

# Metodyki zwinne wytwarzania oprogramowania

## Wykład 2

Marcin Młotkowski

12 października 2016

# Plan wykładu

- 1 Programowanie ekstremalne
  - Wstęp
- 2 Planowanie w XP
- 3 Określanie wymagań — opowieści użytkownika
  - Wymagania — wprowadzenie
  - Wymagania w XP

# eXtreme Programming (XP)

## Naczelna zasada

Rezygnacja z dokumentacji na rzecz stałej komunikacji programistów, klientów i menadżerów.

# Co to takiego to XP

Stały kontakt z klientem

# Co to takiego to XP

Stały kontakt z klientem

Minimalna specyfikacja (lub jej brak)

# Co to takiego to XP

Stały kontakt z klientem

Minimalna specyfikacja (lub jej brak)

Wspólna własność kodu

# Kiedy warto stosować

- małe i średnie projekty
- projekty ryzykowne, tj. nie wiadomo co zrobić i jak

# Organizacja zespołu

## Zespół

Zespół tworzą: programiści, kadra menadżerska i klient.



# Organizacja zespołu

## Zespół

Zespół tworzą: programiści, kadra menadżerska i klient.

## kto jest klientem

- ten kto finansuje projekt;
- grupa analityków biznesowych;
- przedstawiciele użytkownika.

# Organizacja zespołu

## Zespół

Zespół tworzą: programiści, kadra menadżerska i **klient**.

## kto jest klientem

- ten kto finansuje projekt;
- grupa analityków biznesowych;
- przedstawiciele użytkownika.

## Zadanie klienta

- klient jest członkiem zespołu;
- klient powinien pracować blisko (najlepiej w tym samym pomieszczeniu) co zespół programistów;
- klient współpracuje w zakresie określenia wymagań dotyczących powstającego systemu.

# Plan wykładu

- 1 Programowanie ekstremalne
  - Wstęp
- 2 Planowanie w XP
- 3 Określanie wymagań — opowieści użytkownika
  - Wymagania — wprowadzenie
  - Wymagania w XP

# Wstęp

## Iteracje

W czasie realizacji projektu należy dostarczać klientowi działające oprogramowanie co dwa tygodnie.

# Dygresja: opowieści użytkownika

Opowieść użytkownika

Nieformalny opis jakiejś funkcjonalności

# Dygresja: opowieści użytkownika

## Opowieść użytkownika

Nieformalny opis jakiejś funkcjonalności

## Cechy opowieści użytkownika

- Każda opowieść użytkownika ma przypisany "koszt"
- Do każdej opowieści są testy akceptacyjne

## Etap 1. iteracji

### Wybór rozmiaru iteracji

- klient wskazuje, które opowieści użytkownika mają być zrealizowane w ramach iteracji
- łączna wartość punktowa opowieści nie może przekroczyć określonego limitu (rozmiaru iteracji)

## Etap 1. iteracji

### Wybór rozmiaru iteracji

- klient wskazuje, które opowieści użytkownika mają być zrealizowane w ramach iteracji
  - łączna wartość punktowa opowieści nie może przekroczyć określonego limitu (rozmiaru iteracji)
- 
- nie wolno zmieniać listy opowieści podczas iteracji
  - nie wolno zmieniać samych opowieści



## Etap 2. iteracji

### Podział na zadania

- opowieści użytkownika są dzielone na zadania (z klientem)
- kolejność implementacji opowieści i zadań jest wewnętrznym ustaleniem zespołu
- programiści mogą sami wybierać, jakie chcą wykonywać zadania

### Zadanie

Jest to mechanizm, który programista może zaprogramować w ciągu 4–16 godzin pracy.

# Komentarz

- iteracja kończy się ustalonego dnia, niezależnie od zakresu wykonanych prac
- częste i krótkie iteracje umożliwiają zdobycie wiedzy o wydajności zespołu
- ważne jest równe tempo kolejnych iteracji
- zaimplementowane opowieści użytkownika są testowane testami akceptacyjnymi

## Połowa iteracji

W połowie iteracji powinna być gotowa połowa opowieści użytkownika. Jeśli nie, to

- następuje ponowny podział zadań
- reneguje się z klientem, które opowieści można usunąć z iteracji

## Połowa iteracji

W połowie iteracji powinna być gotowa połowa opowieści użytkownika. Jeśli nie, to

- następuje ponowny podział zadań
- renegukuje się z klientem, które opowieści można usunąć z iteracji

### Ważne

Należy dążyć do realizacji kompletnych opowieści użytkownika.

## Zakończenie iteracji

Demonstracja bieżącej wersji, ew. dopisanie kolejnych opowieści.

# Wydania

Wydanie to większy kawałek działającego oprogramowania.

# Wydania

Wydanie to większy kawałek działającego oprogramowania.

- wydanie obejmuje wynik prac zespołu zrealizowany w ciągu 2 – 4 miesięcy
- decyzja o tym, które opowieści powinny wejść do wydania należy do klienta
- Plan wydań nie jest sztywny

## Śledzenie postępu

Analiza, ile punktów (liczą się tylko zaakceptowane opowieści!) zrealizowano w trakcie każdej iteracji.



# Wykres wypalania

Wykres, ile zostało do wykonania zadań (w ramach iteracji) lub ile zostało opowieści (w punktach).

# Plan wykładu

- 1 Programowanie ekstremalne
  - Wstęp
- 2 Planowanie w XP
- 3 Określanie wymagań — opowieści użytkownika
  - Wymagania — wprowadzenie
  - Wymagania w XP

# Definicja

## Wymaganie

Udokumentowana możliwość w zakresie funkcjonalności, jakości, normy, którą musi dostarczyć system.

