

OPIS MODUŁU KSZTAŁCENIA (przedmiot lub grupa przedmiotów)

Nazwa modułu/ przedmiotu Narzędzia, techniki i metodyka pracy programisty	Przedmiot/y Projektowanie graficznych interfejsów użytkownika Zarządzanie zmianą i systemy kontroli wersji Inżynieria oprogramowania I Inżynieria oprogramowania II
---	--

Nazwa jednostki prowadzącej przedmiot
Instytut Matematyki

kierunek	specjalność	specjalizacja	semestr/y	poziom kształcenia/ profil kształcenia	forma studiów
Informatyka	Programowanie	-	4,5,6	SPS/ praktyczny	stacjonarne/ niestacjonarne

Nazwisko osoby prowadzącej (osób prowadzących)

dr inż. Zbigniew Ledóchowski, dr Ryszard Motyka, mgr Ireneusz Lewandowski

Formy zajęć	Liczba godzin				Liczba punktów ECTS
	N (nauczyciel)		S (student)		
	studia stacjonarne	studia niestacjonarne	studia stacjonarne	studia niestacjonarne	
Projektowanie graficznych interfejsów użytkownika	30	18	45	57	3
(CL) ćwiczenia laboratoryjne	30	18	45	57	3
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)			10	15	
Przygotowanie projektu (także zespołowego)			20	27	
Przygotowanie do kolokwium			15	15	
Zarządzanie zmianą i systemy kontroli wersji	10	6	20	24	1
(CAU) ćwiczenia audytorijne	10	6	20	24	1
Przygotowanie do zajęć			10	12	
Przygotowanie domowej pracy kontrolnej			10	12	
Inżynieria oprogramowania I	35	21	75	89	4
(W)wykład	15	9	35	41	2
Przygotowanie do zaliczenia z oceną			25	30	
Studiowanie literatury			10	11	
(CL) ćwiczenia laboratoryjne	20	12	40	48	2
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)			10	10	
Przygotowanie projektu (także zespołowego)			20	28	
Przygotowanie do kolokwium			10	10	
Inżynieria oprogramowania II	20	12	80	88	4
(CL) ćwiczenia laboratoryjne	20	12	80	88	4
Przygotowanie do zajęć (w tym rozwiązywanie zadań domowych)			20	20	
Przygotowanie projektu (także zespołowego)			40	48	

Przygotowanie do kolokwiów			20	20	
Razem	95	57	220	258	12

Metody dydaktyczne

- (W)wykład: wykład problemowy wspomagany pokazem multimedialnym
- (CL) ćwiczenia laboratoryjne: ćwiczenia praktyczne - rozwiązywanie zadań z wykorzystaniem komputera, metoda projektu, metoda problemowa
- (CAU) ćwiczenia audytoryjne: wykład wspierany prezentacjami multimedialnymi, metoda problemowa

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

- A. Wymagania formalne:
Algotytm i struktury danych, Programowanie I-V, Wstęp do gromadzenia i przetwarzania danych, Bazy danych
- B. Wymagania wstępne:
wiadomości i umiejętności z przedmiotów związanych z architekturą komputera i podstawami informatyki oraz wiedza z zakresu podstaw programowania

Cele modułu

- zdobycie wiedzy o rodzajach interakcji i typach interfejsów graficznych użytkownika
- wytworzenie umiejętności posługiwania się narzędziami do projektowania makiet
- nabycie umiejętności projektowania makiet GUI
- budowanie umiejętności projektowania skoncentrowanego na użytkownika.
- przekazanie podstawowej wiedzy o metodach przeprowadzania jakościowych badań z użytkownikami z zakresu testowania użyteczności i ewaluacji stworzonych rozwiązań GUI
- zapoznanie z cyklem życia oprogramowania i jego szczegółowymi etapami.
- wytworzenie kompetencji umożliwiających analizowanie i tworzenie wymagań dotyczących projektów informatycznych
- zapoznanie z pojęciami dotyczącymi projektowania (zwłaszcza z wykorzystaniem modelu obiektowego), wytwarzania i testowania oprogramowania oraz ewolucji oprogramowania
- nabycie podstawowych umiejętności w zakresie wykorzystywania wzorców projektowych
- zapoznanie z możliwościami wykorzystania narzędzi informatycznych wspierających definiowanie, prowadzenie i organizację projektów informatycznych
- wskazanie na wagę zapewniania jakości jako niezbędnego elementu w procesie wytwarzania oprogramowania.
- przekazanie wiedzy dotyczącej różnych metodyk prowadzenia projektu informatycznego.
- zapoznanie z wybranymi zasadami zarządzania przedsięwzięciami programistycznymi
- zapoznanie się z podstawowymi pojęciami i zagadnieniami związanymi z zarządzaniem zmianą w projekcie informatycznym oraz systemami kontroli wersji
- promowanie wiedzy i umiejętności związanych z pracą w zespołach projektowych

