

Metodyki zwinne wytwarzania oprogramowania

Wykład 3

Marcin Młotkowski

19 października 2016

Plan wykładu

- 1 Styl pracy w programowaniu ekstremalnym
 - Organizacja miejsca pracy
 - Praca w parach
 - Praca nad oprogramowaniem
- 2 Styl programowania w XP
 - Projektowanie
 - Programowanie
 - Metafora

Miejsce pracy

Wspólne pomieszczenie dla wszystkich

- stanowiska pracy dla par programistów
- ściany przeznaczone na diagramy wypalania, diagramy UML etc

Komunikacja między parami

Można "podśluchiwać", co robią inne pary, albo jakie mają kłopoty.

Komunikacja między parami

Można "podśluchiwać", co robią inne pary, albo jakie mają kłopoty.

Można się "wtrącać".

Komunikacja między parami

Można "podśluchiwać", co robią inne pary, albo jakie mają kłopoty.

Można się "wtrącać".

To działa!

Sposób programowania

Programiści pracują w parach, przy jednym komputerze.

Sposób programowania

Programiści pracują w parach, przy jednym komputerze.

Jedna osoba programuje, a druga kontroluje i sugeruje usprawnienia.

Zmiany par

Programiści powinni zmieniać się w parach:

- w ramach pary co jakiś czas jest zamiana ról

Zmiany par

Programiści powinni zmieniać się w parach:

- w ramach pary co jakiś czas jest zamiana ról
- powinno się co jakiś czas zmienić partnera (np. raz dziennie)

Zmiany par

Programiści powinni zmieniać się w parach:

- w ramach pary co jakiś czas jest zamiana ról
- powinno się co jakiś czas zmienić partnera (np. raz dziennie)
- co jakiś czas przy zmianie pary można zmienić zadanie

Zmiany par

Programiści powinni zmieniać się w parach:

- w ramach pary co jakiś czas jest zamiana ról
- powinno się co jakiś czas zmienić partnera (np. raz dziennie)
- co jakiś czas przy zmianie pary można zmienić zadanie

Wskazówka

Przy zmianie zadania, np. na implementację GUI, "nowy" powinien być w parze z osobą mającą doświadczenie w zakresie GUI.

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;
- można poznać coś nowego;

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;
- można poznać coś nowego;
- kod jest krótszy i zawiera mniej błędów

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;
- można poznać coś nowego;
- kod jest krótszy i zawiera mniej błędów

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;
- można poznać coś nowego;
- kod jest krótszy i zawiera mniej błędów

Wady

- większy koszt wytwarzania kodu

Komentarz

Zalety

- programiści mają wiedzę o całym projekcie;
- dzielenie się wiedzą;
- można poznać coś nowego;
- kod jest krótszy i zawiera mniej błędów

Wady

- większy koszt wytwarzania kodu
- problem z indywidualistami

Wspólna własność kodu

Para programistów ma prawo pobrać z repozytorium **dowolny** fragment oprogramowania i nad nim pracować.

Wspólna własność kodu

Para programistów ma prawo pobrać z repozytorium **dowolny** fragment oprogramowania i nad nim pracować.

Stosuje się nieblokujące systemy zarządzania wersjami.

Praca na kodem źródłowym

Schemat pracy zespołu

- para programistów pobiera kod jakiegoś modułu;

Praca na kodem źródłowym

Schemat pracy zespołu

- para programistów pobiera kod jakiegoś modułu;
- para pracuje zarówno nad samym kodem jak i nad przypadkami testowymi;

Praca na kodem źródłowym

Schemat pracy zespołu

- para programistów pobiera kod jakiegoś modułu;
- para pracuje zarówno nad samym kodem jak i nad przypadkami testowymi;
- co jakiś czas moduł jest kompilowany i testowany, a następnie zwracany do repozytorium

Praca na kodem źródłowym

Schemat pracy zespołu

- para programistów pobiera kod jakiegoś modułu;
- para pracuje zarówno nad samym kodem jak i nad przypadkami testowymi;
- co jakiś czas moduł jest kompilowany i testowany, a następnie zwracany do repozytorium
- kompilowany i testowany jest cały program

Uwagi

Przy zwracaniu do repozytorium może się zdarzyć konflikt z inną wersją

- konieczne jest scalenie z inną wersją (i być może konsultacje z innym zespołem)

Uwagi

Przy zwracaniu do repozytorium może się zdarzyć konflikt z inną wersją

- konieczne jest scalenie z inną wersją (i być może konsultacje z innym zespołem)
- aby uniknąć długich sesji scalania trzeba często testować i zwracać do repozytorium

Uwagi

Przy zwracaniu do repozytorium może się zdarzyć konflikt z inną wersją

- konieczne jest scalenie z inną wersją (i być może konsultacje z innym zespołem)
- aby uniknąć długich sesji scalania trzeba często testować i zwracać do repozytorium

Uwagi

Przy zwracaniu do repozytorium może się zdarzyć konflikt z inną wersją

- konieczne jest scalenie z inną wersją (i być może konsultacje z innym zespołem)
- aby uniknąć długich sesji scalania trzeba często testować i zwracać do repozytorium

W repozytorium zawsze powinien być kod, który się kompiluje.

