Studia Podyplomowe: Java EE - produkcja oprogramowania

PRODUKCJA I TESTOWANIE (PTE)

Anna Derezińska
A.Derezinska@ii.pw.edu.pl
Instytut Informatyki
Politechnika Warszawska

Semestr zimowy 2010/2021
Grupa: JA20Z
(zesp. JA1-A, JA1-B) - semestr pierwszy

Plan zajęć

- Wstęp do testowania, testy jednostkowe z Junit, wytwarzanie oprogramowania sterowane testami (TDD)
- Metody testowania
- Wzorce projektowe
- Refaktoryzacja kodu
- Testowanie z wykorzystaniem zaślepek/obiektów pozornych (mock objects)

Literatura podstawowa

1. K. Beck TDD. Sztuka tworzenia dobrego kodu, Helion 2014 (K. Beck, *Test Driven Development by Example*, Addison-Wesley, 2002)
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Wzorce projektowe, WNT 2005, II wyd. WNT 2008, Helion 2010
3. M. Fowler i inni: Refaktoryzacja ulepszanie struktury istniejącego kodu, WNT 2006

Literatura uzupełniająca

4. J. W. Cooper: Java Wzorce projektowe, Helion 2001
5. A. Hunt, D. Thomas, JUnit. Pragmatyczne testy jednostkowe w Javie, Helion 2006

Materiały

Na MSTeams
dostępne na kanale: JA1-A-PTE przedmiot
w katalogu: Dokumenty

Dokumenty (pdf)

- PTE
- PTE_TestyJedn
- PTE_TDD
- PTE_MetodyTest
- PTE_WzorceProj
- java-pte2-v3 (*Fabryka Abstrakcji)*
- PTE_MockObjects
- PTE_Refaktoryzacja
- EclEmma – do pokrycia kodu w Javie
- JUnitRefact – o testach jednostkowych i refaktoryzacji

Przykłady

Na MSTeams
dostępne na kanale: JA1-A-PTE przedmiot, w katalogu: Przyklady

Projekty (Eclipse)

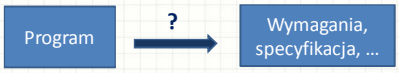
- JUnit5Examples .....zip
- TDDJUnit5Example.....zip
- DesignPatterns.....zip
- abstractFactory.zip
- Refactoring.....zip
- MockExamples.....zip

Wprowadzanie projektów do Eclipse

1. Otwieramy Eclipse
2. Importujemy projekt
   File->Import->General->Existing Projects into Workspace
   Next
3. W oknie ***Import Projects*** zaznaczamy ***Select archive file***:
   oraz wskazujemy wybrane archiwum projektu (.zip)

Weryfikacja & walidacja oprogramowania (V&V)

Å **Weryfikacja** – czy budujemy produkt właściwie
Weryfikujemy względem czegoś – np. specyfikacji



Weryfikacja:
Statyczna – przegląd, inspekcje kodu
Dynamiczna - testowanie

Å **Walidacja** (ocena, zatwierdzanie) – czy budujemy właściwy produkt, spełniający potrzeby klienta

Testowanie

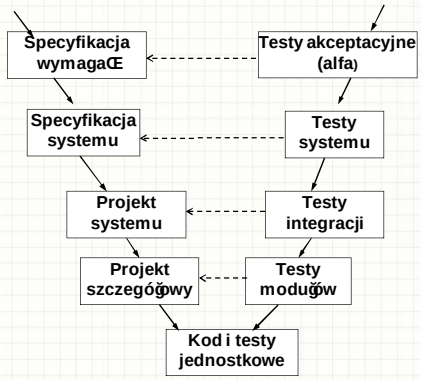
CELE testowania (poprawa jakości):
1. wykrycie defektów, zmniejszenie ryzyka użycia
2. ocena niezawodności i kosztów konserwacji oprogramowania (testy statystyczne)

Testowanie - stwierdza istnienie defektu, ale nie dowodzi braku błędów!

Osobnym zadaniem jest lokalizacja i naprawa

Dobry test:
wg 1 celu (*defect test*) - wykrywa błąd oprogramowania
wg 2 celu – symuluje typowe działanie

Fazy Testowania – model V



Testowanie w cyklu życia oprogramowania

Å W modelu kaskadowym/wodospadowym jako odrębna faza procesu (na końcu)

Å W modelach przyrostowych (inkrementalnych), jako jedna z podfaz każdego przyrostu

Å W metodykach zwinnych, np. opis testów jako część dokumentacji wymagań, promocja testów jednostkowych i TDD (*test-driven development*)

Przypadek testowy (Test case)

Przypadek testowy (Test case) – specyfikacja WE/WY (wartości testowe, spodziewane wyniki)

Wykrywamy błąd, gdy program zachowuje się niezgodnie z oczekiwaniem:
- generuje wyniki niezgodne ze specyfikacją,
- wykonuje czynności, których nie powinno robić, albo o których specyfikacja nie wspomina,
- nie spełnia określonych niefunkcjonalnych wymagań, np. działa za długo,
- nie działa w innym środowisku (inny komputer, system operacyjny, …)

Wymagania

1. Aktywna obecność na zajęciach i realizacja ćwiczeń
2. Wykonanie i prezentacja prac domowych

Ćwiczenia i prace domowe zwracane w postaci zadań (assignment) przez MSTeams