

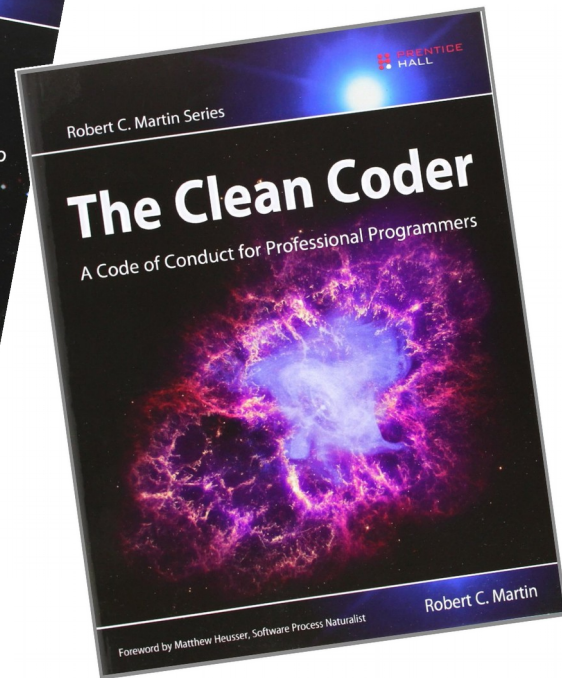
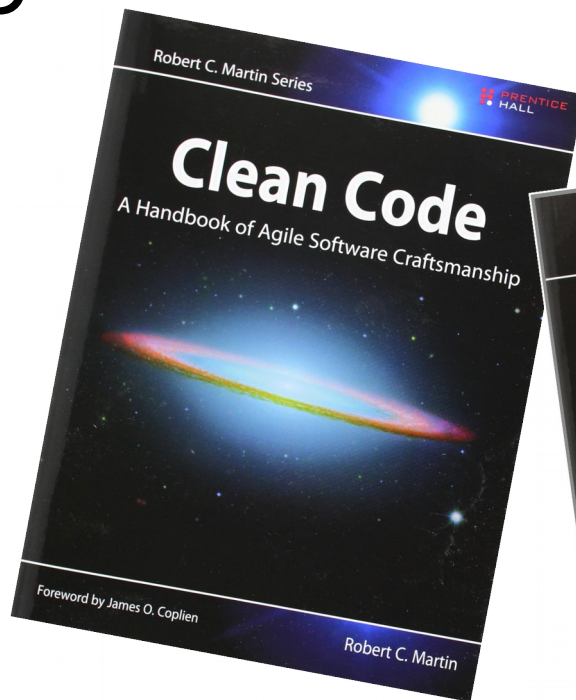


