

# **Zarządzanie Projektami Informatycznymi**

dr hab. inż. Krzysztof Pieńkosz

e-mail: [pienkosz@wit.edu.pl](mailto:pienkosz@wit.edu.pl)

# Zarządzanie Projektami Informatycznymi

## Literatura uzupełniająca

- Wysocki R.K., McGary R.: *Efektywne zarządzanie projektami*, HELION, 2005.
- Szyjewski Z.: *Metodyki zarządzania projektami informatycznymi*, PLACET, 2004.
- Flasiński M.: *Zarządzanie projektami informatycznymi*, PWN, 2006.
- Philips J.: *Zarządzanie projektami IT*, HELION, 2011.
- Stellman A., Greene J.: *Agile. Przewodnik po zwinnych metodykach programowania*, HELION, 2015.
- Rubin K.S.: *Scrum, Praktyczny przewodnik po najpopularniejszej metodyce Agile*, HELION, 2014.
- Chrapko M.: *Scrum, O zwinnym zarządzaniu projektami*, HELION, 2013.
- Shore J., Warden S.: *Agile Development, Filozofia programowania zwinnego*, HELION, 2008.

# **Procesy inżynierii oprogramowania w projekcie informatycznym**

- specyfikacja wymagań systemu
- analiza
- projektowanie
- implementacja
- integracja i testowanie oprogramowania
- integracja i testowanie systemu
- instalacja
- pielęgnacja (konserwacja) systemu i oprogramowania

## **Procesy zarządzania w projekcie informatycznym**

- zarządzanie wymaganiami
- organizowanie zespołów projektowych
- planowanie projektu
- zarządzanie ryzykiem
- kontrola postępu prac i budżetu
- zarządzanie jakością
- zarządzanie zespołem

# **Zarządzanie Projektami Informatycznymi**

## **Główny cel:**

### **realizacja projektu**

- na czas
- w budżecie
- odpowiedniej jakości
- spełniającego potrzeby klienta

## **Kluczowe parametry projektu**

- zakres
- czas
- koszt
- jakość

# Przyczyny niepowodzeń projektów informatycznych

- wynikające ze sposobu zarządzania
  - zbyt mało kompetencji i doświadczenia (zbyt ambitne cele)
  - specyfikacje są niekompletne lub niejednoznaczne
  - za mały udział użytkowników – *“stworzenie użytecznego produktu informatycznego bez udziału użytkowników jest niemożliwe”*
  - zła komunikacja
  - za mało planowania
  - złe oszacowania
  - nieobiektywna ocena postępu prac
  - brak jasnego określenia zakresu odpowiedzialności
  - nieskończone ulepszanie
  - mało uwagi poświęcone problemom jakości
  - słaba dokumentacja

## **Przyczyny niepowodzeń projektów informatycznych c.d.**

- wynikające z natury projektów informatycznych
  - duże zróżnicowanie – projekty mają charakter jednostkowy
  - interdyscyplinarność – konieczność komunikacji i współpracy ekspertów z różnych dziedzin
  - zmienność technologii – praktycznie każdy nowy projekt w innej technologii (często słabo rozpoznanej)
  - niematerialny charakter oprogramowania – trudności w szacowaniu kosztów i produktywności oraz w kontrolowaniu
  - zwykle brak znajomości dziedziny problemu przez decydentów (niejasność celów)
  - konieczność współdziałania z innymi systemami i technologiami (problemy integracji)
  - duża zmienność wymagań i uwarunkowań w trakcie realizacji projektów

## **Ogólne zasady skutecznego działania**

- ustal wyraźny cel
- przeanalizuj wszystkie kierunki działań i środki, za pomocą których można osiągnąć założony cel
- ułóż **plan** działań zmierzających do celu, przy zastosowaniu najlepszych w danych warunkach środków
- wykonaj założony plan (dokonując ewentualnych korekt)

skontroluj osiągnięte wyniki i porównaj z założonym celem – **wyciągnij wnioski** na przyszłość



## **Kluczowe czynniki powodzenia**

- wspólna wizja produktu – atmosfera zaangażowania i współodpowiedzialności za projekt
- doświadczona i kompetentna kadra
- praca zespołowa – w szczególności **współpraca z użytkownikiem**
- planowanie działań
- myślenie przyszłościowe – zapobieganie zagrożeniom, nie zaś tylko usuwanie skutków (zarządzanie zagrożeniami)
- dobra komunikacja i przepływ informacji
- kontrola zmian
- regularne przeglądy projektu

## **Standardy – kompendium wiedzy**

- ogólne wytyczne i wskazówki odnośnie sposobów zarządzania projektami
- opis powszechnie uznanych praktyk znajdujących zastosowanie w zarządzaniu projektami
- zbiór wiedzy i technik zarządzania projektami
- podstawa do tworzenia metodyk dopasowanych do konkretnych zastosowań

Przykład: PMBoK Guide

## **Metodyki - sformalizowane metody zarządzania projektem (metodyki)**

- ustrukturalizowane zarządzanie projektem
- zdefiniowany proces postępowania
- charakter nakazowy
- stosowanie ściśle określonych procedur
- zdefiniowany proces postępowania
- stosowanie ustalonych standardów pracy
- dokumentowanie działań

## **Korzyści ze stosowania metodyk**

- większa “przewidywalność” projektów
- większe możliwości kontroli
- lepsze ukierunkowanie projektów na osiągnięcie właściwych celów i korzyści
- ograniczanie (minimalizacja) ryzyka
- koordynacja działań i lepsze wykorzystanie zasobów
- jednolita praktyka realizacji
- uniezależnienie realizacji projektu od wykonawców (fluktuacji wykonawców)

## **Niedogodności**

- pracochłonność i koszty procesów zarządzania
- konieczność prowadzenia dokumentacji
- usztywnienie sposobu realizacji projektu
- spowolnienie realizacji projektu (w skali krótkiego horyzontu czasu)

# Podejścia do zarządzania projektami

## Tradycyjne

- na początku projektu tworzona jest pełna specyfikacja wymagań
- projekty są realizowane według planu tworzonego przed przystąpieniem do jego realizacji
- cel projektu jest osiągany poprzez skrupulatne wykonywanie planu
- ogromna rola dokumentacji w projekcie
- mała podatność na zmiany wymagań

## Zwinne (lekkie, adaptacyjne, *ang. agile*)

- zorientowane na projekty innowacyjne
- mają charakter adaptacyjny – realizacja iteracyjna (przyrosty funkcjonalności)
- pod koniec każdego etapu są wyciągane wnioski i definiowane są cele następnej iteracji
- samoorganizacja i samozarządzanie zespołów

# **Przykłady podejść do zarządzania projektami IT**

## **Standardy i metodyki tradycyjne**

- PRINCE2 (PRojects IN Controlled Environments)
- PMBOK (Project Management Body Of Knowledge)  
PMI (Project Management Institute)
- RUP (Rational Unified Process)

## **Podejścia zwinne**

- SCRUM
- eXtreme Programming (XP)

## **Metodyki dużych firm wykonawczych i konsultingowych (ORACLE, IBM, HP, NCR, SIEMENS)**

- oparte na znanych metodykach, dostosowane do konkretnych potrzeb firmy i specyfiki analizowanych projektów
- rozwinięte o dodatkowe standardy firmowe (metodyki są na poziomie ogólnym)

**Najważniejsze, podstawowe zasady zapewniające sukces są te same**

## **Cele standardów i metodyk**

- usprawnienie organizacji i prowadzenia projektów
- stosowanie jednolitej praktyki realizacji projektów
- uzyskanie większej “przewidywalność” projektów
- lepsze ukierunkowanie projektów na osiągnięcie właściwych celów i korzyści
- ograniczanie (minimalizacja) ryzyka
- lepsza koordynacja działań i lepsze wykorzystanie zasobów
- uniezależnienie realizacji projektu od wykonawców (fluktuacji wykonawców)

# **PMBOK Guide**

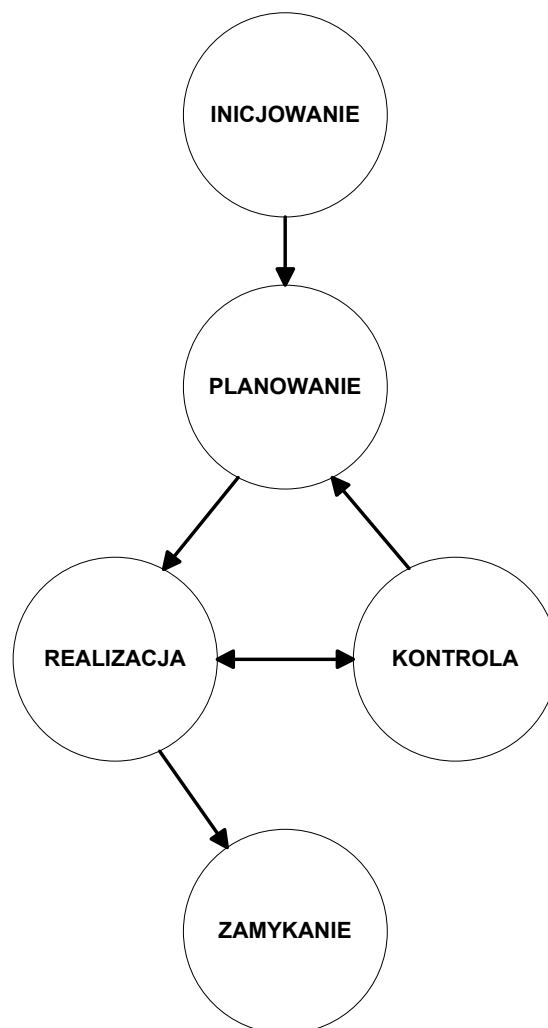
## **(Project Management Body Of Knowledge)**

- opracowywany przez PMI (Project Management Institute)
- kompendium wiedzy (zbiór standardów i rozwiązań)
- zbiór wiedzy i technik zarządzania projektami
- opis „dobrych praktyk”
- ma charakter nienakazowy (można stosować wybiórczo)
- podstawa do zbudowania dopasowanej metodyki do konkretnego projektu
- nastawiony raczej na Kierownika Projektów
- upowszechniony niemal na całym świecie



# PMBOK Guide – grupy procesów zarządzania

- Procesy inicjowania
- Procesy planowania
- Procesy realizacji
- Procesy kontroli
- Procesy zamykania (zakończenia)



# **PMBOK Guide – procesy**

## **Procesy inicjowania**

- Opracowanie karty projektu
- Opracowanie deklaracji zakresu projektu

## **Procesy planowania**

- Opracowanie planu zarządzania projektem
- Planowanie zarządzania zakresem projektu
- Definiowanie zakresu projektu
- Opracowanie struktury podziału prac (WBS)
- Zdefiniowanie zadań
- Określenie relacji między zadaniami
- Szacowanie zasobów dla zadań
- Szacowanie czasu trwania zadań
- Opracowanie harmonogramu
- Szacowanie kosztów
- Opracowanie planu kosztów kosztów
- Planowanie jakości
- Planowanie zasobów ludzkich
- Planowanie komunikacji
- Planowanie zarządzania ryzykiem
- Identyfikacja ryzyka
- Jakościowa analiza ryzyka
- Ilościowa analiza ryzyka
- Planowanie reakcji na ryzyko
- Planowanie zaopatrzenia
- Planowanie kontraktów

## **PMBOK Guide – procesy c.d.**

### **Procesy realizacji**

- Kierowanie i zarządzanie realizacją projektu
- Zapewnienie jakości
- Budowanie zespołu
- Rozwój zespołu
- Dystrybucja informacji
- Gromadzenie ofert od sprzedawców
- Wybór sprzedawców

### **Procesy kontroli**

- Monitorowanie i nadzór nad pracami projektu
- Zintegrowane zarządzanie zmianami
- Weryfikacja zakresu
- Sterowanie zakresem
- Nadzór nad harmonogramem
- Nadzór nad kosztami
- Kontrola jakości
- Zarządzanie zespołem
- Raportowanie postępu prac
- Zarządzanie interesariuszami
- Monitorowanie i nadzór nad ryzykiem
- Administrowanie kontraktem

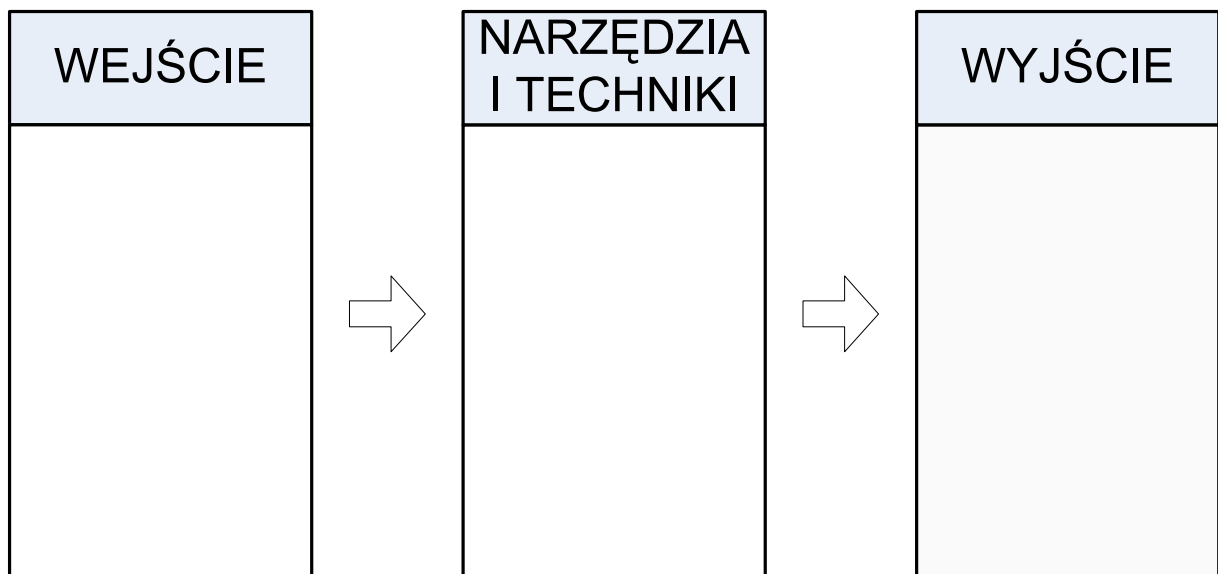
### **Procesy zamykania**

- Zamknięcie projektu
- Zamknięcie kontraktu

## **PMBOK Guide – obszary wiedzy**

- Zarządzanie integralnością projektu – ogólny plan zarządzania projektem, będący połączeniem pozostałych
- Zarządzanie zakresem
- Zarządzanie czasem
- Zarządzanie kosztami
- Zarządzanie jakością
- Zarządzanie zasobami ludzkimi
- Zarządzanie komunikacją
- Zarządzanie ryzykiem
- Zarządzanie zamówieniami

## **PMBOK Guide – opis sposobu realizacji procesów**



# **PMBOK Guide – podstawowe dokumenty**

## **Karta projektu**

- oficjalna nazwa projektu
- dane sponsora
- dane kierownika projektu
- cel projektu
- uzasadnienie biznesowe projektu – powody dla których projekt powinien być zrealizowany
- główne produkty
- kluczowe terminy
- budżet
- informacje dotyczące zasobów niezbędnych do realizacji projektu
- ogólne założenia i ograniczenia
- ogólne ryzyko

## **Karta formalnie definiuje projekt**

- tworzona jest przede wszystkim po to, by przekonać kierownictwo do zaakceptowania projektu.
- jest także podstawą do rozwinięcia szczegółowego planu projektu

## **PMBOK Guide – podstawowe dokumenty**

### **Deklaracja zakresu projektu**

- opis zakresu projektu
- wyłączenia z projektu – co jest poza zakresem projektu
- kryteria akceptacji
- produkty częściowe projektu
- ograniczenia projektu – do czego się trzeba dostosować
- założenia projektu – warunki niezbędne do pomyślnego zrealizowania projektu

Deklaracja zakresu definiuje wszystkie produkty projektu

# **PMBOK Guide – podstawowe dokumenty**

## **Plan zarządzania projektem**

- plan zarządzania zakresem
- harmonogram
- plan kosztów
- plan jakości
- plan zarządzania zespołem
- plan komunikacji
- plan zarządzania ryzykiem
- plan zarządzania zamówieniami

Plan zarządzania projektem określa sposób realizacji projektu



# **PRINCE2**

## **(Projects IN Controlled Environments)**

### **Obszary PRINCE2**

- Zasady (pryncypia) – nakazy przewodnie
- Tematy – zagadnienia uwzględniane przy zarządzaniu projektem
- Procesy – działania podejmowane w trakcie realizacji projektu
- Środowisko projektu – adaptacja metodyki do warunków projektu

## **Zasady PRINCE2**

- projekt ma ciągle aktualne uzasadnienie biznesowe
- korzystanie z doświadczeń - opieranie się na doświadczeniach, gromadzenie doświadczeń, wyciąganie wniosków na przyszłość
- zdefiniowane role i obowiązki – określona struktura organizacyjna
- zarządzanie etapowe – projekt jest planowany, monitorowany i kontrolowany etap po etapie
- zarządzanie z wykorzystaniem tolerancji
- koncentracja na produktach – formalny opis produktów
- dostosowanie do warunków projektu

## **Tematy PRINCE2**

- Uzasadnienie Biznesowe (dlaczego ?)
- Organizacja (kto ?)
- Jakość (co osiągniemy ?)
- Plany (jak ? za ile ? kiedy ?)
- Ryzyko (co, jeśli ?)
- Zmiany (jaki wpływ ?)
- Postępy (gdzie jesteśmy ? dokąd zmierzamy ?)

