**Nazwa przedmiotu:**

.Software Engineering 2

**Koordynator przedmiotu:**

dr inż. Krzysztof Kaczmarski

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

SE2

**Semestr nominalny:**

6 / rok ak. 2009/2010

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 0h |
| Ćwiczenia: | 0h |
| Laboratorium: | 45h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Theory of software engineering, Object oriented programming, Operating systems, Basic knowledge on networking

**Limit liczby studentów:**

**Cel przedmiotu:**

Students will be familiar with typical CASE tools: source code repositories, testing environments, modelling tools. Laboratories will also prepare them for team work, designing, costs forecasting and strict planning of software development process as well as following one explicit methodology.

**Treści kształcenia:**

Student teams develop software product in chosen methodology working under time limits and using team working tools. Whole process of software development is strictly controlled, from analysis to testing and deployment.

**Metody oceny:**

Students have to participate in all laboratories and prepare a single project following a chosen methodology working in teams of four persons. The project is evaluated every 3 weeks and gets points in several categories. All points collected during the semester influence the final mark but points from the last week are the most significant. Only state of the developed application at the last week of the semester is taken into consideration. There are no chances to modify or improve it afterwards.

**Egzamin:**

**Literatura:**

1.M. Fowler and K. Scott. UML distilled. Addison-Wesley, 1998 2.Grady Booch. Object-oriented analysis and design with applications. The Benjamin Cummings Publ. Comp., 1994. 3.David E. Brumbaugh. Object-oriented development : building CASE tools with C++. John Wiley, 1994. 4.Samuel D. Conte, H. E. Dunsmore, V. Y. Shen. Software Engineering Metrics and Models. The Benjamin Cummings Publ, 1986. 5.Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994 6.C. Ghezi, M. Jazayeri, D. Mandrioli. Fundamentals of Software Engineering. Prentice-Hall, 1991. 7.J. Rumbaugh, M. Blaha, W. Premerlani, W. Lorensen. Object-Oriented Modeling and Design. Prentice Hall International, 1991 8.Edward Yourdon, Carl Argila. Case studies in object-oriented analysis and design. Yourdon Press, 1996.

**Witryna www przedmiotu:**

**Uwagi:**

## Efekty przedmiotowe