Test Driven Development

Gdańsk, 20 stycznia 2018

www.infoshareacademy.com

Trzy zasady TDD

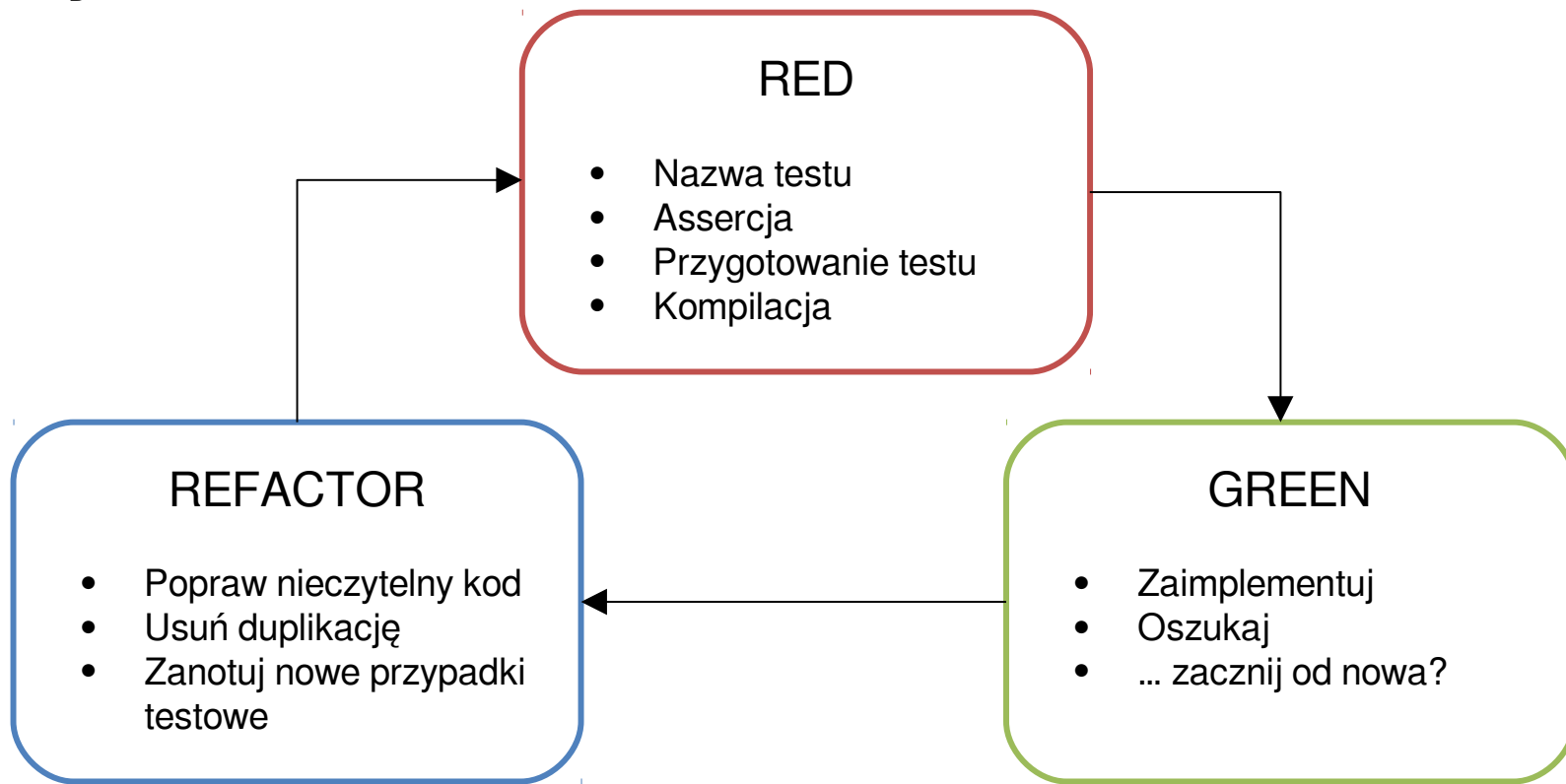


Trzy zasady TDD

1. Możesz pisać kod produkcyjny tylko, jeżeli napisałeś do niego test jednostkowy
2. Możesz napisać tylko tyle kodu testowego, żeby test uruchamiał się, ale nie przechodził
3. Możesz napisać tylko tyle kodu produkcyjnego, żeby przeszedł test, który napisałeś

<http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>

Cykl TDD



Bowling game kata

Punktacja w kręglach:

1. gra ma dziesięć rund (frame)
2. w każdej rundzie gracz ma dwie rzuty (roll), żeby strącić dziesięć kręgli
3. gracz zdobywa tyle punktów ile strącił kręgli, plus bonusy za “strike” i “spare”
4. “strike” jest wtedy, gdy gracz strąci dziesięć kręgli w jednym rzucie, dostanie wtedy dodatkowo tyle punktów ile strąci kręgli w dwóch następnych rzutach
5. “spare” to sytuacja, gdy gracz strąci dziesięć kręgli w dwóch rzutach, dodatkowo dostaje wtedy tyle punktów ile strąci kręgli w następnym rzucie
6. jeżeli gracz zdobędzie “strike” albo “spare” w dziesiątej rundzie to ma dwa (jeden w przypadku “spare”) dodatkowe rzuty na zdobycie dodatkowych punktów
7. punkty zdobyte po ostatniej rundzie naliczane są tylko raz

<http://www.bowlinggenius.com/>

Bowling game kata

```
git checkout $9_bowling_game_kata  
com/infoshare/junit/bowling/Game.java  
com/infoshare/junit/bowling/BowlingGameTest.java
```

1. Program nie powinien sprawdzać poprawności gry tzn. punktacja naliczana jest tylko po wszystkich rzutach
2. Pair programming - zamieniacie się klawiaturą po każdej fazie Red i Refactor
3. Zaczniście od najprostszyc przypadków
4. Po zaimplementowaniu dwóch testów zaczynamy dyskusję na temat rozwiązania

Dziękuję za uwagę