

# Projektowanie Systemów Informatycznych

Bogumiła Hnatkowska

# PSI – formy

- Wykład – zakończony egzaminem: 23.06, 30.06;  
Do egzaminu można przystąpić bez zaliczenia projektu  
Możliwość zwolnienia z egzaminu z oceną 5.0 w  
przypadku, gdy projekt oceniony na 5.5

## PSI – formy, c.d.

- Projekt – zaliczanie fazowe; celem jest projekt architektury oraz implementacja zgodnie z projektem wybranych funkcjonalności i cech нефunkcjonalnych złożonego systemu informatycznego

# Literatura podstawowa

- L. Bass, P. Clements, R. Kazman, Architektura oprogramowania w praktyce. Wydanie II, Helion 2011.
- N. Rozanski, E. Woods, Software Systems Architecture: working with stateholders using viewpoints and perspectives, Second Edition, Pearson Education 2012.
- P. Clements, R. Kazman, M. Klein, Architektura oprogramowania. Metody oceny oraz analiza przypadków, Helion 2003.
- R. C. Martin, Clean Architecture, Prentice Hall 2018
- D. Trowbridge at al. Integration patterns. Microsoft 2004 (patterns & practices)
- Materiały przygotowane przez prowadzącego kurs.

# Cele i zakres wykładu

- Zrozumienie roli modelowania w wytwarzaniu oprogramowania.
- Zapoznanie z procesem projektowania złożonych systemów informatycznych i stosowanymi w tym zakresie rozwiązaniami.
- Zapoznanie z metodami oceny jakości projektów informatycznych i ocena jakości artefaktów powstających podczas wytwarzania oprogramowania.

# Projekt

- **Projekt (czynność)** – czasowy wysiłek podejmowany do stworzenia unikalnego produktu lub usługi [pmbok]
- **Projekt (artefakt)** – plan, szkic czegoś [słownik j. pol.]
- **Projekty w Inżynierii Oprogramowania:**
  - Projekt architektury systemu
  - Projekt architektury oprogramowania
  - Projekt szczegółowy

# System informatyczny

- Zbiór powiązanych ze sobą elementów, którego funkcją jest przetwarzanie danych przy użyciu techniki komputerowej
- Typowe komponenty systemu informatycznego:
  - sprzęt
  - oprogramowanie
  - bazy danych
  - interfejsy (np. innymi programami)
  - procedury korzystania z systemu informatycznego, instrukcje robocze itp.

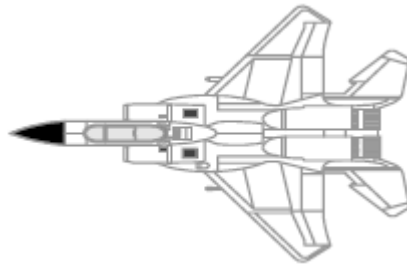
# System informatyczny, cont.

- Termin „system informatyczny” stosuje się często do systemów wspomagających funkcjonowanie (zarządzanie) przedsiębiorstw
- Przykłady zintegrowanych systemów informatycznych:
  - ERP (enterprise resource planning),
    - Magazynowanie
    - Księgowość
    - Finanse
    - Zarządzanie relacjami z klientami
    - ..



# Model

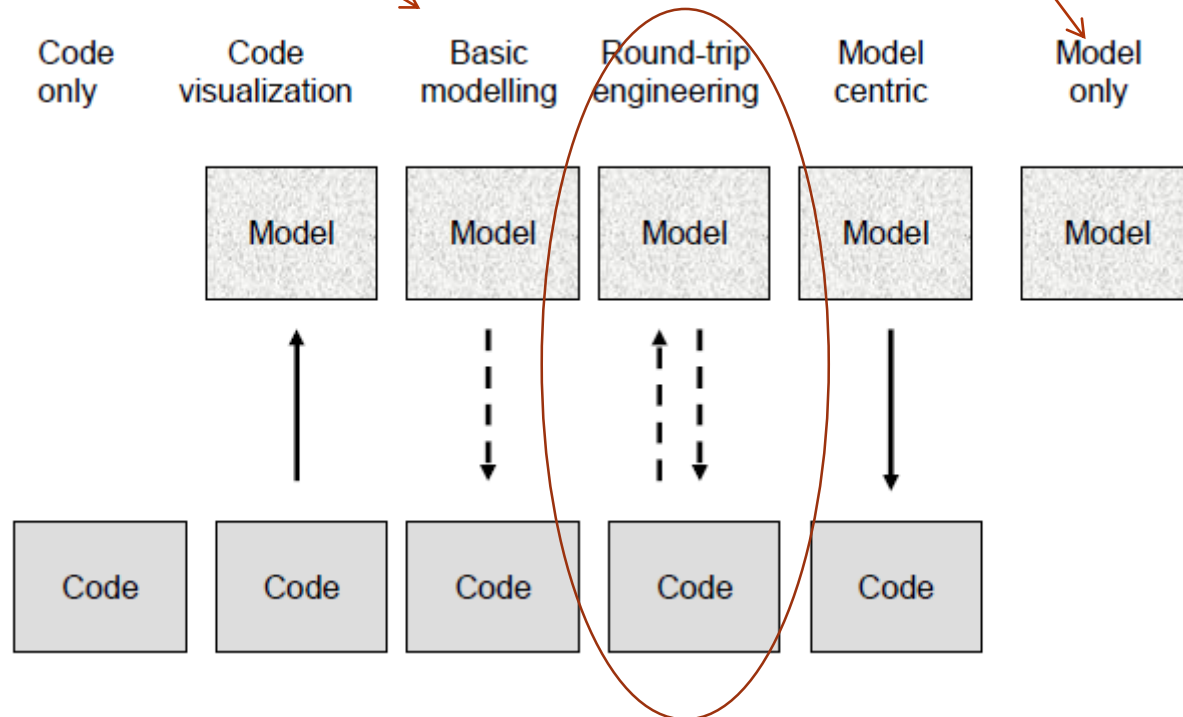
- Model – uproszczona reprezentacja systemu (oprogramowania) lub jego otoczenia na przyjętym poziomie abstrakcji wykonana w określonym celu, przedstawiona z wykorzystaniem języka modelowania; semantycznie zamknięta abstrakcja systemu



