Useful Software Engineering Tools

Useful Software Engeneering Tools

- Object Calisthenics
- Code Smells
- Test Driven Development
- Transormation Priority Premise

Object Calisthenics

1. Use only one level of indentation per method
2. Don’t use the else keyword
3. Wrap all primitives and strings
4. Use only one dot per line
5. Don’t abbreviate
6. Keep all entities small
7. Don’t use any classes with more than two instance variables
8. Use first-class collections
9. Don’t use any getters/setters/properties

Object Calisthenics

Goal:
Best practices to empower full potential of object oriented programming languages

Do‘s:
- Katas
- High Quality Code

Don‘ts:
- Take for granted
- Overusing

Code Smells

Bloaters

Change Preventers

Dispensables

Couplers

Object-Orientation Abusers

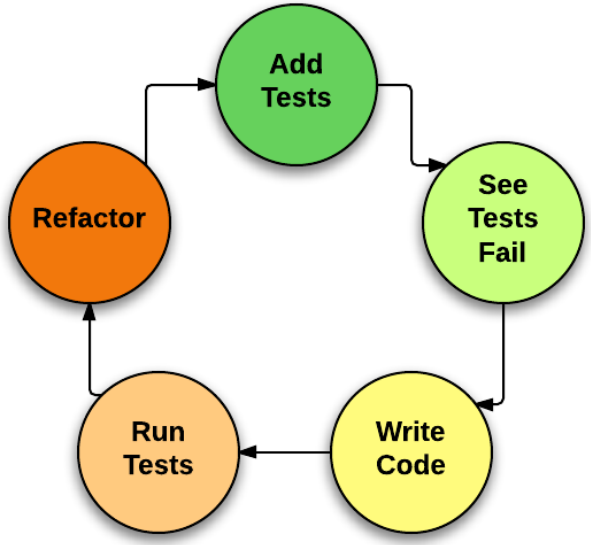
Code Smells

Goal:
Identify potentially bad-designed code by distinctive patterns

Usage:
- Refactoring
- Vocabulary extension in Pair/Mob
- Code Reviews

Pitfalls:
- Banning smells in general
- Ignoring smells for the right reasons

Test Driven Development



Test Driven Development

Goal:
Guiding developer to focus on one thing at a time (test, implement, refactor)

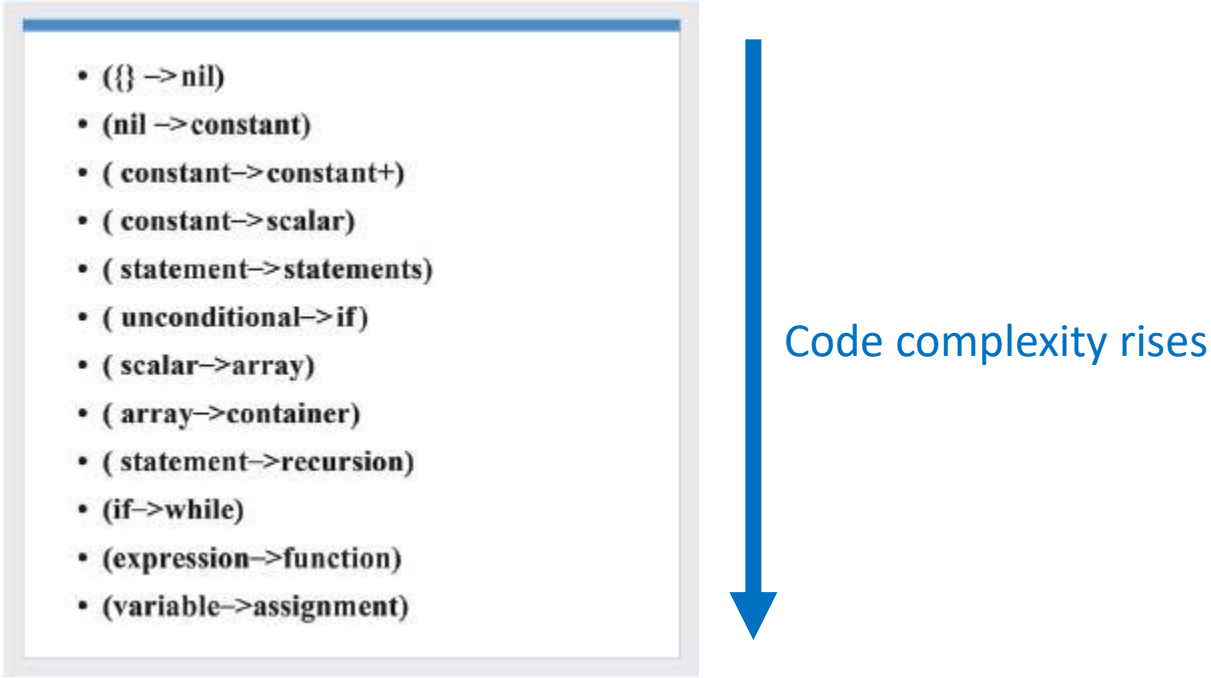
Usage:
- Pair/Mob
- New fatures (greenfields)

Pitfalls:
- Legacy code
- Needs previous knowledge
- Doesn‘t garantee good design

Transormation Priority Premise

- ({} –>nil)
- (nil –>constant)
- (constant–>constant+)
- (constant–>scalar)
- (statement–>statements)
- (unconditional–>if)
- (scalar–>array)
- (array–>container)
- (statement–>recursion)
- (if–>while)
- (expression–>function)
- (variable–>assignment)

Transormation Priority Premise

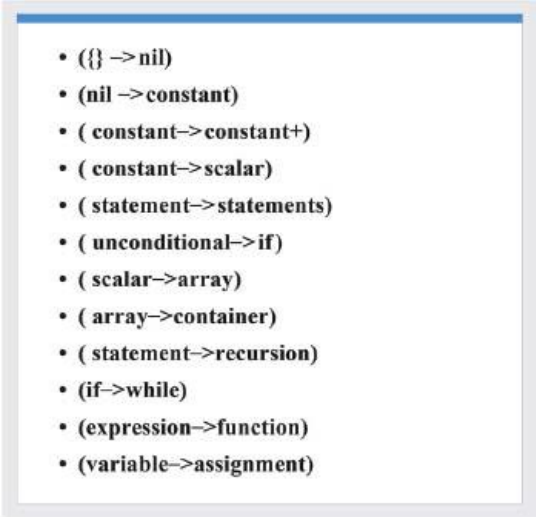


Transormation Priority Premise

Goal:
Change behavoir of code by transform it from to specific into more generic

Usage:
- Complements TDD approach
- Cheet sheet
- Guideline after exposing a code smell

Pitfalls:
- Useless without concrete „bad“ example
- Don‘t confuse with Refactoring



Summary

- Tool benefits may have similarities
- Each tool has different motivations
- Use the right tools in right context

Sources:
Code Smell Images from https://sourcemaking.com/refactoring/smells

Contact:
samuel.degelo@gmail.com