

Test Driven Development w formie Coding Dojo

Dlaczego?

Wierzę, że należy tworzyć dobre jakościowo oprogramowanie.

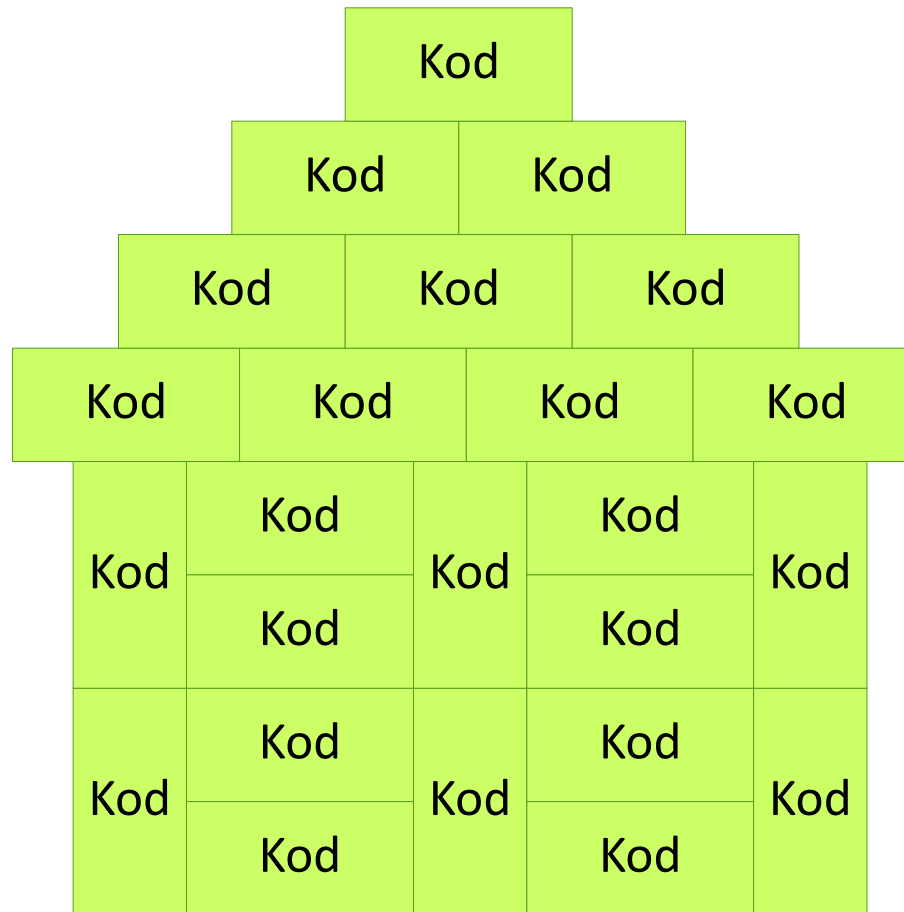
Jak?

