

Projektowanie Systemów Informatycznych II semestr Studiów II stopnia

1. Założenia wstępne

- Dobra znajomość UML
- Znajomość w zakresie analizy biznesowej i systemowej

2. Cele

- Projektowanie oparte o modelowanie, w tym wykorzystanie modeli do generacji kodu
- Ocena jakości artefaktów, np. architektury
- Implementacja fragmentu systemu informatycznego zgodnie z projektem i sprawdzenie jego jakości

3. Realizacja

3.1. Założenia

- Jeden temat wiodący dla całej grupy projektowej
- Praca w grupach 2-3 osobowych

3.2. Narzędzia realizacji

- Modele UML – Visual Paradigm 14.1
- Język implementacji: dowolny (narzędzia dostępne na licencji OpenSource lub w ramach DreamSpark)

3.3. Warunki zaliczenia i sposób oceniania

W harmonogramie prac (rozdział 3.4) zaznaczono punkty kontrolne – w których należy przygotować artefakty do oceny (na żółto). Spóźnienie z oddaniem artefaktów z dowolnych przyczyn (poza usprawiedliwioną nieobecnością) powoduje obniżenie o 0.5 oceny cząstkowej za każdy tydzień. W przypadku usprawiedliwionej nieobecności (choroba) artefakty należy dostarczyć tak szybko, jak to możliwe.¹

Obecność na zajęciach nie jest obowiązkowa za wyjątkiem terminów zaznaczonych na żółto w poniższym harmonogramie.

Ocena końcowa będzie ustalona zgodnie z następującymi zasadami (zaokrąglana):

$$0,3 * F1 + 0,3 * F2 + 0,4 * F3$$

F1 – Ocena za artefakty modelowania biznesowego i analizy wymagań

F2 – Ocena za definicję i ocenę architektury (SAD)

F3 – Ocena za implementację i testy

Brak oddania danego etapu oznacza, że składowa F1, F2, F3 jest równa 0!

Przy ocenie pod uwagę będą brane elementy jakościowe oddanych do oceny artefaktów (kompletność, spójność, etc.) oraz terminowość wykonania zadań.

¹ Uwaga. Prowadzący może wyrazić zgodę, aby artefakty do oceny wzajemnej były przekazane 2-3 dni po punkcie kontrolnym, w którym student może skonsultować projekt.

3.4. Harmonogram prac

Harmonogram sugeruje kolejność i tempo wykonywania prac. Zakłada się, że artefakty wymienione dla danego tygodnia będą w tym tygodniu konsultowane (należy przygotować co najmniej ich wersję roboczą).

Zjazd	Projekt	Wykład
1 – 25.02N	Zajęcia organizacyjne. Organizacja zespołów. Zapoznanie z wizją produktu. Wybór obszaru zainteresowań. Zespół wskazuje 2n cech do analizy i implementacji (n = liczba członków zespołu) – zakres wymaga akceptacji prowadzącego	Wprowadzenie do tematyki przedmiotu. Metodyki. Modele i ich funkcja.
2 – 4.03P	Słownik + model domenowy + reguły biznesowe (Word) - z obszaru zainteresowań.	
3 – 18.03N	Specyfikacja wymagań funkcjonalnych (model PU wraz z opisami streszczającymi w VP lub historii użytkownika pogrupowane w epiki w Excelu) – z obszaru zainteresowań. Model informacyjny, prototyp interfejsu – z obszaru zainteresowań. Wybór min. n+1 wymagań jakościowych różnych typów (jeżeli trzeba – ich uszczegółowienie); zakres wymaga uzgodnienia z prowadzącym	Definicja architektury systemu. Perspektywy i widoki architektoniczne. Style architektoniczne.
4 – 25.03P	F1 - Modelowanie biznesowe i specyfikacja wymagań; Definicja wybranych widoków architektonicznych	
5 – 8.04N	Definicja wybranych widoków	Taktyki. Wpływ wymagań niefunkcjonalnych na architekturę. Zawartość dokumentu opisującego architekturę.
6 – 22.04P	SAD – do konsultacji Implementacja – w miarę możliwości wykorzystać narzędzia DevOps	
7 – 13.05N	F2a – Definicja architektury Implementacja – j.w.	Metody oceny architektury.
8 – 27.05P	Implementacja i testy – j.w.	
9 – 10.06N	F2b – Ocena architektury F3 – Implementacja i testy	MDE, DSL.

3.5. Elementy przedkładane do oceny

Deadline 1: Modelowanie biznesowe i specyfikacja wymagań. Dokument tekstowy podzielony na odpowiednie rozdziały: słownik biznesowy, reguły biznesowe, model przypadków użycia/historijki użytkownika, model informacyjny, prototypy interfejsu oraz dokument przedstawiający wymagania niefunkcjonalne (* za wygenerowanie odpowiednich raportów w ramach używanych narzędzi możliwe podniesienie oceny o 0,5).

Deadline 2a: Architektura systemu. Dokument tekstowy zgodny z szablonem: architecture_notebook_tpl – minimum 3 widoki.

Deadline 2b: Wyniki oceny architektury. Dokument tekstowy przygotowany zgodnie z zaleceniami podanymi na wykładzie.

Deadline 3: Kody źródłowe wygenerowane z narzędzi + kody źródłowe napisane ręcznie + testy jednostkowe/integracyjne (min. 3n). Wersja uruchamialna (w przypadku braku możliwości umieszczenia działającej wersji w dostępnym środowisku produkcyjnym, demonstracja rozwiązania na konsultacjach). Raport wskazujący na sposób realizacji projektu w zakresie wymagań niefunkcjonalnych.