- Korzyści z modelowania:
  - Wizualizacja systemu (polepsza zrozumienie)
  - Dokumentacja decyzji projektowych
  - Możliwość oceny rozwiązania (wybranych cech) przed jego zbudowaniem
- Modelowanie vs projektowanie

# Rola modelowania w IO

- Fowler: szkic, blueprint, język programowania
- Brown:



# Pojęcia podstawowe

- Celem firm IT jest **zaspokojenie potrzeb użytkowników** przez dostarczenie im działającego systemu **wysokiej jakości** w ramach ustalonego **budżetu i harmonogramu**
- Realizacja celu jest możliwa z wykorzystaniem **inżynierii oprogramowania** – **systematyczne podejście** do wytwarzania, działania, pielęgnacji i wycofania oprogramowania
- Systematyczne podejście przejawia się wykorzystaniem **metodyk wytwarzania oprogramowania** (zbiór zdefiniowanych procesów, dotyczących różnych obszarów IO)

# Pojęcia podstawowe

- **Metoda** – świadomie stosowany sposób postępowania mający prowadzić do osiągnięcia zamierzonego celu [słownik j. polskiego]
- **Metodyka** – zbiór zasad, sposobów wykonywania określonej pracy albo osiągnięcia określonego celu; szczegółowe normy postępowania właściwe danej nauce [słownik wyrazów obcych].
- **Metodyka** – zestaw pojęć, modeli, technik i sposobów postępowania służący do analizy dziedziny stanowiącej przedmiot projektowanego systemu oraz do projektowania pojęciowego, logicznego i fizycznego. Metodyka jest powiązana z notacją służącą do dokumentowania wyników faz projektu [Subieta].
- **Metodologia** – nauka o metodach badań naukowych stosowanych w danej dziedzinie wiedzy [słownik języka polskiego]; wyraz praktycznie nie występuje poza tekstami naukowymi, a typowy jego kontekst to metodologia badań (lub nazwa dyscypliny naukowej)

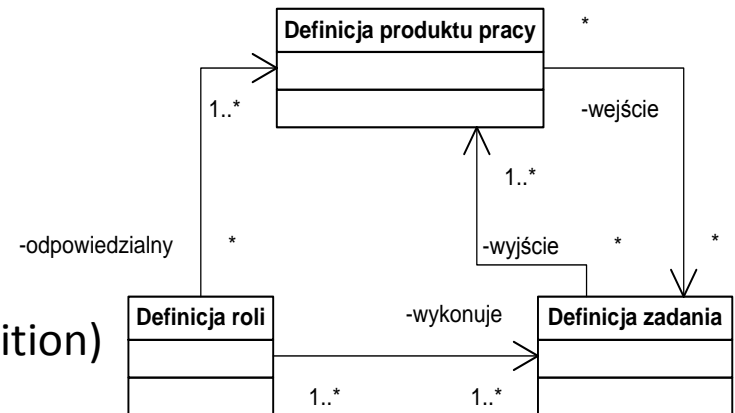
# Opis metodyki – podstawowe pojęcia

- Specyfikacja Meta-modelu Inżynierii Procesów Systemu i Oprogramowania (Software & Systems Process Engineering Meta-Model Specification, SPEM 2.0)