Kod

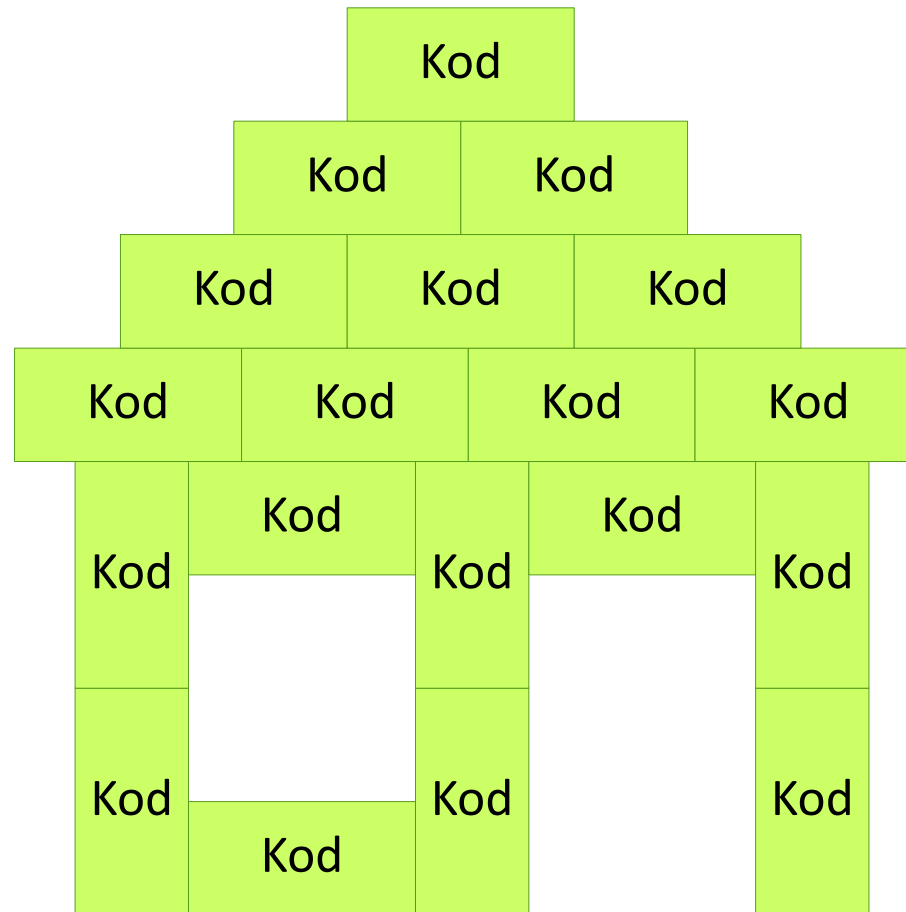
Napisz kod

Kod	Kod	Kod	Kod	Kod
	Kod		Kod	
Kod	Kod	Kod	Kod	Kod
	Kod		Kod	

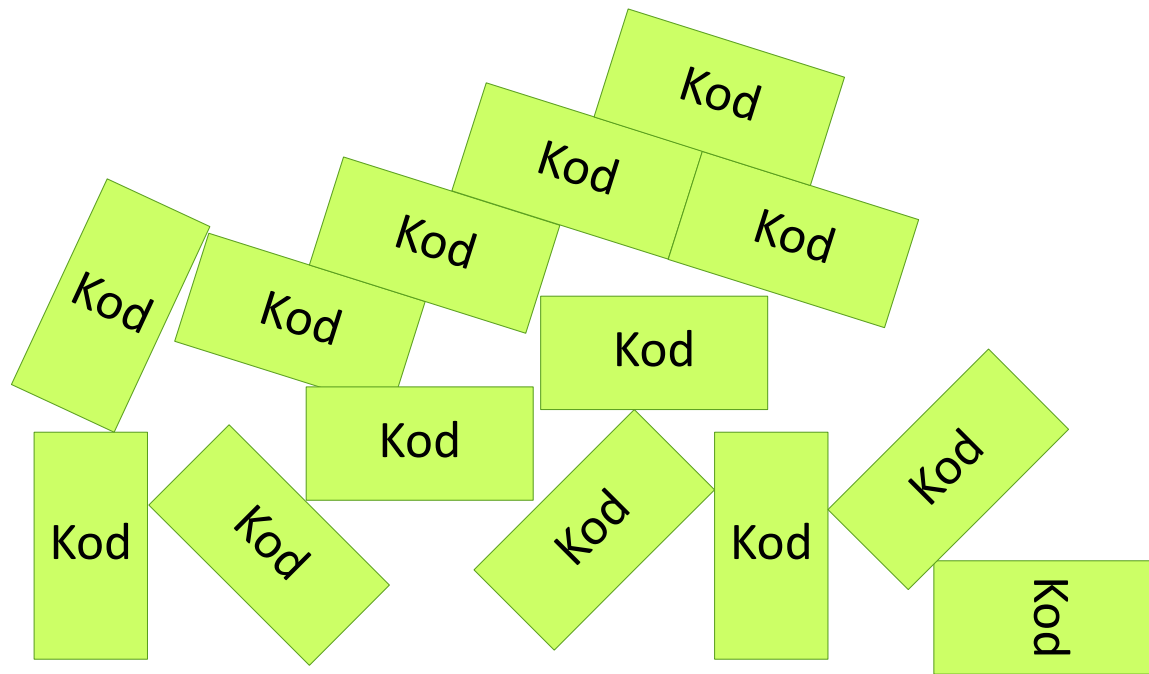
Nowa funkcjonalność



Zmiana funkcjonalności



Ups



Co dalej?

- Przywracamy poprawną wersję z repozytorium
- Debuggujemy i naprawiamy kod
- Dopisujemy testy automatyczne

Jak dodać testy?

