

# Inżynieria oprogramowania (Software Engineering)

## Wykład 3

Studium wykonalności

Definicja wymagań

# Studium wykonalności (feasibility study)

Prowadzone przed rozpoczęciem projektu, krótkie, niekosztowne badanie w celu identyfikacji:

- ▶ Klientów
- ▶ Zakresu
- ▶ Potencjalnych korzyści
- ▶ Niezbędnych zasobów: personel, czas, sprzęt itp.
- ▶ Potencjalnych przeszkód: Gdzie są zagrożenia? Jak mogą być zminimalizowane?

# Studium wykonalności

- ▶ Studium wykonalności prowadzi do decyzji w rodzaju:
  - ▶ Śmiało!
  - ▶ Nie idź tędy!
  - ▶ Pomyśl jeszcze raz!
- ▶ W realizacji projektów produkcyjnych studium wykonalności często prowadzi do wniosku budżetowego.
- ▶ W projektach badawczych studium wykonalności jest często w formie propozycji albo wniosków.

# Klient

- ▶ Pojęcie wieloznaczne:
  - ▶ Obywatel starożytnego Rzymu?
  - ▶ Ubogi szlachcic uzależniony od możnowładcy?
  - ▶ Osoba zwracająca się o wsparcie materialne i niematerialne do instytucji pomocy i integracji społecznej?
  - ▶ Osoba nabywająca towar lub usługę, konsument?
- ▶ W projektach IT ma sens ostatnie znaczenie terminu.

# Zakres projektu

- Co stanowi granice projektu?
  - Statyczne/dynamiczne strony internetowe z otwartym/ograniczonym dostępem w sieci lokalnej/globalnej [Web Profiler]
  - Używane przez publiczność czy tylko przez grono studentów i pracowników [Digital Collections]
  - Zróżnicowane formaty danych [Informacje prawne]
  - Tysiące czujników [Data mining]
  - Wsparcie dla Windows, Mac, Unix [SALSA]

# Potencjalne korzyści

- Dlaczego wykonujemy dany projekt?
  - Tworzenie produktu zbywalnego.
  - Poprawa efektywności organizacji.
  - Kontrola systemu, który jest zbyt skomplikowany, aby sterować ręcznie.
  - Dodanie nowych lub ulepszonych usług.
  - Minimalizacja zagrożenia bezpieczeństwa.
  - Inne powody?

# Zasoby

- Personel:
  - Ile specjalistów? Ile godzin pracy w tygodniu?  
Jakie umiejętności będą wymagane?
- Czas:
  - Wersja beta, testy akceptacyjne, dokumentacja, prezentacja wersji finalnej.
- Sprzęt i oprogramowanie:
  - Jakie specjalne potrzeby?
- Klient:
  - Czy klient jest wystarczająco dostępny i będzie pomocny?

# Przeszkody

- Okres rozruchu:
  - Stworzenie zespołu, planowanie spotkań, nabywanie oprogramowania, uczenia się nowych systemów, ...
- Względy biznesowe:
  - Licencje, tajemnice handlowe, ...
- Ambicja:
  - Nic nie wskazuje na koniec pracy.
- Zmieniające się okoliczności.
- Co jeszcze?



# Jak zminimalizować ryzyko projektowe?

- Kilka docelowych poziomów funkcjonalności:
  - Wymagany,
  - Pożądany,
  - Opcjonalny.
- Widoczny proces wytwarzania oprogramowania:
  - Pośrednie rezultaty.
- Dobra komunikacja w zespole i asysta.

