

# Metodyki zwinne wytwarzania oprogramowania

## Wykład 5

Marcin Młotkowski

9 listopada 2016

# Plan wykładu

- 1 Testowanie akceptacyjne
  - FitNesse
  - Testowanie graficznego interfejsu użytkownika (GUI)
- 2 Zapewnienie jakości kodu
  - Code review
  - Ciągła integracja
  - Czysta inżynieria oprogramowania

# Co to jest

- fork **FIT**: Framework for Integrated Test
- portal w stylu Wiki
- Skąd można pobrać: [www.fitnessse.org](http://www.fitnessse.org)

## Istotne cechy

### Przeznaczenie

Testy warstwy zaraz "za interfejsem graficznym".

## Istotne cechy

### Przeznaczenie

Testy warstwy zaraz "za interfejsem graficznym".

### Łatwy w użyciu

Rodzaj serwisu WWW w stylu *wiki*, który może być "programowany" również przez klienta (użytkownika) niebędącego programistą

## Istotne cechy

### Przeznaczenie

Testy warstwy zaraz "za interfejsem graficznym".

### Łatwy w użyciu

Rodzaj serwisu WWW w stylu *wiki*, który może być "programowany" również przez klienta (użytkownika) niebędącego programistą

### Znaczniki

Prosty język znaczników umożliwia łatwe tworzenie testów.

# Przykład

Przeliczanie temperatur

Przeliczanie stopni Celsjusza na Fahrenheita

## Przykład testu

```
|Fahrenheit|  
|Celsius|temp?|  
|0        | 32.0      |  
|100     | 212.0     |  
|40      | 104.0     |
```



## Przykład testu

### Uzupełnienie

```
!define TEST_SYSTEM {slim}  
!path /home/marcinm/fitnesse/FahrenheitTest
```

```
|Fahrenheit|  
|Celsius|temp?|  
|0        | 32.0        |  
|100     | 212.0       |  
|40      | 104.0       |
```

# Implementacja funkcjonalności

## Przypomnienie

Fahrenheit		
Celsius temp?		
0	32.0	
100	212.0	
40	104.0	

```
public class Fahrenheit {  
    private double cels;  
  
    public void setCelsius(double c) { this.cels = c; }  
    public double temp() { return 32 + (9/5)*this.cels; }  
}
```

## Rodzaje testów

- **Decision table:** tabela argumentów i poprawnych wartości
- **Query table:** zapytania w stylu zapytań do bazy danych
- **Script table:** tworzenie scenariuszy

## Dodatkowe wtyczki

Fit JfcFixtures

Testy akceptacyjne dla programów Java Swing

## Dodatkowe wtyczki

### Fit JfcFixtures

Testy akceptacyjne dla programów Java Swing

### HtmlFixture

Testy akceptacyjne dla serwisów webowych.

# Rodzaje GUI

- Testowanie interfejsów lokalnych
- Testowanie serwisów WWW

# Selenium

- wtyczka do Firefoxa
- może zapisywać scenariusze ("wyklikane" przez użytkownika) i je odtwarzać
- scenariusze mogą być zapisywane w Ruby, Python, .NET, Java, PHP

# CubicTest

Graficzna wtyczka do Eclipse wspierająca tworzenie testów  
Selenium



# AutoHotKey

## Cele programu

Symulowanie użytkownika poprzez "udawanie" naciskania klawiszy  
czy klikania myszką

# AutoHotKey

## Cele programu

Symulowanie użytkownika poprzez "udawanie" naciskania klawiszy czy klikania myszką

- przede wszystkim dla systemów Windows
- są wersje Mono/Linux
- **AutoKey**: wzorowany na AutoHotkey, na Linux/X11

# AutoHotKey

## Przykład skryptu

```
WinGet, wowid, list, World of Warcraft
;2locks

1::
KeyWait, 1, D
    ControlSend,, 1 , ahk_id %wowid1%
    ControlSend,, h , ahk_id %wowid2% ; assist macro /target
Sleep, 500
    ControlSend,, 1 , ahk_id %wowid2%

Return
```

# Xnee

Dedykowany system do X11, z możliwością zapisywania o odtwarzania akcji użytkownika (naciśnięcia klawiszy, ruchy myszki).

# System dedykowany: Linux Desktop Testing Project

Testowanie programów graficznych:

- Gnome
- KDE
- Java (SWING)
- Firefox

# System dedykowany: Linux Desktop Testing Project

Testowanie programów graficznych:

- Gnome
- KDE
- Java (SWING)
- Firefox

Cechy

- możliwość zapisywania i odtwarzania akcji użytkownika
- programowanie akcji: Python

# Test Automation FX

Środowisko do Visual Studio. Cechy:

- możliwość zapisywania akcji użytkownika
- implementacja własnych skryptów w C#

# Plan wykładu

- 1 Testowanie akceptacyjne
  - FitNesse
  - Testowanie graficznego interfejsu użytkownika (GUI)
- 2 Zapewnienie jakości kodu
  - Code review
  - Ciągła integracja
  - Czysta inżynieria oprogramowania



# Inspekcja kodu

Systematyczne przeglądanie kodu w celu znalezienia błędów.

# Sposoby code review

- podczas programowania w parach;

## Sposoby code review

- podczas programowania w parach;
- wyznaczenie osoby odpowiedzialnej za przeglądy;

## Sposoby code review

- podczas programowania w parach;
- wyznaczenie osoby odpowiedzialnej za przeglądy;
- wykorzystanie oprogramowania do wyszukiwania błędów;

## Sposoby code review

- podczas programowania w parach;
- wyznaczenie osoby odpowiedzialnej za przeglądy;
- wykorzystanie oprogramowania do wyszukiwania błędów;
- organizowanie spotkań.

# Ciągła integracja

Łączenie wyników pracy programistów. Najlepiej łątwo i kilka razy dziennie.

# Ciągła integracja

Łączenie wyników pracy programistów. Najlepiej łątwo i kilka razy dziennie.

## Dlaczego

Wczesne wykrywanie problemów integracji (liczba argumentów w wywołaniach, różnice kompilatorów/bibliotek, etc).

# Elementy integracji

- kompilacja całej aplikacji;
- testowanie całej aplikacji;



# Praktyczna realizacja ciągłej integracji

- wspólne repozytorium kodu;

# Praktyczna realizacja ciągłej integracji

- wspólne repozytorium kodu;
- automatyczna budowa kodu wynikowego;

## Praktyczna realizacja ciągłej integracji

- wspólne repozytorium kodu;
- automatyczna budowa kodu wynikowego;
- testy automatyczne;

## Praktyczna realizacja ciągłej integracji

- wspólne repozytorium kodu;
- automatyczna budowa kodu wynikowego;
- testy automatyczne;
- automatyczne uruchamianie integracji po commicie;

# Idea

najlepiej robic wszystko dobrze od razu.

# Realizacja

- formalne definiowanie specyfikacji;
- dowodzenie twierdzeń o poprawności;
- statystyczna kontrola jakości (oparta na statystycznych przewidywaniach dot. wykorzystania aplikacji);