Brak testów

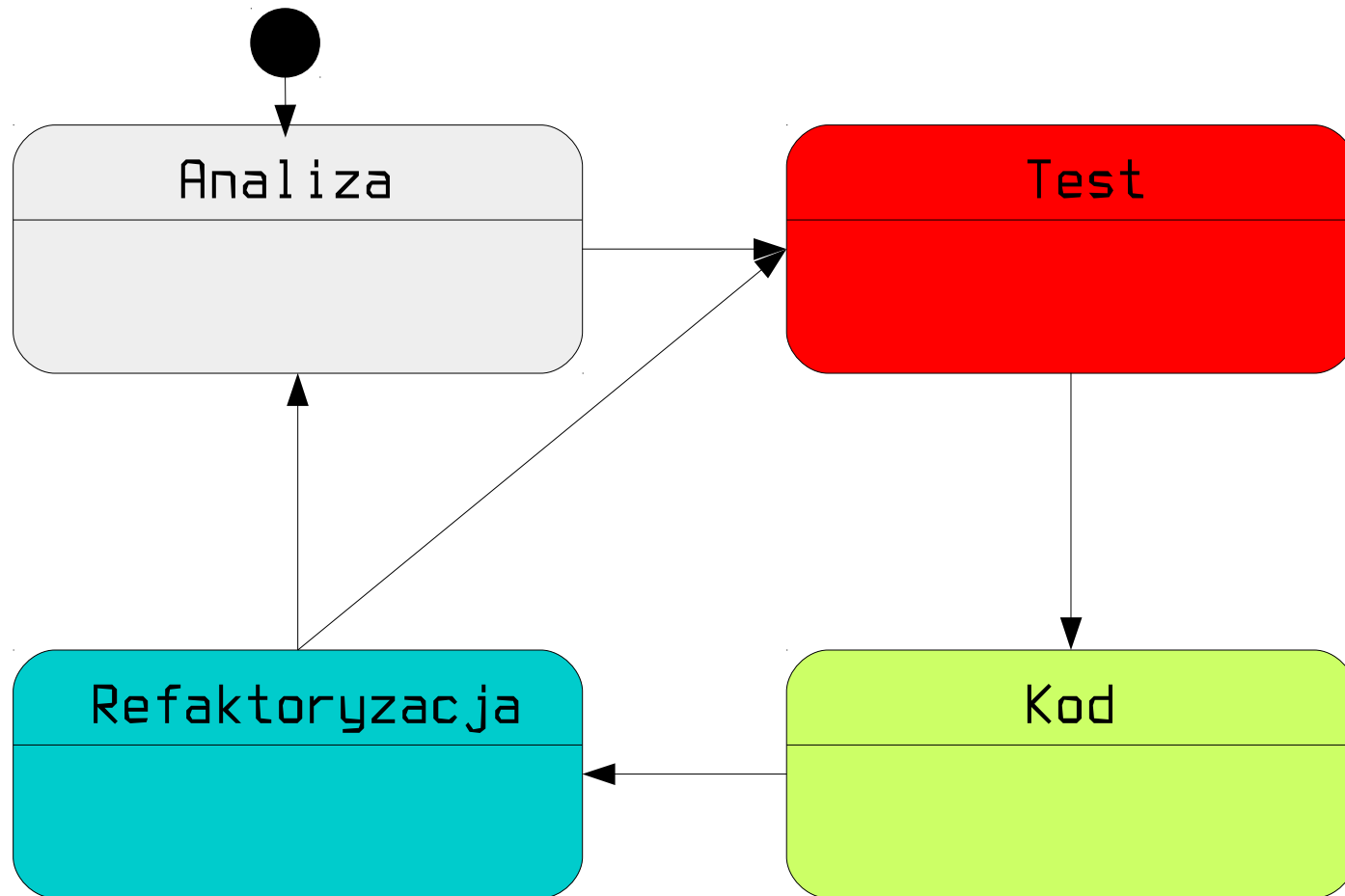
- Brak refaktoryzacji
- Kod spaghetti
- Trudne do wdrożenia testy

- Brak dokumentacji?
- Kod odziedziczony?
- Przepisać od zera?
- Jak nie powtórzyć błędów?

