



Nazwa modułu zajęć:	Zarządzanie Systemem Informatycznym		
Rok akademicki:	2019/2020	Kod:	AMAT-2-302-MZ-s Punkty ECTS: 2
Wydział:	Matematyki Stosowanej		
Kierunek:	Matematyka	Specjalność:	Matematyka w zarządzaniu
Poziom studiów:	Studia II stopnia	Forma studiów:	Stacjonarne
Język wykładowy:	Polski	Profil:	Ogólnoakademicki (A) Semestr: 3
Strona www:	—		
Prowadzący moduł:	dr Mielczarek Dominik (dmielcza@wms.mat.agh.edu.pl)		

### Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Element inżynierii oprogramowania. Systemy informatyczne.

### Opis efektów uczenia się dla modułu zajęć

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Powiązania z KEU	Sposób weryfikacji i oceny efektów uczenia się osiągniętych przez studenta w ramach poszczególnych form zajęć i dla całego modułu zajęć
Wiedza: zna i rozumie			
M_W001	Student identyfikuje zagrożenia dla typowego projektu informatycznego oraz metody stosowane do ich uniknięcia lub zmniejszenia skutków.	MAT2A_W11	Egzamin
M_W002	Student rozpoznaje pojęcia z dziedziny inżynierii oprogramowania, rozumie cele standardowych procesów.	MAT2A_W11	Egzamin
Umiejętności: potrafi			
M_U001	Student potrafi zebrać wymagania dla systemu informatycznego oraz zapisać je w postaci umożliwiającej dalsze przetwarzanie.	MAT2A_K03, MAT2A_K02, MAT2A_U20, MAT2A_U19	Wykonanie ćwiczeń laboratoryjnych
M_U002	Student potrafi wykonać analizę oczekiwań interesariuszy oraz ich priorytetyzację.	MAT2A_K02, MAT2A_U16	Egzamin

M_U003	Student potrafi użyć notacji UML do przedstawienia aspektów projektowanego systemu oraz wykonać model w narzędziu CASE.	MAT2A_U20	Projekt, Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych
Kompetencje społeczne: jest gotów do			
M_K001	Student identyfikuje kontekst społeczny zaplanowanego systemu, w jaki sposób wykonana praca wpłynie na pracę i przyzwyczajenia innych ludzi.	MAT2A_K03	Wykonanie ćwiczeń laboratoryjnych
M_K002	Student aktywnie proponuje uproszczenia projektowanego systemu, biorąc pod uwagę ograniczenia zasobów.	MAT2A_K03, MAT2A_K02	Wykonanie ćwiczeń laboratoryjnych
M_K003	Student aktywnie bierze udział w dyskusji nad wizją projektowanego systemu, zadaje pytania prowadzącemu.	MAT2A_K04, MAT2A_K03, MAT2A_K02	Wykonanie ćwiczeń laboratoryjnych

### Liczba godzin zajęć w ramach poszczególnych form zajęć

Suma	Forma zajęć dydaktycznych										
	Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
30	15	0	15	0	0	0	0	0	0	0	0

### Matryca kierunkowych efektów uczenia się w odniesieniu do form zajęć i sposobu zaliczenia, które pozwalają na ich uzyskanie

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Forma zajęć dydaktycznych										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
Wiedza: zna i rozumie												
M_W001	Student identyfikuje zagrożenia dla typowego projektu informatycznego oraz metody stosowane do ich uniknięcia lub zmniejszenia skutków.	+	-	+	-	-	-	-	-	-	-	-
M_W002	Student rozpoznaje pojęcia z dziedziny inżynierii oprogramowania, rozumie cele standardowych procesów.	+	-	+	-	-	-	-	-	-	-	-