## **Procesy PRINCE2**

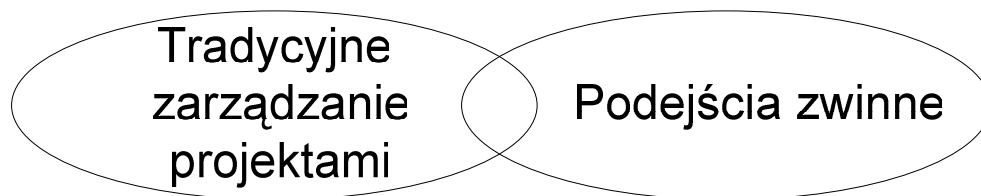
- Przed projektem – przygotowanie organizacyjne projektu, decyzja o uruchomieniu projektu
- Etap inicjowania – planowanie
- Kolejne etapy realizacji – sterowanie etapem, zarządzanie dostarczaniem produktów, zarządzanie końcem etapu
- Finalny etap realizacji – zamykanie projektu

## **Podejścia zwinne (lekkie, *ang. agile* )**

Koncepcja zaproponowana w 2001 roku w **Agile Manifesto**

- zbiór wartości, zasad i praktyk
- koncentracja na czynnościach związanych z jak najszybszym dostarczaniem nowej funkcjonalności (zamiast na czynnościach wymaganych przez formalne procedury)
- realizacja iteracyjna (przyrosty funkcjonalności)
- mają charakter adaptacyjny
- otwartość na zmiany
- samoorganizacja i samozarządzanie zespołów
- wymaganie samodyscypliny, egalitaryzmu
- zorientowane na projekty innowacyjne

# **Podejścia tradycyjne i zwinne**



---

**Cel i rozwiązanie  
jasno określone**

**Cel i rozwiązanie  
niejasne**

## **Filozofia zwinnego zarządzania projektami (wartości)**

- Działające produkty ważniejsze niż kompleksowa dokumentacja
- Ludzie i ich interakcje ważniejsze niż procedury i narzędzia
- Współpraca z klientem ważniejsza niż negocjowanie kontraktu
- Reagowanie na zmiany ważniejsze niż trzymanie się planu

# Przywiązywanie wagi do działających produktów

- przejście od dostarczania dokumentacji do dostarczania iteracyjnie kolejnych, coraz pełniejszych wersji rzeczywistego produktu
- koncentracja na czynnościach związanych z dostarczaniem nowych funkcjonalności
  - wytwarzanie iteracyjne
  - oparte na dostarczaniu elementów funkcjonalności
  - zamknięte w określonych, krótkich okresach czasu (*timeboxach*)
  - inkrementalnie – budowanie produktów częściowych
- długoterminowe plany są płynne, a dokładnie określa się plany na bardzo krótki okres czasu – plan jednej iteracji



## **Koncepcja samoorganizacji, samorządzenia, samodyscypliny i egalitaryzmu**

- w zespole samorganizującym się poszczególne osoby podejmują odpowiedzialność za zarządzanie własnym obciążeniem, uczestniczą w zespołowym podejmowaniu decyzji
- członkowie zespołu mają stosunkowo wolną rękę w kwestii sposobu dostarczania wyniku, ale są zdyscyplinowani wewnętrznie w kwestii odpowiedzialności za wyniki i pracę w elastycznej strukturze
- atmosfera egalitaryzm - szacunku dla jednostki, jej kompetencji i potrzeb

## **Relacje z klientem**

- zwinne zarządzanie projektem prowadzi do zupełnie nowych relacji z klientem i nowych zasad funkcjonowania zespołu
- orientacja na klienta – w centrum zainteresowania projektu jest klient i jego potrzeby
- współudział klienta – zespół klienta i zespół wykonawców tworzą związek partnerski, w którym każda ze stron spełnia określone role, bierze na siebie obowiązki i odpowiedzialność (podobne relacje z głównymi dostawcami)
- w pracach nad nowym produktem, które są z natury nerwowe, wielowątkowe, niejasne i niepewne relacje między klientem a wytwórcą muszą być oparte na współpracy

## **Adaptacyjność**

- zmiana w sposobie wytwarzaniu produktów od wytwarzania opartego na planowaniu długookresowym i specyfikacjach do wytwarzania adaptacyjnego
- praca w trybie adaptacji i ewolucji
- zakres projektu może się zmieniać w kolejnych etapach realizacji projektu

# **Zasady zwinnego zarządzania projektami**

1. Priorytetem jest osiągnięcie satysfakcji klienta poprzez szybkie i ciągłe dostarczanie wartościowego oprogramowania
2. Otwartość na zmiany nawet w późnych etapach prac
3. Działające oprogramowanie jest dostarczane okresowo (raczej tygodniowo niż miesięcznie)
4. Bezpośredni kontakt jako najlepsza forma komunikacji w zespole i poza nim
5. Odbiorcy i programiści muszą codziennie wspólnie pracować nad projektem
6. Należy uwierzyć, że zmotywowane osoby, którym zapewni się potrzebne środowisko i wsparcie, wykonają swoje zadania

## **Zasady zwinnego zarządzania projektami**

7. Podstawową miarą postępu prac jest działające oprogramowanie
8. Sponsorzy, programiści i użytkownicy powinni utrzymywać stałe tempo pracy
9. Ciągła troska o techniczną doskonałość oraz dobry projekt
10. Prostota rozumiana jako sztuka minimalizowania zbędnej pracy
11. Samoorganizujące się zespoły
12. Zespół regularnie zastanawia się jak zwiększyć swoją efektywność, a następnie odpowiednio usprawnia i dostosowuje swoje działania

## **Procesy zwinnego zarządzania projektem**

- **Rozpoczęcie**
  - **Spekulacja** – określenie kierunku działania
  - **Eksploracja** – realizacja przyrostu funkcjonalności
  - **Przegląd, ocena i wnioski** – ewentualna modyfikacja wizji

# **Zarządzanie Projektami Informatycznymi**

---

## **PODEJŚCIA ZWINNE**

# Scrum



# Scrum – zespół i role

---

## ❑ Właściciel Produktu

- jest centralnym punktem zarządzania – jest odpowiedzialny za to, co będzie wytwarzane produkowane i w jakiej kolejności
- pośrednik między zespołem a klientem (użytkownikiem)
- jego zadaniem jest zrozumienie potrzeb i priorytetów użytkowników oraz informowanie na bieżąco zespołu o tym, czego najbardziej potrzebują od oprogramowania i w jakiej kolejności
- odpowiada za wymagania – zna wizję projektu, zarządza rejestrem produktu (listą wymagań)
- ustala priorytety
- określa kryteria akceptacji
- odbiera pracę w sprintach
- pilnuje aby powstało właściwe rozwiązanie – jest odpowiedzialny za całkowite powodzenie wytwarzanego lub rozwijanego rozwiązania
- chcąc mieć pewność, że zespół buduje wyłącznie to, na czym mu zależy, aktywnie współpracuje z zespołem i ScrumMasterem – musi być dostępny i szybko odpowiadać na wszelkie pojawiające się pytania
- może mieć asystenta

# Scrum – zespół i role

---

## ❑ Właściciel Produktu – główne obowiązki

- współpraca z przedstawicielami klienta – reprezentowanie ich interesów („głos klienta”)
- współpraca na co dzień z zespołem – zgłaszanie wytycznych, udzielanie odpowiedzi i wsparcia
- dbanie o podejmowanie właściwych ekonomicznie decyzji
- pielęgnacja rejestru produktu: tworzenie i uszczegóławianie elementów rejestru, nadawanie im ocen oraz priorytetów
- udział w planowaniu – współpraca z zespołem
- definiowanie kryteriów akceptacji i weryfikowanie czy zostały spełnione

# Scrum – zespół i role

---

## ❑ Właściciel Produktu – cechy i umiejętności

- posiada ekspercką wiedzę z zakresu biznesu i domeny produktu
- umiejętności interpersonalne: dobre relacje z interesariuszami, jest dobrym negocjatorem (buduje konsensus) , umiejętności komunikowania się i przekazywania informacji, potrafi skutecznie motywować
- zdolność podejmowania trudnych decyzji (wymagających często kompromisów), jest zdecydowany, potrafi zachować równowagę pomiędzy potrzebami biznesowymi i możliwościami inżynierskimi
- odpowiedzialność – podejmowanie odpowiedzialności za dostarczenie dobrego produktu
- pełne zaangażowanie i stała dostępność

# Scrum – zespół i role

---

## ☐ Właściciel produktu

## ☐ **ScrumMaster** (Mistrz Młyna)

- pomaga zespołowi w zrozumieniu i przestrzeganiu wartości, zasad i praktyk Scruma (zmianie mentalności)
- nie jest kierownikiem projektu! (nie wyznacza bezpośrednio szczegółowych zadań, nie ma władzy nad zespołem)
- wspomaga zespół i Właściciela Produktu jako doradca (trener metod zwinnych) poprzez zadawanie pytań, udzielanie wskazówek, rad itp.
- „wielcy Mistrzowie Młyna niemal nigdy nie odpowiadają na pytanie w sposób bezpośredni, zamiast tego odpowiadają zawsze kolejnym pytaniem”
- pomaga w sprawach organizacyjnych i usuwaniu przeszkód

# Scrum – zespół i role

---

- ❑ **Właściciel Produktu**
- ❑ **ScrumMaster** – odpowiedzialny zarządzaniem procesem Scrum i usuwanie przeszkód
- ❑ **Zespół** - od 5 do 9 osób
  - Scrum bazuje na *samoorganizacji* zespołu i zaangażowaniu ludzi (pracy zespołowej)
  - podstawą realizacji prac jest *wspólne zobowiązanie* – nikt nie narzuca i nie przypisuje prac z góry (członkowie zespołu samodzielnie się ich podejmują i dzięki temu czują się bardziej zobowiązani)
  - zespół sam decyduje o metodach swojej pracy
  - wielofunkcyjny (interdyscyplinarny) i samowystarczalny – zróżnicowane i uzupełniające się umiejętności typu T
    - prostsza komunikacja
    - redukcja nieporozumień
    - szersza perspektywa

# Scrum – zasady pracy zespołu

---

- ☐ wspólne zobowiązanie – każdy czuje się zobowiązany do realizacji celów projektu (wspólne zobowiązanie)
- ☐ wzajemny szacunek – kultura egalitaryzmu, otwartość na pomysły i uwagi innych
- ☐ skoncentrowanie na pracy – praca tylko w jednym projekcie, unikanie wielozadaniowości
- ☐ otwartość (przejrzystość komunikacji) – każdy członek zespołu ma pełny przegląd sytuacji, wczesne ujawnianie problemów, utrwalanie zaufania (zarówno do procesu, jak i pomiędzy członkami zespołu)
- ☐ odwaga – otwartość wymaga odwagi
- ☐ podejmowanie decyzji w ostatnim możliwym terminie – zapewnia to swobodę wyboru właściwej osoby do wykonania potrzebnej pracy w odpowiednim czasie

# Scrum – praktyki

---

- ❑ **krótkie iteracje (sprinty)** do 30 dni ale stałej długości – efektem każdej iteracji jest przyrost funkcjonalności produktu

# Zalety ograniczenia czasowego sprintu

---

- ☐ Ograniczenie pracy częściowej – szybszy przepływ prac i dostarczanie produktów
- ☐ Szybsza korzyść z inwestycji
- ☐ Wymuszenie ustalenia priorytetów i wykonywania pracy o największym znaczeniu – wytworzenie czegoś wartościowego w szybkim tempie (bez ograniczenia w czasie realizuje się więcej funkcji o niskiej wartości)
- ☐ Szybkie weryfikowanie produktów i założeń oraz uzyskiwanie informacji zwrotnych (np. sygnałów do zaprzestania prac zanim doprowadzą one do poważnych strat ekonomicznych)
- ☐ Motywowanie domykania prac – bez ograniczenia czasowego znika poczucie pilności zakończenia prac
- ☐ Poprawianie przewidywalności – łatwiejsze planowanie
- ☐ Unikanie zbędnego perfekcjonizmu („done is better than perfect”) – ograniczenie czasowe wymusza zakończenie prac i „połączania”
- ☐ Ograniczanie propagacji błędów poprzez szybsze ich wykrywanie



# Scrum – praktyki

---

- ❑ **krótkie iteracje (sprinty)** – efektem każdej iteracji jest przyrost funkcjonalności produktu
- ❑ na początku iteracji – **spotkanie planujące sprint**
  - pierwsze 4 godz. – Właściciel Produktu prezentuje zespołowi wymagania i ich priorytety; zespół zadaje pytania i określa co da się zrobić (gra planistyczna)
  - kolejne 4 godz. – zespół rozplanowuje szczegóły działania

# Scrum – praktyki

---

- ❑ **krótkie iteracje (sprinty)** – efektem każdej iteracji jest przyrost funkcjonalności produktu
- ❑ na początku iteracji – **spotkanie planujące sprint** (max 8 godz.)
- ❑ w trakcie iteracji **codzienne ok. 15 min. spotkania** wszystkich członków zespołu (scrummy)
  - omawiane są zadania zrealizowane poprzedniego dnia (co robiłem wczoraj?)
  - zadania do wykonania w dniu spotkania (co będę robił dzisiaj?)
  - pojawiające się problemy (co mi przeszkadza?)
  - zespół mówi o tym do siebie – nie raportuje do szefa
  - uaktualnienie rejestru postępu prac (na tablicy zadań i wykresie spalania)
  - celem spotkania jest codzienne zsynchronizowanie pracy członków całego zespołu
  - sposób na wczesne wykrywanie problemów i zatorów

