

Inżynieria oprogramowania II

Wprowadzenie i sprawy organizacyjne



Jerzy Nawrocki
www.cs.put.poznan.pl/jnawrocki/io

Inżynieria oprogramowania II

Definicja

inżynieria

Zastosowanie

- systematycznego,
- zdyscyplinowanego,
- ilościowego

podjęcia do

- rozwoju,
- eksploatacji i
- utrzymania

oprogramowania.


IEEE Std 610.12-1990
IEEE Standard Glossary of Software Eng. Terminology

Wprowadzenie i sprawy org. (2)



Inżynieria oprogramowania II

Cele przedmiotu



Przygotowanie do pracy inżynierskiej


Przekazanie podstawowej wiedzy z zakresu inżynierii oprogramowania

Wprowadzenie i sprawy org. (3)

Inżynieria oprogramowania II

Plan wykładu

- Praca inżynierska
- Minimum programowe
- Plan wykładów
- Regulamin przedmiotu



Wprowadzenie i sprawy org. (4)

Inżynieria oprogramowania II

Harmonogram

Do 30. kwietnia	Promotorzy przedstawiają propozycje tematów prac inżynierskich.
Do 30. maja	Zespoły studenckie wyrażają swoje zainteresowanie wybranymi tematami a promotorzy wybierają spośród zgłaszających się zespołów te, które będą realizowały ich tematy.
Do 30. czerwca	Inicjowanie projektu
Październik - listopad	Zespół realizuje pierwszy etap prac.
Do 15. grudnia	Promotorzy przedstawiają dziekanowi krótką ocenę postępów.
Grudzień - styczeń	Zespół realizuje drugi etap prac i oddaje pracę dyplomową.
Do 30. stycznia	Dziekan powołuje komisje i publikuje szczegółowy harmonogram obron.
Luty	Obrony prac dyplomowych

Wprowadzenie i sprawy org. (5)

Inżynieria oprogramowania II

Rekomendowany spis treści pracy inżynierskiej



- 1 Wprowadzenie
- 2 Opis procesów biznesowych
- 3 Specyfikacja wymagań funkcjonalnych
- 4 Specyfikacja wymagań pozafunkcyjnych
- 5 Architektura systemu
- 6 Opis implementacji
- 7 Zapewnianie jakości
- 8 Zarządzanie zmianami
- 9 Weryfikacja celów biznesowych
- 10 Zebrane doświadczenia

Dodatki

Wprowadzenie i sprawy org. (6)

Inżynieria oprogramowania II

Computing Curricula 2001

Computing Curricula 2001
Computer Science



— Final Report —
(December 15, 2001)

- Struktury dyskretne
- Podstawy programowania
- Algorytmy i złożoność
- Architektura sys. komputerowych
- Systemy operacyjne
- Technologie sieciowe
- Języki i paradygmaty program.
- Komunikacja człowiek-komputer
- Grafika komputerowa
- Sztuczna inteligencja
- Bazy danych
- Problemy społeczne i zawodowe
- Inżynieria oprogramowania**
- Nauki obliczeniowe

Wprowadzenie i sprawy org. (7)

Inżynieria oprogramowania II

Inżynieria oprogramowania

Computing Curricula 2001
Computer Science

— Final Report —
(December 15, 2001)

Obligatoryjne

Opcjonalne

Wprowadzenie i sprawy org. (8)

Inżynieria oprogramowania II

Inżynieria oprogram.



Wymagania	Projektowanie
Walidacja	Ewolucja
Procesy	Zarządzanie
Narzędzia	API
M.formalne	Sys. specjalne
Komponenty	Niezawodn.