Treści programowe

Projektowanie graficznych interfejsów użytkownika

1. Podstawowe pojęcia z zakresu UI, UX oraz projektowania zorientowanego na użytkownika (User Centered Design).
2. Zasady projektowania aplikacji zawierających graficzny interfejs użytkownika oraz mechanizmy generowania zdarzeń i sposoby ich obsługi na różnych platformach (desktop, mobile, web).
3. Reguły projektowania perswazyjnego oraz prewencyjnego wraz z przykładami.
4. Architektura informacji - znaczenie, podstawowe reguły i koncepcje.
5. Rodzaje interakcji i GUI z uwzględnieniem interfejsów aplikacji internetowych, mobilnych, konsolowych, dotykowych, okienkowych.
6. Narzędzia oraz techniki wspomagające projektowanie interfejsów.
7. Tworzenie makiet low-fi oraz hi-fi z uwzględnieniem dobrych praktyk stosowanych w projektowaniu UI uwzględniającym UCD.
8. Projektowania nawigacji, interakcji i metod dialogu z użytkownikami.
9. Projektowanie prezentacji informacji, pomocy dla użytkownika, dashboardów i tabel.
10. Przegląd elementów GUI (kontrolki) ze szczególnym uwzględnieniem ich wad i zalet obejmujących aspekty funkcjonalne oraz kontekst użyteczności.
11. Testy użyteczności - korzyści, sposoby i narzędzia do ich przeprowadzania.
12. Wprowadzenie do techniki RWD (Responsive Web Design) jako standardu projektowania interfejsów stron internetowych przeznaczonych dla różnego typu urządzeń (mobile, desktop).
13. Podstawy projektowania wizualnego (grafika i jej wpływ na odbiór informacji, typografia, ikonografia, teoria kolorów).
14. Rola brandingu i identyfikacji wizualnej w interfejsie użytkownika.

15. Dark patterns a etyka w projektowaniu aplikacji
16. Case Study - analiza współczesnych trendów w projektowaniu interfejsów użytkownika

Zarządzanie zmianą i systemy kontroli wersji

1. Zarządzanie zmianą w projekcie informatycznym.
2. Śledzenie błędów i zmian.
3. Kontrola wersji. Systemy kontroli wersji -zastosowanie i odmiany.
4. Przykładowe systemy kontroli wersji (np. **RCS, CVS, GIT, Mercurial**).

Inżynieria oprogramowania I

1. Przedmiot inżynierii oprogramowania.
2. Cykl życia oprogramowania na poziomie ogólnym.
3. Analiza i inżynieria wymagań. Zbieranie, porządkowanie, wycena i poziomy wymagań. Wymagania pozafunkcjonalne dotyczące systemów informatycznych (wydajność, użyteczność, niezawodność, bezpieczeństwo).
4. Kontekst biznesowy projektów informatycznych. Analiza wymagań biznesowych.
5. Studium wykonalności projektów informatycznych.
6. Sposobu pozyskiwania wymagań. Specyfikacja wymagań. Przypadki użycia. Modelowanie i dokumentowanie wymagań. Wzorzec SRS.
7. Model obiektowy w programowaniu. Najpopularniejsze wzorce projektowe w programowaniu obiektowym.
8. Sposoby projektowania rozwiązania informatycznego na bazie wymagań na oprogramowanie - wprowadzenie do języka UML. Diagramy UML.

Inżynieria oprogramowania II

1. Narzędzia CASE do etapu analitycznego i do projektowania oprogramowania.
2. Zapewnianie jakości na etapie analizy, projektowania i wytwarzania oprogramowania.
3. Metody formalne i empiryczne oceny jakości oprogramowania. Testowanie. Kategorie testów. Organizacja i zasady testowania oprogramowania.
4. Tworzenie przykładowych plików konfiguracyjnych i instalacyjnych – elementy wiedzy o zarządzaniu konfiguracją oprogramowania.
5. Wybrane problemy związane z wdrożeniem oprogramowania.
6. Metodyki tworzenia oprogramowania. Klasyczny model kaskadowy i metodyki zwinne. PRINCE2, SCRUM oraz XP-uwagi ogólne.
7. Elementy wiedzy o prowadzeniu przedsięwzięć informatycznych. Struktury organizacyjne w projektach informatycznych.
8. Ewolucja projektów informatycznych. Zapewnienie jakości projektu informatycznego w kontekście jego cyklu życia. Rola refaktoryzacji.