# Scrum – codzienne spotkania (scrumy)

---

- ☐ wszyscy powinni uczestniczyć w codziennych spotkaniach – włącznie z Właścicielem Produktu i ScrumMasterem
- ☐ na stojąco – z pełnym zaangażowaniem (bez odbierania e-maili, sprawdzania SMS itd.)
- ☐ przed tablicą – żeby każdy widział aktualny stan prac i wszystkie zadania do wykonania
- ☐ za każdym razem zaczyna kto inny – wszyscy uczestniczą na równych prawach (spotkania mają służyć całemu zespołowi)
- ☐ szczegółowe rozmowy i rozwiązywanie problemów po codziennym spotkaniu

# Scrum – praktyki

---

- ❑ **30-dniowe iteracje (sprinty)** – efektem każdej iteracji jest przyrost funkcjonalności produktu
- ❑ na początku iteracji – **spotkanie planujące sprint** (max 8 godz.)
- ❑ w trakcie iteracji **codzienne ok. 15 min. spotkania** wszystkich członków zespołu (scrummy)
- ❑ na końcu iteracji
  - **spotkanie przeglądu sprintu**
    - prezentowany jest produkt wykonany podczas iteracji
    - ocena i adaptacja produktu
    - wspólne określanie tego, co zespół powinien robić w dalszej kolejności
    - rezultatem przeglądu sprintu mogą być decyzje o dodaniu nowych funkcjonalności lub zmiana w istniejących priorytetach
    - dwukierunkowy przepływ informacji (informacja zwrotna)

# Scrum – praktyki

---

- ❑ **30-dniowe iteracje (sprinty)** – efektem każdej iteracji jest przyrost funkcjonalności produktu
- ❑ na początku iteracji – **spotkanie planujące sprint** (max 8 godz.)
- ❑ w trakcie iteracji **codzienne ok. 15 min. spotkania** wszystkich członków zespołu (scrumy)
- ❑ na końcu iteracji
  - **spotkanie przeglądu sprintu**
  - **retrospekcja**
    - ocena przebiegu iteracji
    - określenie działań w celu usprawnienia pracy
    - „oczyszczenie” atmosfery prac
    - celem jest nieustająca poprawa procesu, dzięki której zespół scrumowy stanie się zespołem jeszcze lepszym
    - dostosowanie Scruma do swoich własnych, unikatowych warunków

# Scrum – praktyki

---

- ❑ **30-dniowe iteracje (sprinty)** – efektem każdej iteracji jest przyrost funkcjonalności produktu
- ❑ na początku iteracji – **spotkanie planujące sprint** (max 8 godz.)
- ❑ w trakcie iteracji **codzienne ok. 15 min. spotkania** wszystkich członków zespołu (scrummy)
- ❑ na końcu iteracji
  - **spotkanie przeglądu sprintu**
  - **retrospekcja**

**Przegląd sprintu jest czasem przeznaczonym na ocenę i adaptację produktu (tego co tworzymy)**

**Retrospekcja sprintu stanowi okazję do oceny i adaptacji procesu (tego jak tworzymy)**

# Scrum – porównanie z podejściem tradycyjnym

<b>Sekwencyjnie (kaskadowo)</b>	<b>Scrum</b>
Koncentracja na formalnych procedurach	Nacisk na dostarczanie nadającego się do wdrożenia przyrostu produktu
Postęp mierzony liczbą ukończonych faz projektu	Postęp mierzony działającym oprogramowaniem
Tworzenie oprogramowania sterowane planem (z przewidywaniem)	Działanie w samą porę (just in time), adaptacja
Przewidywanie tego, co jest nieznane	Eksploracja i informacja zwrotna
Projekty o małej niepewności (daty są predykcją terminu zakończenia)	Projekty o dużej niepewności (daty stanowią momenty graniczne! )
Kontrolowanie, minimalizowanie i w miarę możliwości unikanie zmian	Otwartość na zmiany – zmiana to rzecz powszechna (nie da się wyeliminować niepewności)
Inwestowanie w zrobienie wszystkiego dobrze za pierwszym razem	Eksploracja i szybkie adaptacje w oparciu o informacje zwrotne od klienta
Wszystkie wymagania muszą posiadać ten sam poziom uszczegółowienia	Wymagania są uszczegółowiane w samą porę

# Scrum – podsumowanie

---

- ❑ Scrum stanowi pewien szkielet (jak fundament i ściany dla domu) – środowisko ze zbiorem wartości, zasad i praktyk
- ❑ Pozwala zastosować własne metody realizacji prac inżynierskich i sposoby postępowania w ramach tego szkieletu
- ❑ Proces iteracyjny i przyrostowy
- ❑ Istotą Scrum jest ograniczenie czasowe sprintu (które mobilizuje do działania) oraz dostarczanie w tym czasie produktów o określonej funkcjonalności
- ❑ Filozofia Scrum – „sztuka rzeczy możliwych”
- ❑ Scrum bazuje na samoorganizacji zespołu i zaangażowaniu ludzi – wartościach uczciwości, szacunku, zaufania, mobilizacji i pracy zespołowej
- ❑ Równowaga kontroli i elastyczności



# eXtreme Programming (XP)

# eXtreme Programming (XP)

---

W odróżnieniu od Scrum określa też wiele praktyk programistycznych

# eXtreme Programming (XP) – zespół i role

---

- ☐ **Przedstawiciele klienta**
- ☐ **Programiści**
- ☐ **Testerzy**
- ☐ **Coachowie**

**Role w dojrzałych zespołach XP nie są stałe i ściśle określone**

# eXtreme Programming (XP) – praktyki

---

- ☐ krótkie i częste iteracje
- ☐ programowanie w parach
- ☐ wytwarzanie sterowane testami
- ☐ ciągła integracja i testowanie
- ☐ kolektywne prawo do zmian
- ☐ dążenie do prostoty
- ☐ stały kontakt z klientem
- ☐ projektowanie przyrostowe

# eXtreme Programming (XP) – organizacja

---

- ☐ informacyjne miejsce pracy
- ☐ wspólna praca
- ☐ codzienne kilkuminutowe spotkania informacyjne
- ☐ zespół przez cały czas współpracuje z przedstawicielami klienta
- ☐ pełnoetatowość członków zespołu
- ☐ stopniowe zbieranie wymagań
- ☐ przestrzeganie ustalonych standardów pisania kodu
- ☐ szacowanie czasu na podstawie „prędkości” realizacji poprzedniej iteracji
- ☐ energiczna praca

# eXtreme Programming (XP) – otwartość na zmiany

---

W XP liczy się możliwość szybkiego wprowadzania zmian przy jak najmniejszej liczbie błędów

Praktyki wspierające łatwość wprowadzania zmian

- dążenie do prostoty kodu
- przestrzeganie ustalonych standardów pisania kodu
- projektowanie przyrostowe
- współwłasność kodu

Praktyki wspierające wykrywanie błędów:

- programowanie w parach
- programowanie sterowane testami
- ciągła integracja

# eXtreme Programming (XP) – cykl pracy

---

- ☐ Planowanie iteracji
- ☐ Zobowiązanie się do dostarczenia nowej funkcjonalności
- ☐ Tworzenie oprogramowania
- ☐ Udostępnienie kodu
- ☐ Demonstracja iteracji
- ☐ Retrospekcja

# eXtreme Programming (XP) – kwestie kontrowersyjne

---

- ☐ konieczna stała dostępność przedstawiciela klienta (klient „na miejscu”)
- ☐ „wspólna własność” kodu - każdy może zmieniać dowolny fragment systemu
- ☐ projektowanie przyrostowe



## Środowisko projektu

- **klient** – osoba lub instytucja zamawiająca i finansująca produkt
  - decydenci
  - użytkownicy
  - sponsor
- **udziałowcy** – osoby, na które wdrożenie systemu będzie miało bezpośredni wpływ
- **wykonawcy** – osoby lub instytucje tworzące produkt
  - kierownik projektu
  - główny projektant
  - pełnomocnik ds. jakości
  - liderzy zespołów
  - członkowie zespołów
  - konsultanci
  - kooperanci (podwykonawcy)
- **otoczenie**
  - dostawcy wyposażenia
  - opinia publiczna
  - konkurencja

## **Ogólna struktura organizacyjna zarządzania i kontroli projektu**

- **Sponsor**

- finansowanie projektu
- określanie ogólnych wytycznych do projektu
- „strategiczna” kontrola na projektem

- **Komitet Sterujący (rada, zarząd projektu)**

- „taktyczna” kontrola nad projektem
- ogólny nadzór nad realizacją projektu
- ocena i akceptacja poszczególnych etapów projektu
- ustalanie kierunków dalszej realizacji projektu

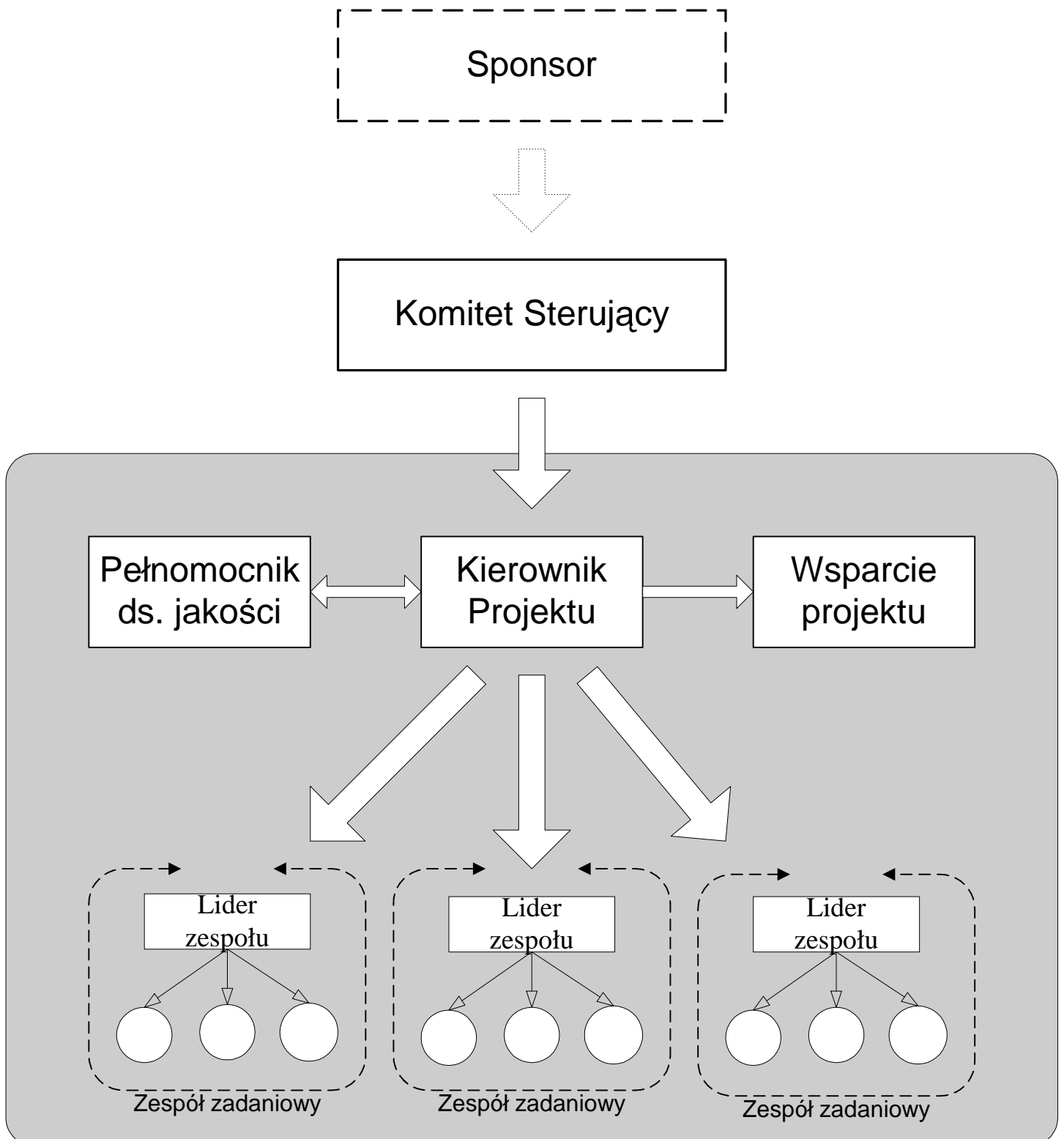
- **Kierownik Projektu (etapu)**

- przygotowanie planów projektu
- organizowanie, przydział i zarządzanie zasobami
- prowadzenie projektu
- „operacyjna” kontrola nad postępami projektu
- raportowanie o stanie prac

- **Zespoły wykonawcze**

- wykonywanie szczegółowych zadań związanych z realizacją projektu

## Struktura zarządzania projektem



## **Zadania sponsora projektu**

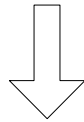
- określanie ogólnych założeń projektu (planów strategicznych)
- zapewnienie odpowiednich środków finansowych niezbędnych do realizacji projektu
- popieranie projektu
- zapewnienie dostępu ekspertów, których wiedza i zaangażowanie jest niezbędne do realizacji projektu
- rozwiązywanie konfliktów wewnątrz organizacji związanych z projektem

## **Zadania Komitetu Sterującego**

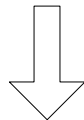
- inicjowanie projektu
- nadzór projektu
  - raporty okresowe
  - raporty o istotnych odchyleniach
  - przeglądy etapów
- ustalanie priorytetów działań
- zatwierdzanie planów projektu na następne etapy
  - harmonogramu i budżetu
  - planu jakości
  - planów naprawczych
- określanie parametrów tolerancji
- wsparcie projektu i pomoc w podejmowaniu kluczowych decyzji
- zamykanie projektu
- opracowanie raportu o stanie projektu dla sponsora
  - stopień realizacji celów projektu
  - wykorzystany i planowany budżet projektu
  - planowany harmonogram realizacji projektu z kluczowymi datami

## Ocena etapu

- Plan realizacji projektu
- Raport końcowy etapu
- Plan etapu następnego
- Uzasadnienie biznesowe
- Rejestr ryzyka