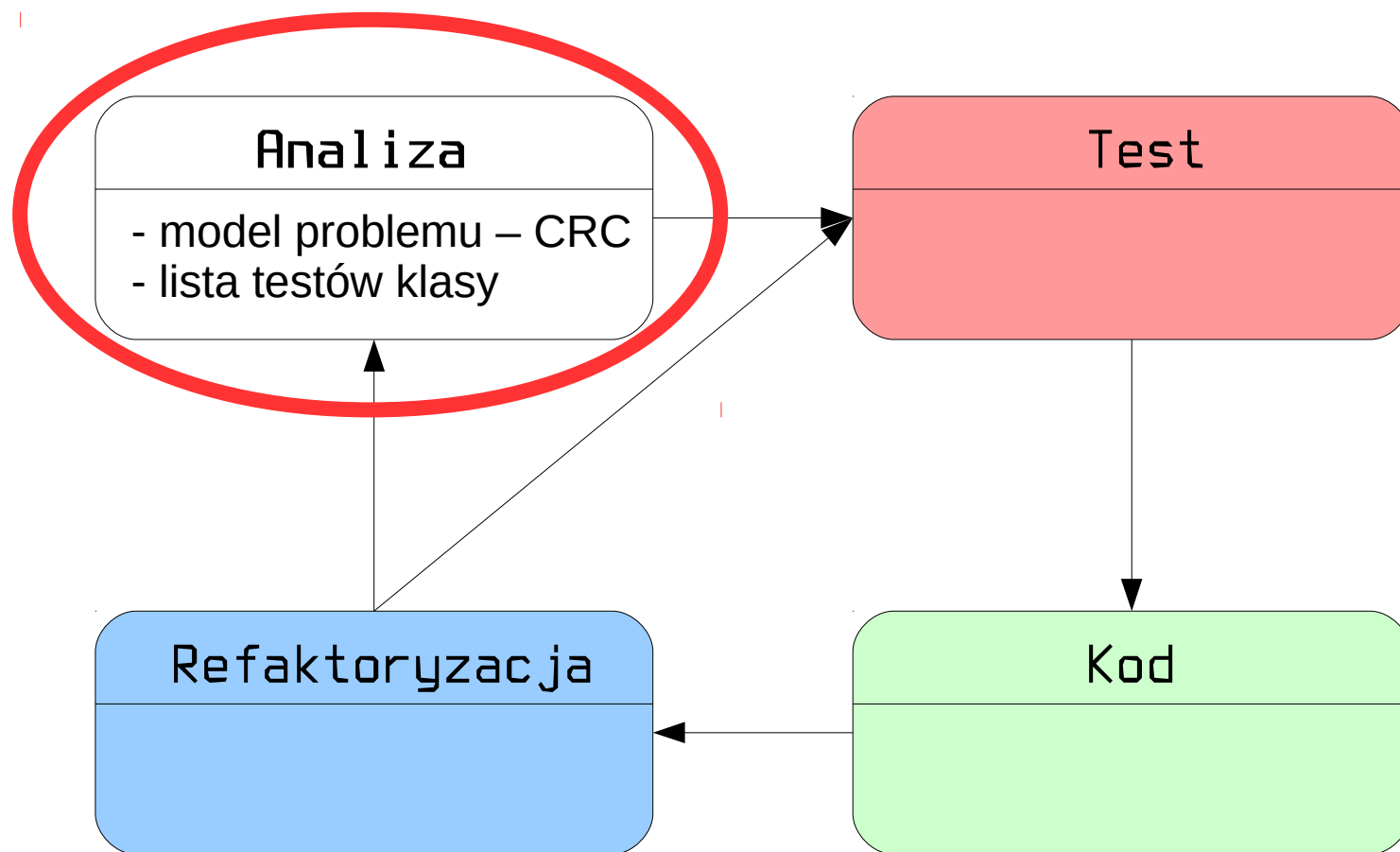
Test Driven Development

(TDD)