Efekty kształcenia

Wiedza

W_01 zna podstawowe pojęcia z zakresu projektowania interfejsów użytkownika oraz rodzaje interakcji z użytkownikiem.

W_02 zna zasady projektowania aplikacji zawierających graficzny interfejs użytkownika oraz mechanizmy generowania zdarzeń i sposoby ich obsługi uwzględniające reguły projektowania skoncentrowanego na użytkowniku.

W_03 zna popularne elementy GUI.

W_04 opisuje etapy tworzenia i funkcjonowania oprogramowania, także ewolucję oprogramowania

W_05 opisuje najważniejsze wymagania dotyczące systemów informatycznych

W_06 zna metody modelowania wymagań, projektowania i wytwarzania oprogramowania w tym uwzględniające wykorzystanie wzorców projektowych oraz wspomaganie procesu tworzenia oprogramowania przez narzędzia informatyczne

W_07 charakteryzuje formalne i empiryczne metody oceny jakości

Sposób zaliczenia oraz formy i podstawowe kryteria oceny/wymagania egzaminacyjne

A. Sposób zaliczenia

a) Przedmiot Projektowanie graficznych interfejsów użytkownika

CL – zaliczenie z oceną

b) Przedmiot Zarządzanie zmianą i systemy kontroli wersji

CAU – zaliczenie z oceną

c) Przedmiot Inżynieria Oprogramowania-I

W – zaliczenie z oceną

<p>oprogramowania, w tym organizację i zasady testowania oprogramowania</p> <p>W_08 ocenia jakość systemów informatycznych ze szczególnym uwzględnieniem interfejsu użytkownika</p> <p>W_09 opisuje najważniejsze metodyki realizacji projektów informatycznych, w tym możliwości ich doboru do konkretnych przedsięwzięć</p> <p>W_10 opisuje zasady i struktury organizacyjne charakterystyczne dla projektów informatycznych</p> <p>W_11 zna podstawowe pojęcia związane z zarządzaniem zmianą w projekcie informatycznym.</p> <p>W_12 zna zalety i wady różnych odmian systemów kontroli wersji</p> <p>Umiejętności</p> <p>U_01 posługuje się narzędziami i technikami wspomagającymi projektowanie UI</p> <p>U_02 umie zaprojektować graficzny interfejs użytkownika.</p> <p>U_03 przygotowuje makiety low-fi oraz hi-fi z uwzględnieniem dobrych praktyk stosowanych w projektowaniu UI uwzględniającym UCD</p> <p>U_04 dokonuje analizy wymagań dotyczących projektu interfejsu graficznego</p> <p>U_05 tworzy specyfikację wymagań dla projektowanego systemu informatycznego wykorzystując różne techniki dokumentowania wymagań</p> <p>U_06 konstruuje diagramy UML związane z modelowaniem oprogramowania na bazie wymagań</p> <p>U_07 wykorzystuje wzorce projektowe oraz narzędzia CASE do wspomagania procesu analizy i projektowania oprogramowania.</p> <p>U_08 stosuje przy modelowaniu rozwiązania informatycznego techniki obiektowe.</p> <p>U_09 konstruuje plan testowania oprogramowania zgodnie z uzyskaną wiedzą</p> <p>U_10 tworzy pliki będące elementem konfigurowania i zarządzania oprogramowaniem</p> <p>U_11 konstruuje plany wdrożenia oprogramowania</p> <p>U_12 dobiera odpowiednie metodyki realizacji projektów informatycznych do zaaranżowanych problemów</p> <p>U_13 potrafi stosować system kontroli wersji w projekcie</p> <p>Kompetencje społeczne</p> <p>K_01 wykazuje kreatywność przy rozwiązywaniu problemów</p> <p>K_02 pracuje w zespole</p> <p>K_03 ma świadomość ograniczeń wynikających z