Wprowadzenie i sprawy org. (9)


Inżynieria oprogramowania II

Plan wykładów

22.02	11:45	1	Wprowadzenie
29.02	11:45	2	API (A. Jaskiewicz)
07.03	11:45	3	Refaktoryzacja kodu w IBM Eclipse (B. Walter)
14.03	11:45	4	Model-Driven Architecture (Ł. Olek)
21.03			<i>Wielki Piątek</i>
28.03	11:45	5	Wprowadzenie do XPrince
04.04			<i>Finale Olimpiady Informatycznej</i>
11.04	11:45	6	Przypadki użycia
18.04	11:45	7	Wymagania pozafunkcjonalne
25.04	11:45	8	Analiza procesów biznesowych
02.05			<i>Dzień Rektorski (1-3 maja)</i>
09.05	11:45	9	Architektura oprogramowania
09.05	13:30	10	Kontrola jakości

Inżynieria oprogramowania II

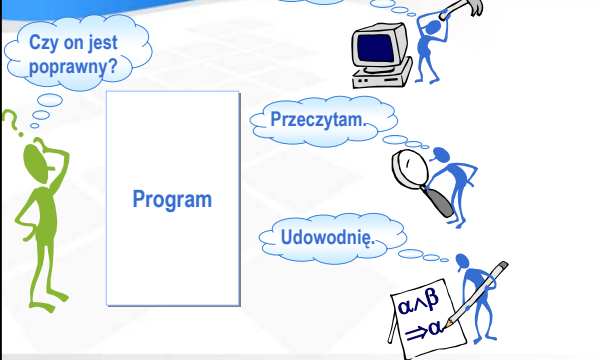
Artefakty



Wprowadzenie i sprawy org. (11)

Inżynieria oprogramowania II

Metody formalne



Wprowadzenie i sprawy org. (12)

Inżynieria oprogramowania II

Rodzaje kontroli jakości



Testowanie



Przeglądy

Wprowadzenie i sprawy org. (13)

Inżynieria oprogramowania II


Plan wykładów

22.02	11:45	1	Wprowadzenie
29.02	11:45	2	API (A. Jaszekiewicz)
07.03	11:45	3	Refaktoryzacja kodu w IBM Eclipse (B. Walter)
14.03	11:45	4	Model-Driven Architecture (Ł. Olek)
21.03			<i>Wielki Piątek</i>
28.03	11:45	5	Wprowadzenie do XPrinca
04.04			<i>Finale Olimpiady Informatycznej</i>
11.04	11:45	6	Przypadki użycia
18.04	11:45	7	Wymagania pozafunkcjonalne
25.04	11:45	8	Analiza procesów biznesowych
02.05			<i>Dzień Rektorski (1-3 maja)</i>
09.05	11:45	9	Architektura oprogramowania
09.05	13:30	10	Kontrola jakości
16.05			<i>30th International Conf. on Software Eng. - Lipsk</i>
23.05			<i>Dzień Rektorski (Boże Ciało)</i>
30.05	11:45	11	Zasady skutecznego działania
30.05	13:30	12	Planowanie projektu
06.06	11:45	13	Zarządzanie wydaniem

Inżynieria oprogramowania II

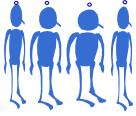
Najprostszy system zarządzania zmianami

Zmieńmy wymagania.



Klient

OK. OK. OK. OK.



Programiści

Wprowadzenie i sprawy org. (15)

Inżynieria oprogramowania II

Standard kodowania

Nagłówek programu

```

/*****
/* Program: KolorGraf
/* Autor: Jerzy Nawrocki
/* Data: 20.05.2006
/* Funkcja: Program koloruje węzły podanego grafu nieskierowanego w taki sposób, aby każda para węzłów połączonych luką miała różny kolor.
/* Wejście: Liczba naturalna N>0 określająca liczbę węzłów.
/* Sekwencja par liczb A,B (0 < A,B <= N). Para taka oznacza, że węzły A,B są połączone luką.
/* Wyjście: Minimalna liczba potrzebnych kolorów
/*Efekt ub: Brak
/* Uwagi: Program koloruje graf metoda brutalnej siły.
*****/

```

Wprowadzenie i sprawy org. (16)

Inżynieria oprogramowania II

Standard kodowania

Identyfikatory

```

int N; /* Liczba węzłów w grafie: 0 < N <= MaxN */
int A, B; /* Numery rozważanych węzłów: 0 < A,B <= N */
int Kolor; /* Numer rozważanego koloru: 0 < Kolor <= N */
int x41;
int PopKG;

```

Wprowadzenie i sprawy org. (17)

