

2ND EDITION

Podstawy inżynierii oprogramowania

Wprowadzenie

Zespół przedmiotowy



dr inż. Krystian Wojtkiewicz
adiunkt
Katedra Informatyki Stosowanej
<https://i-cu.eu>
krystian.wojtkiewicz@pwr.edu.pl

Wykłady:

Krystian Wojtkiewicz

Ćwiczenia:

Iwona Dubielewicz

Anita Walkowiak-Gall

Krystian Wojtkiewicz

Piotr Zabawa

Laboratorium:

Iwona Dubielewicz

Urszula Staszak

Podstawy Inżynierii Oprogramowania

C1 – Uzyskanie podstawowej wiedzy z zakresu inżynierii oprogramowania w tym znajomość: procesów i metodyk wytwarzania oprogramowania, modeli cyklu życia, zagadnień inżynierii wymagań oraz problemów testowania oprogramowania

C2 – Zdobywanie praktycznych umiejętności w zakresie inżynierii wymagań i specyfikowania testów oprogramowania.

Podstawy Inżynierii Oprogramowania

Wiedza + kompetencje

Umiejętności

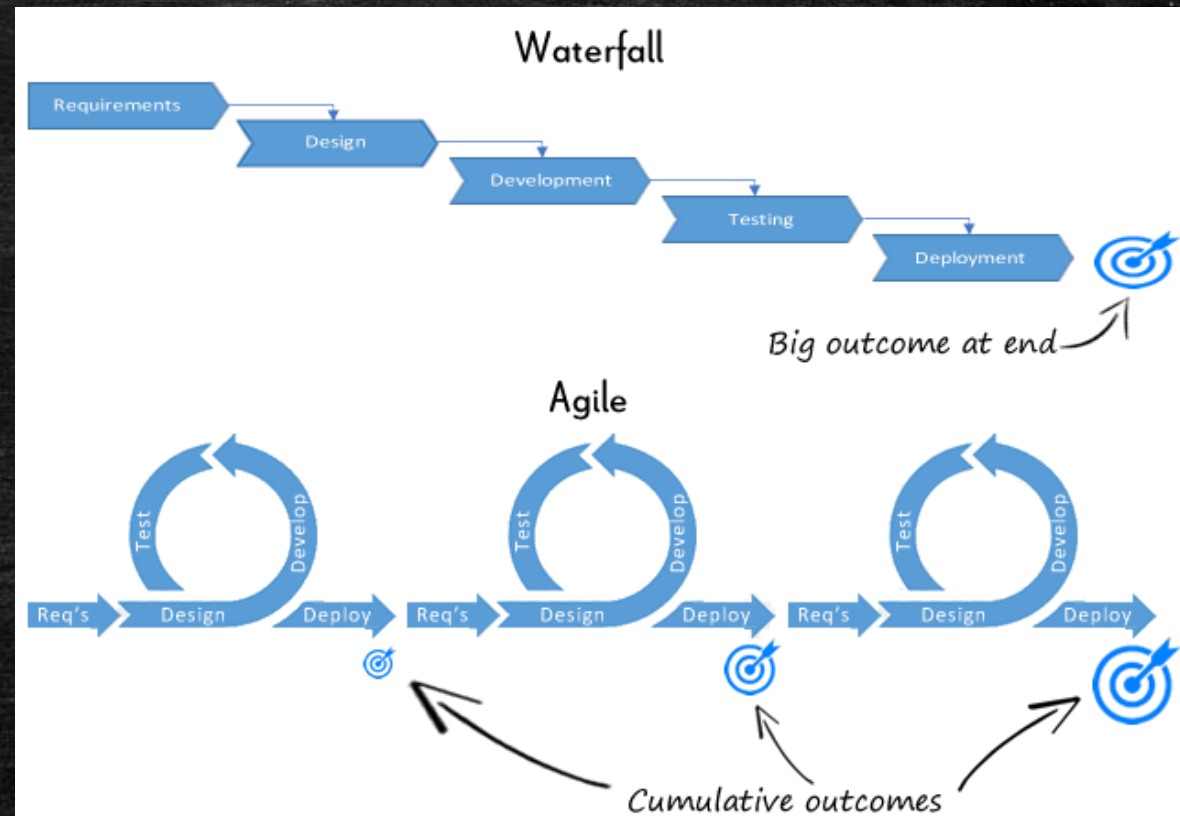
Wykład
(15h)

Ćwiczenia
(30h)

Laboratorium
(15h)