Plan wykładu

- 1 Styl pracy w programowaniu ekstremalnym
 - Organizacja miejsca pracy
 - Praca w parach
 - Praca nad oprogramowaniem
- 2 Styl programowania w XP
 - Projektowanie
 - Programowanie
 - Metafora

Pytanie

Jeśli są krótkie iteracje, kiedy tworzy się infrastrukturę (warstwy pośrednie, bazy danych, wielodostęp, ORB)?

Zasada 1

Prosta architektura

Przy implementacji korzysta się z najprostszych rozwiązań wystarczających do realizacji zadania

Zasada 1

Prosta architektura

Przy implementacji korzysta się z najprostszych rozwiązań wystarczających do realizacji zadania

Przykłady

- zwykłe pliki zamiast bazy danych;
- gniazda sieciowe zamiast ORB;
- jeden wątek zamiast wielu.

Zasada 2

Wiadomo, że zaawansowane technologie będą kiedyś potrzebne

Zaawansowane technologie wdraża się gdy:

- zespół jest przekonany, że wdrożenie danej technologii jest niezbędne;
- opóźnianie wdrożenia może być kosztowne.

Powtórzenia kodu źródłowego

Każde zauważone powtórzenie kodu jest natychmiast eliminowane.

Techniki eliminacji kodu

Wzorzec projektowy

Pewien kod, który pozwala rozwiązać różne podobne problemy.

Refaktoryzacja

Ciąg drobnych transformacji poprawiających strukturę programu bez naruszania jego funkcjonalności.

Przykład wzorca

Zadanie

Chcemy zaimplementować sortowanie bąbelkowe tablic typu `int` i `float`, nie używając tzw. generyków.

```
class IntSort {  
    public static int Sort(int [] array) {  
        ...  
        return result;  
    }  
}
```

```
class FloatSort {  
    public static float Sort(float [] array) {  
        ...  
        return result;  
    }  
}
```

Wzorzec *Template Method*

```
abstract class BubbleSorter {
    int DoSort() {
        ...
    }
    abstract void Swap(int indeks):
    abstract bool OutOfOrder(int indeks);
}

class IntSorter : BubbleSorter {
    public int Sort(int[] arr) { ... }
    override void Swap(int indeks) { ... }
    override bool OutOfOrder(int indeks) { ... }
}
```

Przykład refaktoryzacji: wydzielenie funkcji

Sito Eratostenesa

```
void sito() {  
    int [] array = new int[100];  
    for(i = 0; i < 100; i++)  
        array[i] = 0;  
  
    for(d = 2; d < 100; d++) {  
        licz = 0;  
        while(licz < 100) {  
            array[licz] = 0;  
            licz = licz + d;  
        }  
    }  
}
```

Poprawiony kod

Wydzielenie kodu

```
void inicjowanie() { ... }  
void odsiewanie { ... }  
  
void sito() {  
    inicjowanie();  
    odsiewanie();  
}
```


Pojęcie metafory

Syntetyczny opis funkcjonalności czy architektury, posługujący się terminami zrozumiałymi przez klienta.

Przykłady

Rekordy z tabeli PurchaseItems o Id równym \$_SESSION['id']

Przykłady

Rekordy z tabeli PurchaseItems o Id równym \$_SESSION['id']

Zawartość koszyka w sklepie internetowym.

Przykłady

Rekordy z tabeli PurchaseItems o Id równym \$_SESSION['id']

Zawartość koszyka w sklepie internetowym.

Podkatalog zawierający pliki, do których nie było odwołania przez program wyświetlający od co najmniej 10 dni.

Przykłady

Rekordy z tabeli PurchaseItems o Id równym \$_SESSION['id']

Zawartość koszyka w sklepie internetowym.

Podkatalog zawierający pliki, do których nie było odwołania przez program wyświetlający od co najmniej 10 dni.

Archiwum zawierające dawno nieoglądane zdjęcia.

Przykłady

Rekordy z tabeli PurchaseItems o Id równym \$_SESSION['id']

Zawartość koszyka w sklepie internetowym.

Podkatalog zawierający pliki, do których nie było odwołania przez program wyświetlający od co najmniej 10 dni.

Archiwum zawierające dawno nieoglądane zdjęcia.