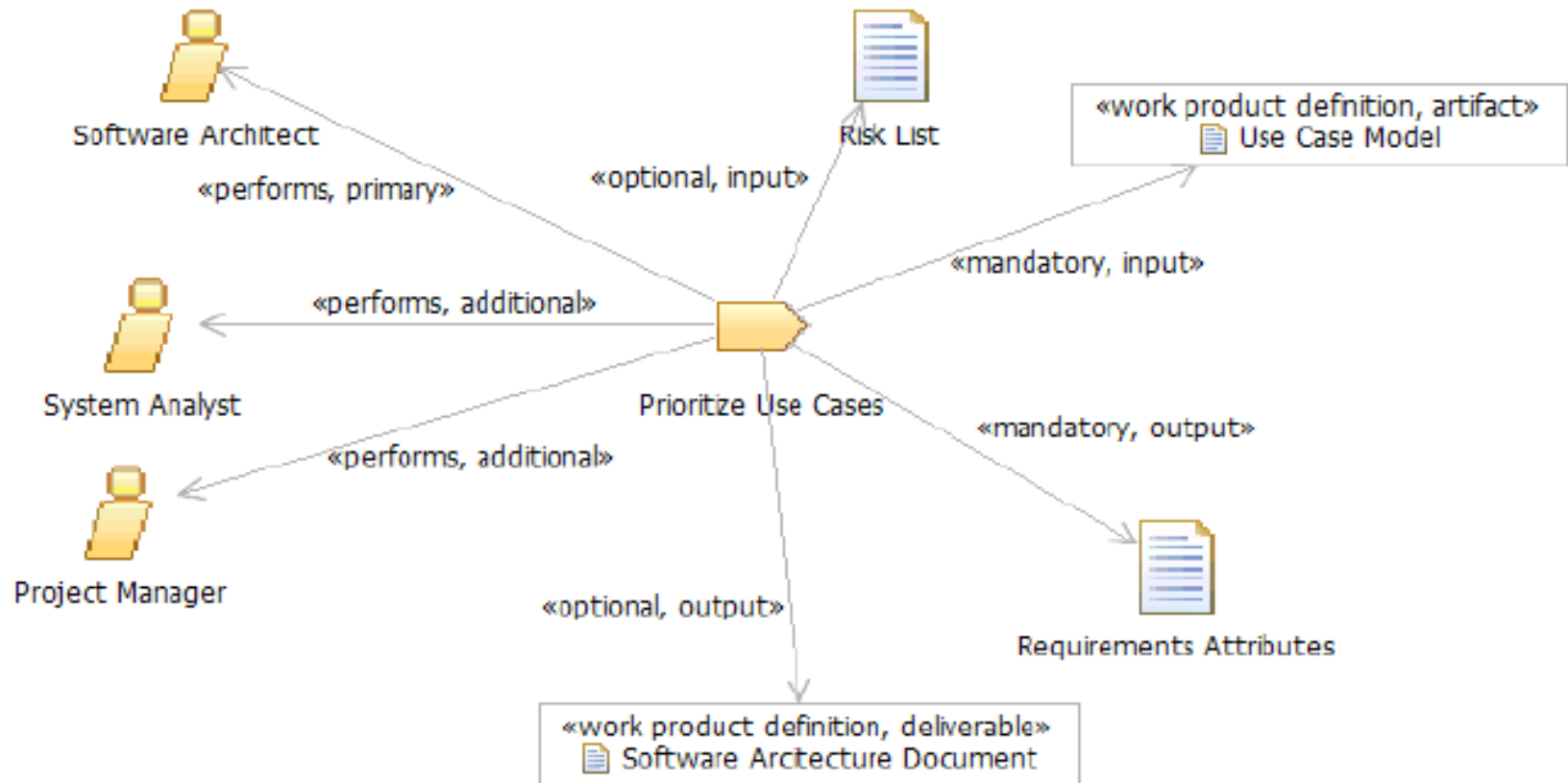
- **Definicja roli (Rola)** (ang. role definition) reprezentuje jednostkę lub zespół, która wykonuje określone aktywności i jest odpowiedzialna za pewne artefakty.

- **Definicja zadania (zadanie)** (ang. task definition) opisuje czynności podejmowane przez rolę w celu wyprodukowania określonych (znaczących) rezultatów.