niedostatecznej wiedzy i umiejętności i rozumie potrzebę dalszego kształcenia, jest otwarty na poszukiwanie niestandardowych rozwiązań</p>	<p>CL – zaliczenie z oceną</p> <p>d) Przedmiot Inżynieria Oprogramowania II CL – zaliczenie z oceną</p> <p>B. Sposoby weryfikacji i oceny efektów</p> <p>a) Przedmiot Projektowanie graficznych interfejsów użytkownika</p> <p>(CL) Ćwiczenia laboratoryjne - kolokwium - efekty: W_01, W_02, W_03, U_01, U_02, U_03, U_04 - projekt zespołowy - efekty: U_01, U_02, U_03, U_04 K_01, K_02</p> <p>Maksymalna liczba punktów to x. Ocena K z kolokwium, projektu jest wyliczona według zasady:</p> <table border="0"> <tr> <td>K ∈ [0% x, 50% x)</td> <td>niedostateczna</td> </tr> <tr> <td>K ∈ [50% x, 60% x)</td> <td>dostateczna</td> </tr> <tr> <td>K ∈ [60% x, 70% x)</td> <td>dostateczna plus</td> </tr> <tr> <td>K ∈ [70% x, 80% x)</td> <td>dobra</td> </tr> <tr> <td>K ∈ [80% x, 90% x)</td> <td>db plus</td> </tr> <tr> <td>K ∈ [90% x, 100% x]</td> <td>bardzo dobra</td> </tr> </table> <p>Ocena zaliczenia ćwiczeń jest obliczona jako średnia arytmetyczna ocen z kolokwium pisemnego oraz oceny z projektu.</p> <p>Końcowa ocena z zaliczenia przedmiotu jest oceną za ćwiczenia laboratoryjne.</p> <p>b) Przedmiot Zarządzanie zmianą i systemy kontroli wersji</p> <p>(CAU) - domowa praca kontrolna - efekty: W_11, W_12, U_13, K_01</p> <p>Maksymalna liczba punktów to a. Ocena K z domowej pracy kontrolnej jest wyliczona według zasady:</p> <table border="0"> <tr> <td>K ∈ [0% a, 50% a)</td> <td>niedostateczna</td> </tr> <tr> <td>K ∈ [50% a, 60% a)</td> <td>dostateczna</td> </tr> <tr> <td>K ∈ [60% a, 70% a)</td> <td>dostateczna plus</td> </tr> <tr> <td>K ∈ [70% a, 80% a)</td> <td>dobra</td> </tr> <tr> <td>K ∈ [80% a, 90% a)</td> <td>db plus</td> </tr> <tr> <td>K ∈ [90% a, 100% a]</td> <td>bardzo dobra</td> </tr> </table> <p>Ocena zaliczenia ćwiczeń jest oceną z domowej pracy kontrolnej.</p> <p>c) Przedmiot Inżynieria Oprogramowania-I</p> <p>(W) Wykład –test końcowy – efekty: W_04, W_05, W_06, W_07, W_08, W_09, W_10, K_03</p>	K ∈ [0% x, 50% x)	niedostateczna	K ∈ [50% x, 60% x)	dostateczna	K ∈ [60% x, 70% x)	dostateczna plus	K ∈ [70% x, 80% x)	dobra	K ∈ [80% x, 90% x)	db plus	K ∈ [90% x, 100% x]	bardzo dobra	K ∈ [0% a, 50% a)	niedostateczna	K ∈ [50% a, 60% a)	dostateczna	K ∈ [60% a, 70% a)	dostateczna plus	K ∈ [70% a, 80% a)	dobra	K ∈ [80% a, 90% a)	db plus	K ∈ [90% a, 100% a]	bardzo dobra
K ∈ [0% x, 50% x)	niedostateczna																								
K ∈ [50% x, 60% x)	dostateczna																								
K ∈ [60% x, 70% x)	dostateczna plus																								
K ∈ [70% x, 80% x)	dobra																								
K ∈ [80% x, 90% x)	db plus																								
K ∈ [90% x, 100% x]	bardzo dobra																								
K ∈ [0% a, 50% a)	niedostateczna																								
K ∈ [50% a, 60% a)	dostateczna																								
K ∈ [60% a, 70% a)	dostateczna plus																								
K ∈ [70% a, 80% a)	dobra																								
K ∈ [80% a, 90% a)	db plus																								
K ∈ [90% a, 100% a]	bardzo dobra																								

Oceną zaliczenia wykładu jest ocena uzyskana za test końcowy -ocena A.