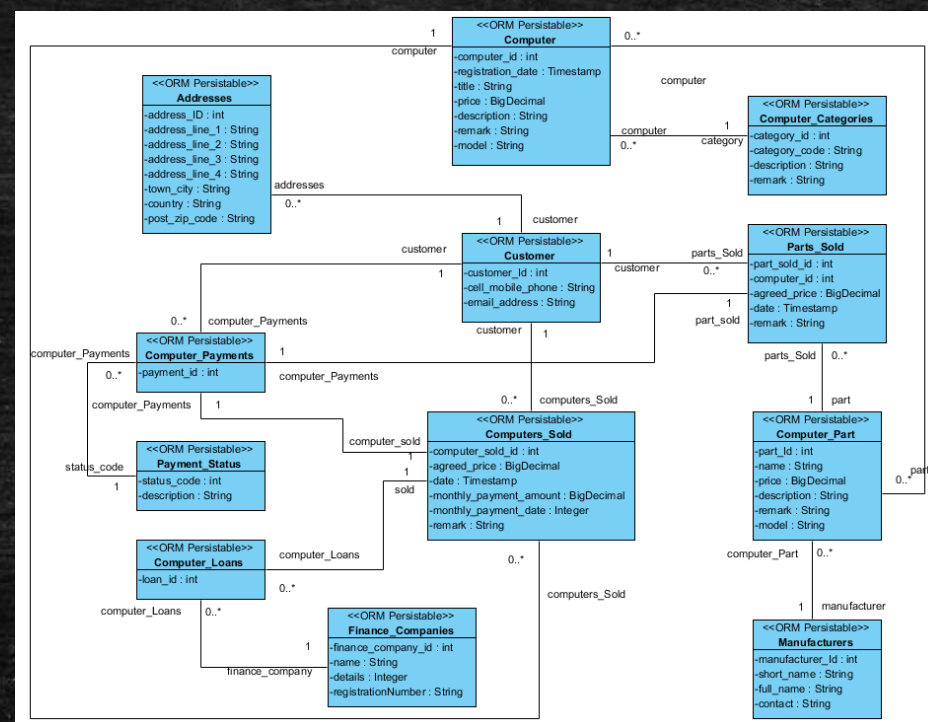
Tematyka wykładu

- Wprowadzenie do przedmiotu. Prezentacja programu kursu, organizacji zajęć i zasad zaliczania. Wprowadzenie podstawowych pojęć inżynierii oprogramowania.
- Procesy cyklu życia oprogramowania. Modele cyklu życia oprogramowania. Metodyki wytwarzania oprogramowania.
- Inżynieria wymagań. Modelowanie zachowania systemów (model przypadków użycia).
- Modelowanie struktury (model danych) systemów informatycznych.
- Język OCL jako formalny język specyfikacji ograniczeń systemu.
- Makietowanie interfejsu systemu jako technika weryfikacji wymagań: normy, projektowanie, narzędzia makietowania.
- Jakość oprogramowania; Testowanie – typy, techniki, testy jednostkowe; narzędzia testowania i pomiarów jakości oprogramowania.
- Zagadnienia zarządzania konfiguracją i zmianą.

