

# Testowanie i rozwój aplikacji - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Testowanie i rozwój aplikacji
Kod przedmiotu	06.0-WE-INFP-TiRA
Wydział	<a href="#">Wydział Informatyki, Elektrotechniki i Automatyki</a>
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2021/2022

Informacje o przedmiocie	
Semestr	6
Liczba punktów ECTS do zdobycia	4
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	<ul style="list-style-type: none"><li>dr inż. Michał Doligalski</li></ul>

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	15	1	9	0,6	Zaliczenie na ocenę
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę

## Cel przedmiotu

Zapoznanie studentów z nowoczesnymi technikami testowania i rozwoju aplikacji.

Ukształtowanie świadomości konieczności przeprowadzenia testów oraz nabycie umiejętności automatyzacji testów.

Planowanie i realizacja testów.

## Wymagania wstępne

Język Java i technologie Web

Zarządzanie przemysłowym projektem informatycznym

## Zakres tematyczny

**Podstawowe zasady testowania programów, miejsca testowania w inżynierii komputerowej i inżynierii oprogramowania. Rola testów w procesie rozwoju systemów informatycznych. Inspekcja kodu źródłowego i tworzenie przypadków testowych. Testowanie poszczególnych modułów aplikacji, testowanie integracyjne. Testowanie funkcjonalne, system, akceptacja i instalacja. Testowanie atrybutów нефункциональных, testowanie strukturalne. Ekstremalne testy. Testowanie aplikacji internetowych. Ciągła integracja. Automatyzacja testów., w tym narzędzia do automatyzacji testów. Rola testów w metodykach zwinnych, rozwój systemów z wykorzystaniem metodyk zwinnych.**

## Metody kształcenia

**Wykład:** wykład konwencjonalny i multimedialny.

**Laboratorium:** ćwiczenia laboratoryjne

## Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrąfi kreatywnie zaplanować testy i zinterpretować jego wyniki. W świetle wyników zidentyfikować obszar wadliwego działania, jak i zaproponować metodę jego eliminacji	<ul style="list-style-type: none"><li><a href="#">K_W19</a></li><li><a href="#">K_U15</a></li></ul>	<ul style="list-style-type: none"><li>bieżąca kontrola na zajęciach</li><li>sprawdzian</li></ul>	<ul style="list-style-type: none"><li>Laboratorium</li></ul>
Ma podstawową wiedzę na temat cykli życia systemów informatycznych oraz metod i narzędzi do weryfikacji i testowania systemów informatycznych	<ul style="list-style-type: none"><li><a href="#">K_U10</a></li></ul>	<ul style="list-style-type: none"><li>kolokwium</li><li>zaliczenie - ustne, opisowe, testowe i inne</li></ul>	<ul style="list-style-type: none"><li>Wykład</li></ul>
Ma świadomość wpływu poszczególnych etapów procesu projektowania na występowanie błędów w projekcie informatycznym i jego koszt eliminacji	<ul style="list-style-type: none"><li><a href="#">K_W05</a></li></ul>	<ul style="list-style-type: none"><li>kolokwium</li></ul>	<ul style="list-style-type: none"><li>Wykład</li></ul>

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrafi wykorzystać w sposób praktyczny, aktualne narzędzia i rozwiązania testowania oprogramowania i projektowania systemów informatycznych	<ul style="list-style-type: none"> <li>• <a href="#">K_W09</a></li> <li>• <a href="#">K_U07</a></li> </ul>	<ul style="list-style-type: none"> <li>• bieżąca kontrola na zajęciach</li> <li>• sprawdzian</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratorium</li> </ul>

## Warunki zaliczenia

**Wykład** - Warunkiem zaliczenia jest uzyskanie pozytywnej oceny z kolokwium realizowanego w formie pisemnej.

**Laboratorium** - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium (80%) oraz aktywności na zajęciach (20%).

**Składowe oceny końcowej** = wykład: 50% + laboratorium: 50%

## Literatura podstawowa

1. How Google Tests Software, James A. Whittaker, Jason Arbon, Jeff Carollo, Addison-Wesley Professional, 2012
2. The Digital Quality Handbook: Guide for Achieving Continuous Quality in a DevOps Reality, Eran Kinsbruner, 2017
3. Test Automation in the Real World: Practical Lessons for Automated Testing  
by Greg Paskal
4. Testing of Digital Systems, N. K. Jha (Author), S. Gupta (Author), Cambridge University Press, 2003

## Literatura uzupełniająca

### Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 20-04-2021 08:55)

Wygenerowano automatycznie z systemu SylabUZ