- **Definicja produktu pracy (Artefakt lub produkt pracy)** (ang. work product definition) jest wynikiem pracy pewnej roli w ramach określonego zadania:
  - dokument
  - model
  - element modelu
  - kod źródłowy



# Opis metodyki – podstawowe pojęcia, c.d.



# Własności metodyk

- Zakres (scope) metodyki
- Rozmiar metodyki (size)
- Dokładność metodyki (ceremony)
- Waga metodyki (weight)
- Widzialność (visibility)

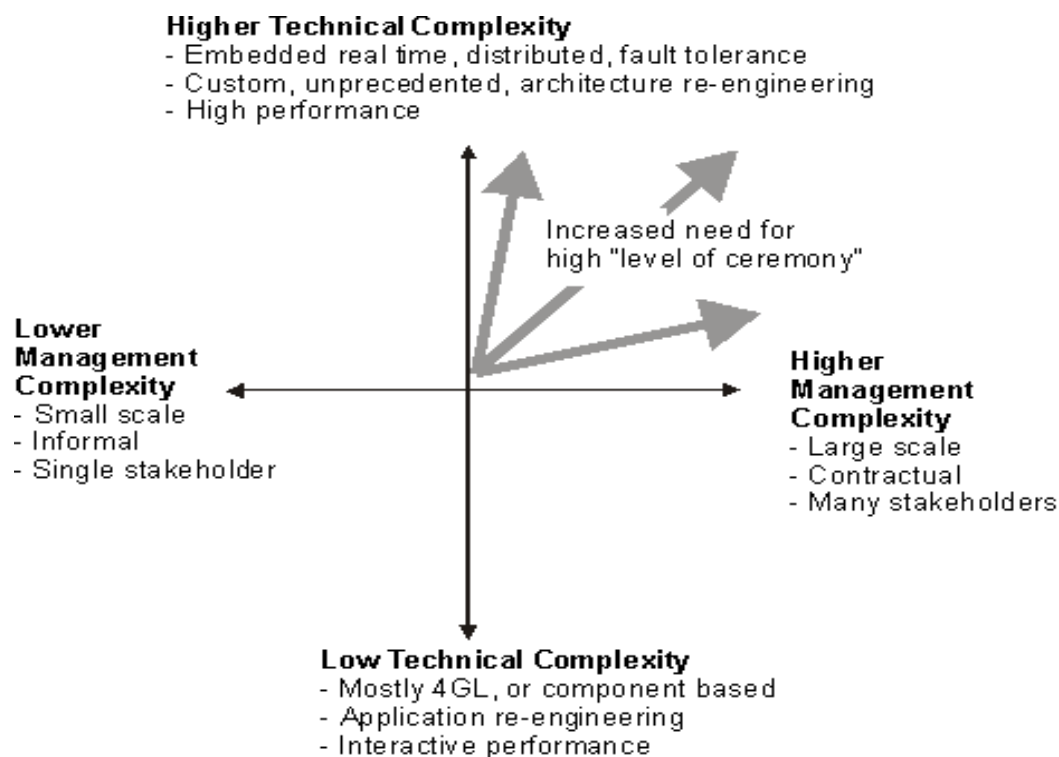
# Klasyfikacja metodyk

- Klasyfikacja metodyk ze względu na wagę:
  - ciężkie (tradycyjne), np. USDP, RUP, Enterprise RUP
  - lekkie (zwinne), np. Scrum, XP, Crystal, Agile UP, Agile RUP
- Klasyfikacja metodyk ze względu na rodzaj procesów cyklu życia, które metodyka obejmuje:
  - metodyki zarządcze, np. SCRUM, Prince 2
  - metodyki wytwórcze, np. USDP



# Parametry wpływające na dobór metodyki

- Cięższe metodyki dla: (a) wieloosobowych zespołów (b) dużych projektów (c) projektów o zastosowaniach krytycznych (d) projektów kontraktowych



# Metodyka stosowana w ramach projektu

- Unified Process z przedstawicielami:
  - Unified Software Development Process (USDP), twórcy UML
  - **OpenUP, projekt Eclipse → szablony dokumentów, wybrane aktywności w obrębie dyscyplin**
  - Rational Unified Process (RUP)
  - Enterprise Unified Process (EUP), Ambler
  - Agile Unified Process (AUP), Ambler
- Charakterystyka
  - Generyczny framework
  - Przedstawiciele współdzielą cykl życia