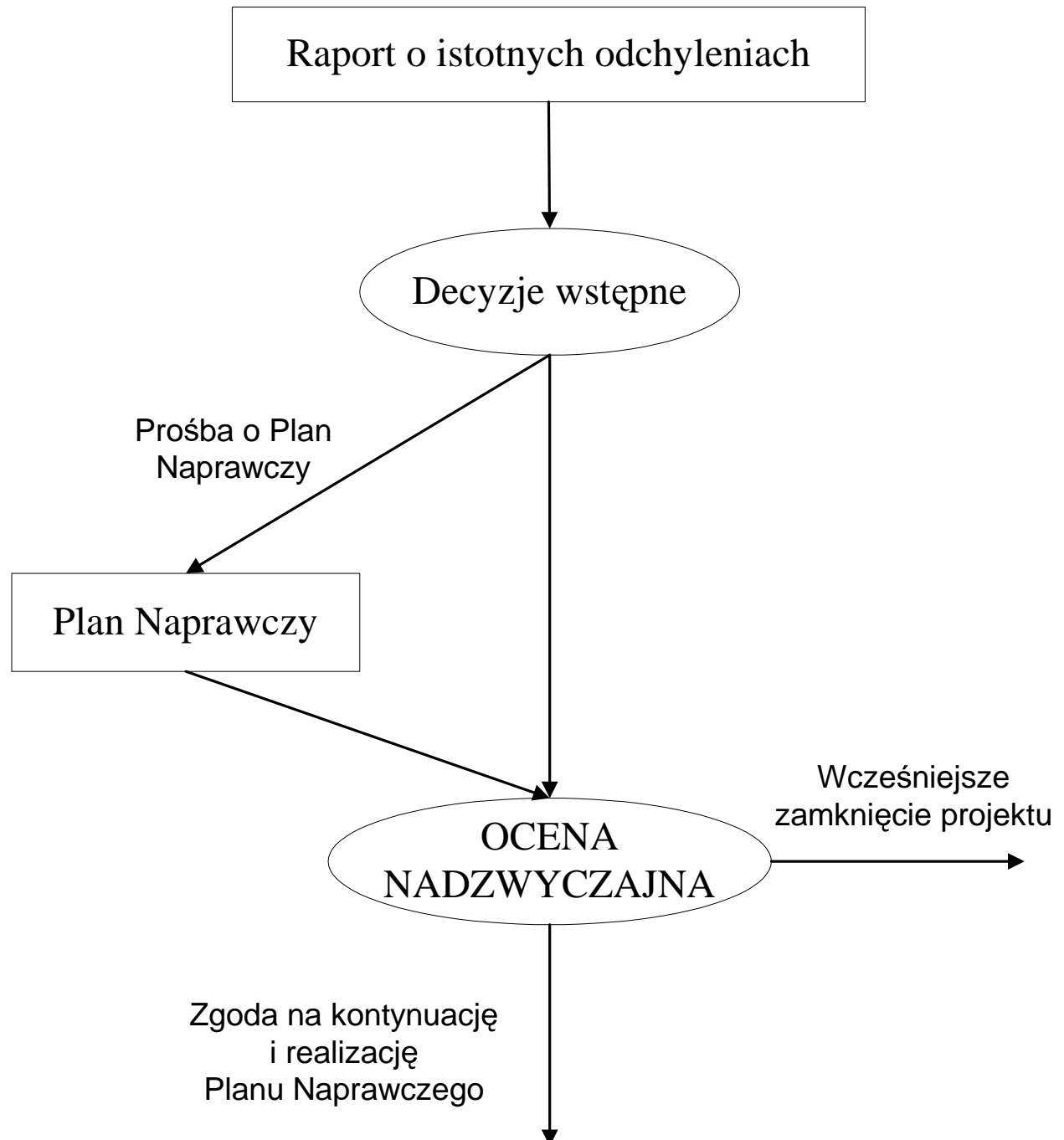


OCENA ETAPU



- Decyzje dotyczące dalszych losów projektu
  - kontynuowanie prac - inicjowanie następnego etapu
  - powtórzenie części lub całego etapu
  - zawieszeniem realizacji projektu
  - przerwanie projektu
- Zatwierdzone plany dla następnego etapu
- Parametry tolerancji dla odchyleń harmonogramu i kosztów
- Zalecenia dla kierownika projektu

## Ocena nadzwyczajna



## **Po co Komitet Sterujący ?**

### **Może zabezpieczyć projekt przed:**

- przekroczeniami budżetu i czasu (poprzez nadzór osób spoza grona wykonawców)
- nieosiągnięciem celów projektu – po każdym przeglądzie Komitet Sterujący może zmienić kierunek projektu tak, by osiągnąć cele
- brakiem lub nadmierną reakcją na zmiany w wymaganiach, środowisku lub zasobach

### **Częstotliwość spotkań Komitetu Sterującego**

- raz na jeden-dwa miesiące oraz na zakończenie każdego etapu projektu
- w sytuacjach krytycznych



## **Zadania Kierownika Projektu**

- zarządzanie projektem
  - organizowanie
  - planowanie
  - kierowanie i koordynacja prac
  - kontrolowanie
  - nadzorowanie dokumentacji projektu
- raportowanie
  - raporty okresowe
  - raport końcowy etapu
  - raport o istotnych odchyleniach

## **Raporty Kierownika Projektu**

- raport okresowe
  - postęp prac osiągnięty w danym okresie
  - produkty wykonane
  - stan budżetu
  - aktualne lub potencjalne problemy
  - uaktualniony rejestr zagrożeń
  - produkty do wykonania w następnym etapie
- raport końcowy etapu
  - terminy planowane i faktyczne
  - nakłady pracy planowane i poniesione
  - koszty planowane i poniesione
  - produkty dostarczone i potwierdzenie, że spełniają określone dla nich kryteria jakości
- raport o istotnych odchyleniach
  - informacja o przekroczeniu granic tolerancji
  - powody odchyleń
  - wpływ na etap i projekt
  - wpływ na uzasadnienie biznesowe
  - rekomendacje

## **Zadania liderów zespołów**

- określanie szczegółowych zadań w krótkoterminowym (np. tygodniowym) horyzoncie czasu
- ustalanie krótkoterminowego harmonogramu i przydział zadań w zespole
- ocena pracy członków zespołu
- raportowanie

## **Zadania członków zespołów zadaniowych**

- wykonywanie zadań w terminie
- udział w spotkaniach zespołowych
- dokumentacja prac

## **Wsparcie projektu**

(Biuro projektu)

- administrowanie poszczególnych procesów
- pilnowanie przestrzegania standardów
- zarządzanie dokumentacją projektu
- rozsyłanie komunikatów i dokumentów
- zarządzanie konfiguracją
- gromadzenie raportów i danych o faktycznym postępie prac
- organizacji i obsługa zebrań
- koordynacja kontaktów z klientem

## **Hierarchia ról u wykonawców**

- **Kierownik Projektu i Główny Projektant**

- ta sama osoba

- zwykle tylko w małych projektach
- mało jest osób mających tak wszechstronne zdolności
- trudno pogodzić pracowitość obu ról

- **Kierownik Projektu ponad Głównym Projektantem**

- powinny być duże uprawnienia dla Głównego Projektanta w zakresie podejmowania decyzji technicznych

- **Główny Projektant ponad „Kierownikiem Projektu”**

- „Kierownik Projektu” pełni funkcje administracyjne

- **Pełnomocnik ds. jakości**

- przeciwwaga dla Kierownika Projektu

## **Organizacja struktury zespołu wykonawczego**

### **Zalecenia**

- jasno określone cele działania i kryteria oceny
- przydział ról adekwatnych do posiadanych kompetencji i wagi podejmowanych decyzji
- jednoznacznie i precyzyjnie określony zakres odpowiedzialności
- przydział uprawnień decyzyjnych
- zdefiniowane procedury komunikacji

## **Organizacja struktury zespołu wykonawczego**

### **Procedura**

- identyfikacja potrzeb projektu
- identyfikacja ról
- opis roli
  - nazwa
  - wymagane kwalifikacje (preferencje)
  - profil realizowanych zadań
  - ograniczenia
- określenie uprawnień i zakresu odpowiedzialności dla poszczególnych ról
- schemat organizacyjny projektu
  - hierarchia zależności
  - schemat raportowania postępu prac i sygnalizowania sytuacji krytycznych
- plan obsady ról w projekcie – może być rozłożony w czasie

## **Trudności z doborem zespołu**

- personel z odpowiednimi kompetencjami i doświadczeniem może nie być dostępny w ramach firmy, a nawet w jej otoczeniu
- budżet projektu nie pozwala na zatrudnienie takich wykonawców
- firma może oczekiwać, aby do projektu angażować mniej doświadczonych pracowników w celu doskonalenia ich umiejętności (należy w pierwszej kolejności wykorzystywać zasoby firmy)



## **Zarządzanie wymaganiami (zakresem)**

- usystematyzowane podejście do ustalania i utrzymywania porozumienia między klientem i zespołem wykonawców
- zarządzanie procesem pozyskiwania, specyfikacji i weryfikacji wymagań
- opracowanie standardu specyfikacji wymagań
- zarządzanie specyfikacją wymagań w różnych fazach realizacji projektu (analiza, projektowanie, programowanie) – utrzymywanie zgodności
- stworzenie systemu identyfikacji i śledzenia wymagań w dokumentacji
- zarządzanie zmianami wymagań

## **Trudności w określeniu wymagań**

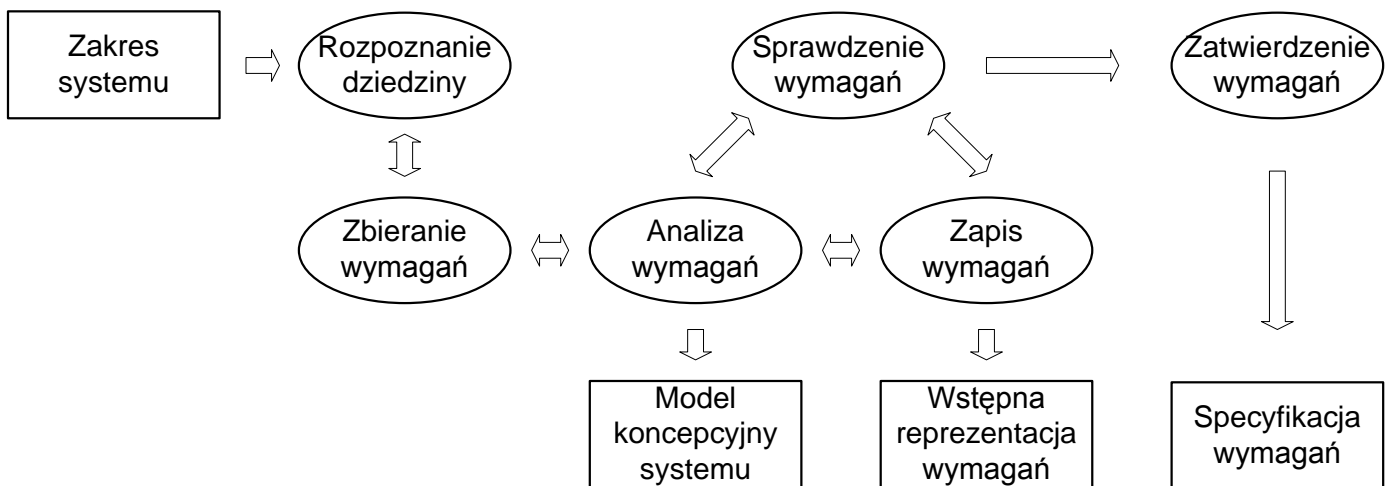
- klienci i użytkownicy często nie wiedzą dokładnie czego chcą i zaczynają rozumieć swoje potrzeby dopiero wtedy, kiedy widzą system podczas pracy
- duże systemy są wykorzystywane przez wielu użytkowników z różnymi punktami widzenia (mogą być bardzo różne nawet sprzeczne wymagania)
- klient z reguły nie wie dokładnie w jaki sposób osiągnąć założone cele – konieczność negocjacji wymagań
- różnice środowiskowe i pojęciowe – trudności we wzajemnym zrozumieniu
- klient może nie „dostrzegać” pewnych ograniczeń lub może uznawać je za oczywiste (powszechnie znane)
- konieczność wynegocjowania zmiany organizacji pracy w przedsiębiorstwie (żeby nie informatyzować bałaganu)
- niestabilne koncepcje systemu lub uwarunkowania zewnętrzne (np. wymagania prawne) – „pełzanie” wymagań

## **Pozyskiwanie wymagań (w metodykach tradycyjnych)**

- ustalenie zakresu systemu
- identyfikacja źródeł pozyskiwania wymagań i różnych punktów widzenia
- wydobywanie wymagań
- analiza wymagań
- wstępna reprezentacja wymagań
- potwierdzenie właściwego zrozumienia wymagań
- konsolidacja i specyfikacja wymagań
- analiza specyfikacji: weryfikacja i walidacja
- ustalenie hierarchii wymagań
- zatwierdzenie wymagań

### **Proces iteracyjny**

# Proces określania wymagań



## **Źródła informacji**

- bezpośredni użytkownicy – funkcje systemu i sposób użytkowania
- wyższe kierownictwo – cele strategiczne, bezpieczeństwo
- administrator systemu – niezawodność, funkcje wykrywania i diagnozy błędów itp.
- kierownik marketingu – nowoczesność rozwiązań
- kierownik finansowy – koszty

## **Ponadto**

- dokumenty – instrukcje, procedury
- przykłady działania starego systemu lub systemów podobnych

# Metody zbierania wymagań

- wywiady
  - z wyższym kierownictwem odnośnie strategii firmy i celów systemu
  - z kadrami inżynierską odnośnie potrzeb i ograniczeń technicznych
  - z „bezpośrednimi” użytkownikami
- warsztaty wymagań, „burze mózgów”
- ankiety (ograniczona użyteczność)
- analiza scenariuszy zdarzeń (przypadków użycia)
- poznawanie środowiska docelowego, w którym zostanie zainstalowany system
- odgrywanie ról użytkowników
- studiowanie innych istniejących systemów
- stosowanie prototypów

## **Analiza wymagań**

- głębsze poznawanie wymagań
- kształtowanie modelu przyszłego systemu
- wykrywanie braków w zestawie wymagań
  - błędów (wymagania błędnie wyspecyfikowane)
  - sprzeczności (konfliktów)
  - nadmiarowości
  - luk (wymagania niewyspecyfikowane)
  - wymagań spoza zakresu systemu
- badanie wzajemnych związków między wymaganiami
- analiza sytuacji nietypowych

## **Pożądane cechy Specyfikacji Wymagań**

- jednoznaczność – w interpretacji wymagań
- kompletność
- poprawność
- spójność (niesprzeczność)
- weryfikowalność – istnieją procedury sprawdzenia spełnienia wymagań
- modyfikowalność – możliwość łatwego uwzględnienia zmian w specyfikacji
- jasny system identyfikacji i odwołań



# **Specyfikacja wymagań w podejściach tradycyjnych i zwinnych**

## **W podejściach tradycyjnych**

- kompletne wymagania tworzone są na samym początku
- są one bardzo szczegółowe – wszystkie wymagania muszą jednocześnie posiadać ten sam poziom uszczegółowienia
- poświęca się wiele wysiłku żeby „zrobić wszystko dobrze za pierwszym razem” i potem „za wszelką cenę” unika się zmian

## **W podejściach zwinnych**

- uważa się, że podczas tworzenia innowacyjnych produktów nie da się pozbyć niepewności i z góry stworzyć wyczerpującej listy wymagań poprzez wydłużenie czasu prac lub wzmożenie wysiłków
- nie inwestujemy dużo czasu i pieniędzy w wytworzenie szczegółowych wymagań na samym początku projektu
- na początku tworzymy jedynie wskaźniki wymagań nazywane historyjkami użytkownika
- szczegóły są precyzowane później w samą porę

## **Zapis wymagań w metodykach zwinnych**

- zbiór historyjek (opowieści) użytkownika (ang. *user stories*)
- historyjki są uporządkowane według priorytetów
- rejestr produktu jest „żywy” i podlega ciągłym zmianom – zmieniają się priorytety, modyfikowany jest zakres historyjek itd.