Inżynieria oprogramowania II

Standard kodowania

Wcięcia

```

while (scanf("%d %d", &A, &B)>0) {
    if (0<A && A<=N && 0<B && B<=N)
        Luk[A,B]= True;
        LiczbaLukow++;
}

```

Wprowadzenie i sprawy org. (18)

Inżynieria oprogramowania II

Standard kodowania

Komentowanie instrukcji

```

while (scanf("%d %d", &A, &B)>0) { /* Czytaj kolejne pary */
    /* wezlow A,B i dla kazdej,*/
    if (0<A && A<=N && 0<B && B<=N) /* o ile jest poprawna, */
        Luk[A,B]= True; /* zapamiętaj, że jest luk */
    /* między tymi wezłami. */
}

while (scanf("%d %d", &A, &B)>0) { /* Jak długo funkcja scanf,*/
    /* czytając wartości A,B, */
    /* zwraca wartość dodatnia,*/
    if (0<A && A<=N && 0<B && B<=N) /* sprawdź, czy A,B należą */
        /* do przedziału [1, N] i */
        /* zapamiętaj wartość True */
        Luk[A,B]= True; /* w tablicy Luk o współ- */
    /* rzędnych A,B. */
}

```

Wprowadzenie i sprawy org. (19)

Inżynieria oprogramowania II

Zarządzanie konfiguracją

Wprowadzenie i sprawy org. (20)

Inżynieria oprogramowania II

Plan wykładów

22.02	11:45	1	Wprowadzenie
29.02	11:45	2	API (A. Jaszkiwicz)
07.03	11:45	3	Refaktoryzacja kodu w IBM Eclipse (B.Walter)
14.03	11:45	4	Model-Driven Architecture (Ł. Olek)
21.03			<i>Wielki Piątek</i>
28.03	11:45	5	Wprowadzenie do XPrinse
04.04			<i>Finale Olimpiady Informatycznej</i>
11.04	11:45	6	Przypadki użycia
18.04	11:45	7	Wymagania pozafunkcjonalne
25.04	11:45	8	Analiza procesów biznesowych
02.05			<i>Dzień Rektorski (1-3 maja)</i>
09.05	11:45	9	Architektura oprogramowania
09.05	13:30	10	Kontrola jakości
16.05			<i>30th International Conf. on Software Eng. - Lipsk</i>
23.05			<i>Dzień Rektorski (Boże Ciało)</i>
30.05	11:45	11	Zasady skutecznego działania
30.05	13:30	12	Planowanie projektu
06.06	11:45	13	Zarządzanie wydaniem
13.06	11:45	14	Zamknięcie projektu

Wprowadzenie i sprawy org. (22)

Inżynieria oprogramowania II

Regulamin przedmiotu

Wymiar przedmiotu:
2w, 2lab, Egzamin, 5 ECTS

Obecność:
na wykładach – nieobowiązkowa
na ćwiczeniach – obowiązkowa

Sposób bieżącej kontroli wyników nauczania:
ćwiczenia: sprawdziany i bieżąca ocena

Wprowadzenie i sprawy org. (22)

Inżynieria oprogramowania II

Regulamin przedmiotu

Konsultacje:
J.Nawrocki: wt. (co drugi), godz. 11:15 – 12:00, pok. 109 CW
www.cs.put.poznan.pl/jnawrocki/lo
jerzy.nawrocki@put.poznan.pl

Oszacowanie pracochłonności (godz.):
30 wykłady
15 bieżąca powtórka treści wykładowych
35 studia literaturowe
30 laboratoria
30 przygotowanie do ćw. lab.
10 przygotowanie do egzaminu
150 RAZEM

Wprowadzenie i sprawy org. (23)

Inżynieria oprogramowania II

Egzamin

Termin: Ustali dziekanat

5-10 zadań; od 0 do 10 pkt za zadanie

Od 56% pkt 3.0; od 66% pkt 3.5; od 76% pkt 4.0, ...

Indeks lub legitymacja

Kalkulatory zabronione

Pisemny: 1 kartka „ściąg” formatu A4 (obie strony)

Wprowadzenie i sprawy org. (24)