# Unified Process – struktura procesu

**FAZY**

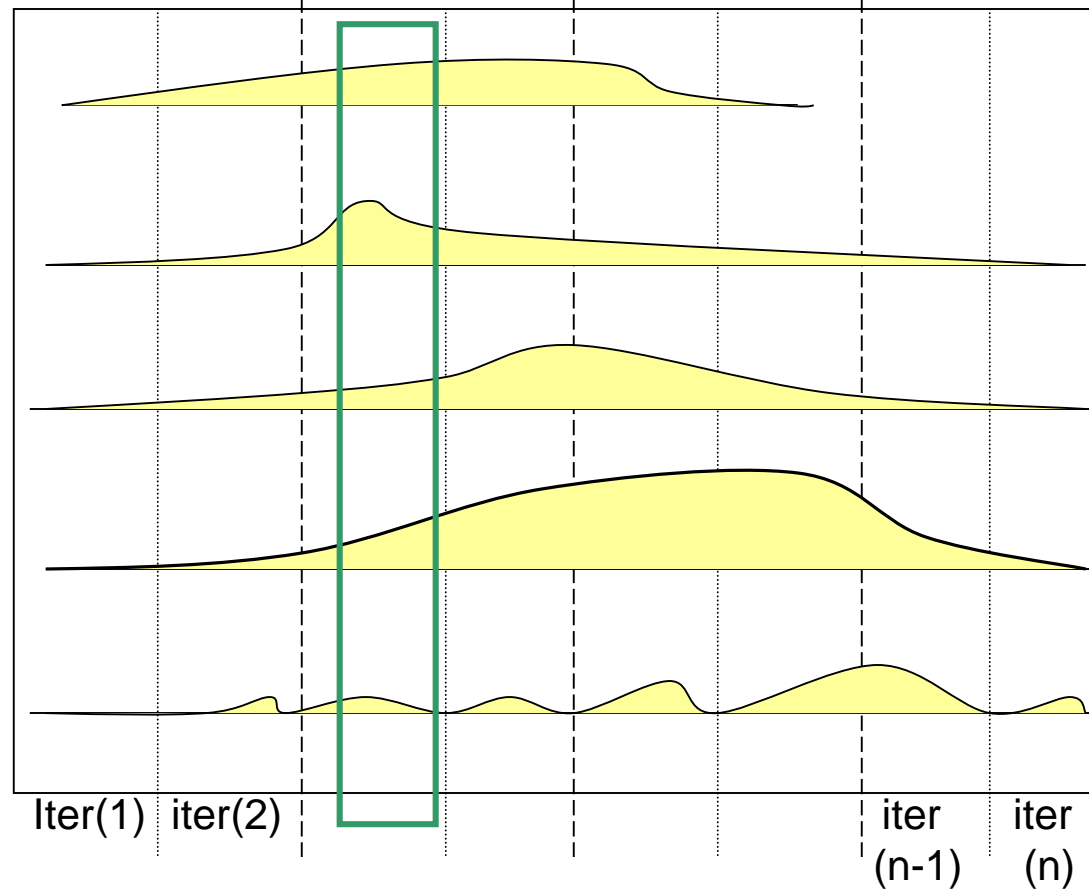
Zapoczątkowania  
Inception

Opracowania  
Elaboration

Konstrukcji  
Construction

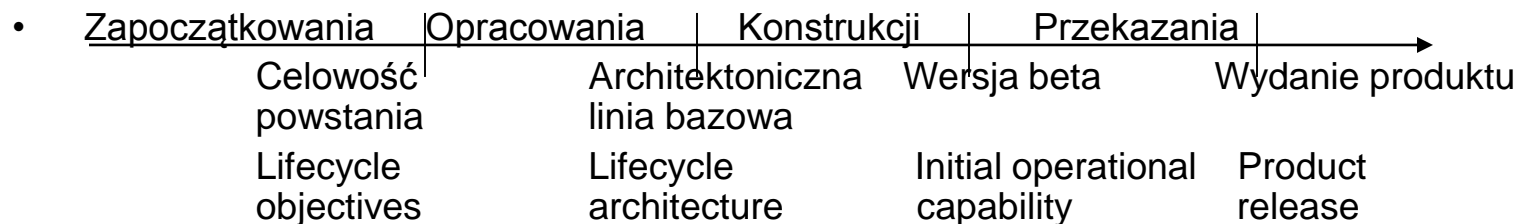
Przekazania produktu  
Transition

Dyscypliny



# Unified Proces – fazy i kamienie milowe

FAZA:



**Table 2: The focus of RUP phases**

Phase	Inception	Elaboration	Construction	Transition
Key question	Should we build it?	Can we build it?	Are we building it?	Have we delivered it?
Focus	Scope	Risk	Functionality	Delivery

**Risk reduction with the RUP Phase Plan, [Mark Aked](#)**

# Unified Process – struktura procesu

- Dyscyplina – pakiet grupujący zadania wykonywane w ramach danego etapu wytwórczego, np. specyfikacja wymagań, analiza, projekt etc.
- Zadania mogą być grupowane w większe struktury – aktywności
- Kolejność wykonywania aktywności w obrębie iteracji i/lub zadań w ramach dyscypliny jest pokazana za pomocą przepływów prac (workflow).