### **Historyjka (opowieść) użytkownika**

- prosty jedno lub dwuwierszowy opis funkcjonalności požądanej przez klienta
- historyjki zapisuje się na osobnych kartkach (kartonikach)
- od zapisu historyjki dużo ważniejsza jest rozmowa, o której papierowy kartonik ma nam przypomnieć
- historyjka powinna być napisana w języku klienta i opisywać efekt wartościowy dla klienta, a nie szczegóły techniczne
- historyjki powinny mieć jasne kryteria ukończenia (obiektywne testy akceptacyjne) – zapisywane zwykle na odwrocie kartonika

Bezpośredni kontakt i rozmowa zamiast szczegółowego dokumentu ze specyfikacją wymagań

## Zapis wymagań w metodykach zwinnych

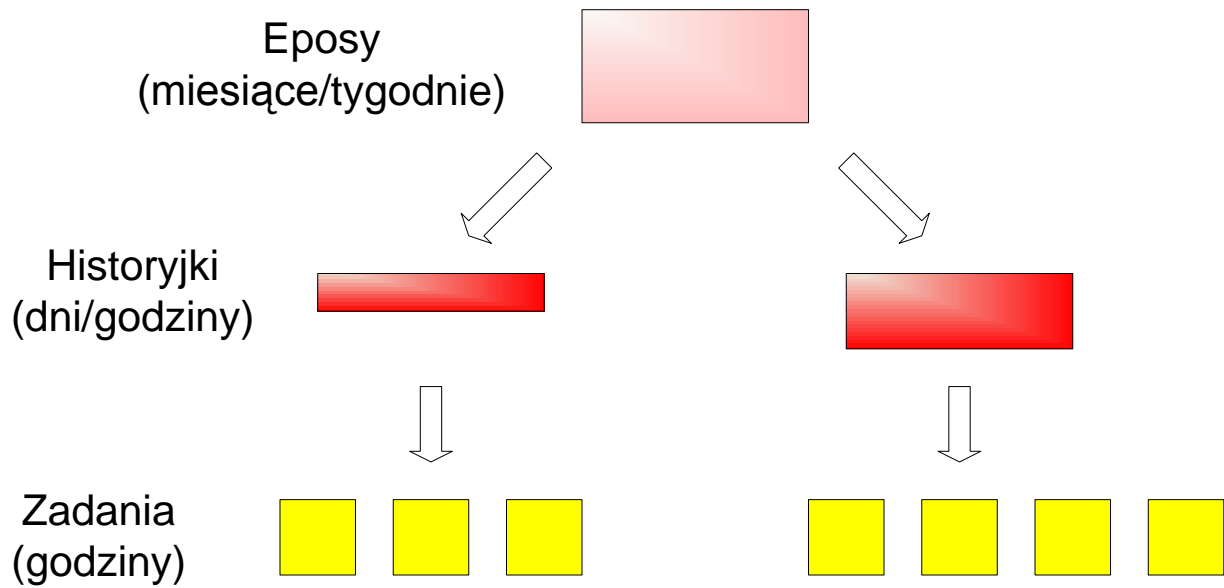
### Atrybuty dobrze sformułowanych historyjek - INVEST

- Niezależne (ang. *Independent*) – historyjki są niezależne jeżeli możemy implementować je w różnej kolejności
- Negocjowalne (ang. *Negotiable*) – szczegóły dopracowuje się w bezpośredniej rozmowie
- Wartościowe dla odbiorcy (ang. *Valuable for Users and Customers*) - reprezentują wartość dla klienta i są napisane w jego języku
- Możliwe do oszacowania (ang. *Estimatable*) - wymaga to możliwości zrozumienia istoty historyjki, umiarkowanego rozmiaru historyjki, doświadczenia i wiedzy zespołu
- Małe (ang. *Small*) - można je łatwo zaplanować i oszacować (np. zaimplementować w jednej iteracji)
- Testowalne (ang. *Testable*) – mają jasno określone kryteria ukończenia

## **Zapis wymagań w metodykach zwinnych**

- zbiór historyjek (opowieści) użytkownika (ang. user stories)
- historyjki są uporządkowane według priorytetów (tylko początek listy)
- historyjki w rejestrze mają różne rozmiary i są w różnym stopniu uszczegółowione
- historyjki z największym priorytetem (nad którymi planujemy pracować wkrótce) mają najmniejszy rozmiar i są bogate w detale
- stosuje się strategię stopniowego uszczegółowiania historyjek – zbliżając się do pracy nad większymi elementami, rozbijamy je na zbiory mniejszych historyjek nadających się do implementacji w jednej iteracji
- rejestr produktu jest „żywy” i podlega ciągłym zmianom – zmieniają się priorytety, modyfikowany jest zakres historyjek itd. (ciągła pielęgnacja rejestru produktu)

## Poziomy szczegółowości elementów rejestrów



# Poziomy szczegółowości elementów rejestrów

## Rejestr produktu

- **Eposy** – duże historyjki wymagające kilka miesięcy/tygodni pracy (bez szczegółów)
- (małe) **Historyjki** – kilka godzin/dni pracy (poziom szczegółowości odpowiedni do realizacji w ramach iteracji)

## Rejestr iteracji

**Historyjki z Zadaniem** (prace inżynierskie niezbędne do realizacji poszczególnych historyjek)

Zadania nie są historyjkami – specyfikują, *jak* należy coś zbudować, w przeciwieństwie do tego *co* należy zbudować (określają to eposy i historyjki)

## **Planowanie (harmonogramowanie)**

### **Po co planować ?**

- w celu określenia wszystkich zadań do wykonania
- w celu efektywnego wykorzystania czasu i zasobów (określenia niezbędnych zasobów i kosztów)
- planowanie ujawnia wąskie gardła i zagrożenia
- planowanie umożliwia sformułowanie cząstkowych celów oraz hierarchii ich ważności
- plan określa wizję całego projektu – wagę zadań i udział poszczególnych wykonawców, co zwykle wpływa na większe zaangażowanie w projekcie
- planowanie określa reżim realizacji projektu ograniczając wpływ chwilowych opinii i emocji
- plan jest podstawą do koordynacji i kontroli projektu
- plan stanowi dokumentację, która może być poddana ocenie oraz może być wykorzystywana do doskonalenia procesu planowania w przyszłości

## **Niedogodności planowania**

- planowanie wymaga czasu
- planowanie jest kosztowne (wymaga dobrych fachowców)
- planowanie jest oparte na oszacowaniach i prognozach przyszłych wydarzeń
- stosowanie planów „usztynia” funkcjonowanie przedsiębiorstwa



## **Najważniejsze elementy planowania**

- uzgodnienie celów i ograniczeń projektu
- strategia realizacji – określenie etapów i zadań
- zdefiniowanie produktów końcowych i pośrednich
- określenie infrastruktury projektu
- oszacowanie pracochłonności i kosztów
- opracowanie harmonogramu, budżetu i przydziału zasobów
- określenie punktów kontrolnych

## Hierarchia harmonogramów

- harmonogram negocjacyjny – estymacja zgrubna na poziomie całego projektu
  - czasu
  - budżetu
- harmonogram kontraktowy – bardziej szczegółowy harmonogram w oparciu o wymagania użytkownika z podziałem na etapy
  - kontrakt zwykle obejmuje pierwsze etapy, a następne etapy warunkowo z możliwością renegotjacji
- harmonogram szczegółowy – na poziomie etapu
  - czynności
  - budżetu
  - zasobów
  - procedur kontroli

## W trakcie realizacji projektu

Struktura dwupoziomowa

- aktualny etap – harmonogram **szczegółowy**
- następne etapy – harmonogram zgrubny (np. kontraktowy)

## **Czynniki decydujące o długości etapu**

- struktura projektu
- poziom ryzyka
- możliwość oceny przebiegu, sensowności i kosztów projektu
- stopień poinformowania kierownictwa
- czas niezbędny na przeglądy etapów (przygotowanie dokumentacji, przeprowadzenie przeglądu itd.)
- utrzymywanie zaangażowania

## **Elementy planowania**

- określenie zakresu prac
- zaproponowanie „technologii” – sposobu wykonywania prac (kolejności, standardów, narzędzi, zasobów itd.)
- zdefiniowanie produktów końcowych i pośrednich
- oszacowanie pracochłonności i kosztów
- rozplanowanie zadań w czasie z uwzględnieniem dostępnych zasobów, ograniczeń i czynników ryzyka

## **Etapy harmonogramowania szczegółowego**

- uzgodnienie celu, zakresu i ograniczeń etapu  
(TOR - Terms of Reference)
- podział etapu i zadania na zadania  
(WBS - Work Breakdown Structure)
  - specyfikacja zadań
  - określenie ograniczeń kolejnościowych zadań
  - szacowanie czasów wykonania zadań
  - określenie efektów wykonania zadań (produktów)
  - przydział zasobów do zadań
- sporządzenie i analiza sieci czynności
  - analiza ścieżki krytycznej
  - analiza rozdziału zasobów
- opracowanie wstępnego harmonogramu projektu
  - szacowanie i harmonogramowanie kosztów
  - zdefiniowanie punktów kontrolnych
- weryfikacje i korekty (proces iteracyjny)
- akceptacja harmonogramu

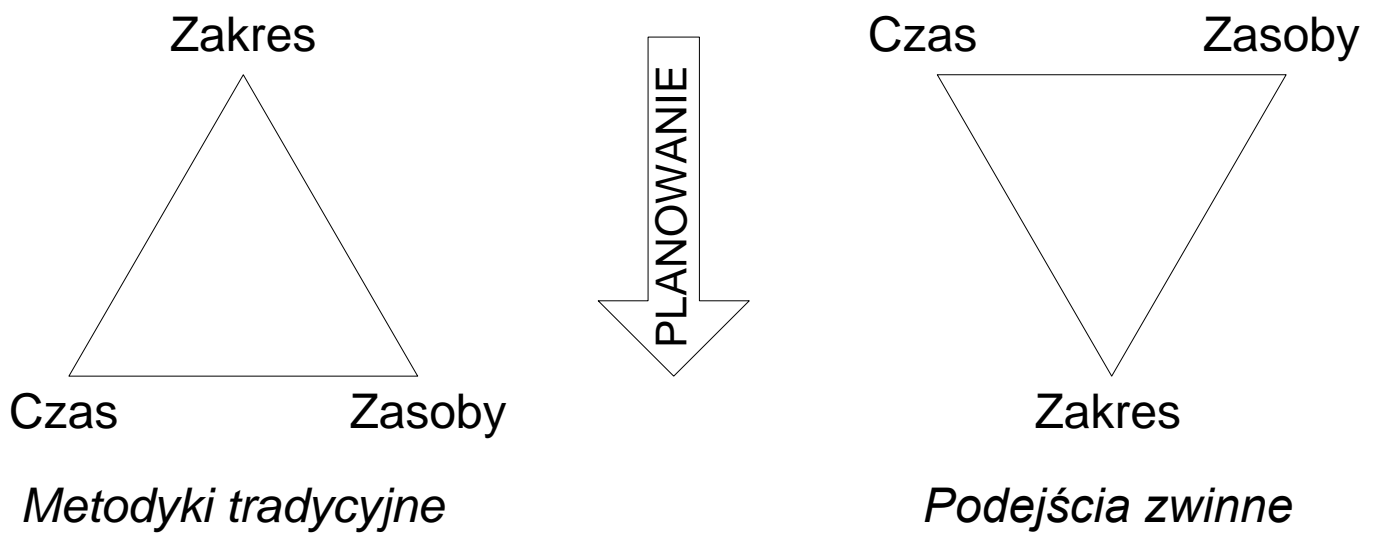
## **Specyfikacja zadań**

- cel zadania
- procedura – opis czynności wchodzących w skład zadania
- CZAS przeznaczony na realizację zadania
- dane wejściowe – materiały źródłowe dla zadania
- produkty – efekt realizacji zadania
- kryteria rozpoczęcia – kiedy, przy jakich warunkach można rozpocząć realizację zadania
- kryteria zakończenia – kryteria akceptacji (odbioru)
- role – kto realizuje zadanie (jaka komórka) i jest za nie odpowiedzialny
- narzędzia, zasoby niezbędne do realizacji
- metryki – opis danych o zadaniu, które należy gromadzić w trakcie jego realizacji
- inne – inne dane, dokumenty, standardy, wymagania, ograniczenia itp., które mają wpływ na sposób realizacji zadania

## **Zasady planowania zwinnego**

- Przyjmuje się, że nie da się wszystkiego przewidzieć z góry
- Odpowiednia równowaga pomiędzy planowaniem z góry i szczegółowym planowaniem w samą porę
- Planowanie z góry powinno być pomocne, ale nie rozległe
- Wstrzymywanie się z planowaniem szczegółowym w samą porę do ostatniego rozsądnego momentu
- Planowanie w sposób ciągły (wielokrotny) z adaptacją
- Planowanie wielopoziomowe – różne poziomy szczegółowości planów
- Planowanie z ograniczeniem czasowym

## Planowanie z ograniczeniem czasowym





## **Planowanie wielopoziomowe w podejściach zwinnych**

Warstwy planowania zwinnego

- **Planowanie produktu** – tworzenie wizji produktu
- **Planowanie wydania** – grupowanie historyjek, które po zaimplementowaniu stworzą dla klienta nową wersję
- **Planowanie iteracji** – zespół z Właścicielem Produktu zastanawia się ile jest w stanie zrobić w ramach iteracji i jak podzielić pracę na konkretne zadania
- **Planowanie codzienne** – codzienne spotkania

## **Planowanie produktu**

- Tworzenie wizji produktu – do czego zmierzamy?
- Nie musimy znać wszystkich detali dotyczących produktu
- Informacje niezbędne dla decydentów do podjęcia decyzji o uruchomieniu i finansowaniu projektu (karta wizji)
- Stworzenie rejestru produktu wysokiego poziomu – początkowej wersji rejestru produktu na dużym poziomie ogólności (eposów)
- Skupienie się głównie na zdefiniowaniu minimalnego zakresu pierwszego wydania produktu
- Dążenie do szybkiego uzyskania informacji zwrotnej potwierdzającej lub odrzucającej założenia odnośnie docelowego produktu (strategia „szybkiej porażki”)

## **Planowanie wydania w podejściach zwinnych**

Cel: zbilansowanie zakresu z czasem dostarczenia  
(planowanie długoterminowe)

I

Informacje potrzebne do konstruowania planu wydania

- rejestr produktu – podstawa
- oczekiwania klienta
- wielkość pracy – wymaga oszacowania
- długość iteracji i prędkość zespołu

**Prędkość** zespołu określa ile zespół jest w stanie zrobić w ciągu jednej iteracji pracując w podtrzymywalnym tempie

**Tempo podtrzymywalne** – prędkość zapewniająca komfort pracy w długim okresie (sprawnie ale bez pośpiechu)

## **Planowanie wydania w podejściach zwinnych**

### **Rezultat planowania**

- lista planowanych funkcjonalności, które chcielibyśmy, żeby znalazły się w wydaniu poukładane są według priorytetów.  
(niektóre zespoły dzielą je również na te, które za wszelką cenę muszą zrobić - zakładając pesymistyczny wariant prędkości i te które może dostarczymy, pod warunkiem że się uda osiągnąć lepsze tempo)
- planowane daty rozpoczęcia i zakończenia sprintów oraz planowana data zakończenia wszystkich prac w wydaniu

**Plan wydania wymaga ciągłej aktualizacji, np. po każdej iteracji**

## **Planowanie iteracji w podejściach zwinnych**

1. wybór historyjek na podstawie ich wartości dla klienta (priorytetów), szacunkowego czasu pracy nad nimi i prędkości (gra planistyczna zespołu i Właściciela Produktu)
2. podział historyjek na zadania inżynierskie (wskazane aby zadania były do wykonania w ciągu jednego dnia – ułatwia to śledzenie postępu prac)
3. szacowanie zadań w roboczogodzinach (idealnych)
4. podjęcie zobowiązania
5. umieszczenie karteczek z historyjkami i zadaniami na tablicy zadań w kolumnie „Do zrobienia” (rejestr sprintu)
6. przydział zadań i ustalenie strategii realizacji zadań

Inny wariant 6: przydział zadań do realizacji w samą porę

Preferowane są zadania tych historyjek, nad którymi już rozpoczęto prace

## **Co należy oszacować ?**

- pracochołoność
- harmonogram – czas trwania zadań i całego projektu
- zasoby ludzkie – jak duży zespół
- budżet
  - koszty wynagrodzeń
  - koszty sprzętu i oprogramowania
  - koszty materiałów
  - koszty usług obcych (podwykonawców)
  - koszty szkoleń, wyjazdów itp.

