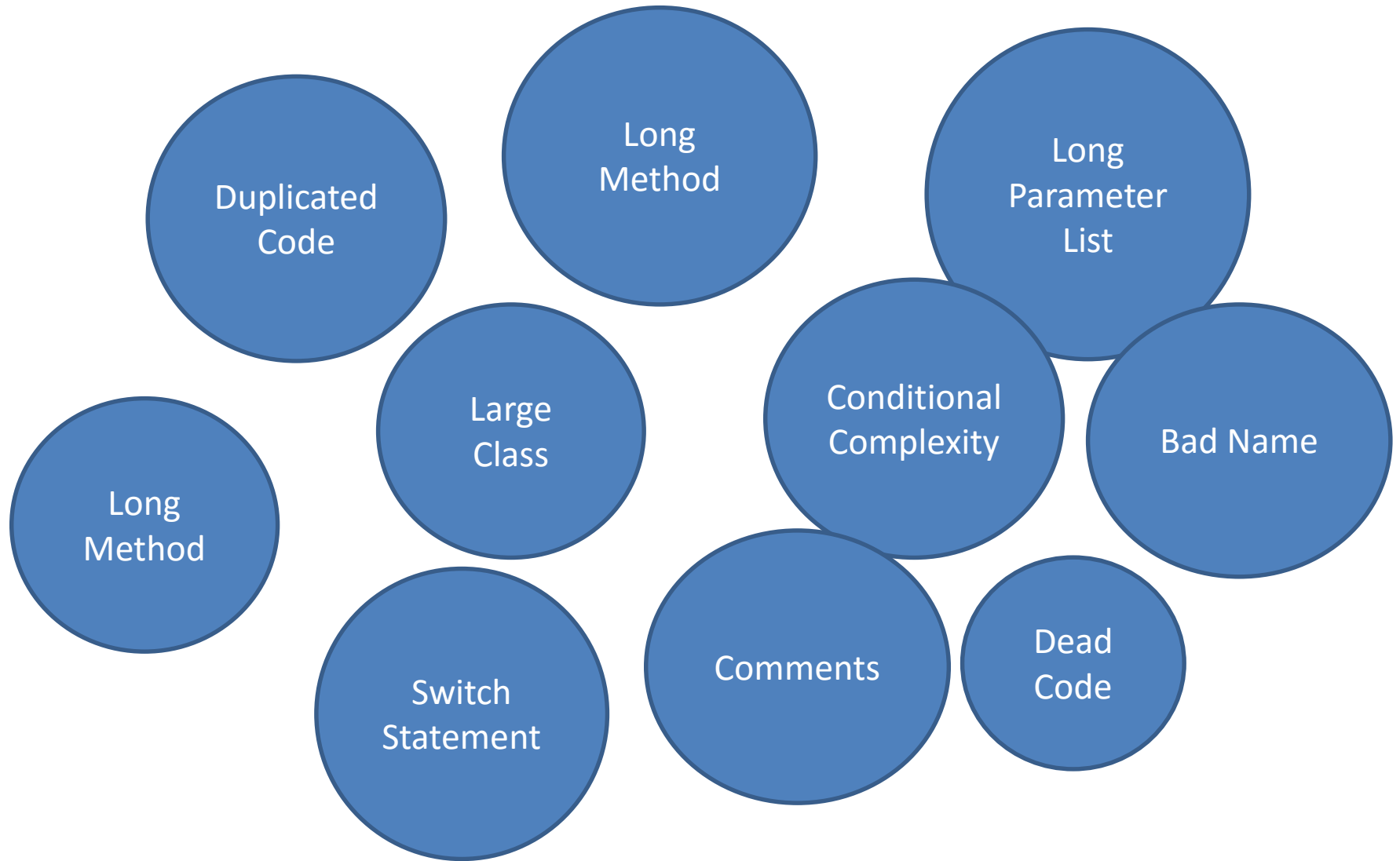
- Extract interface
- Extract superclass



Refactoring to Design Pattern



Code Smells



Summary



Simplicity is beauty