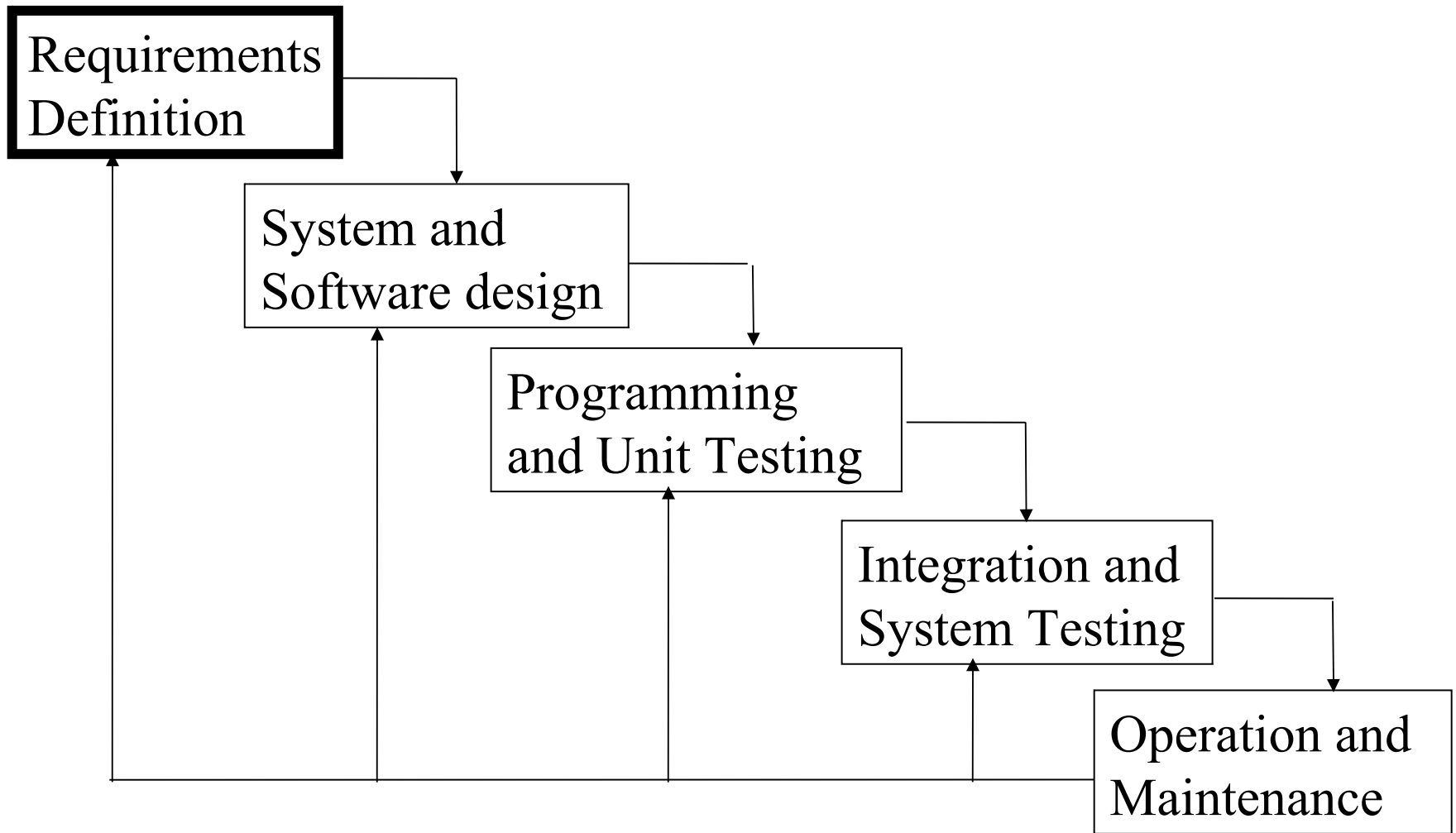
**Good processes lead to good software!**

