

# Metodyki zwinne wytwarzania oprogramowania

## Wykład 12

Marcin Młotkowski

11 stycznia 2017

# Plan wykładu

- 1 eXtreme programming
- 2 Scrum
  - Wprowadzenie
  - Role w Scrum'ie
  - Cykl pracy w Scrum'ie
  - Dokumentacja
- 3 Crystal methods
  - Elementy Crystal
  - Kolory w metodologii Crystal
- 4 XPrince
  - Organizacja zespołu
  - Cykl życia projektu

## Przypomnienie z poprzednich wykładów

- podział na iteracje;
- brak hierarchii;
- opowieści użytkownika;
- programowanie w parach.

# Plan wykładu

- 1 eXtreme programming
- 2 Scrum
  - Wprowadzenie
  - Role w Scrum'ie
  - Cykl pracy w Scrum'ie
  - Dokumentacja
- 3 Crystal methods
  - Elementy Crystal
  - Kolory w metodologii Crystal
- 4 XPrince
  - Organizacja zespołu
  - Cykl życia projektu

# Definicja

Scrum, ang:

*młyn* w grze rugby.

# Definicja

Scrum, ang:

*młyn* w grze rugby.

Twórcy

Ogólne założenia: Hirotaka Takeuchi i Ikujiro Nonaka.

Sformalizowanie: Ken Schwaber i Jeff Southerland.

## Podstawowe informacje

### Główne zadanie

Równoważenie potrzeb klienta i możliwości zespołu.

### Kiedy stosować

Nie do końca są znane wymagania klienta, albo ich ustalenie jest niemożliwe.

## Zespół realizujący projekt (*świnki*)

### Mistrz młyna (scrum master)

- nie jest kierownikiem zespołu;
- odpowiada za tworzenie dobrych wymagań i listy priorytetów;
- wspomaga pracę zespołu;
- kontaktuje się z innymi działami firmy (marketing, zarząd etc)



## Zespół realizujący projekt (*świnki*)

### Mistrz młyna (scrum master)

- nie jest kierownikiem zespołu;
- odpowiada za tworzenie dobrych wymagań i listy priorytetów;
- wspomaga pracę zespołu;
- kontaktuje się z innymi działami firmy (marketing, zarząd etc)

### Członkowie zespołu (ok. 5–9 osób)

Programiści, testerzy, analitycy, graficy i inni. Członkowie zespołu są zaangażowani tylko do tego projektu.

## Zespół realizujący projekt (*świnki*)

### Mistrz młyna (scrum master)

- nie jest kierownikiem zespołu;
- odpowiada za tworzenie dobrych wymagań i listy priorytetów;
- wspomaga pracę zespołu;
- kontaktuje się z innymi działami firmy (marketing, zarząd etc)

### Członkowie zespołu (ok. 5–9 osób)

Programiści, testerzy, analitycy, graficy i inni. Członkowie zespołu są zaangażowani tylko do tego projektu.

### Właściciel produktu

Osoba reprezentująca interesy klienta. Odpowiada za historie użytkownika, nadawanie im priorytetów, a także kontroluje postępy prac.

## Osoby pośrednio zaangażowane w projekt (*kurczaki*)

- kierownictwo
- reprezentanci klienta

## Podstawowe pojęcia

sprint

ustalanej długości (zwykle 15–30 dni) iteracja

# 1. etap

## Planowanie (1 dzień)

- spotkanie z właścicielem produktu celem ustalenia priorytetów nowych i zaległych zadań, oraz listy zadań na najbliższy sprint na podstawie szacunkowego czasu;
- zespół dokładnie planuje pracę na bieżący sprint z podziałem pracy pomiędzy członków zespołu

## 2. etap: realizacja projektu

### Codzienny SCRUM

- każdy może przyjść, ale tylko "świnki" mogą się wypowiadać;
- informacja od członków zespołu o wykonanej pracy, o zadaniach na dzisiaj, oraz o problemach;
- Scrum master na podstawie tych informacji natychmiast reaguje i stara się rozwiązać istniejące problemy.

