

Oskar Szrajer

TDD - unit test

<http://gotar.info>

<http://ragnarson.com>

oskarszrajer@gmail.com

gotar@ragnarson.com

Test jednostkowy

*Zautomatyzowany test pisany przez programistę
testujący pojedynczy element systemu w izolacji*

Test jednostkowy

Zautomatyzowany test pisany przez programistę testujący pojedynczy element systemu w izolacji

1. Automatyzacja

Test jednostkowy

Zautomatyzowany test pisany przez programistę testujący pojedynczy element systemu w izolacji

1. Automatyzacja
2. Programista (autor programu)

Test jednostkowy

Zautomatyzowany test pisany przez programistę testujący pojedynczy element systemu w izolacji

1. Automatyzacja
2. Programista (autor programu)
3. Pojedynczy element systemu (“unit”)

Test jednostkowy

Zautomatyzowany test pisany przez programistę testujący pojedynczy element systemu w izolacji

1. Automatyzacja
2. Programista (autor programu)
3. Pojedynczy element systemu (“unit”)
4. Izolacja

Test jednostkowy

Zautomatyzowany test pisany przez programistę testujący pojedynczy element systemu w izolacji

1. Automatyzacja
2. Programista (autor programu)
3. Pojedynczy element systemu (“unit”)
4. Izolacja

Na co komu testy jednostkowe

- natychmiastowy feedback
- wyraźna lokalizacja błędu
- lepszy design kodu
- ...

Automatyzacja

rspec

```
# Gemfile
```

```
group :development do
```

```
  gem 'rspec'
```

```
end
```

```
$ bundle
```

Automatyzacja

guard

```
# Gemfile
```

```
group :development do
```

```
  gem 'rspec'
```

```
  gem 'guard'
```

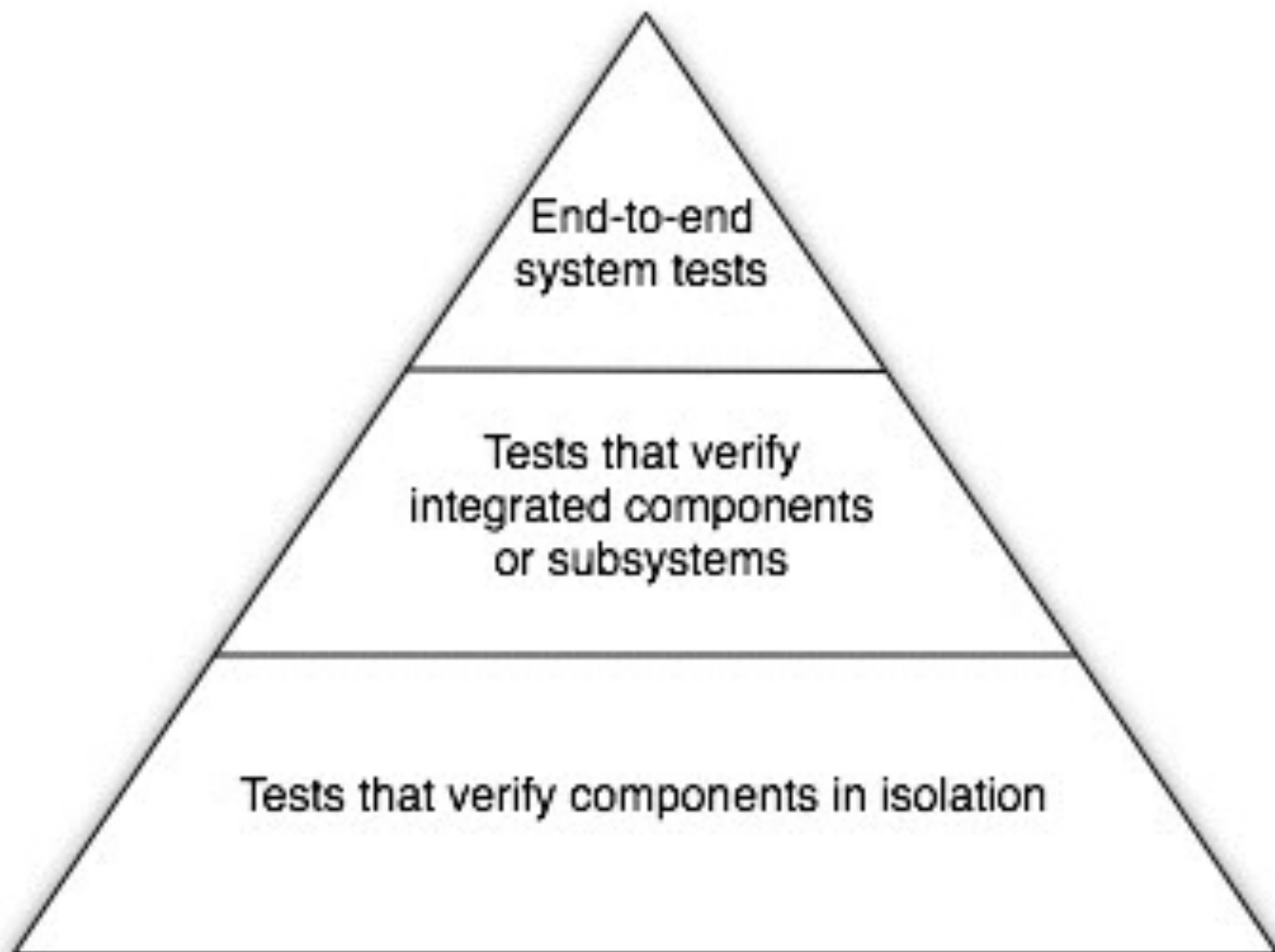
```
  gem 'guard-rspec'
```

```
end
```

```
$ bundle
```