**Good processes reduce risk!**

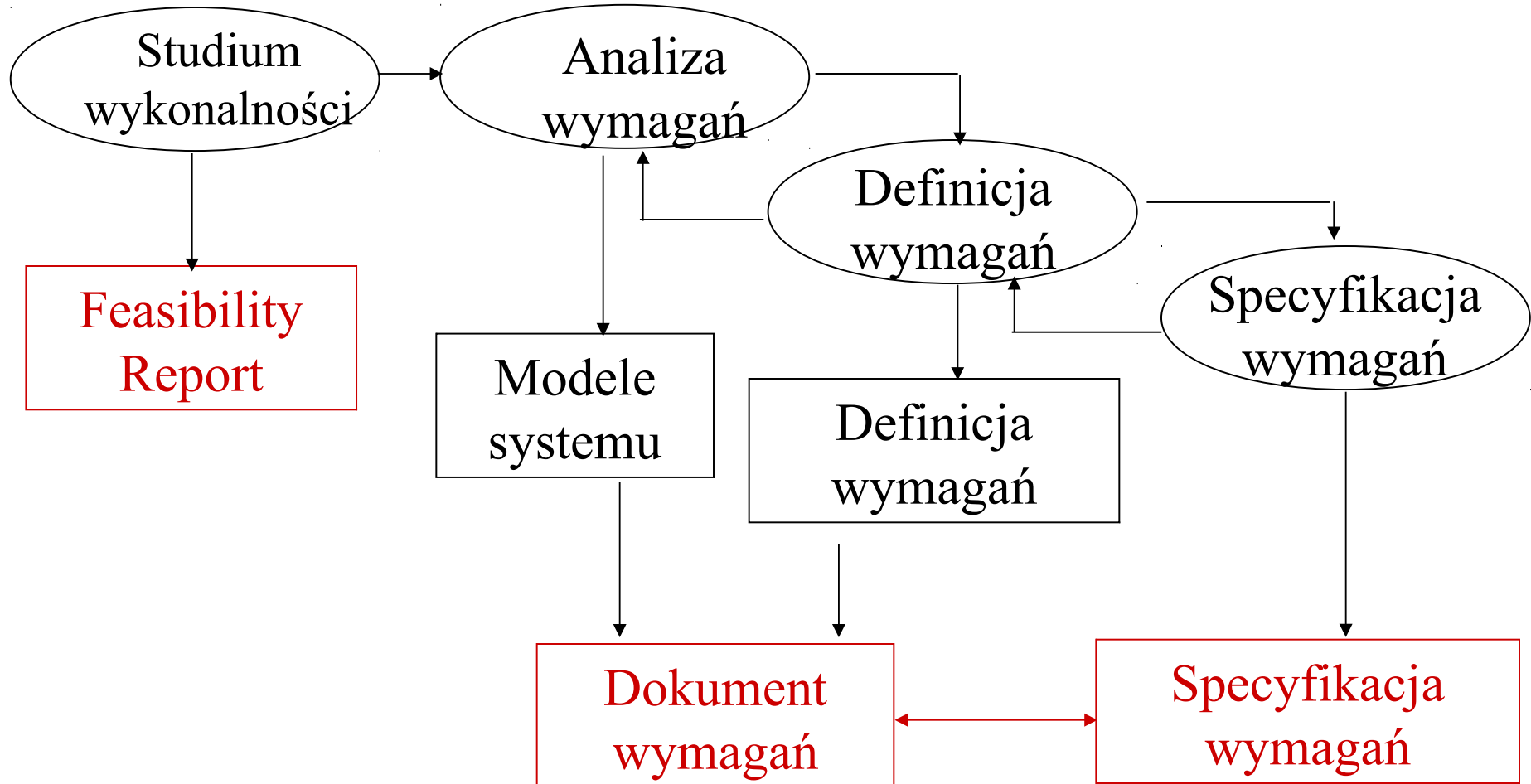
# Raport wykonalności

- Dokument pisemny:
  - W audyencji generalnej: klient, zarządzania finansowe, zarządzanie techniczne, itp.
  - Krótki na tyle, że każdy przeczyta.
  - Długi na tyle, że nie ważne tematy mogą być pomijane.

# Definicja i analiza wymagań



# Proces inżynierii wymagań



# Definicja wymagań

- Jest to wysokopoziomowy, abstrakcyjny opis wymagań:
  - Określa zewnętrzne zachowanie systemu.
  - Jest zrozumiały dla klienta, menadżerów i użytkowników.
  - Powinien dokładnie odzwierciedlać to, co klient chce:
    - Usługi, które system zapewni.
    - Ograniczenia w ramach których będzie działać.

# Wymaganie funkcjonalne

- Udokumentowana potrzeba określonego produktu czy usługi, albo sposobu ich działania.
- Stwierdzenie identyfikujące potrzebne cechy, możliwości, charakterystyki lub jakość oprogramowania, aby było ono wartościowe i pożyteczne dla użytkownika.
- Stwierdzenie, jakie usługi ma oferować system, jak ma reagować na określone dane wejściowe oraz jak ma się zachowywać w określonych sytuacjach. W niektórych wypadkach wymagania funkcjonalne określają, czego system nie powinien robić.

# Wymagania niefunkcjonalne

- Wynikają pośrednio ze zgłoszonych potrzeb użytkownika lub bezpośrednio z wymagań funkcjonalnych.
- Realizacja wymagania niefunkcjonalnego może wymagać realizacji kilku zależnych wymagań niefunkcjonalnych.
- Wymagania niefunkcjonalne powstają w wyniku zaprojektowania i uzgodnienia przez architektów produktu specyficznej realizacji wymagań funkcjonalnych.

# Specyfikacja wymagań

- Dokument opisujący wymagania na oprogramowanie.
- Dobrze napisana specyfikacja wymagań ułatwia komunikację pomiędzy klientem i dostawcą.
- IEEE Recommended Practice for Software Requirements Specifications



# Analiza wymagań

- Po wykonaniu identyfikacji wymagań należy przeprowadzić ich analizę celem uszczegółowienia, ewentualnego odrzucenia wymagań sprzecznych lub nieprawidłowych oraz określenia priorytetu i ryzyka.
- Analiza wymagań funkcjonalnych umożliwia zidentyfikowanie i opisanie pożądanego zachowania systemu oprogramowania.

# Ewolucja wymagań

- Jeżeli definicja wymagań jest błędna, projekt systemu zakończy się porażką.
- Dla systemów złożonych, zrozumieniu potrzeb klienta zawsze będzie się poprawiać stopniowo.
  - Definicje wymagań muszą ewoluować.
  - Dokumentacja wymagań musi być aktualna (wersjonowanie wymagań).