

# Metodyki zwinne wytwarzania oprogramowania

## Wykład 14

Marcin Młotkowski

18 stycznia 2017

# Plan wykładu

- 1 Szacowanie rozmiaru projektów
  - Rozmiar projektu
  - Szacowanie kosztów projektu
  
- 2 Zarządzanie ryzykiem

# Koszty projektu

Składowe kosztów projektu informatycznego:

- projektowanie i implementacja oprogramowania;
- opracowanie dokumentacji (np. dokumentacja przedwdrożeniowa);
- zakup narzędzi (sprzęt, licencje, szkolenia);
- podręcznik instalacji/administratora/użytkownika;
- ...

# Jak oszacować rozmiar projektu (metryki)

- sufit;
- LOC: lines of code;
- punkty funkcyjne;
- ocena ekspertów;
- analogia z innymi;
- metody algorytmiczne
-

# Zła wiadomość

## Prawo Parkinsona

Prawo Parkinsona mówi, że praca rozciąga się na cały przeznaczony na to czas. Jeśli więc 5-osobowy zespół ma wykonać jakiś projekt w ciągu 12 miesięcy, to projekt szacuje się na 60 osobomięsi.

# Metoda LOC

## LOC: Lines of Code

Liczba wierszy programu. Oblicza się fizyczne wiersze kodu źródłowego wraz z komentarzami i wierszami pustymi.

## LLOC - logical LOC

Liczba instrukcji.

# Zastosowanie

Można obliczać średnią wydajność programisty.

# Zastosowanie

Można obliczać średnią wydajność programisty.

## Ograniczenia metody

- zależność od języka programowania;
- zależność od charakteru projektu, np. jak obliczać modyfikacje, gdzie powstaje mało nowego kodu.



# Metoda punktów funkcyjnych

Szacowanie oparte tylko na wymaganej funkcjonalności, bez implementacji.

## Punkty funkcyjne

- interakcje z użytkownikiem;
- udostępniane metody i funkcje zewnętrznym aplikacjom;
- pliki używane przez system.

Każdemu punktowi przypisuje się odpowiednią wagę, a następnie sumuje się. Wadą tej metody jest jednak subiektywna ocena wag przypisywanych poszczególnym punktom.

## Zła wiadomość

Nie ma dobrej metody szacowania kosztów. Wszystkie z nich dają jedynie przybliżoną wartość. Co więcej, każda z metod da inną wartość. W związku z tym wyników takiego oszacowania należy traktować jedynie jako wartość pomocniczą.

# Metody "niealgorytmiczne"

- ocena ekspertów: prosi się kilku ekspertów o niezależną ocenę. Następnie omawiane są istotne różnice i następuje ponowna ocena, aż do uzyskania podobnych wyników;
- szacowanie przez analogię: na podstawie wcześniejszego, podobnego projektu.

# Metoda algorytmiczna COCOMO II

## COCOMO II (COConstructive COst MOdel)

### punkty obiektowe (NOP)

- liczba różnych ekranów, przy czym proste ekrany to 1 punkt obiektowy, średnio złożone to 2 punkty, a najbardziej złożone to 3 punkty;
- liczba różnych raportów, gdzie prosty raport to 2 punkty obiektowe, średni: 5 punktów, a najbardziej skomplikowane to 8 punktów;
- liczba modułów, gdzie każdy moduł to 10 punktów obiektowych.

## COCOMO II: poziom wczesnego prototypowania

Na początku projektu przy nieustalonych wymaganiach oblicza się pracę liczoną w osobomiesiącach:

$$PM = (NOP * (1 - \%reuse/100))/PROD$$

gdzie

$\%reuse$  szacowany wskaźnik procentowy ponownego użycia

PROD wskaźnik produktywności programisty [4, 50].

## COCOMO II: poziom wczesnego projektowania

Znane są już wymagania oraz gdy jest już jakiś wstępny projekt.

$$PM = 2,5 \times \text{Wielkość}^B \times M + PM_m$$

gdzie **B** to współczynnik między 1,1 a 1,24 zależny od trudności projektu i sprawności zespołu

**M** zależy m.in. od niezawodności i złożoności produktu, wymagania wielokrotnego użycia czy harmonogramu. **M** jest z przedziału [1, 6]

$PM_m$  współczynnik związany z automatycznym generowaniem kodu

**Wielkość** to oszacowanie punktów funkcyjnych.

## COCOMO II: poziom postarchitektoniczny

Znana jest architektura systemu. Wzór jest ten sam, wyliczenie współczynnika  $M$  zależy od 17 atrybutów.