Maksymalnie w teście można uzyskać x punktów, a ocena A jest ustalana na podstawie następujących kryteriów.

A ∈ [0% x, 50% x)	niedostateczna
A ∈ [50% x, 60% x)	dostateczna
A ∈ [60% x, 70% x)	dostateczna plus
A ∈ [70% x, 80% x)	dobra
A ∈ [80% x, 90% x)	db plus
A ∈ [90% x, 100% x]	bardzo dobra

(CL) Ćwiczenia laboratoryjne

- praca projektowa nr 1 realizowana podczas zajęć dotycząca studium wykonalności projektów informatycznych - efekty: U_05 (20%)

-praca projektowa nr 2 realizowana w domu dotycząca tworzenia specyfikacji wymagań- efekty: U_05 (30%)

- praca projektowa nr 3 (zespołowa) realizowana w domu dotycząca modelowania oprogramowania na bazie wymagań - efekty: U_06, U_07,U_08, K_01, K_02 (50%)

Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y.

W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:

P ∈ [0% y, 50% y)	niedostateczna
P ∈ [50% y, 60% y)	dostateczna
P ∈ [60% y, 70% y)	dostateczna plus
P ∈ [70% y, 80% y)	dobra
P ∈ [80% y, 90% y)	db plus
P ∈ [90% y, 100% y]	bardzo dobra

Końcowa ocena z zaliczenia przedmiotu jest wyliczona jako średnia ważona ocen otrzymanych za wykład (A) i ćwiczenia laboratoryjne (B), dla których wagami są przypisane im liczby punktów ECTS.

d) Przedmiot Inżynieria Oprogramowania-II

(CL) Ćwiczenia laboratoryjne

- kolokwium dotyczące problematyki tworzenia planu testowego oraz związane z tworzeniem plików konfiguracyjnych – efekty: U_09, K_01 (40 %)

	<p>- praca projektowa zespołowa realizowana w domu dotycząca planu wdrożeniowego dla oprogramowania oraz doboru metodyki jego tworzenia - efekty: U_11, U_12,, K_01, K_02 (60%)</p> <p>Każda z form oceny CL jest punktowana, a suma punktów możliwych do uzyskania to y. W nawiasach przy każdej z form oceniania CL podano jaki procent y można uzyskać maksymalnie z tej formy oceniania. Ocena B dla zaliczenia CL jest ustalana na podstawie uzyskanej przez studenta sumy punktów P według zasady:</p> <table data-bbox="906 560 1514 784"> <tr> <td>$P \in [0\% y, 50\% y)$</td> <td>niedostateczna</td> </tr> <tr> <td>$P \in [50\% y, 60\% y)$</td> <td>dostateczna</td> </tr> <tr> <td>$P \in [60\% y, 70\% y)$</td> <td>dostateczna plus</td> </tr> <tr> <td>$P \in [70\% y, 80\% y)$</td> <td>dobra</td> </tr> <tr> <td>$P \in [80\% y, 90\% y)$</td> <td>db plus</td> </tr> <tr> <td>$P \in [90\% y, 100\% y]$</td> <td>bardzo dobra</td> </tr> </table> <p>Końcową oceną z zaliczenia przedmiotu jest ocena za ćwiczenia laboratoryjne.</p> <p>Ocena modułu jest średnią ważoną z ocen końcowych dla wszystkich przedmiotów. Wagami jest łączna liczba punktów ECTS dla każdego przedmiotu.</p>	$P \in [0\% y, 50\% y)$	niedostateczna	$P \in [50\% y, 60\% y)$	dostateczna	$P \in [60\% y, 70\% y)$	dostateczna plus	$P \in [70\% y, 80\% y)$	dobra	$P \in [80\% y, 90\% y)$	db plus	$P \in [90\% y, 100\% y]$	bardzo dobra
$P \in [0\% y, 50\% y)$	niedostateczna												
$P \in [50\% y, 60\% y)$	dostateczna												
$P \in [60\% y, 70\% y)$	dostateczna plus												
$P \in [70\% y, 80\% y)$	dobra												
$P \in [80\% y, 90\% y)$	db plus												
$P \in [90\% y, 100\% y]$	bardzo dobra												