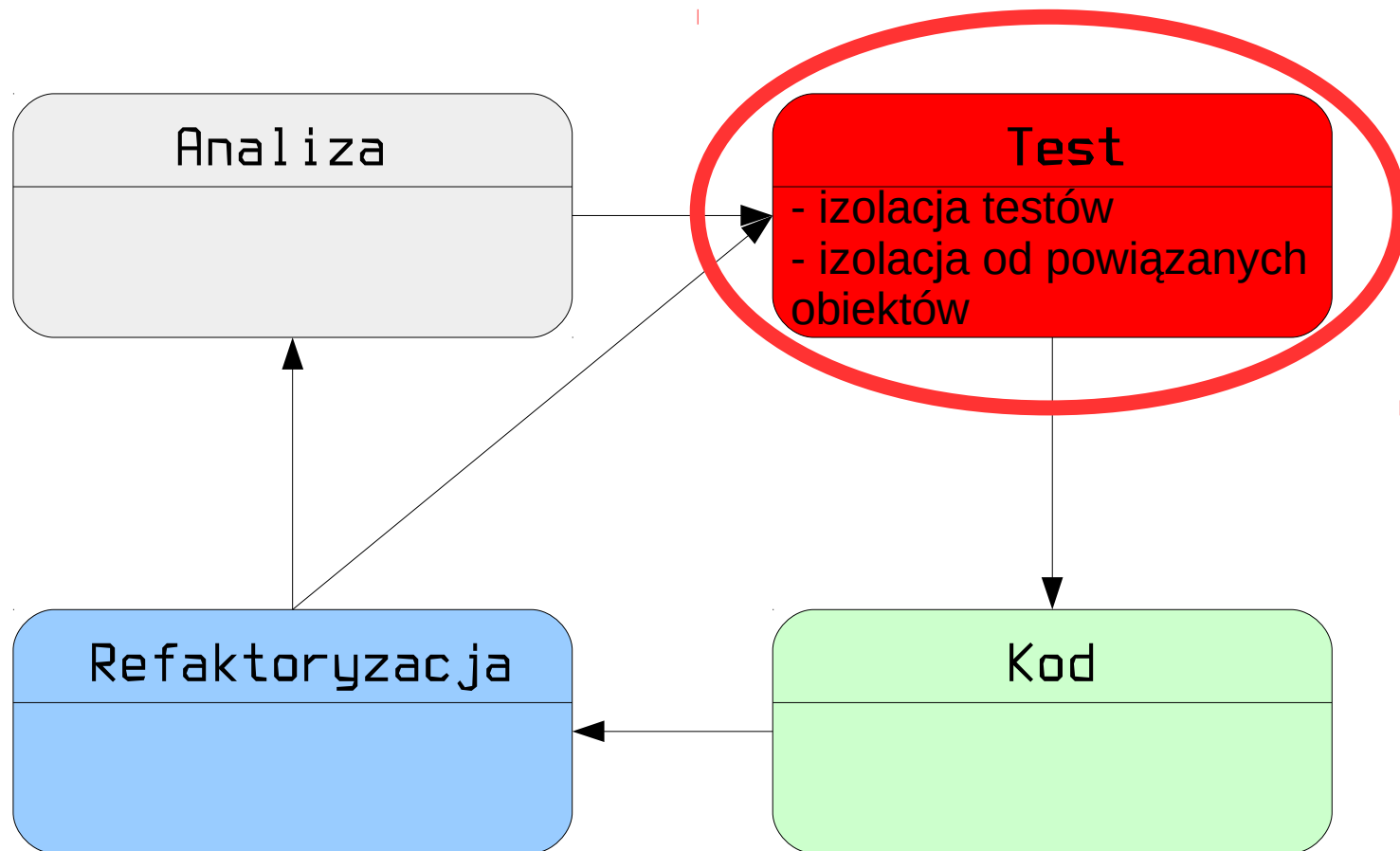
Cykl TDD



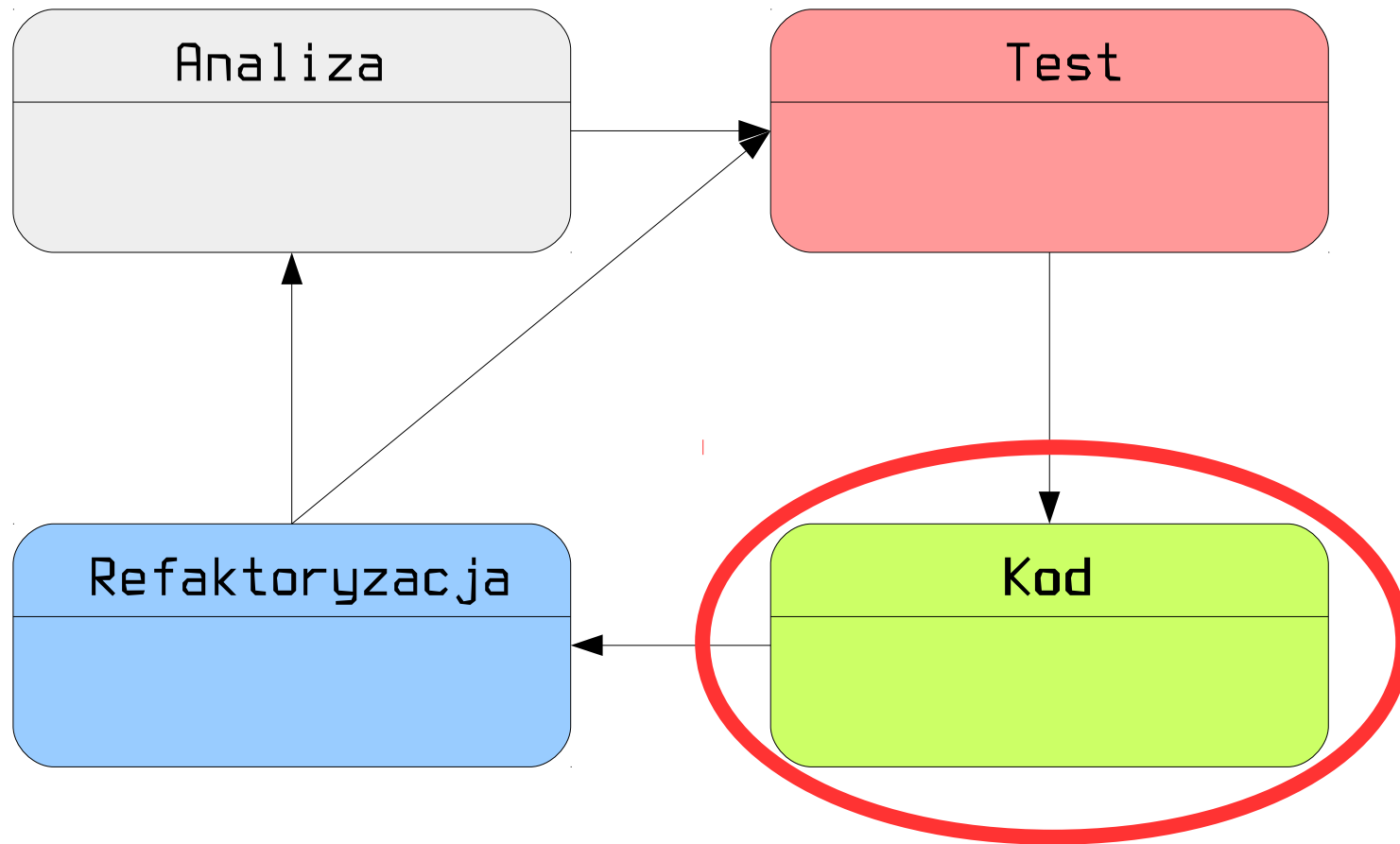
Cykl TDD



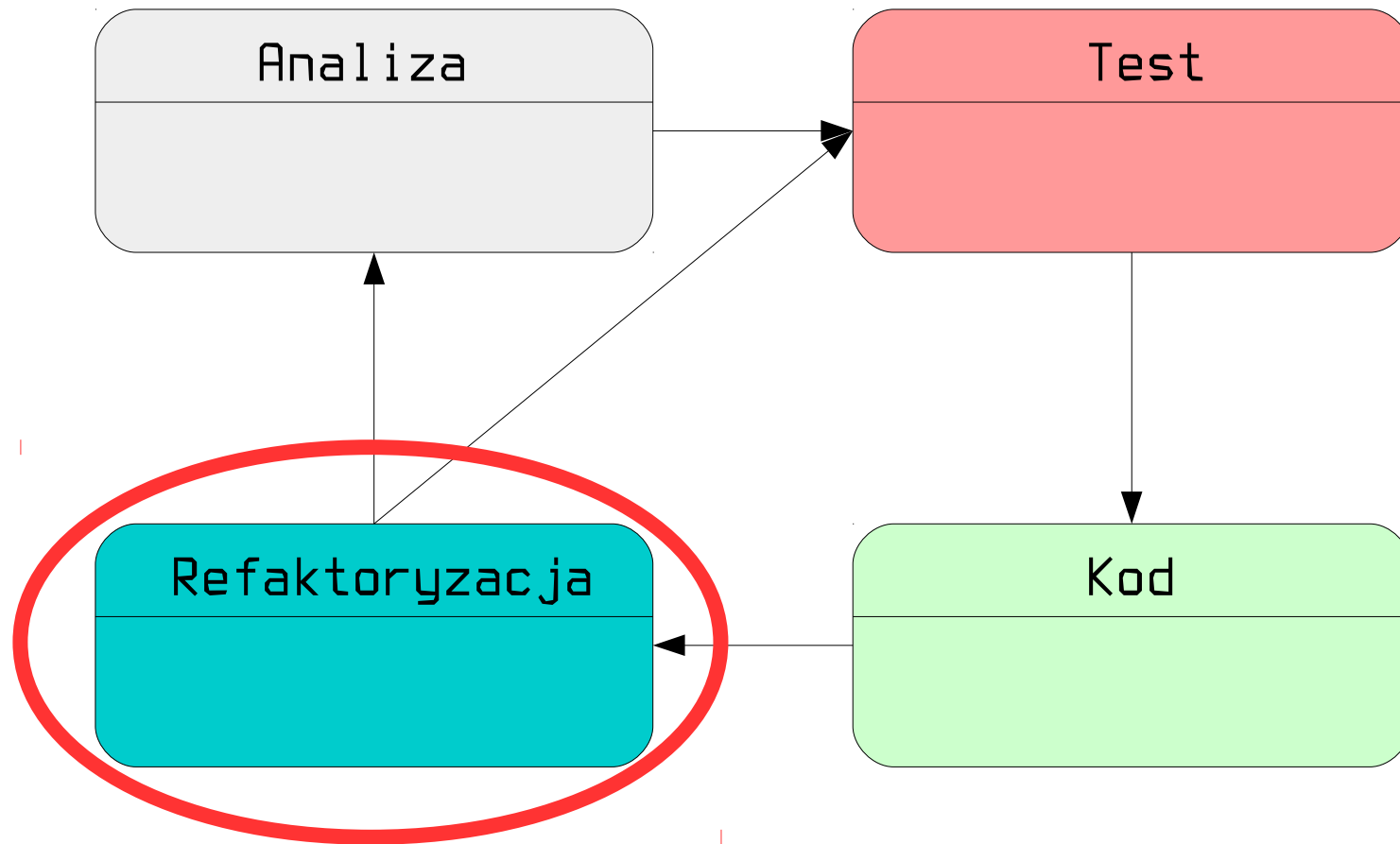
Cykl TDD



Cykl TDD



Cykl TDD



Furia TDD

- Test – Kod – Refaktoryzacja – 10%
- Test – Kod – Refaktoryzacja – 20%
- Test – Kod – Refaktoryzacja – 30%
- Test – Kod – Refaktoryzacja – 40%
- Test – Kod – Refaktoryzacja – 50%
- Test – Kod – Refaktoryzacja – 60%
- Test – Kod – Refaktoryzacja – 70%
- Test – Kod – Refaktoryzacja – 80%
- Test – Kod – Refaktoryzacja – 90%

Demo



Nie działa

- Klient: „Coś nie działa...”
- My: „Hmm, przecież przechodzi testy!”
- Klient: „Jakie testy? Właśnie testuję i nie działa! Poza tym to nie robi tego co chciałem.”

Co poszło nie tak?