# Rodzaje wymagań

## Wymagania funkcjonalne

określają zachowanie systemu, głównie w zakresie wejścia/wyjścia (np. interfejs użytkownika) lub w zakresie przetwarzania danych

# Rodzaje wymagań

## Wymagania funkcjonalne

określają zachowanie systemu, głównie w zakresie wejścia/wyjścia (np. interfejs użytkownika) lub w zakresie przetwarzania danych

## Wymagania niefunkcjonalne

- efektywność (wydajność)
- niezawodność
- użyteczność
- łatwość modyfikacji

## Rodzaje wymagań, cd

### Wymagania projektowe

- narzucenie konkretnej technologii (.NET czy Java) lub bazy danych;
- wymóg stosowania odpowiednich przepisów (np. ochrona danych osobowych)

# Dokumentowanie wymagań funkcjonalnych

Przypadek użycia

## Przykład: system *Zapisy na zajęcia*

### Krótki opis

Ten przypadek użycia opisuje zapisy na wybrane wykłady oraz modyfikację i usuwanie zapisów.



## Przykład: system *Zapisy na zajęcia*

### Krótki opis

Ten przypadek użycia opisuje zapisy na wybrane wykłady oraz modyfikację i usuwanie zapisów.

### Przepływ zdarzeń

- Student wybiera *zapisy na wykłady*
- student wyszukuje przedmioty z *katalogu przedmiotów*
- student wybiera przedmiot
- system sprawdza, czy
  - 1 student spełnia warunki wstępne
  - 2 czy są wolne miejscajeśli tak, student jest dodawany do listy uczestników zajęć

## Inne elementy wymagań

- słownik pojęć z zakresu analizowanej dziedziny
- spis aktorów

# Obserwacje

Nie jest konieczna wiedza o całości i o wszystkich szczegółach wymagań.

# Obserwacje

Nie jest konieczna wiedza o całości i o wszystkich szczegółach wymagań.

Bo to wymaga długiego czasu.

## Zbieranie wymagań od klienta

### Opowieści (historie) użytkownika

Uzyskana podczas rozmowy niezbyt precyzyjny opis tego jakiejś funkcjonalności.

## Uwagi o opowieściach użytkownika

- nie zawierają szczegółów (w końcu klient jest na miejscu ;)

## Uwagi o opowieściach użytkownika

- nie zawierają szczegółów (w końcu klient jest na miejscu ;)
- każdej opowieści użytkownika przypisuje się szacowaną wartość (na przykład liczbę potrzebnych do implementacji godzin)

## Uwagi o opowieściach użytkownika

- nie zawierają szczegółów (w końcu klient jest na miejscu ;)
- każdej opowieści użytkownika przypisuje się szacowaną wartość (na przykład liczbę potrzebnych do implementacji godzin)
- do opowieści użytkownika przypisuje się automatyczne *testy akceptacyjne*



## Uwagi o opowieściach użytkownika

- nie zawierają szczegółów (w końcu klient jest na miejscu ;)
- każdej opowieści użytkownika przypisuje się szacowaną wartość (na przykład liczbę potrzebnych do implementacji godzin)
- do opowieści użytkownika przypisuje się automatyczne *testy akceptacyjne*
- opowieść użytkownika jest zapisywana na fiszce, jeśli się nie mieści to znaczy że trzeba ją podzielić

# Przykład

## Opowieść użytkownika, system zapisy

Student zapisuje się na zajęcia. System weryfikuje warunki wstępne a także *zajętość* wykładu.

# Szacowanie czasu wykonania

Popularna metoda

Metoda SWAG

Scientific Wild-Assed Guess

# Szacowanie czasu wykonania

Popularna metoda

Metoda SWAG

Scientific Wild-Assed Guess

Polskie tłumaczenie

Zaawansowana sufitologia

## Szacowanie rozmiaru projektu

Przykładowe metryki oprogramowania:

### LOC (LLOC)

Lines of Code (LOC): wiersze kodu źródłowego z wierszami pustymi i komentarzami

Logical LOC (LLOC): liczba instrukcji.

# Szacowanie rozmiaru projektu

Przykładowe metryki oprogramowania:

## LOC (LLOC)

Lines of Code (LOC): wiersze kodu źródłowego z wierszami pustymi i komentarzami

Logical LOC (LLOC): liczba instrukcji.

## Punkty funkcyjne

- interakcje z użytkownikiem;
- udostępniane funkcje;
- zewnętrzne dane wejściowe i wyjściowe;
- pliki używane przez system.

# Szacowanie pracochłonności

Czas wykonania to rozmiar projektu i wydajność.

Szacowanie

- metody algorytmiczne, np. COCOMO
- szacowanie przez ekspertów;
- szacowanie przez analogię.

# Szacowanie pracochłonności

Czas wykonania to rozmiar projektu i wydajność.

Szacowanie

- metody algorytmiczne, np. COCOMO
- szacowanie przez ekspertów;
- szacowanie przez analogię.

*The Mythical Man Month*



# Dzielenie i scalanie opowieści

## obserwacja 1

Szacowany koszt wielkich opowieści jest zwykle zaniżany.

# Dzielenie i scalanie opowieści

## obserwacja 1

Szacowany koszt wielkich opowieści jest zwykle zaniżany.

## obserwacja 1

Szacowany koszt małych opowieści jest zwykle zawyżony.

# Dzielenie opowieści

Jeśli jest kłopot z szacowaniem nieprecyzyjnej opowieści należy ją podzielić.

## Przykład: zapisy

Zamiast ogólnej opowieści *studenci zapisują się na zajęcia*:

- student może się logować
- student może się wylogowywać
- student może się zapisać
- student może się wypisać