## **Trudności w szacowaniu**

- bardzo duże zróżnicowanie i złożoność projektów informatycznych
- nieliniowy wzrost złożoności oprogramowania przy zwiększaniu zakresu projektu
- zmienność wymagań, środowiska, organizacji przy każdym nowym projekcie
- zmienność technologii – praktycznie każdy nowy projekt w innej technologii
- rosnący udział kosztu opracowania oprogramowania w ogólnych kosztach systemu
- „niematerialny” charakter oprogramowania trudny z natury do oszacowania
- brak doświadczenia zespołów projektowych – zwykle młodzi ludzie
- brak dojrzałych metryk oprogramowania dobrze skorelowanych z procesem tworzenia oprogramowania

## **Dodatkowo**

- konieczność dokonywania oszacowań przed
  - specyfikacją wymagań
  - wykonaniem projektu, który określa technologię, narzędzia i architekturę

## **Przy szacowaniu trzeba uwzględnić**

- złożoność zadania
- umiejętności i doświadczenie pracowników
- znajomość i dostępność technologii
- czas nieproduktywny (praca w innych projektach, administrowanie, wakacje, choroby, szkolenia)
- czynności i czas związany z zarządzaniem (spotkania, przygotowywanie raportów, sprawozdań itp.)
- czas niezbędny dla komunikacji w zespole
- czas przeznaczony na kontrolę jakości (np. audyty, przeglądy, inspekcje)



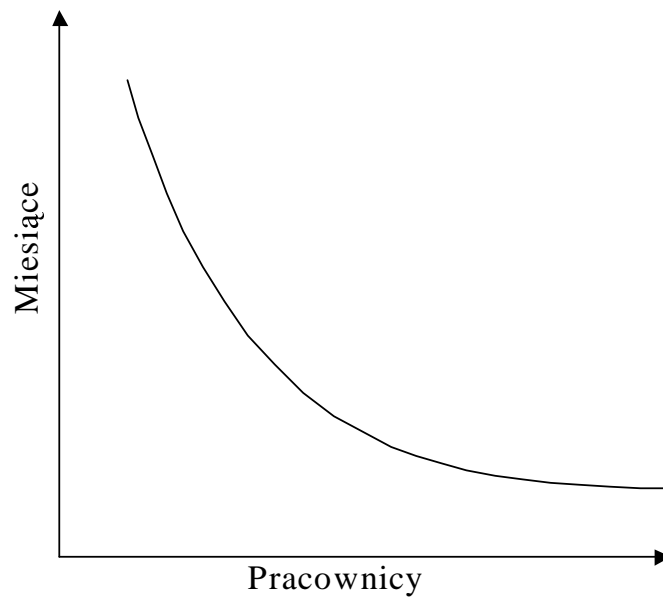
## **Czas realizacji a liczba pracowników**

### **Zależność nieliniowa**

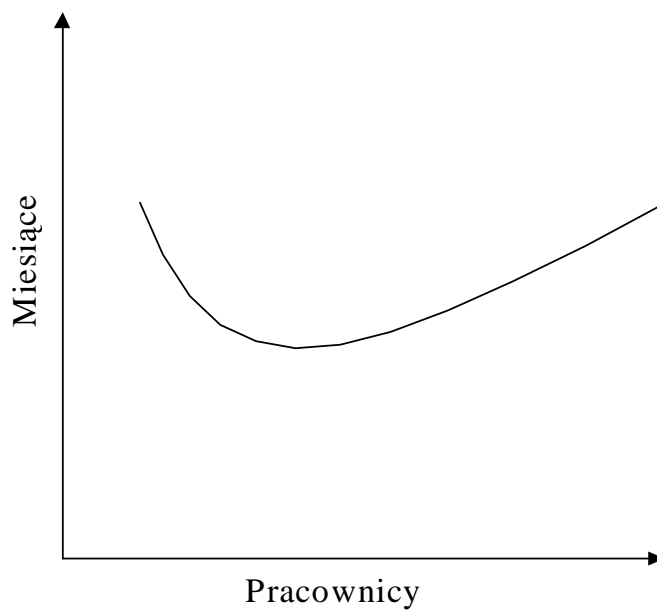
### **Przyczyny**

- wraz z liczbą pracowników rośnie nakład czasu na komunikację
- niepodzielność zadań

## **Czas realizacji a liczba pracowników**



Projekty wymagające niewielkiej komunikacji



Projekty wymagające intensywnej komunikacji

## **Zalecenia przy szacowaniu**

- dekompozycja
- zlicz, oblicz a oceniaj w ostateczności
- zbieranie (zapisywanie) doświadczeń - danych historycznych
- opieranie się na charakterystyce produktywności zespołu
- szacowanie z określaniem prawdopodobieństwa
- uaktualnianie oszacowań
- stosowanie różnorodnych oszacowań
- unikanie zbyt ostrożnego i zbyt napiętego szacowania

## **Metody szacowania pracochłonności**

- szacowanie wstępujące
- szacowanie zstępujące – określenie stopnia pracochłonności zadania względem większej całości (np. na zasadzie reguły 40-20-40)
- na zasadzie analogii z podobnymi przedsięwzięciami
- ocena ekspertów
  - metoda delficka
- szacowanie według kategorii
- metoda AQUA
- szacowanie oparte na zliczaniu obiektów zastępczych
  - zliczanie lub szacowanie liczby skorelowanych obiektów zastępczych reprezentujących wielkość projektu (np. linii kodu – COCOMO II, funkcji – metoda punktów funkcyjnych, punktów historyjek – podejścia zwinne)
  - konwersja liczby i charakteru obiektów zastępczych na końcowe oszacowanie na podstawie danych historycznych
- użycie i porównanie kilku metod

## **Szacowanie na zasadzie analogii z podobnymi projektami**

- 
- ustalenie wielkości, pracochłonności i ostatecznego kosztu starego projektu
- porównanie wielkości nowego projektu ze starym
- oszacowanie wielkości nowego projektu w stosunku do starego projektu (procentowe)
- oszacowanie pracochłonności nowego projektu na podstawie porównania wielkości nowego projektu ze starym
- uwzględnienie pozostałych różnic między projektami (np. stosowanej technologii, doświadczenia zespołu itp.)

## **Metoda delficka**

Stosowana raczej w początkowej fazie projektu, gdy mamy do czynienia z nieznanym systemem i gdy projekt obejmuje kilka odmiennych dziedzin - w sytuacji dużej niepewności

### **Założenia**

- niezależność ocen ekspertów
- anonimowość ocen
- wieloetapowość postępowania
- kontrolowane sprzężenie zwrotne
- dążenie do uzgadniania ocen ekspertów i osiągnięcia konsensusu

## **Metoda delficka**

- szacowania dokonuje grupa ekspertów – każdy z osobna (w sposób niezależny)
- wyniki zbiera się i wyznacza
  - wartość średnią oszacowań
  - ewentualnie najbardziej pesymistyczne i optymistyczne oszacowanie
- powyższe wielkości przedstawia się ekspertom i prosi się ich znowu o dokonanie (zrewidowanie) oszacowania
- proces kontynuuje się aż się uzyska zbieżność ocen (albo się nie uzyska)

## **Kryteria zakończenia**

- oceny ekspertów są zbieżne – różnice między nimi są mniejsze od założonego progu zbieżności
- wykonano 3 lub 4 rundy – zwykle po tylu rundach oceny się już nie zmieniają
- wszyscy eksperci nie chcą zmienić swoich ocen
- wyczerpano limit czasu

## **Metoda delficka – modyfikacje**

- eksperci, którzy podali skrajne oceny proszeni są o uzasadnienie wszystkim swoich opinii
- może też być dyskusja
- po wysłuchaniu argumentów następna runda
- są trzy rundy
- średnia ocena z trzeciej rundy stanowi końcowe oszacowanie



## Szacowanie według kategorii

- klasyfikacja projektu (zadania) względem wielkości, np. mały, średni, duży
- klasyfikacja projektu (zadania) względem złożoności, np. prosty, umiarkowany, złożony
- tabelaryczna ocena pracochłonności (kosztu)

### Przykład

„Przygotowanie planu realizacji projektu”

Złożoność projektu

Rozmiar projektu	Złożoność projektu		
	Prosty	Umiarkowany	Złożony
	Mały	10 osobodni	13 osobodni
	Średni	12 osobodni	15 osobodni
Duży	15 osobodni	18 osobodni	21 osobodni

## **Metoda AQUA**

- opiera się na analizie danych historycznych
  - wartości atrybutów
  - pracochłonność
- badany jest stopień podobieństwa aktualnego projektu do projektów historycznych na podstawie porównywania wartości atrybutów
- określenie  $N$  najbardziej podobnych przypadków w oparciu o wartości miary podobieństwa ( $N$  wyliczane automatycznie)
- wyliczana jest pracochłonność nowego projektu na podstawie pracochłonność  $N$  projektów najbardziej podobnych (jako średnia ważona względem miar podobieństwa)

## Metoda punktów funkcyjnych

- obliczenie liczby wstępnych punktów funkcyjnych (*UFP– Unadjusted Function Points*)
- obliczenie sumarycznego poziomu technicznej złożoności (*TCF– The Technical Complexity Factor*)
- obliczenie punktów funkcyjnych (*FP – Function Points*)

$$FP = UFP * TCF$$

- wyznaczenie pracochłonności projektu na podstawie wyliczonej liczby punktów funkcyjnych
- określenie czasu trwania projektu
- określenie liczebności zespołu

## Wstępne punkty funkcyjne

(UFP – Unadjusted Function Points)

- klasyfikacja składników systemu
  - wejścia do systemu
  - wyjścia systemu
  - obsługa wewnętrznych zbiorów danych
  - obsługa zewnętrznych zbiorów danych
  - zapytania
- określenie kategorii złożoności funkcji
  - proste
  - średnie
  - złożone
- obliczenie sumy wag złożoności na podstawie tabeli

	Proste	Średnie	Złożone
<b>Wejścia</b>	3	4	6
<b>Wyjścia</b>	4	5	7
<b>Zbiory wewn.</b>	7	10	15
<b>Zbiory zewn.</b>	5	7	10
<b>Zapytania</b>	3	4	6

## Poziom technicznej złożoności systemu

(TCF– *The Technical Complexity Factor*)

- szacowanie wpływu czynników korygujących ( w skali 0, 1, 2, 3, 4, 5 – im większy wpływ tym więcej punktów)
  1. szybkość przesyłania danych
  2. rozproszenie przetwarzania
  3. wydajność
  4. stopień obciążenia systemu
  5. częstotliwość wykonywania transakcji
  6. ilość danych wprowadzanych on-line
  7. wydajność użytkownika końcowego
  8. aktualizacja danych w trybie on-line
  9. złożoność przetwarzania danych
  10. przenaszalność oprogramowania
  11. łatwość instalacji
  12. prostota obsługi systemu
  13. liczba lokalizacji stanowisk
  14. łatwość wprowadzania zmian
- obliczenie złożoności przetwarzania  $F$  (suma stopni wpływu poszczególnych czynników)
- wyznaczenie współczynnika korygującego, np.
$$TCF = 0,65 + (0.01 * F)$$

## **Szacowanie w metodykach zwinnych**

- w tradycyjnych metodykach dąży się do szacowania pracochłonności, która przekłada się na czas trwania
- w metodykach zwinnych szacuje się raczej wielkość i złożoność historyjek używając „abstrakcyjnych” mierników: dni idealnych lub punktów
- określa się zakres prac w kolejnych iteracjach na podstawie prędkości zespołu wyrażonej w dniach idealnych lub punktach pracy
- prędkość wyraża liczbę dni idealnych lub punktów, które zespół jest w stanie zaimplementować w ciągu jednej iteracji
- wykorzystuje się sprzężenie zwrotne z poprzednich iteracji do korekty parametru szybkości
- każdy zespół może szacować dni idealne lub punkty inaczej
- ważna jest nie tyle dokładność szacunków, co ich stabilność

## **Szacowanie w metodykach zwinnych - dni idealne**

- dni idealne – czas potrzebny do realizacji zadania przy założeniu, jest on poświęcony wyłącznie do realizacji zadania i wszystkie zasoby są dostępne
- nie uwzględnia się czynności pobocznych, np. spotkań, telefonów, przerwań, odczytywania i odpisywania listów, itd.
- czas szacowany w dniach idealnych odnosi się do możliwości zespołu jako całości, bez porównywania indywidualnych zdolności

## **Szacowanie w metodykach zwinnych - punkty**

- są miarą relatywną (względną)
- szacowanie w punktach polega na określeniu relatywnej wielkości historyjki w stosunku do pozostałych
- określają jak bardzo stworzenie funkcjonalności danej historyjki jest trudniejsze od innej (szacowanie przez analogię)
- stosowane skale punktowe
  - 0, 1, 2, 3, 5, ... – każdy kolejna liczba sumą dwóch poprzednich
  - 0, 1, 2, 4, 8, ... – każdy kolejna liczba podwojeniem poprzedniej

### **Podejścia do wyboru punktu odniesienia**

- wybranie najłatwiejszej do realizacji historyjki i przypisanie jej 1 punktu
- wybranie historyjki o średniej złożoności i przypisanie jej wartości ze środka używanej skali
- używanie puli historyjek wzorcowych



## **Porównanie jednostek szacowania w metodykach zwinnych**

### **Przewagi szacowania w punktach**

- wartość punktów nie zmienia się wraz ze wzrostem produktywności zespołu
- punkty mierzą wielkość i złożoność funkcjonalności i nie zależą od indywidualnej umiejętności i doświadczenia wykonawców
- szacowanie w dniach idealnych sprowadza się w istocie do historyjek rozbicia na zadania i szacowania czasu - jest bardziej skomplikowane

### **Przewagi szacowania w dniach idealnych**

- dni idealne są miarą bardziej naturalną, łatwiejszą na starcie i lepiej zrozumiałą dla osób spoza zespołu

## **Techniki szacowania w metodykach zwinnych**

- zmodyfikowana metoda delficka (raczej przy stosowaniu dni idealnych)
- Planning Poker (raczej przy stosowaniu punktów)
  - używa się kart ze skalą punktową
  - po wyjaśnieniach dotyczących analizowanej historyjki każdy wyciąga kartę z szacowaną przez niego liczbą punktów
  - osoby które wybrały najniższe i najwyższe oszacowania uzasadniają swój wybór i następuje dyskusja mająca na celu uzyskanie konsensusu
  - może być kilka rund
- grupowanie historyjek
  - historyjki zapisane na kartonikach są rozmieszczane na stole
  - każdą kolejną historyjkę porównujemy do pozostałych i próbujemy tworzyć grupy historyjek o podobnej wielkości
  - przypisujemy poszczególnym grupom historyjek punkty
  - w trakcie szacowania przesuwamy kartoniki na stole

# Zarządzanie ryzykiem

*(zarządzanie w warunkach występowania ryzyka)*

**Ryzyko** – *możliwość, prawdopodobieństwo, że coś się nie uda (zagrożenie).*

## Kategorie ryzyka

- **znane** – gdy wystąpienie zagrożenia jest prawie pewne;
- **przewidywalne** – gdy wystąpienie zagrożenia jest prawdopodobne i może być wcześniej wykryte/rozpoznane;
- **nieprzewidywalne** – gdy wystąpienia zagrożenia nie można przewidzieć z wyprzedzeniem (przypadki losowe).