## 3. etap: prezentacja sprintu

Prezentacja służy weryfikacji efektów pracy zespołu.

## 3. etap: prezentacja sprintu

Prezentacja służy weryfikacji efektów pracy zespołu.

- spotkanie po wykonaniu zadań;
- każdy może przyjść (również spoza zespołu);
- prezentowane są tylko działające funkcjonalności.



## 4. etap: retrospekcja

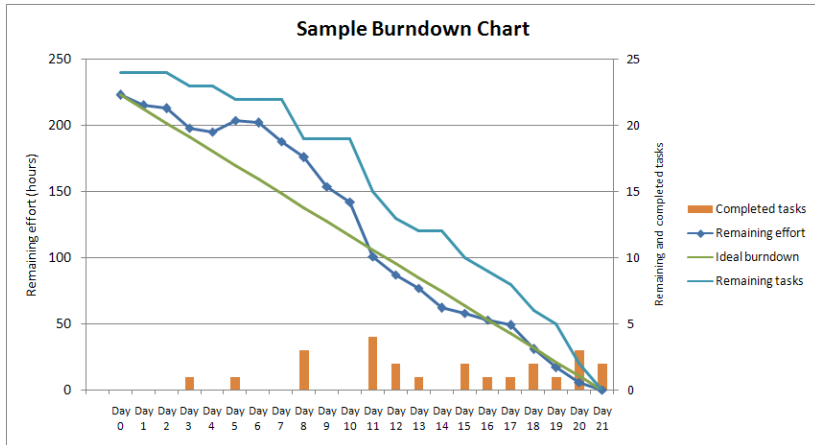
Jest to wewnętrzne spotkanie zespołu, które ma ocenić metodykę pracy, a także efekty.

## Zaległość produktowa (backlog)

### Lista prac dla sprintu

- nazwa zadań;
- priorytet;
- szacowana pracochłonność;
- procent wykonania;
- status;
- osoba wykonująca.

# Wykres wypalania



# Plan wykładu

- 1 eXtreme programming
- 2 Scrum
  - Wprowadzenie
  - Role w Scrum'ie
  - Cykl pracy w Scrum'ie
  - Dokumentacja
- 3 Crystal methods
  - Elementy Crystal
  - Kolory w metodologii Crystal
- 4 XPrince
  - Organizacja zespołu
  - Cykl życia projektu

# Co to jest?

## Metodologie Crystal

Jest to grupa metodologii dostosowanych do projektów o różnych wielkościach.

## Element 1.

### Gra zespołowa

Tworzenie oprogramowania gra zespołowa komunikacji w zespole i pomysłowości. Zespół sam zastanawia się jakie ruchy czy strategie są najefektywniejsze do osiągnięcia celu.

## Element 2.

### Priorytety metodologii

- zakończenie projektu sukcesem;
- wydajność procesu;
- przystosowanie procesu do indywidualnych przyzwyczajęń, nawet kosztem wydajności.

## Element 3.

### Projektowanie własnej metodologii

Przystosowanie sposobu pracy przy uwzględnieniu:

- ludzi i ich umiejętności i przyzwyczajzeń;
- różnych rodzajów projektów czy klientów z różnych krajów;
- adaptacja metodologii pracy do zmieniającego się zespołu, nabywanych doświadczeń czy technologii.



## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;
- szybki kontakt z osobami mogącymi udzielić odpowiedzi na różne pytania;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;
- szybki kontakt z osobami mogącymi udzielić odpowiedzi na różne pytania;
- przyjazna atmosfera, pozwalająca zawczasu postawić konfliktowe zagadnienia;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;
- szybki kontakt z osobami mogącymi udzielić odpowiedzi na różne pytania;
- przyjazna atmosfera, pozwalająca zawczasu postawić konfliktowe zagadnienia;
- organizacja pracy, dająca możliwość długiego nieprzerywanego czasu na pracę nad projektem;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;
- szybki kontakt z osobami mogącymi udzielić odpowiedzi na różne pytania;
- przyjazna atmosfera, pozwalająca zawczasu postawić konfliktowe zagadnienia;
- organizacja pracy, dająca możliwość długiego nieprzerywanego czasu na pracę nad projektem;
- dobra komunikacja z użytkownikami/ekspertami z dziedziny projektu;