# Dyscypliny

## USDP

- [Modelowanie biznesowe]
- Specyfikacja wymagań
- Analiza
- **Projekt**
- Implementacja
- Testy

Elementy pogrubione –  
zakres objęty wykładem;  
kursywą - przypomnienie

## OpenUP

- Specyfikacja wymagań
- **Architektura**
- Wytwarzanie
- Testy
- Zarządzanie projektem

# Dyscypliny

## RUP

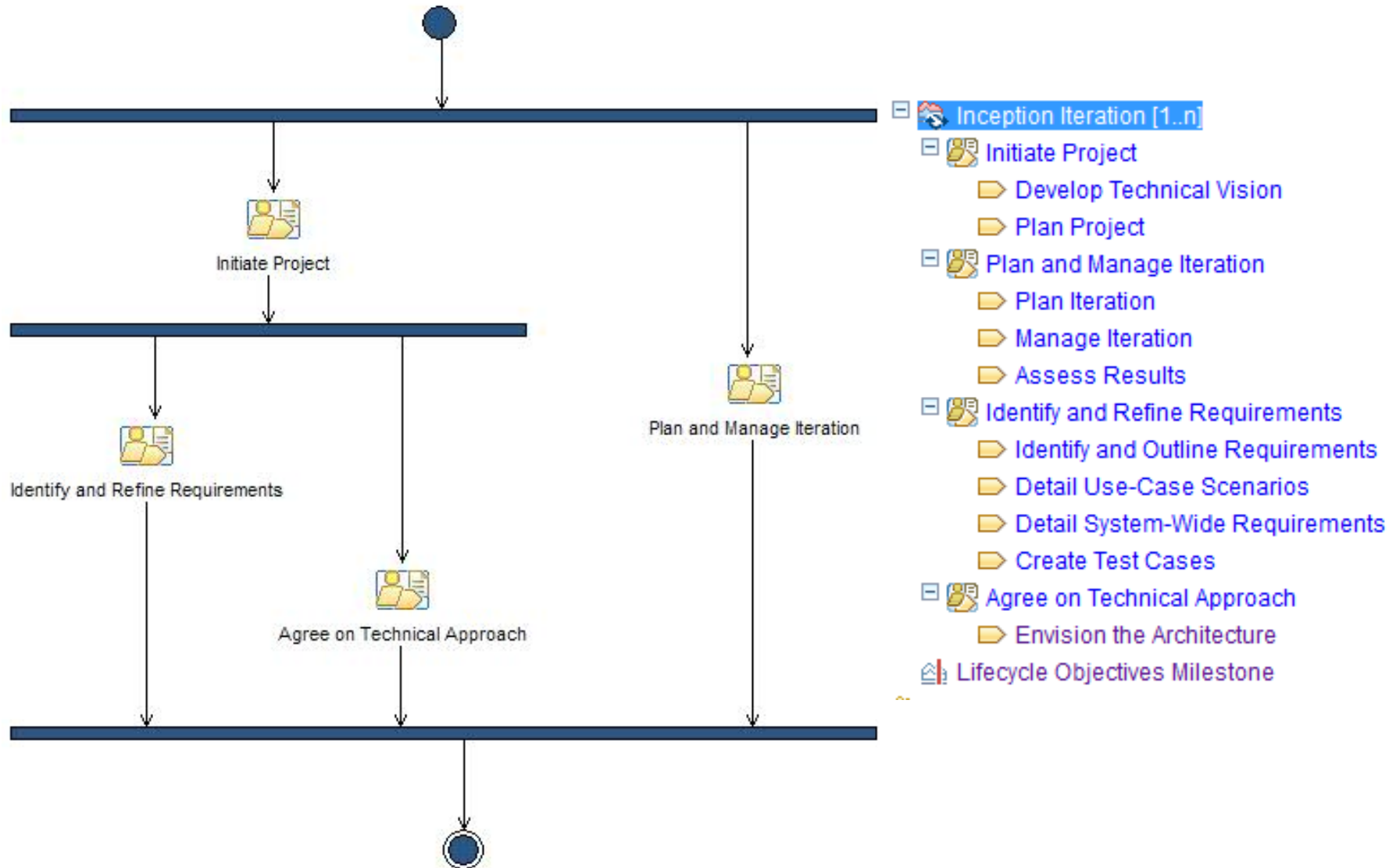
- Modelowanie biznesowe
- Specyfikacja wymagań
- **Analiza i projekt**
- Implementacja
- Testy
- Rozmieszczenie
  