Umiejętności: potrafi												
M_U001	Student potrafi zebrać wymagania dla systemu informatycznego oraz zapisać je w postaci umożliwiającej dalsze przetwarzanie.	-	-	+	-	-	-	-	-	-	-	-
M_U002	Student potrafi wykonać analizę oczekiwań interesariuszy oraz ich priorytetyzację.	-	-	+	-	-	-	-	-	-	-	-
M_U003	Student potrafi użyć notacji UML do przedstawienia aspektów projektowanego systemu oraz wykonać model w narzędziu CASE.	-	-	+	-	-	-	-	-	-	-	-
Kompetencje społeczne: jest gotów do												
M_K001	Student identyfikuje kontekst społeczny zaplanowanego systemu, w jaki sposób wykonana praca wpłynie na pracę i przyzwyczajenia innych ludzi.	+	-	+	-	-	-	-	-	-	-	-
M_K002	Student aktywnie proponuje uproszczenia projektowanego systemu, biorąc pod uwagę ograniczenia zasobów.	+	-	+	-	-	-	-	-	-	-	-
M_K003	Student aktywnie bierze udział w dyskusji nad wizją projektowanego systemu, zadaje pytania prowadzącemu.	+	-	+	-	-	-	-	-	-	-	-

## Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach dydaktycznych/praktyka	30 godz
Przygotowanie do zajęć	4 godz
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	20 godz
Egzamin lub kolokwium zaliczeniowe	1 godz
Dodatkowe godziny kontaktowe	2 godz
Sumaryczne obciążenie pracą studenta	57 godz
Punkty ECTS za moduł	2 ECTS

## Pozostałe informacje

### Szczegółowe treści kształcenia w ramach poszczególnych form zajęć (szczegółowy program wykładów i pozostałych zajęć)

#### Wykład

Treść wykładu odpowiada zajęciom laboratoryjnym.

## **Ćwiczenia laboratoryjne**

### Ćwiczenia laboratoryjne

Zajęcia laboratoryjne mają przygotować do wykonania projektu przez zajęcia z kontaktu z klientem, uzupełnianie wiedzy praktycznej. Część czasu zostanie przeznaczona na konsultacje wykonywanych projektów i przedyskutowanie problemów pojawiających się podczas wykonania.

1. Analiza systemu – 1,5 godz.

Analiza interesariuszy. Wywiad z klientem. Definiowanie wymagań funkcjonalnych i niefunkcjonalnych.

2. Przygotowanie dokumentacji projektowej – 1,5 godz.

Wykonanie (na podstawie szablonu) pierwszej wersji specyfikacji funkcjonalnej i zapisu decyzji architektonicznych.

3. Przygotowanie i wykorzystanie modelu systemu – 1,5 godz.

Wykonanie modelu systemu w oparciu o notację UML w narzędziu CASE.

4. Tworzenie aplikacji z użyciem wybranych technologii (np. HTML, CSS i PHP, C++) – 4,5 godz.

Obsługa narzędzi potrzebnych do wykonania projektu. Przygotowanie interfejsu użytkownika. Analiza użyteczności interfejsu użytkownika.

5. Weryfikacja i akceptacja projektu – 1,5 godz.

Przekonanie klienta, że wykonany system jest odpowiedzią na zgłoszone potrzeby. Przedstawienie możliwości dalszego rozwoju.

### Seminarium

1. Wprowadzenie do Inżynierii Oprogramowania – 1,5 godz.

Historia inżynierii oprogramowania. Zakres, podstawowe pojęcia, nomenklatura. Wyzwania stojące przed projektami.

2. Inżynieria wymagań – 1,5 godz.

Analiza interesariuszy. Wymagania niefunkcjonalne i funkcjonalne. Sposoby definiowania wymagań. Błędy popełniane przy specyfikowaniu wymagań. Praktyki przydatne w zarządzaniu wymaganiami.

3. Modelowanie i dokumentacja projektowa – 3 godz.

Rodzaje modeli. Modelowanie obiektowe. Modelowanie zachowania systemu. Modelowanie dziedziny systemu. Modelowanie z użyciem UML. Przykład modelu systemu opartego na przypadkach użycia. Zintegrowane narzędzia do modelowania.