## Element 4.

Właściwości projektów odnoszących sukcesy:

- częste dostarczanie użytecznej wersji użytkownikom;
- spotkania poświęcone organizacji zespołu, aby przedyskutować jego słabe i mocne strony i co powinno być zmienione;
- szybki kontakt z osobami mogącymi udzielić odpowiedzi na różne pytania;
- przyjazna atmosfera, pozwalająca zawczasu postawić konfliktowe zagadnienia;
- organizacja pracy, dająca możliwość długiego nieprzerywanego czasu na pracę nad projektem;
- dobra komunikacja z użytkownikami/ekspertami z dziedziny projektu;
- środowisko programistyczne z testami automatycznymi, częstą integracją i wspólnym repozytorium kodu.

## Element 5.

Wybranie tych technik, które w danym zespole i projekcie mogą być użyteczne, np:

- dokładna analiza biznesowa, wymagania, możliwości zespołu, plan zespołu etc;
- *szybkie zwycięstwo*: szybkie uzyskanie działającego szkieletu;
- programowanie w parach;
- TDD
- codzienne spotkania zespołu;
- wykresy postępu prac;
- dynamiczna lista priorytetów;
- ...



## Element 6.

### Układanie własnej metodologii

Używanie tych elementów, które znamy/lubimy/sądzimy że się przydadzą...

## Zasady Crystal Clear

- projektant i 2–7 programistów;
- pracują razem w pokoju lub obok;
- łatwy dostęp do użytkownika;
- dostarczają użyteczny kod co miesiąc lub dwa;
- dokumentacja to: harmonogramy oceny użytkownika, opisane przypadki użycia, szkice ekranów, przypadki podręcznik użytkownika.

# Crystal Yellow

- programiści pracują w sąsiednich pokojach;
- jest pokój wspólny do spotkań zespołu;
- miesięczne iteracje zaczynające się od planowania i kończące się analizą prac;
- spotkania zespołu co kilka dni;
- spotkanie z klientami raz w tygodniu.

# Crystal Orange

- zespół 10–140 osób, czas trwania projektu: 1–2 lata;
- podział na role: sponsor, ekspert biznesowy, projektant, projektant interfejsu, architekt, ...
- podział na mniejsze zespoły;
- więcej dokumentacji, m.in.: dokument wymagań, dokument projektu interfejsu użytkownika, specyfikacje międzyzespołowe;

# Plan wykładu

- 1 eXtreme programming
- 2 Scrum
  - Wprowadzenie
  - Role w Scrum'ie
  - Cykl pracy w Scrum'ie
  - Dokumentacja
- 3 Crystal methods
  - Elementy Crystal
  - Kolory w metodologii Crystal
- 4 XPrince
  - Organizacja zespołu
  - Cykl życia projektu

# XPrince

## XPrince

e**X**treme **P**Rogramming **I**N **C**ontrolled **E**nvironments

## Czy to coś nowego?

XPrince bazuje na trzech innych metodykach: XP, PRINCE2 oraz Rational Unified Process.

## Kto to wymyślił

Jerzy Nawrocki z Instytutu Informatyki Politechniki Poznańskiej

## Motywacje do powstania XPrince

### Słabości XP

intensywna współpraca z klientem

klient często jest nie ma na to czasu.

## Motywacje do powstania XPrince

### Słabości XP

intensywna współpraca z klientem

klient często nie ma na to czasu.

brak dokumentacji papierowej

kłopot ze zmianami w dużych systemach.



## Motywacje do powstania XPrince

### Słabości XP

intensywna współpraca z klientem

klient często nie ma na to czasu.

brak dokumentacji papierowej

kłopot ze zmianami w dużych systemach.

planowanie

krótka perspektywa planowania.

## Naczelne hasło

Równowaga między zwinnością i dyscypliną.