- Zarządzanie konfiguracją i zmianami
- Zarządzanie projektem
- Środowisko

## AUP

- **Modelowanie**
- Implementacja
- Testy
- Rozmieszczenie
  
- Zarządzanie konfiguracją i zmianami
- Zarządzanie projektem
- Środowisko

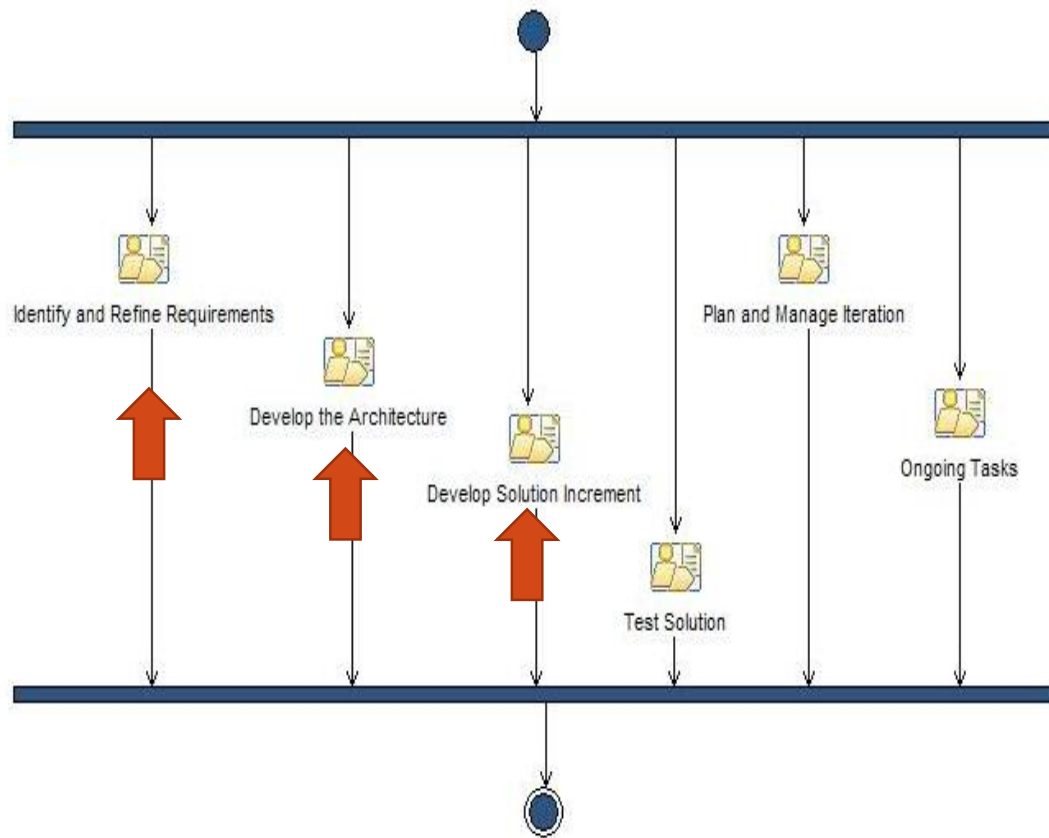


# OpenUP – przepływ prac (aktywności) w iteracji fazy Zapoczątkowania





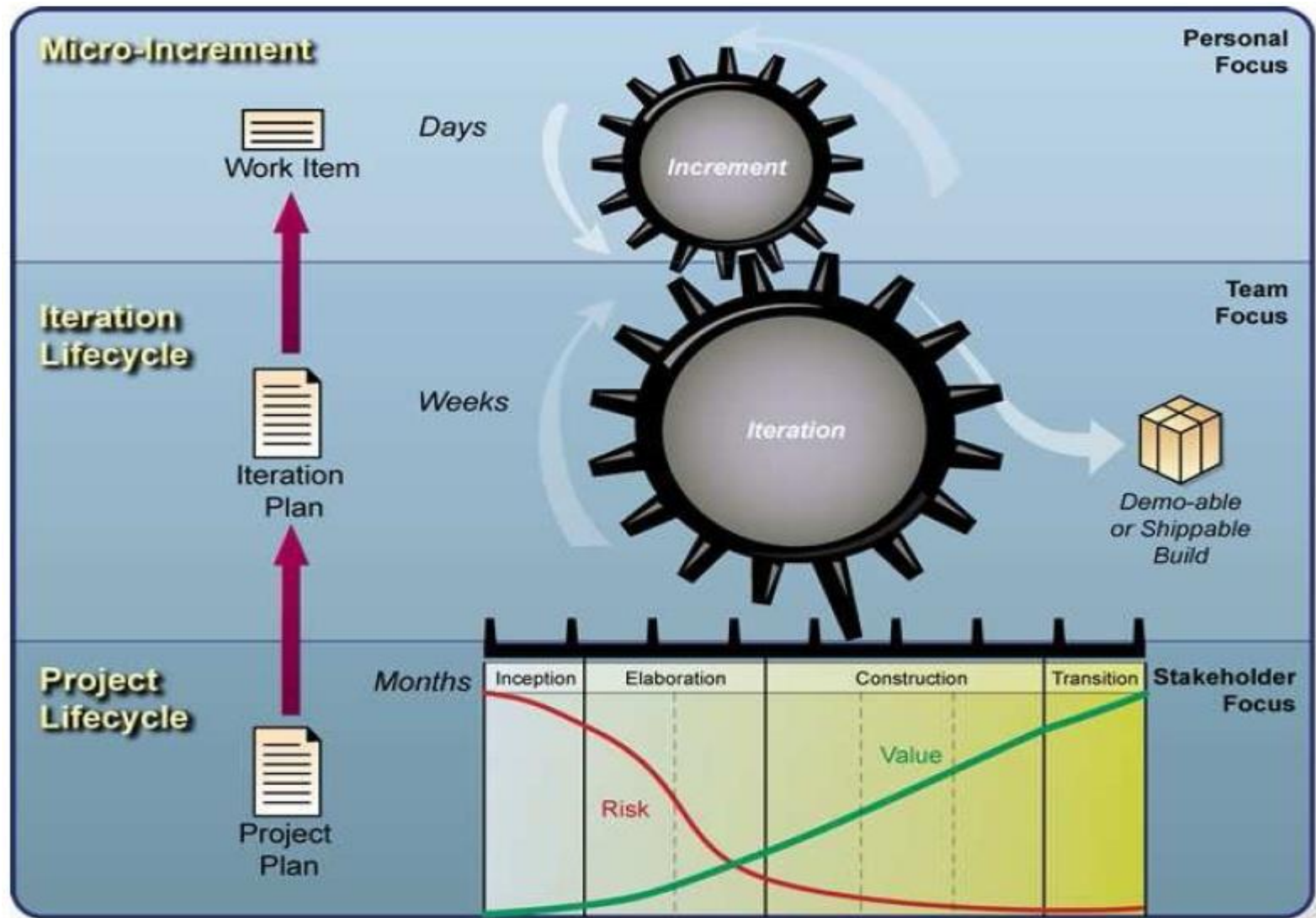
# OpenUP – przepływ prac (aktywności) w iteracji fazy Opracowania