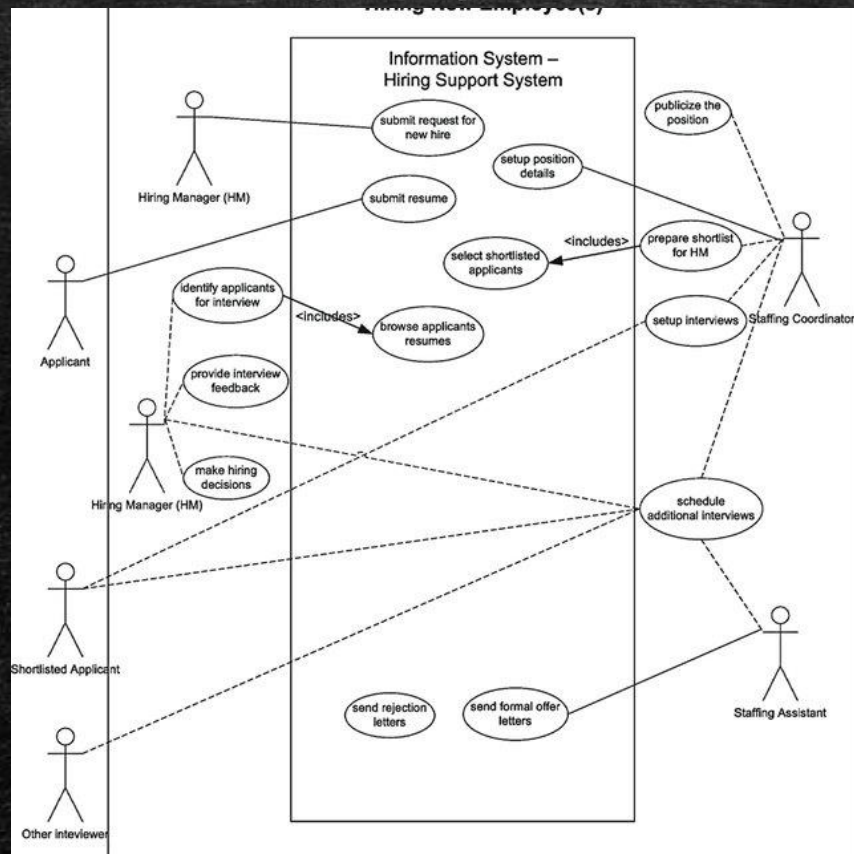


Tematyka ćwiczeń

- Klasa, obiekt
- Relacja generalizacji
- Asocjacja: podstawowe właściwości
- Asocjacja: zaawansowane aspekty (asocjacja rekurencyjna, asocjacja n-arna)
- Typ danych
- Właściwość statyczna (w tym read only)
- Właściwość wnioskowana (derived property)
- Krotności i liczności
- Właściwość identyfikująca obiekty klasy (ograniczenie ID)
- Wartość domyślna właściwości
- Właściwość wielowartościowa (ograniczenie ORDERED, NOUNIQUE)
- Klasa asocjacji
- Agregacja, Kompozycja



Tematyka ćwiczeń



Aktor

Przypadek użycia

Asocjacja

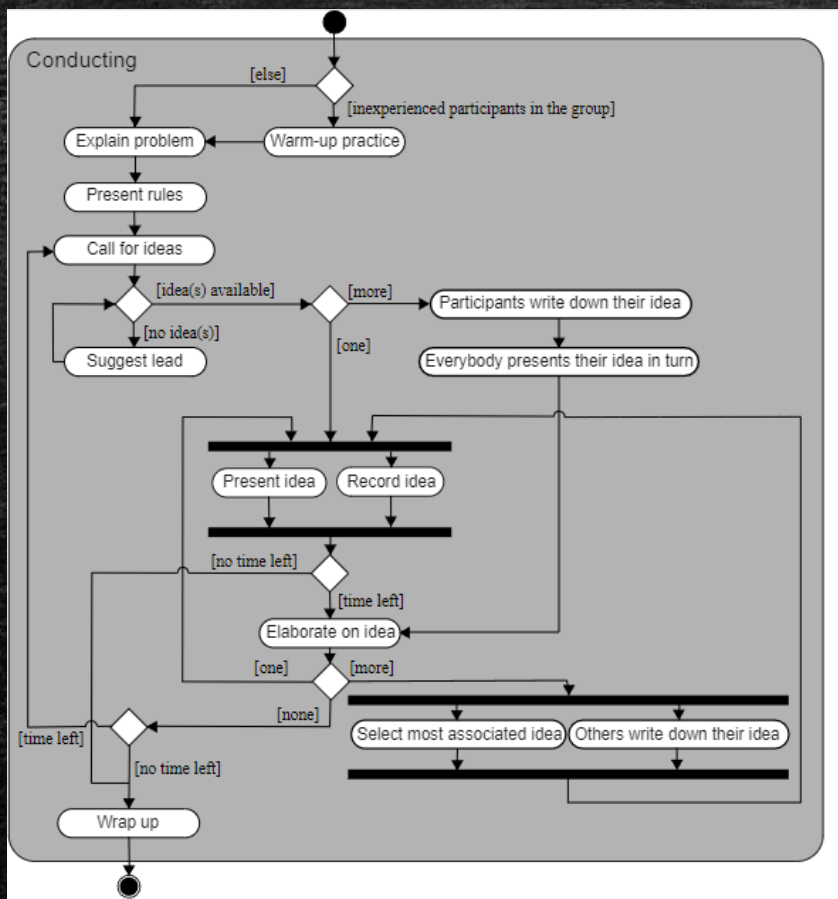
Relacja zawierania (include dependency)

Relacja rozszerzania (extend dependency)