# Zespół w XPrince

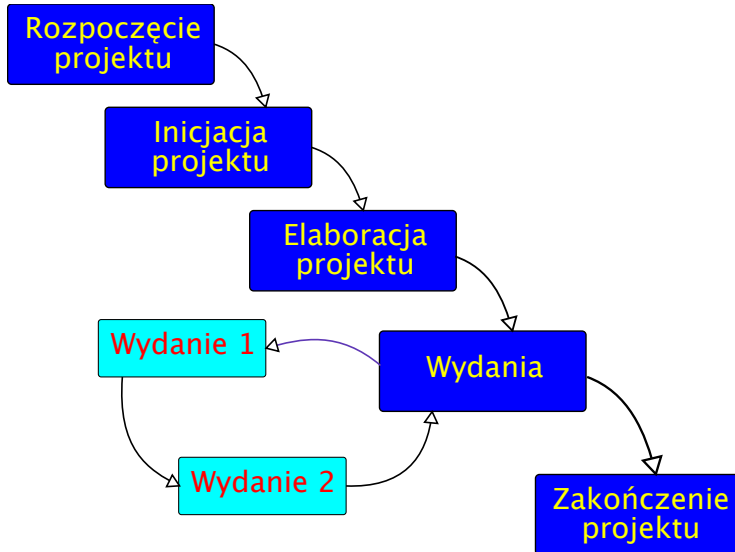
## Zespół zarządzania projektem

- **Dyrektor:** reprezentuje inwestora
- **Główny użytkownik:** kieruje użytkownikami końcowymi, odpowiada za użyteczność programu;
- **Główny dostawca:** reprezentuje wykonawcę

## Role w zespole wykonawczym

**menadżer projektu** odpowiada za środowisko pracy  
**architekt** główny projektant systemu  
**główny analityk** odpowiada za analizę biznesową  
**programiści**

# Schemat cyklu życia projektu



# Rozpoczęcie projektu

Projekt rozpoczyna manager projektu.

## Cele fazy *rozpoczęcie projektu*

### Organizacja zespołu zarządzającego projektem

Powołanie dyrektora, głównego użytkownika, głównego dostawcę.

### Stworzenie wizji systemu

Zawiera zarówno wstępną wizję techniczną jak i biznesową.

### Plan inicjacji systemu

Zaplanowanie działań potrzebnych do zainicjowania działań.

## Inicjacja projektu: cele fazy

### Zrozumienie co należy zrobić

- wstępny model działania oparty na przypadkach użycia;
- lista kluczowych funkcjonalności;
- kryteria oceny jakości.

### Przedstawienie propozycji architektury

- to wysokopoziomowy opis służący dalszemu planowaniu projektu;
- lista potrzebnych narzędzi.

# Planowanie projektu

- plan wydań z przydzielonymi funkcjonalnościami;
- długość wydań



## Faza elaboracji

- Architekt: opracowanie mechanizmów architektonicznych (w tym rozpoznanie ryzyk za pomocą eksperymentów);
- Analityk: dopracowuje wymagania;
- Manager projektu: dopracowuje plan projektu.

## Fazy wydania

- architekt i programiści implementują przypadki testowe;
- analityk: tworzy testy akceptacyjne i ocenia kod pod kątem wymagań.

## Przyrosty w ramach wydania

### Przyrost

Wewnętrzny (w ramach wydania) punkt kontrolny.

## Przyrosty w ramach wydania

### Przyrost

Wewnętrzny (w ramach wydania) punkt kontrolny.

### Koniec wydania

Nowa wersja jest przekazywana użytkownikowi końcowemu.

## Przyrosty w ramach wydania

### Przyrost

Wewnętrzny (w ramach wydania) punkt kontrolny.

### Koniec wydania

Nowa wersja jest przekazywana użytkownikowi końcowemu.

### Długość wydania

Wydania są podobnej długości.

## Faza zamknięcie projektu

- sprawdzenie kompletności projektu (wykonanie i przekazanie);
- sporządzenie dokumentacji technicznej;
- zaplanowanie oceny systemu "w działaniu".