4. Procesy tworzenia oprogramowania – 1,5 godz.

Modele rozwoju oprogramowania. Porównanie modelu kaskadowego, spiralnego, unified process. Cechy procesu fazowego i iteracyjnego na podstawie Agile Unified Process i Enterprise Unified Process. Aspekty faz początkowej, rozwinięcia, budowy, dostarczenia.

5. Lekkie metody tworzenia oprogramowania – 1,5 godz.

Geneza praktyk zwinnych. Porównanie metod preskryptywnych i adaptacyjnych.

Praktyki stosowane w różnych obszarach aktywności projektowej.

6. Testowanie oprogramowania i jakość – 1,5 godz.

Kryteria jakości. Poziomy testów. Techniki testowania. Pokrycie systemu testem. Rozwój oprogramowania sterowany testami.

7. (opcjonalne) Studium przypadków zastosowania inżynierii oprogramowania – 1,5 godz.

Analiza wykonania dwóch projektów. Użyte narzędzia i metody pracy.

#### Projekt

Celem zadań projektowych jest zaplanowanie, zaprojektowanie i wykonanie systemu informatycznego. Może zostać przez wykonawców wybrany dowolny temat i technologia, jeżeli tylko wyniki prac mogą być prezentowane podczas zajęć i prowadzący podejmie się pełnienia roli klienta w takim zestawieniu.

Studenci wykonują projekt pracując w grupach 3-osobowych. W ramach projektu przygotowują:

- dokumentację analityczną tworzonego systemu
- opis architektury systemu
- działający prototyp systemu, wykonujący przynajmniej część przewidzianej w dokumentacji funkcjonalności i zgodny z przyjętymi założeniami

#### **Metody i techniki kształcenia:**

Wykład: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.

Ćwiczenia laboratoryjne: W trakcie zajęć laboratoryjnych studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia. Prowadzący stymuluje grupę do refleksji nad problemem, tak by otrzymane wyniki miały wysoką wartość merytoryczną.

#### **Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:**

dwa zaliczenia poprawkowe

#### **Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:**

Wykład:

- Obecność obowiązkowa: Nie

- Zasady udziału w zajęciach: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne:

- Obecność obowiązkowa: Tak

- Zasady udziału w zajęciach: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

### **Sposób obliczania oceny końcowej**

Oceny z projektu (P) oraz z egzaminu (E) obliczane są następująco: procent uzyskanych punktów przeliczany jest na ocenę zgodnie z Regulaminem Studiów AGH.

Ocena końcowa (OK) obliczana jest jako średnia ważona ocen z egzaminu (E) i z wykonania projektu (P):

$$OK = 0.35 \times E + 0.65 \times P$$

### **Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:**

indywidualnie

### **Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności modułów**

Uczestnicy kursu powinni wcześniej przejść przez podstawowe kursy obsługi komputera, oprogramowania biurowego i programowania. Przydatne również będzie znajomość konstrukcji baz danych, znaczników HTML.

Nawet dogłębna znajomość zagadnień poruszanych w ramach kursu nie stanowi przeszkody w uczestnictwie.

### **Zalecana literatura i pomoce naukowe**

1) IEEE Computer Society, Guide to the Software Engineering Body of Knowledge (SWEBOK), 2004, <http://www.computer.org/portal/web/swebok/htmlformat>

2) Scott W. Ambler, Agile Modeling: Effective Practices for Extreme Programming and the Unified Process, J. Wiley., 2002

3) Alistair Cockburn, Writing Effective Use Cases, A. Addison Wesley, 2001.

4) Karl Wieggers, Software Requirements 2, O'Reilly Media, 2009

### **Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu**

Nie podano dodatkowych publikacji

### **Informacje dodatkowe**

Wszystkie zajęcia są prowadzone przez przedstawicieli firmy Ericpol:

Jacek Rybicki ([jacek.rybicki@ericpol.com](mailto:jacek.rybicki@ericpol.com))

Michał Szancer

Wiesław Chmielnicki