Paweł Rajba
pawel@ii.uni.wroc.pl
http://www.kursy24.eu/

Podstawy testowania

Agenda

- Wprowadzenie
- Czym wspierać testowanie?
- Rodzaje testów?
- Co testujemy?
- Kiedy testować?
- Jak budować testowalne systemu?
- Moq

Wprowadzenie

- Debuggowanie i testowanie to nie to samo
- Testowanie to systematyczne próby „zepsucia” aplikacji
- Programiści nie są przyzwyczajeni do poświęcania 40% swojego czasu na testy
  - Dla krytycznych projektów (np. kontrola lotów, monitorowanie reaktorów) testy mogą kosztować 5 razy więcej niż wytworzenie samego kodu

Wprowadzenie

- Co daje testowanie?
  - Wykrycie błędów, których nikt się nie spodziewał
  - Automatyczne testy pozwalają na szybkie wykrycie błędów i niezgodności
  - Upewnienie się, że system działa zgodnie ze specyfikacją
  - Utrzymywanie poprawnie działającego kodu
    - Gdy ktoś coś zmieni, łatwo wykryć, czy dalej jest ok

Czym wspierać testowanie?

- Dummy
  - Np. napis przekazywany do metody
- Fake
  - Np. sztuczna implementacja BD oparta na kolekcjach
- Stub
  - Obiekt, którego uczymy konkretnych zachowań
- Mock
  - Zaprogramowany „gotowiec” do udawania

Rodzaje testów

- Jednostkowe
- Integracyjne
- Wydajnościowe
- Obciążeniowe
- Akceptacyjne

Co testujemy?

- Stan (state verification)
- Zachowanie (behaviour verification)

Kiedy testować?

- Testy szybkie (np. jednostkowe)
  - Po każdym „check-in”
- Testy wolne (np. integracyjne)
  - Np. raz dziennie

- Continuous Integration

Jak budować testowalne sytemy?

- Znaczenie interfejsów
- Struktura projektu
  - Wzorzec „objects mother”
  - Rozkład projektów

Moq

- Oglądamy przykłady na
  - http://code.google.com/p/moq/wiki/QuickStart