Matryca efektów kształcenia dla przedmiotu		
Numer (symbol) efektu kształcenia	Odniesienie do efektów kształcenia dla programu	Odniesienie do charakterystyki drugiego stopnia PRK dla obszaru/ obszarów
W_01	K1_W21	P6S_WG
W_02	K1_W21	P6S_WG
W_03	K1_W21	P6S_WG
W_04	K1_W23	P6S_WG
W_05	K1_W23	P6S_WG
W_06	K1_W18, K1_W21, K1_W23	P6S_WG
W_07	K1_W13, K1_W21, K1_W23	P6S_WG, P6S_WK
W_08	K1_W13, K1_W21, K1_W23	P6S_WG ,P6S_WK
W_09	K1_W23	P6S_WG
W_10	K1_W11, K1_W13, K1_W23	P6S_WG ,P6S_WK
W_11	K1_W21	P6S_WG
W_12	K1_W21	P6S_WG
U_01	K1_U04, K1_U32	P6S_UW
U_02	K1_U05, K1_U08, K1_U09, K1_U19	P6S_UW

U_03	K1_U04, K1_U05, K1_U32	P6S_UW
U_04	K1_U06, K1_U09	P6S_UW
U_05	K1_U30	P6S_UW
U_06	K1_U04, K1_U30, K1_U31	P6S_UW
U_07	K1_U04, K1P_U32	P6S_UW
U_08	K1_U26	P6S_UW
U_09	K1_U14, K1_U27	P6S_UW
U_10	K1_U28, K1_U32	P6S_UW
U_11	K1_U08, K1_U27	P6S_UW
U_12	K1_U06, K1_U08, K1_U19, K1_U31	P6S_UW
U_13	K1_U04, K1_U32	P6S_UW
K_01	K1K02, K1_08	P6S_KO
K_02	K1_K04, K1_K07, K1_K08	P6S_KR, P6S_KO
K_03	K1_K01, K1_08	P6S_KK, P6S_KO

Wykaz literatury

A. Literatura wymagana do ostatecznego zaliczenia zajęć (zdania egzaminu):

1. Anderson J., McRee J., Wilson, R. Effective UI. *The Art of Building Great User Experience in Software*, Helion 2010
2. Beatty J., Wieggers K. *Specyfikacja oprogramowania. Inżynieria wymagań*, Helion, Gliwice 2014
3. Colborne G., *Prostota i użyteczność. Projektowanie rozwiązań internetowych, mobilnych i interaktywnych*, Helion 2011
4. Chrabski B., Żmitrowicz K. *Inżynieria wymagań w praktyce* PWN, Warszawa 2015
5. Krug S., *Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych*, Wydanie III, Helion 2014
6. Sacha K. *Inżynieria oprogramowania*, PWN, Warszawa 2010
7. Shalloway A., Trott J.R. *Programowanie zorientowane obiektowo. Wzorce projektowe*. Helion, Gliwice 2005
8. Tidwell J., *Projektowanie interfejsów. Sprawdzone wzorce projektowe*, Helion 2012
9. Wrycza S., Marcinkowski B., Wyrzykowski K. *Język UML 2.0 w modelowaniu systemów informatycznych, modelowanie biznesowe, metodyki projektowe oparte na UML, narzędzia CASE*, Helion, Gliwice 2005

B. Literatura uzupełniająca

1. Adzic G. *Specyfikacja na przykładach. Poznaj zwinne metody pracy i dostarczaj właściwe oprogramowanie.*, Helion, Gliwice 2014
2. Allen J., Chudley J., *Projektowanie witryn internetowych User eXperience. Smashing Magazine*, Helion 2013
3. Beck K., Cynthia A., *Wydajne programowanie – Extreme Programming*, Mikom 2005
4. Gajda W., *Git. Rozproszony system kontroli wersji*, Helion 2013
5. Hoekman jr R., *Magia interfejsu. Praktyczne metody projektowania aplikacji internetowych*, Helion 2010
6. Jaskiewicz A. *Inżynieria oprogramowania*, Helion. Gliwice 1997
7. Matt W., *The Smashing Book #2*, Helion, 2012
8. Nielsen J., Budi R., *Funkcjonalność aplikacji mobilnych. Nowoczesne standardy UX i UI*, Helion 2013
9. Phillips J., *Zarządzanie projektami IT*, Wydanie III, Helion Gliwice 2011,

Kontakt

dr inż. Zbigniew Ledóchowski, dr Ryszard Motyka
 zbigniew.ledochowski@apsl.edu.pl, ryszard.motyka@apsl.edu.pl