## Zarządzanie ryzykiem

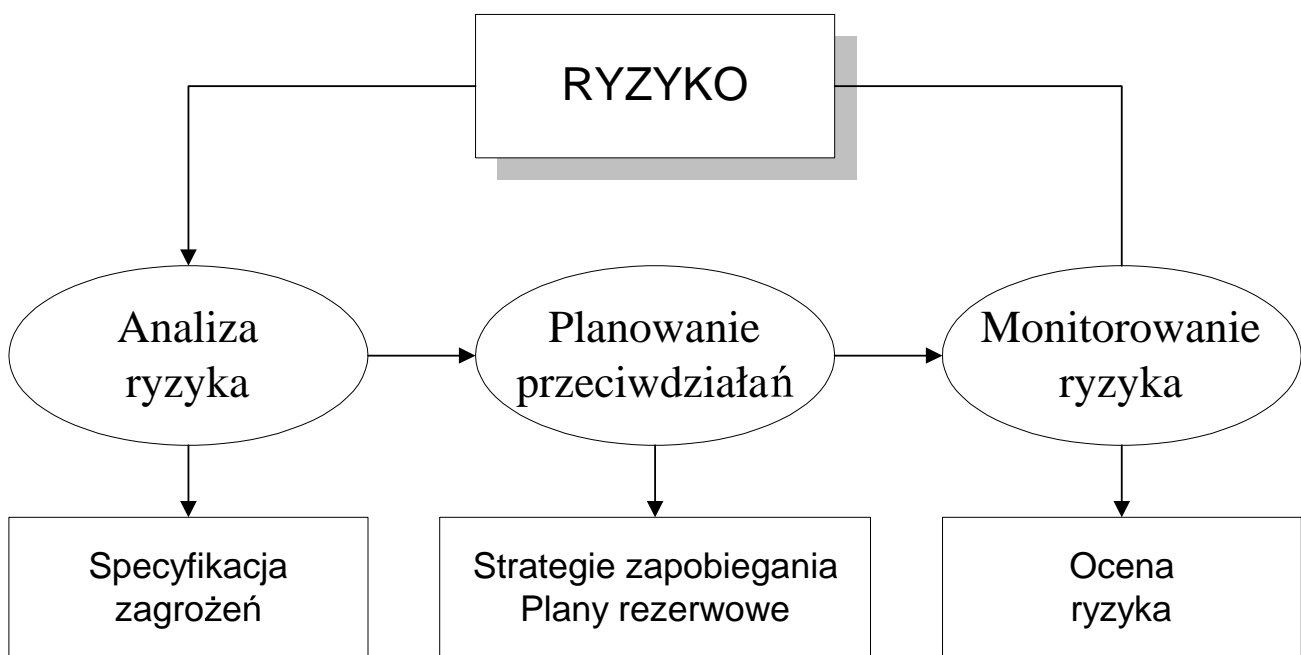
- zajmuje się dwiema pierwszymi kategoriami tzn. ryzykiem znanym oraz przewidywalnym
- koncentruje się na wczesnym wykrywaniu możliwego ryzyka i działaniach zapobiegawczych, nie zaś na usuwaniu skutków powstałych problemów

## Typowe źródła ryzyka

- charakter projektu
  - duży stopień złożoności
  - napięte (mało elastyczne) terminy
  - długi czas trwania projektu – duże prawdopodobieństwo znacznych zmian (technologii, pracowników, otoczenia, kierownictwa, prawa, itd.)
  - zbyt optymistyczny budżet
  - nieznana, nie wypróbowana nowa technologia
- środowisko zespołu wykonawczego
  - doświadczenie i umiejętności zespołu (merytoryczne, w metodykach prowadzenia projektu, w zarządzaniu, itd.)
  - motywacja i zaangażowanie zespołu
  - zdrowie fizyczne i psychiczne
  - koordynacja – duża liczba kooperantów (dostawców, podwykonawców itd.)
  - rozmieszczenie miejsc realizacji projektu
- środowisko klienta
  - zaangażowanie klienta w projekt (specyfikacja wymagań, zmiany wymagań, komunikacja)
  - poziom wiedzy użytkowników
  - organizacja klienta – liczba komórek objętych projektem
  - konieczność wprowadzenia reorganizacji
- czynniki zewnętrzne

# Metodyka zarządzania ryzykiem

- analiza ryzyka
  - identyfikacja zagrożeń
  - ocen ryzyka
  - specyfikacja zagrożeń (z przypisanymi wagami)
- planowanie przeciwdziałań
  - planowanie działań bezwarunkowych (w celu uniknięcia zagrożenia lub zredukowania jego skutków)
  - ustalenie procedury działania w przypadku wystąpienia zagrożenia
- monitorowanie ryzyka
- doskonalenie procesów zarządzania ryzykiem



# Metody identyfikacji ryzyka

- analiza źródeł ryzyka
  - charakter projektu
  - klient
  - wykonawcy
  - podwykonawcy
  - procesy zarządzania i tworzenia oprogramowania
  - technologia (sprzęt, oprogramowanie)
  - inne czynniki wewnętrzne i zewnętrzne
- wywiady, warsztaty, „burze mózgów”
- listy kontrolne – kwestionariusze oceny zagrożeń (ryzyka)
- informacja archiwalna z poprzednich projektów
- analogie do znanych przypadków
- eksperymenty lub testy (np. nowej technologii)

# Ocena ryzyka

- prawdopodobieństwo wystąpienia
  - parametry liczbowe
  - podział na kategorie np. prawdopodobieństwo bardzo wysokie, wysokie, średnie, niskie, bardzo niskie
- skutki wystąpienia
  - liczone w pracochłonności, koszcie, opóźnieniu, jakości produktu itp.
  - podział na kategorie np. skutki katastrofalne, duże, średnie, małe
- określenie wagi ryzyka
  - waga ryzyka = (prawdopodobieństwo) \* (skutek)
  - na podstawie macierzy wag np. ryzyko wysokie, średnie, niskie, nieistotne, itp.

## Prawdopodobieństwo

Skutek	Bardzo duże	Duże	Średnie	Małe	Bardzo małe
<b>Katastrofalny</b>	wysokie	wysokie	wysokie	średnie	niskie
<b>Duży</b>	wysokie	wysokie	średnie	niskie	nieistotne
<b>Średni</b>	średnie	średnie	niskie	nieistotne	nieistotne
<b>Mały</b>	średnie	niskie	niskie	nieistotne	nieistotne

- na podstawie graficznej reprezentacji wagi ryzyka

# **Specyfikacja ryzyka**

*(wynik analizy ryzyka)*

- nazwa ryzyka
- opis ryzyka
- prawdopodobieństwo pojawienia się ryzyka
- wpływ na projekt (skutki)
- waga ryzyka
- symptomy ryzyka
- czas występowania zagrożenia



# Planowanie zarządzania ryzykiem

*(planowanie przeciwdziałania zagrożeniom)*

## Strategie zarządzania ryzykiem

- **unikanie ryzyka** – eliminacja zagrożeń, np. przez usuwanie ich przyczyn, rezygnację z ryzykownych alternatyw
- **łagodzenie wpływu ryzyka**
  - redukcja prawdopodobieństwa wystąpienia, np. poprzez wprowadzenie redundancji, prototypowanie, wybór standardowych rozwiązań
  - redukcja wartości strat
- **przenoszenie ryzyka** – np. na inny etap, na klienta (ujmując to w umowie) lub inną instytucję (ubezpieczenie)
- **akceptacja ryzyka** – zgoda na skutki ryzyka
  - ustalenie procedury działania w przypadku wystąpienia zagrożenia
  - śledzenie poziomu ryzyka

## **Plan postępowania awaryjnego**

- opis zagrożenia, którego plan dotyczy
- metoda śledzenia ryzyka związanego z zagrożeniem
- przypisanie odpowiedzialności za śledzenie ryzyka i realizację planu
- warunki uruchomienia planu
- opis akcji awaryjnych
- przydział zasobów do wykonania planu

## **Monitorowanie ryzyka**

- wykrywanie pierwszych symptomów ryzyka
- uruchamianie akcji naprawczych
- monitorowanie działań związanych z ograniczaniem ryzyka
- ponawianie oceny ryzyka
- korygowanie specyfikacji zagrożeń oraz planów przeciwdziałania zagrożeniom

# **Prowadzenie projektu**

## **Zadania Kierownika Projektu**

- uruchomienie prac
- monitorowanie postępów
  - porównywanie aktualnego stanu prac z planem projektu
  - kontrola czasu (harmonogramu) i budżetu
  - identyfikacja odchyleń od planu
- podejmowanie kroków naprawczych
- monitorowanie zmian
- rozwiązywanie problemów
  - identyfikacja problemu
  - rozwiązywanie
  - raportowanie
- zarządzanie ryzykiem
  - analiza zdarzeń w poprzednim etapie
  - ponowne wykonanie analizy ryzyka dla bieżącego etapu
- zarządzanie zespołem
- kontrola podwykonawców
  - kontrola postępów
  - procedury akceptacji
  - zapewnienie jakości prac zleconych

# **Kontrola postępu prac**

## **Mechanizmy kontroli**

- raporty
- spotkania
- przeglądy produktów

Najważniejsza jest dokładność informacji !!!

Trzeba znaleźć punkt równowagi między poziomem kontroli i ryzykiem wystąpienia niekorzystnych zdarzeń

## **Działania w przypadku opóźnień**

- wykorzystanie rezerwy projektowej
- nadgodziny (zaakceptowanie większych kosztów)
- powiększenie zespołu
- przesunięcie zasobów
- częściowe zrównoleglenie zadań
- zaciąganie długu technicznego
- negocjacje z klientem
  - zmiana harmonogramu
  - ograniczenie zakresu
  - metoda triage – klasyfikacja wymagań
    - zrobić koniecznie
    - powinno się zrobić
    - ewentualnie zrobić

# **Czas realizacji a liczba pracowników**

## **Zależność nieliniowa**

### **Przyczyny**

- wraz z liczbą pracowników rośnie nakład czasu na komunikację
- niepodzielność zadań

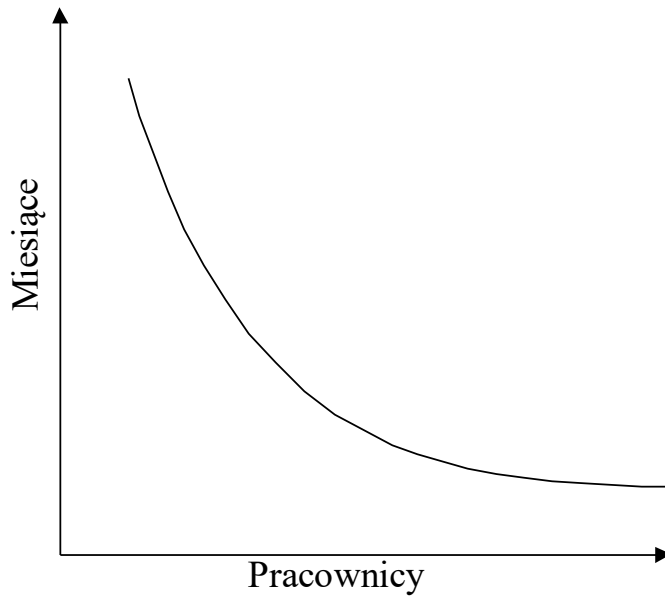
### **Dodatkowo trzeba uwzględnić, że**

- nowych wykonawców trzeba wdrożyć
- żeby ich wdrożyć trzeba oddelegować w tym celu innych wykonawców

### **Możliwy efekt (*Brooks*)**

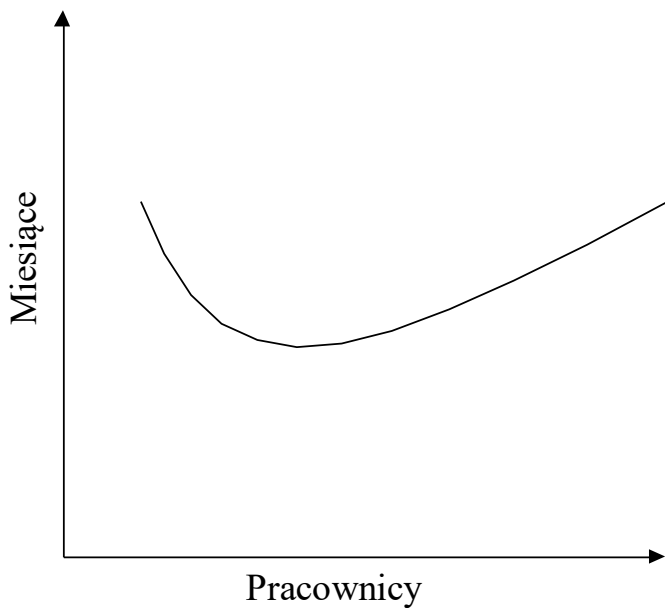
*“dodanie nowych pracowników do opóźnionego projektu zwiększa opóźnienie”*

## Czas realizacji a liczba pracowników



Projekty wymagające niewielkiej komunikacji

Projekty wymagające intensywnej komunikacji





## Dług techniczny

- metafora wprowadzone przez Warda Cunninghama w 1992 r.
- szybsze osiągnięcie celu i informacji zwrotnej poprzez świadome „stosowanie skrótów” (np. pozostawienie problemów z kodem)
- podobnie jak w przypadku długu finansowego dług techniczny wymaga spłaty odsetek – czasu na przywrócenie odpowiedniej jakości kodu
- mała ilość długu przyspiesza pracę tak długo, jak długo jest on spłacany odpowiednio szybko przez refaktoryzację (ulepszenie struktury istniejącego kodu)
- musimy uwzględnić spłatę zaciągniętego długu i liczyć się z tym, że zbyt długie zwleknięcie ze spłatą spowoduje narośnięcie kosztownych odsetek (coraz większych problemów z utrzymaniem i rozwijaniem kodu)
- potrzebna jest więc dogłębna analiza kosztu długu technicznego – trzeba być bardzo ostrożnym

