POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

SYLABUS PRZEDMIOTU

Nazwa przedmiotu:	**Budowa i integracja systemów informatycznych**
Kod przedmiotu:	BYT
Kierunek / Profil:	Informatyka / praktyczny
Tryb studiów:	stacjonarny
Rok / Semestr:	3 / 5
Charakter:	obowiązkowy
Odpowiedzialny:	dr hab. inż. Marta Łabuda
Wersja z dnia:	19.02.2026

1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratori	Z prowadzącym	Praca własna	Łącznie	ECTS
30 h	—	—	30 h	20 h	50 h	**2**

2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Wykład	Egzamin

3. Cel dydaktyczny

Celem cyklu wykładów jest zapoznanie słuchaczy z podstawowymi zagadnieniami inżynierii oprogramowania, w tym z fazami rozwoju i ewolucji oprogramowania oraz metodami podwyższenia jakości oprogramowania. Wykład dotyczy podstawowych aspektów inżynierii oprogramowania i jest zorganizowany według kolejnych faz cyklu powstawania oprogramowania. Omówione są fazy: strategiczna, gromadzenia wymagań, analizy, projektowania, konstrukcji, testowania, instalacji i konserwacji a także jego wdrażanie i pielęgnacji. Wykład przedstawia też zagadnienia związane z architekturą systemów informatycznych i

dobrymi praktykami projektowania za pomocą narzędzi CASE. Znaczna waga przypisana jest dyskusji aktualnych modeli cyklu życia oprogramowania i kryteriom doboru strategii jego wytworzenia, jak też wykorzystaniu wzorców projektowych i standardowych API oraz dobrym praktykom projektowania i implementacji systemów, w szczególności: systemy rozproszone, systemy krytyczne, czasu rzeczywistego oraz wielokrotnego użytku. Omawiane na zajęciach tematy obejmują też jakość oprogramowania i jej miary, podstawowe informacje o procesie testowania oprogramowania oraz o zarządzaniu przedsięwzięciem programistycznym, zarządzaniu konfiguracją oprogramowania, implementację, a także problemy outsourcingu informatycznego. Charakteryzuje informatyczne systemy wsparcia procesów biznesowych omawiając ich najważniejsze cechy. Zajęcia są skorelowane z wykładem i projektem przedmiotu Projekt informatyczny (PRO), zapewniając kompleksowość przedstawianych zagadnień inżynierii oprogramowania i dając studentom możliwość samodzielnego, metodycznego przeprowadzenia niedużych projektów informatycznych - od zdefiniowania problemu, poprzez dobór strategii i zaplanowanie jego rozwiązania - do zaprojektowania i wykonania praktycznych elementów produktu, z użyciem wzorców i komponentów projektowych i przeprowadzeniem testów systemowych i walidacyjnych. Wzorcowym modelem jest wykorzystanie technologii obiektowej w cyklu kaskadowym, komponentowym lub zwinnie, nie są wykluczone inne strategie wytwórcze np. dla specyficznego produktu jakim są gry komputerowe.

4. Przedmioty wprowadzające

Przedmiot	Wymagane zagadnienia
• Wstęp do informatyki i architektura komputerów • Projektowanie systemów informacyjnych • Podstawowa wiedza o organizacji systemu komputerowego i jego architekturze • Umiejętność posługiwania się notacją UML	• Relacyjne bazy danych • Projekt (stowarzyszony) • Znajomość obiektowości, znajomość zagadnień analizy i projektowania SI • Rozumienie potrzeb syste-

5. Treści programowe

1. Wykład:
2. Przedmiot i zagadnienia inżynierii oprogramowania. Podstawowe motywacje, społeczny kontekst przedsięwzięcia. Modelowanie pojęciowe. Pojęcie metodyki. Modele cyklu życia oprogramowania
3. Przebieg i ocena fazy strategicznej. Planowanie projektu. Projekt informatyczny; podstawowe charakterystyki, pojęcia, udziałowcy projektu; cykl życia i zakres projektu. Planowanie zadań. Identyfikacja problemu; Wzbogacony wizerunek (Rich Picture). Decyzje strategiczne i wizja rozwiązania.
4. Studium Wykonalności projektu informatycznego. Cele, płaszczyzny oceny –techniczna, ekonomiczna, organizacyjna i prawna; ryzyko przedsięwzięcia. Przykłady studiów wykonalności
5. Analiza wymagań i proces inżynierii wymagań. Wymagania stawiane oprogramowaniu. Cechy dobrego wymagania. Metody pozyskiwania wymagań, user stories. Podział i klasyfikacja wymagań. Miary i formalne techniki specyfikowania wymagań.
6. Strategie i procesy prowadzenia projektów informatycznych; tradycyjne (model kaskadowy, model V, prototypowanie, przyrostowy, spiralny) i nowoczesne cykle wytwarzania oprogramowania (reuse i komponentowość), MDA, ponowna inżynieria, RUP.
7. Zwinne metodyki wytwarzania oprogramowania. Programowanie ekstremalne. SCRUM: procesy, artefakty, role. Dobór strategii prowadzenia projektu.
8. Analiza i projektowanie architektury systemów; role, procesy, techniki i produkty w podejściu obiektowym oraz agile. Wzorce analizy i projektowania; Klasyfikacja i przykłady wzorców.
9. Optymalizacja projektu dla specyfiki produktu. Projektowanie systemów rozproszonych, krytycznych, czasu rzeczywistego i wielokrotnego użytku.
10. Zarządzanie przedsięwzięciem informatycznym. Infrastruktura - dokumentacja procesu wytwarzania i produktu programistycznego, komunikacja, struktury organizacyjne. Raportowanie. Obszary zarządzania. zarządzanie zespołem. Harmonogramowanie.
11. Implementacja, integracja i wersjonowanie oprogramowania.; środowiska narzędziowe oraz technologie. Zarządzanie konfiguracją oprogramowania.
12. Pojęcie jakości oprogramowania i dróg jej osiągania. Standardy jakości i dobre praktyki jakości. Zapewnianie jakości oprogramowania. Modele i miary jakości oprogramowania. Wprowadzenie do zarządzania jakością. Inspekcja kodu i analiza bezpieczeństwa. Metody szacowania nakładów. Model COCOMO II.
13. Testowanie i walidacja oprogramowania. Rodzaje testów testy jednostkowe, integracyjne, systemowe; walidacja i testy akceptacyjne. Typowe fazy i metody testowania. Testy statyczne, funkcjonalne i strukturalne. Czynniki sukcesu i rezultaty testowania. Dokumentowanie testów. Przeglądy oprogramowania.
14. Instalacja i wdrożenie oprogramowania. Pielęgnacja oprogramowania. Analiza potrzeby wprowadzania modyfikacji. Koszty konserwacji oprogramowania. Kluczowe czynniki sukcesu fazy utrzymania.
15. Outsourcing w inżynierii oprogramowania.