- Testowanie klas i metod TYLKO w izolacji
- Tworzenie kodu, a nie funkcjonalności

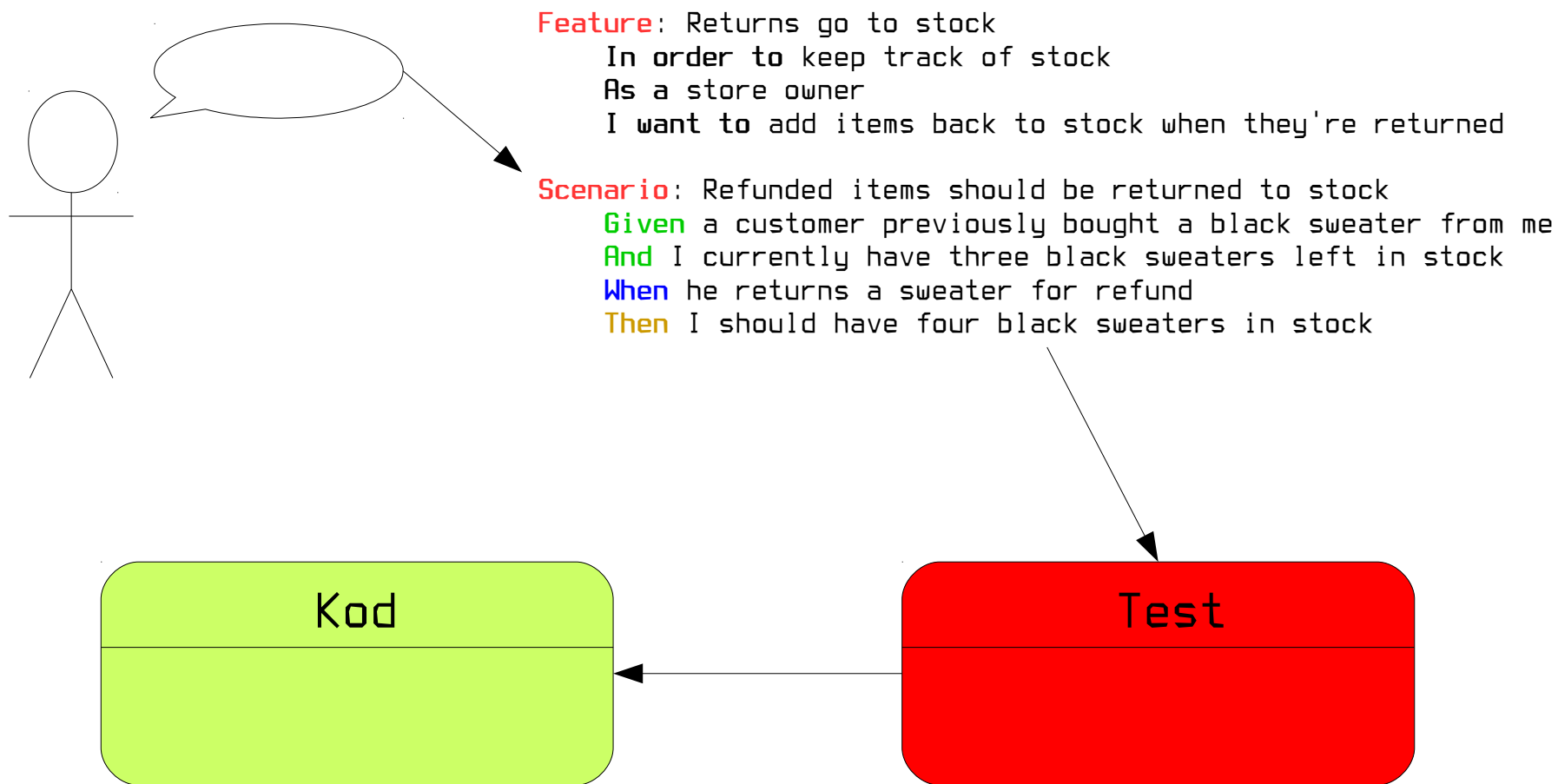
TDD w małym projekcie = łatwo

TDD w dużym projekcie = trudniej

Behaviour Driven Development

(BDD)

Idea BDD



Furia BDD ;)

- Scenariusz – Test – Kod – Refaktoryzacja – 10%
- Scenariusz – Test – Kod – Refaktoryzacja – 20%
- Scenariusz – Test – Kod – Refaktoryzacja – 30%
- Scenariusz – Test – Kod – Refaktoryzacja – 40%
- Scenariusz – Test – Jak te testy powolne... – Kod – Refaktoryzacja – 50%
- Scenariusz – Nie mam czasu na testy – Kod – Refaktoryzacja – 60%
- Scenariusz – Kod – Ups – Debugger...