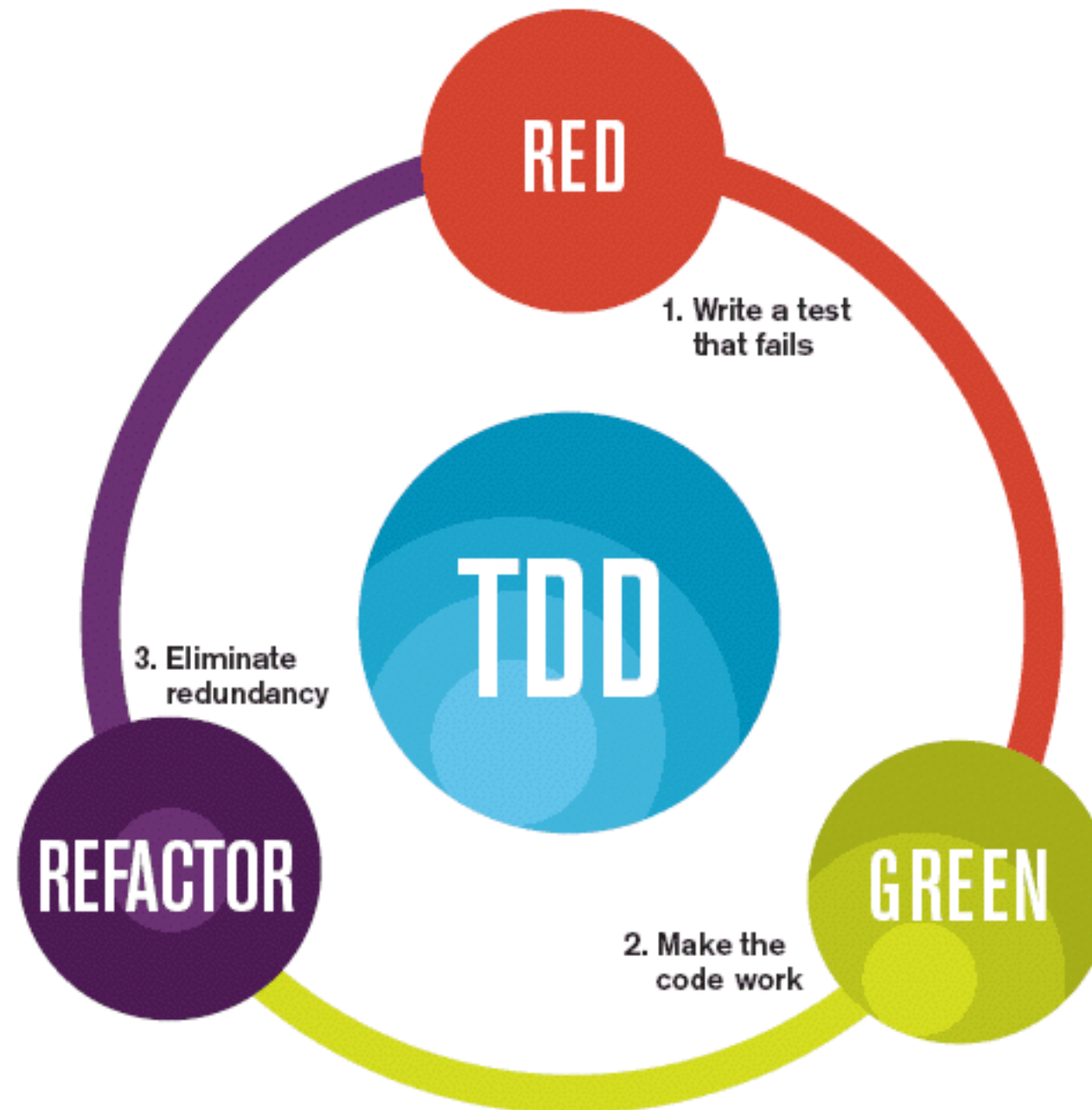
Top-Down/Bottom-Up





Test-first / Test-last

Red -> Green -> Refactor



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Refaktor

- Gdy nasze rozumienie problemu się zwiększa, design ewoluuje i kryteria co traktować jako unit się zmieniają.
- Gdy kod się zmienia, testy też muszą się zmienić.

Izolacja

- Pliki
- bazy danych
- duża ilość require
- fixtures, factory_girl, ...
- spec_helper.rb
- File.open, ...
- system, IO.popen, ...
- ENV, ARGV

Mocki / Stuby

Prawo Demeter

(Zasada minimalnej wiedzy lub Reguła ograniczania interakcji)

"rozmawiaj tylko z (bliskimi) przyjaciółmi"

W pełnej formie mówi ono, iż metoda danego obiektu może odwoływać się jedynie do metod należących do:

- metod tego samego obiektu,
- metod dowolnego parametru przekazanego do niej,
- dowolnego obiektu przez nią stworzonego,
- dowolnego składnika, klasy do której należy dana metoda.



“śmierdzące” testy

- Duża ilość require'ów.
- Duża ilość mocków.
- Wiele linii setupu, długie mocki.
- Warunki (if/case) w kodzie testowym
- `send(:foobar)`

Wyjątki

- nauka, poznawanie bibliotek, eksperymentowanie
- testy integracyjne
- testy wydajnościowe

Pytania?