Relacja generalizacji: aktorzy

Relacja generalizacji: przypadki użycia

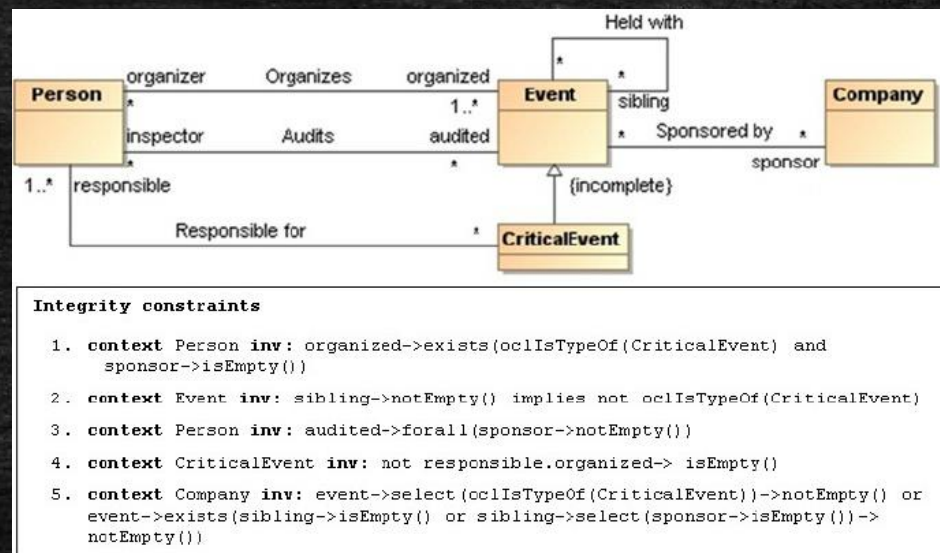
Tematyka ćwiczeń



- Aktywność, a czynność (Activity vs Action)
- Aktywność: przepływ obiektów, węzły obiektów
- Aktywność: przepływ sterowania, węzły sterowania
- Aktywność: węzły wykonawcze (executable nodes)
- Aktywność: grupy aktywności (Activity Groups), partycje aktywności (Activity Partitions)
- Akcja: podział na rodzaje, funkcja przekaźników (Pins)
- Akcja: akcje wywołania (invocation actions)
- Akcja: akcje obiektów (object actions), akcje powiązań (Link End Actions), asocjacji (Link Actions) i akcje obiektów powiązań (Link Object Action)
- Akcja: akcje cech strukturalnych (Structural Feature Action), akcje zmiennych (Variable Actions), akcje odpowiedzi na zdarzenia (Accept Action), pozostałe akcje (Other Actions)
- Akcja: akcje strukturalne (Structured Actions)
- Akcja: obszary rozszerzenia (Expansion Regions)

Tematyka ćwiczeń

- Reguły biznesowe i ich kategorie (definicja terminów biznesowych; fakty pokazujące relacje między terminami; ograniczenia; wnioski)
- Ograniczenia na elementy modelu systemu
 - niezmienniki dla atrybutów klasy;
 - niezmienniki dla asocjacji klasy
 - określenia warunków początkowych i końcowych realizacji operacji
 - ograniczenia dla zbiorów generalizacji
- Stereotypy związane z ograniczeniami w UML: «invariant», «precondition», «postcondition».