Elementy objęte  
wykładem/projektem

- [-] Elaboration Phase
  - [-] Elaboration Iteration [1..n]
    - [+] Plan and Manage Iteration
    - [-] Identify and Refine Requirements
      - Identify and Outline Requirements
      - Detail Use-Case Scenarios
      - Detail System-Wide Requirements
      - Create Test Cases
    - [-] Develop the Architecture
      - [+] Develop Solution Increment
        - Refine the Architecture
    - [-] Develop Solution Increment
      - Design the Solution
      - Implement Developer Tests
      - Implement Solution
      - Run Developer Tests
      - Integrate and Create Build
    - [+] Test Solution
    - [+] Ongoing Tasks
  - [+] Lifecycle Architecture Milestone

# Unified Process – struktura



# Omówienie projektu

**FAZY :**

Zapoczątkowanie

Opracowanie

Konstrukcja

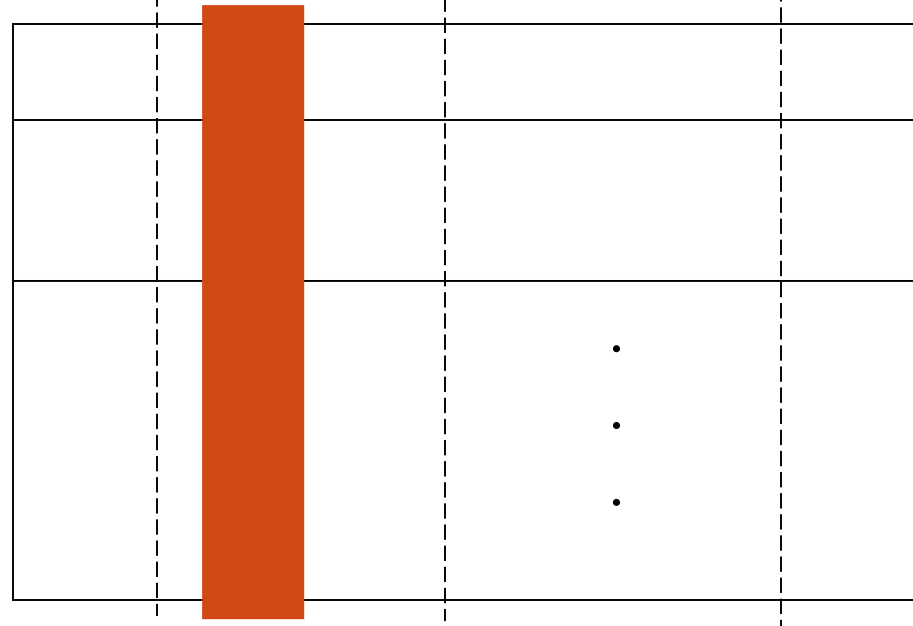
Przekazanie

Specyfikacja  
wymagań

Architektura

Implementacja

Testowanie



Iter (2)

# Pytania kontrolne

- Co to jest system informatyczny?
- Co to jest metodyka?
- Co to jest model i jakie role może pełnić w ramach wytwarzania oprogramowania?
- Czym cechuje się rodzina metodyk Unified Process?
- Omów podstawowe elementy cyklu życia metodyki z rodziny Unified Process.