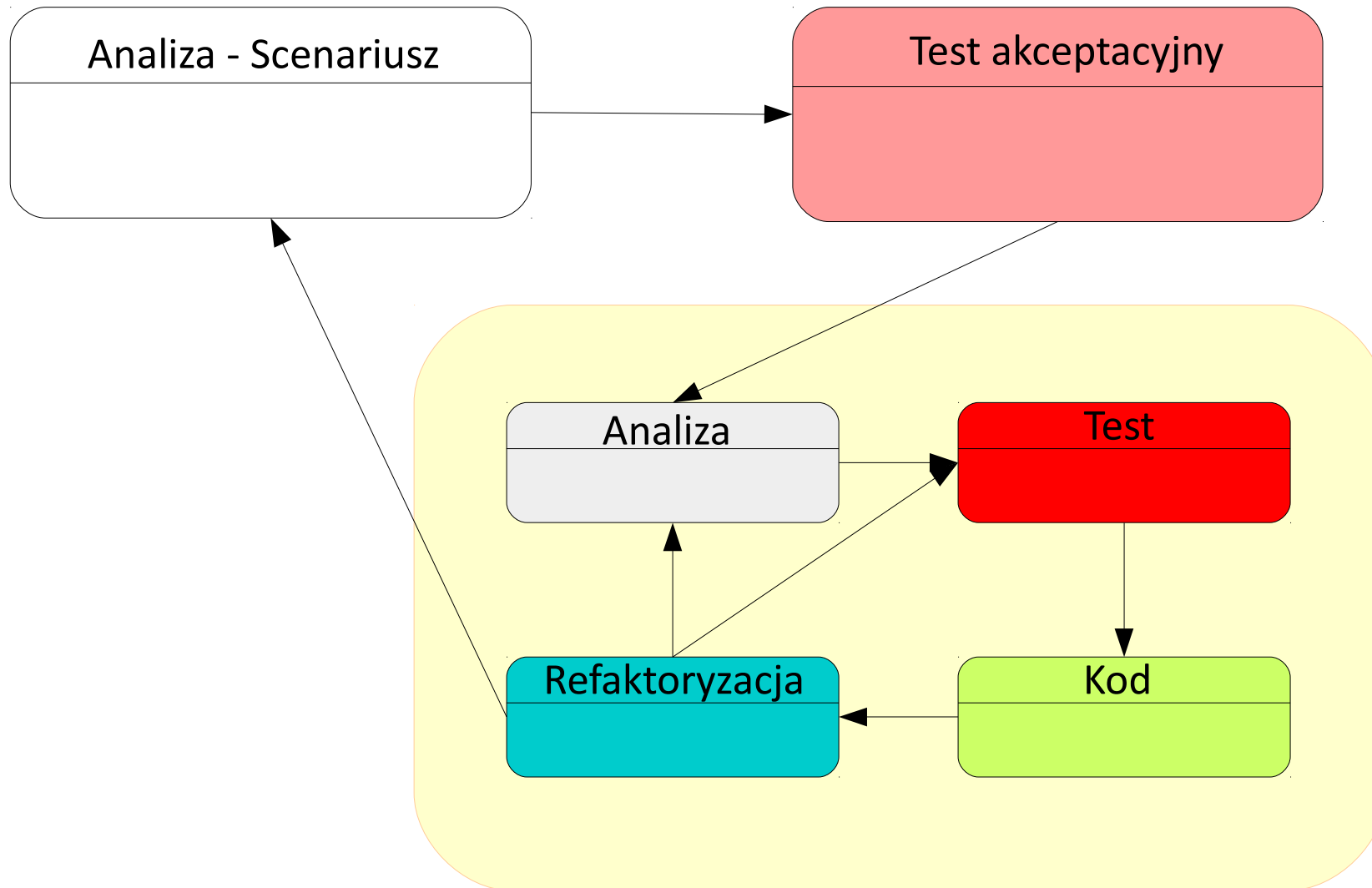
Wooolne BDD

- sprawdzanie wszystkich kombinacji parametrów
- zapisy i odczyty z bazy
- zapisy i odczyty z dysku
- zapisy i odczyty z sieci
- uruchamianie przeglądarki

Sprawdzanie całego scenariusza
dla każdego możliwego rozgałęzienia w kodzie

Więc jak testować?

Double Loop



Zyski

- Rozległe i szybkie testy
- Kod zgodny z wymaganiami
- Bezpieczna refaktoryzacja
- Bardzo dokładne wskazanie miejsca z błędem

Koszty

- Trzeba się tego nauczyć :)

Coding Dojo

Coding Kata

- Szybkie do zakodowania problemy
- Pozornie proste
- Zmuszają do innego spojrzenia na problem
- Zwykle wywołują burzliwą dyskusję :)

Formy ćwiczeń

- Samodzielne kata
- Randori
- Kata zespołowe

Pair programming

- Zapoznanie z innymi sposobami realizacji
- Poprawa komunikacji
- Ujednolicenie bazy pojęć i skrótów
- Propagacja wiedzy i umiejętności

Retrospekcja

- Dyskusja na temat rozwiązań
- Podkreślenie wypracowanych dobrych rozwiązań
- Zwrócenie uwagi na pułapki
- Wyciągnięcie wniosków do zastosowania w pracy zawodowej

Dlaczego?
Dobre jakościowo oprogramowanie

Jak?
Test Driven Development

Co?
Coding Dojo

Dziękuję

Zgłoszenia na Coding Dojo
jkulesza@grupazpr.pl
W temacie proszę dodać: [CD]

Jeżeli masz jakieś pytania, napisz.
Chętnie pomogę.

jakub@stolarscy.com