Tematyka ćwiczeń



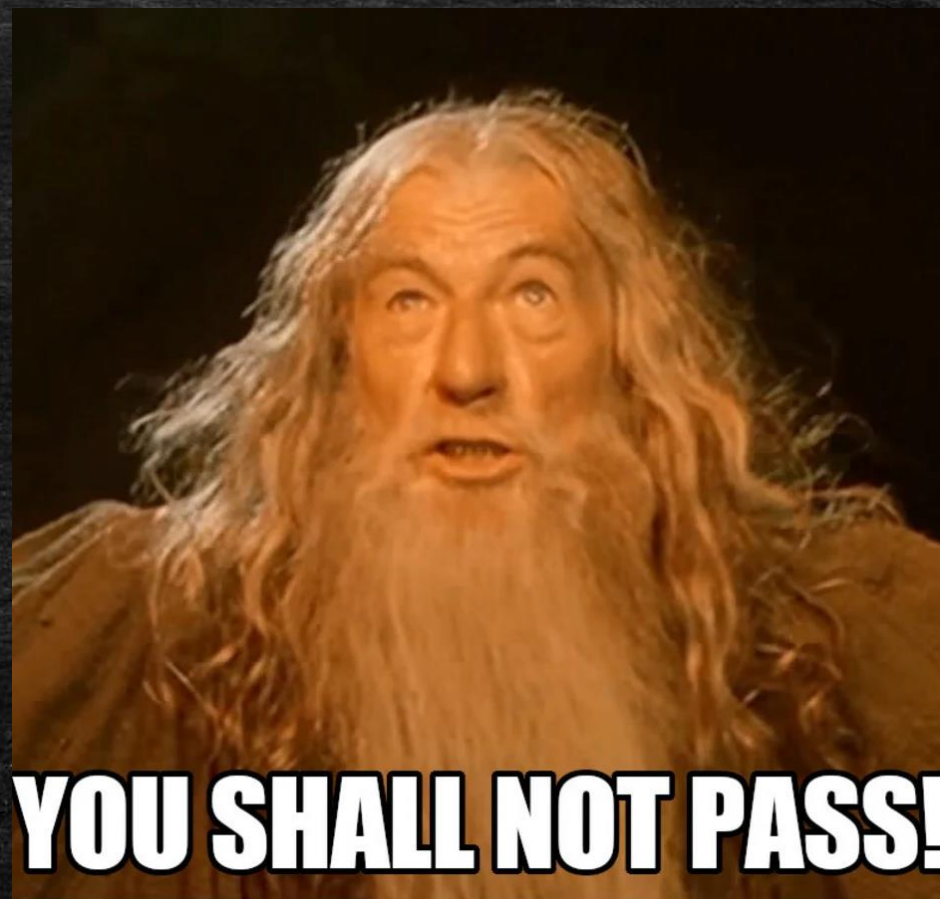
- Wymagania wg. UML (diagram przypadków użycia) i wg sysML (diagram wymagań)
- Elementy składni konkretnej diagramu wymagań wg sysML
- Wymagania wg RUP
- Specyfikowanie treści przypadków użycia wg RUP - szablony dokumentów specyfikacji i podstawowe zasady
- Realizacje przypadków użycia w UML wg RUP (przejście od dyscypliny Requirements do dyscypliny Analysis & Design)
- Klasyfikacja FURPS+ wg RUP w odniesieniu do wymagań
- Klasyfikacja FURPS+ w odniesieniu do testów (przez analogię do klasyfikacji wymagań wg RUP)
- Wybrane klasyfikacje i metodyki opisu wymagań.

Zasady zaliczenia

- Podstawą zaliczenia ćwiczeń jest uzyskanie wymaganej liczby punktów z (P)rezentacji i (A)ktywności w trakcie ćwiczeń oraz z (K)olokwium.
- Punkty (P) można uzyskać za:
 - Przygotowanie prezentacji na podany temat - max 15pkt (należy przedstawić co najmniej jeden temat w semestrze – ok. 10 min).
 - Niezależnie od liczby prezentowanych tematów student może uzyskać max. 15 pkt za prezentację. Prezentacja kolejnych tematów jest wskazana, jeżeli student uzyska małą liczbę punktów przy prezentacji pierwszego tematu.
 - O możliwości prezentacji danego tematu na zajęciach decyduje prowadzący.
- Punkty (A) można uzyskać za:
 - prezentację rozwiązań zadań podanych „na bieżąco” na ćwiczeniach (tylko w wypadku zgłoszenia się studenta) – max 25 (max 4pkt na jednych ćwiczeniach),
 - Odpowiedzi na kartkówki, wejściówki, zadania zlecane przez Prowadzącego.
- Punkty (K) to punkty z kolokwium (max 60)

Zasady udziału w zajęciach

1. Obecność na zajęciach jest obowiązkowa.
2. Każdy student ma obowiązek uczestnictwa w kolokwium zaliczeniowym.
3. Zaliczenie ćwiczeń w normalnym terminie (bez kolokwium poprawkowego) wymaga spełnienia warunków:
 - $(P + A + K) > 40$,
 - $P \geq 8$,
 - $K > 30$.



Zasady zaliczenia

Z = P+A+K	41	52	64	77	90	98
Ocena	3.0	3.5	4.0	4.5	5.0	5.5

*I'm a great believer in luck,
and I find the harder I work,
the more I have of it.*

Thomas Jefferson

Literatura

- ❑ Specyfikacja UML 2.5.1
- ❑ Specyfikacja OCL 2.4
- ❑ Wrycza i in., UML 2.0, Ćwiczenia zaawansowane, Helion 2012
- ❑ Raul Wazlawick, Object-Oriented Analysis and Design for Information Systems, Elsevier, 2014





WHAT'S
NEXT

?

0