# Mierzenie wykorzystania budżetu i postępu prac

## Metoda Wartości Wypracowanej (Earned Value Method)

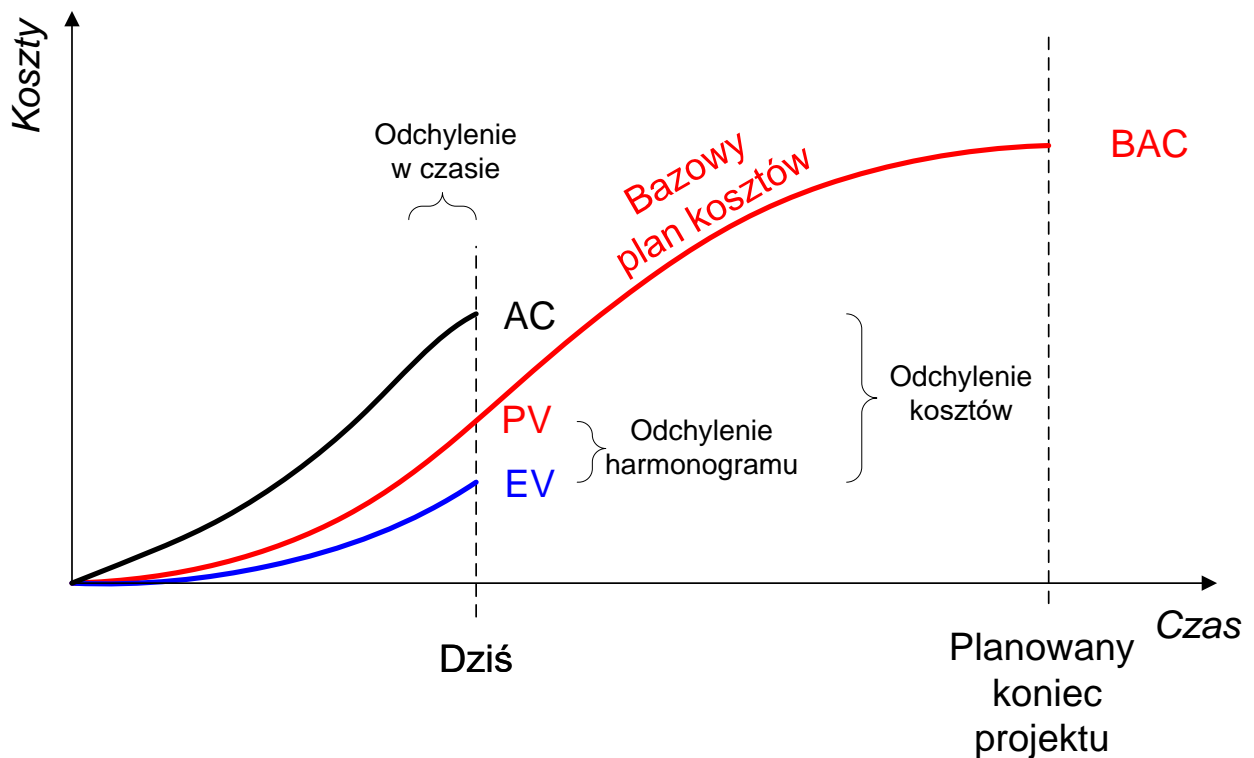
- Określa czy realizacja **harmonogramu** i wykorzystanie **budżetu** są zgodne z planem
- Porównanie **planowanych nakładów** pracy z faktycznie wypracowaną **wartością** i **poniesionymi w związku z tym kosztami**
- Zarówno **koszty**, jak i **postęp w realizacji harmonogramu** są wyrażone w wartościach pieniężnych

## Mierzenie wykorzystania budżetu i postępu prac (Metoda Earned Value)

### Informacje wejściowe (wyrażone w wartościach pieniężnych):

- **BCWS** (Budgeted Cost of Work Scheduled) – wartość planowana (Planned Value – **PV**) planowany koszt prac, które zgodnie z harmonogramem powinny być ukończone do danego dnia projektu
- **BCWP** (Budgeted Cost of Work Performed) – wartość wypracowana (Earned Value – **EV**) planowany koszt prac ukończonych do danego dnia projektu
- **ACWP** (Actual Cost of Work Performed) – koszt rzeczywisty  
(Actual Cost – **AC**) rzeczywisty koszt prac ukończonych do danego dnia projektu

# Metoda wartości wypracowanej



## Podstawowe wskaźniki

- odchylenie od planowanego harmonogramu  
 $SV = BCWP - BCWS$   
(wartość ujemna oznacza opóźnienie w stosunku do planu)
- odchylenie od planowanego kosztu  
 $CV = BCWP - ACWP$   
(wartość ujemna oznacza przekroczenie budżetu)

## **Wskaźniki wydajności realizacji projektu**

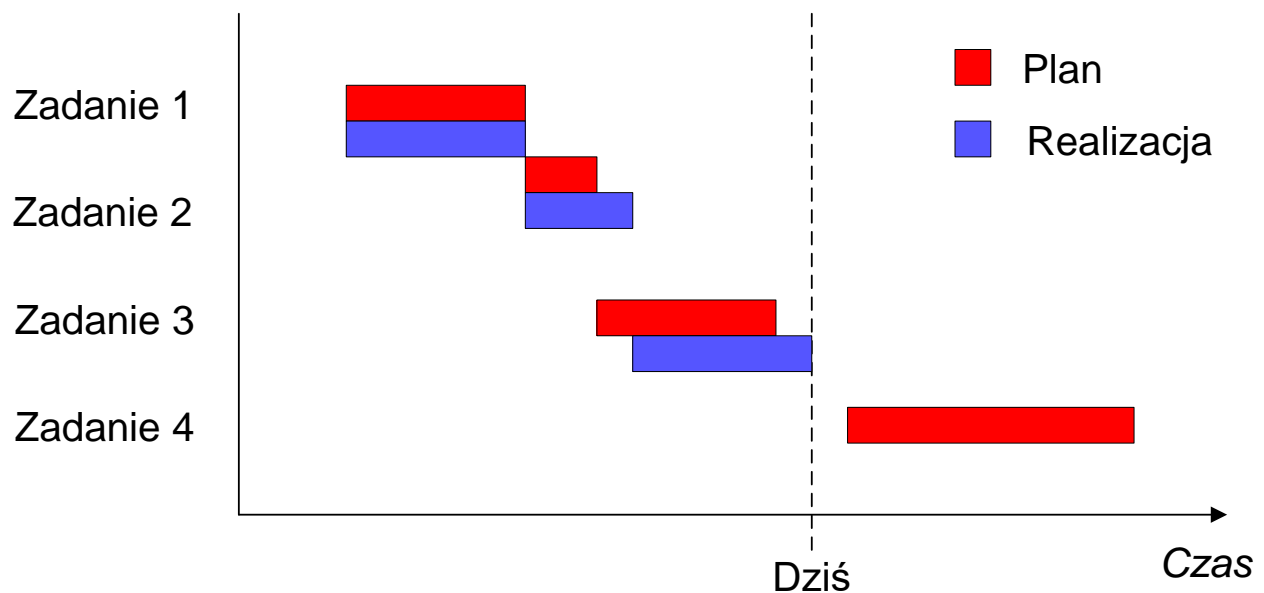
- wskaźnik wydajności realizacji harmonogramu  
$$SPI = BCWP / BCWS$$
- wskaźnik wydajności realizacji budżetu  
$$CPI = BCWP / ACWP$$
- wskaźnik wydajności budżetu dla pozostałych do wykonania prac w celu zmieszczenia się w budżecie  
$$TCPI = (BAC - BCWP) / (BAC - ACWP)$$

## **Szacowane koszty końcowe**

- EAC (Estimate at Completion)  
$$EAC = ACWP + (BAC - BCWP) / CPI$$

## Inne sposoby rejestracja postępów prac w podejściach tradycyjnych

- Monitorowanie kamieni milowych – planowanych i rzeczywistych dat ukończenia kluczowych produktów (etapów)
- Wykresy Gantta



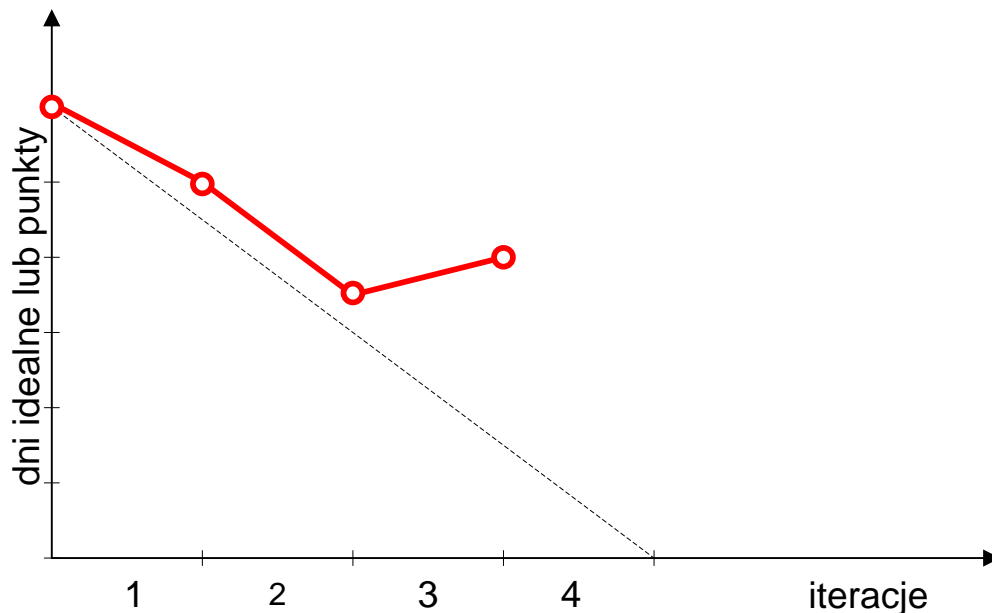
# Monitorowanie postępów prac w podejściach zwinnych

- codzienne spotkania 15 minutowe
  - przeważnie na stojąco
  - zespół organizuje je sam dla siebie
  - celem jest synchronizacja zadań, przygotowanie planu działania na najbliższy dzień i korygowanie na bieżąco wszelkich odchyłeń od planu
  - wykryte problemy są traktowane są jako dodatkowe zadania
  - zaznaczenie postępu na tablicy zadań i wykresie spalania
- przeglądy na koniec iteracji
  - demonstracja efektów pracy
  - zespół dostaje informację zwrotną
  - efektem może być zgłoszenie potrzeb nowych funkcjonalności, zmiana priorytetów (może się zmienić rejestr produktu)
  - zaznaczenie postępu na wykresie spalania wydania

# Rejestracja postępów prac wydania

- **Wykres spalania liniowy**

- oś pozioma – czas mierzony w liczbie iteracji
- oś pionowa – suma dni idealnych (lub punktów) historyjek, które pozostały do realizacji

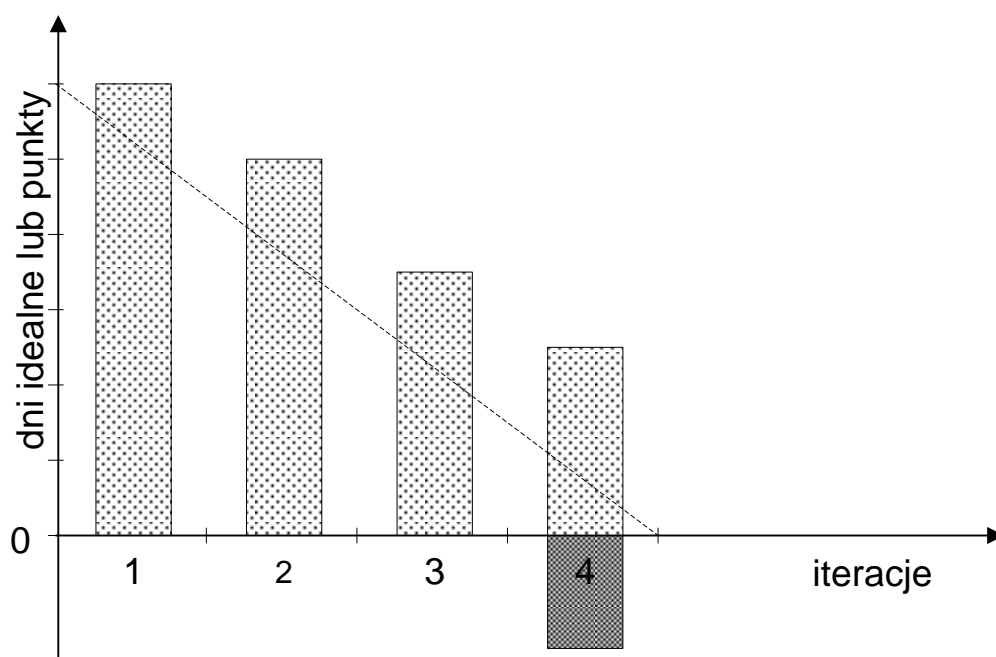


**Niedogodności** – trudno zilustrować zmiany w zakresie wydania (gdy w międzyczasie dodano lub usunięto pewne funkcjonalności)



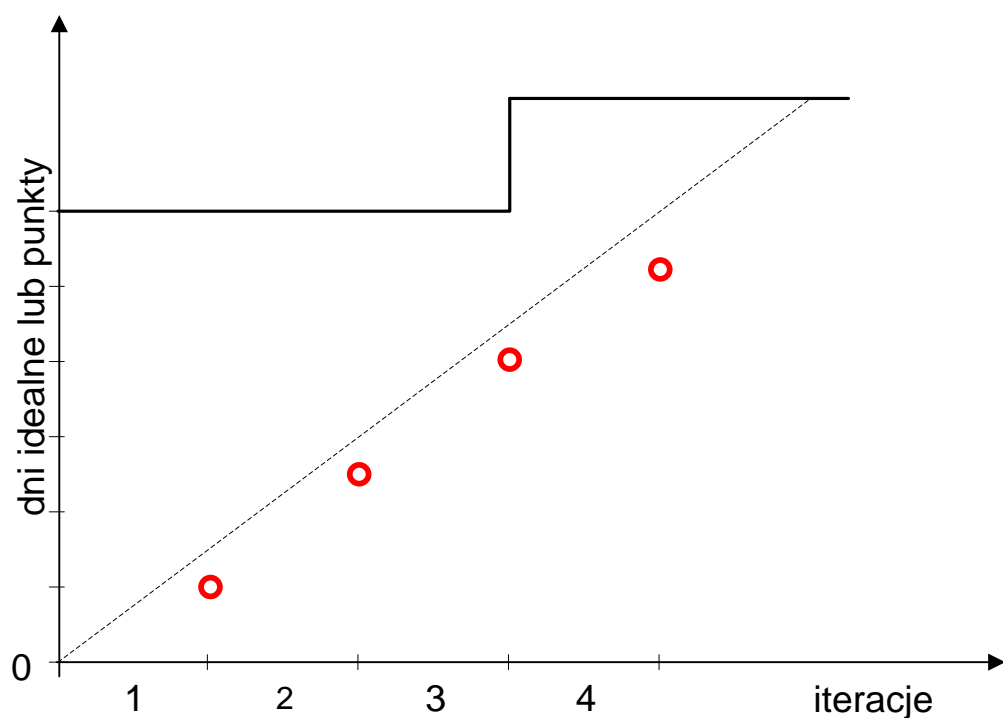
## Rejestracja postępów prac wydania

- Wykres spalania liniowy
- **Wykres spalania słupkowy**
  - oś pozioma – czas (liczba iteracji)
  - słupek górny – ile mamy do zrobienia na początku każdej iteracji ( w dniach idealnych lub punktach)
  - słupek dolny – reprezentuje zmiany w zakresie



# Rejestracja postępów prac wydania

- Wykres spalania liniowy
- Wykres spalania słupkowy
- **Wykres rozpalania** (oś pionowa – ile wykonaliśmy)



# Rejestracja postępów prac w iteracji

- **Tablica zadań**

- historyjki i ich zadania reprezentowane w postaci karteczek przyklejonych do tablicy
- w trakcie realizacji iteracji karteczki z zadaniami są przemieszczane między kolumnami („Do zrobienia”, „W toku”, „Gotowe”)

HISTORYJKI	Do zrobienia	W toku	Gotowe
HIST. A	Z2 Z11	Z3	Z13 Z1
HIST. B	Z8 Z10	Z9	Z4
HIST. C	Z5 Z6 Z7 Z12		

# Rejestracja postępów prac w iteracji

- **Wykres spalania**

- oś pozioma – czas mierzony w dniach
- oś pionowa – suma roboczogodzin dla realizacji zadań które pozostały do wykonania