16. Informatyczne wsparcie procesów biznesowych. Oprogramowanie klasy CRM, ERP. Architektura systemów ERP. Przykłady dostawców systemów CRM i ERP.

6. Efekty kształcenia

Wiedza

- Student zna i rozumie pojęcia w zakresie sieci komputerowych, ich technologii, protokołów komunikacyjnych i zagadnień bezpieczeństwa, telekomunikacji oraz potrzebę przenoszenia dobrych praktyk na grunt informatyki
- Student zna i rozumie pojęcia z zakresu kluczowych zagadnień w zarządzania informacją i modelowania danych; szczegółowo zna i rozumie zagadnienia konstrukcji relacyjnych baz danych, ich programowania i przetwarzania transakcji; ma znajomość aktualnie stosowanych systemów baz danych
- Student zna i rozumie zaawansowane pojęcia z zakresu zagadnień inżynierii oprogramowania, standardów i kształtu cykli wytwórczych oraz ewolucji oprogramowania; zna podstawy zarządzania przedsięwzięciem programistycznym i rozumie problem jakości oprogramowania; rozumie rolę modelowania i ma szczegółową wiedzę o obiektowym wytwarzaniu oprogramowania i notacji UML, zna i rozumie zasady korzystania z wzorców programowych i standardowych API; ma wiedzę o typowych narzędziach i środowiskach wspomagających;
- Student zna i rozumie podstawowe pojęcia z zakresu kluczowych zagadnień inżynierii wymagań, rozumie potrzebę systematycznego budowania i pielęgnacji specyfikacji wymagań; ma szczegółową wiedzę dotyczącą ich specyfikacji, analizy i modelowania z użyciem dostępnych narzędzi
- Student zna i rozumie kluczowe pojęcia z zakresu walidacji i testowania oprogramowania

Umiejętności

- Student potrafi zaplanować i przeprowadzić procesy pozyskiwania, analizy, specyfikacji i modelowania wymagań wobec oprogramowania oraz ich pielęgnacji
- Student potrafi dokonać przeglądu projektu oprogramowania i poprawić jego jakość
- Student potrafi uwzględnić społeczny, etyczny i prawny kontekst przedsięwzięcia informatycznego oraz ocenić związane z nim zagrożenia
- Student potrafi zaplanować i wytworzyć podstawowe dokumenty związane z realizacją prostego przedsięwzięcia informatycznego, wstępnie ocenić efekty ekonomiczne i społeczne przedsięwzięcia oraz ich wpływ na udziałowców;

7. Kryteria oceny

- Kryteria oceny
- Wykład: egzamin pisemny poprzedzony pracami bieżącej oceny, motywującej do systematycznej pracy w trakcie semestru.

8. Metody dydaktyczne

Wykład, laboratoria, praca własna studenta.

9. Literatura

Podstawowa:

- 1. Krzysztof Sacha, Inżynieria Oprogramowania, PWN 2022
- 2. Ian Sommerville, Inżynieria Oprogramowania, wyd.10, PWN 2020
- 3. Max Kanat-Alexander, Zrozumieć oprogramowanie, Helion 2019
- 4. Robert C. Martin, Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów, Helion 2022
- 5. Mike Cohn, Agile. Metodyki zwinne w planowaniu projektów, Helion 2019
- 6. Robert C. Martin, Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki, Helion 2017

Uzupełniająca:

- 1. Bernd Bruegge, Allen H. Dutoit, Inżynieria oprogramowania w ujęciu obiektowym UML, wzorce projektowe i Java Helion 2011
- 2. Keeling Michael,Zostań architektem oprogramowania, PWN 2019
- 3. Piotr Gaczkowski, Adrian Ostrowski, Architektura oprogramowania bez tajemnic, Helion 20224.
- 4. Arnon Axelrod, Automatyzacja testów. Kompletny przewodnik dla testerów oprogramowaniaPWN 2022
- 5. Strony domowe do wybranych narzędzi informatycznych. Instrukcje obsługi, przykłady.