# Plan wykładu

- 1 Szacowanie rozmiaru projektów
  - Rozmiar projektu
  - Szacowanie kosztów projektu
  
- 2 Zarządzanie ryzykiem



# Pojęcie

## Ryzyko

Cechy ryzyka:

- niepewność wystąpienia;
- strata, jaką się ponosi przy wystąpieniu.

# Motywacje

Przykłady zagrożeń w projektach:

- duża absencja (okres jesienno–zimowy) członków zespołu czy klienta;
- opóźnienie w dostarczeniu niezbędnych narzędzi;
- brak dobrej komunikacji;
- niedoszacowanie projektu.

# Dlaczego zarządzamy ryzykiem

Powody:

- (prawie) zawsze coś się zdarzy;<sup>1</sup>
- jeśli się zdarzy, to istotnie wpłynie na koszt/terminy projektu;
- jeśli coś się zdarzy, to lepiej to przewidzieć.

---

<sup>1</sup>prawo Murphy'ego

# Klasyfikacja

## Rodzaje zagrożeń

- *zagrożenia projektowe*, które mogą skutkować przekroczeniem terminów lub budżetu;
- *zagrożenia techniczne* zagrażają jakości i terminowości dostarczenia produktu;
- *zagrożenia ekonomiczne* mają wpływ na sukces ekonomiczny produktu (np. produkt którego nikt nie potrzebuje, brak zainteresowania w kontynuowaniu projektu, zmniejszenie budżetu).

## Inny podział:

- zagrożenia znane, które można przewidzieć na podstawie analizy przedsięwzięcia;
- zagrożenia przewidywalne, które przewidujemy na podstawie doświadczenia z poprzednich projektów,
- zagrożenia nieprzewidywalne.

# Jak zarządzać

Metoda Indiany Jonesa

*nie martw się, coś wymyślę!*

# Jak zarządzać

## Metoda Indiany Jonesa

*nie martw się, coś wymyślę!*

To jest zarządzanie kryzysowe.

# Systematyczne podejście

## Analiza ryzyka:

- identyfikacja różnych zagrożeń;
- określenie typu zagrożenia (kwalifikacje zespołu, narzędzia i środowisko etc);
- szacunkowe prawdopodobieństwo wystąpienia;
- skutki, jakie może wywołać wystąpienie zagrożenia;
- plan zapobiegania lub minimalizowania zagrożeń.



## Przykład tabeli zagrożeń

Opis	typ	prawd.	skutki (1-10)	zapobieganie
awaria serwera z plikami	ŚiN	małe	9	regularne kopie, serwer zapasowy, ćwiczenia z szybkiego odtwarzania środowiska
brak osób z odpowiednimi kwalifikacjami	ZE	duże	6	wyszukanie osoby z odpowiednimi kwalifikacjami, wyśłanie na szkolenie
rotacja zespołu	ZE	średnie	4	zwiększenie czasu potrzebnego na realizację (np. w czasie urlopów)

## Co dalej

### Analiza ryzyka:

- analizuje się tylko zagrożenia o dużym prawdopodobieństwie bądź dużych skutkach. Te ryzyka powinny być stale monitorowane podczas projektu, np. plany urlopowe pracowników;
- określa się *poziom dopuszczalnego ryzyka*, tj. kiedy trzeba podjąć akcję (np. zamknięcie projektu po przekroczeniu budżetu);
- przygotowanie planów awaryjnych;
- można pewnych ryzyk nie analizować (np. klęski żywiołowe).

# Znane ryzyka w projektach informatycznych

- popieranie projektu przez kierownictwo firmy;
- entuzjazm użytkowników do pomysłu stworzenia nowego oprogramowania;
- rozumienie wymagań stawianych produktowi przez zespół i klientów;
- udział klientów w tworzeniu specyfikacji;
- realistyczność oczekiwań klientów;
- niezmiennosc zakresu działania produktu;
- dostateczne umiejętności zespołu;
- doświadczenie i wielkość zespołu;
- motywacja zespołu i klienta.