

**POLYTECHNIC UNIVERSITY OF PUERTO RICO
COMPUTER ENGINEERING CURRICULUM
COURSE SYLLABUS**

Course code: CECS 4204	Course Title: Software Engineering	Classification: Requisite Course	Credits: 3
Pre-requisites: CECS 4202	Co-requisites: None	Schedule: Two two-hour lectures per week	

Course Description:

Familiarity with Windows or Unix applications, and knowledge of advanced object oriented programming is required. An entire software development cycle is executed on a small scale project. The Object Oriented analysis, design, coding, and testing techniques using the Unified Modeling Language (UML) are discussed in detail. Tools to support Software Engineering methods for Project Planning, Software Configuration Management, and OOD are demonstrated and used by the students to create sample Software Engineering work products. Some of these Computer Aided Software Engineering (CASE) tools include MS Project, WinCVS, Visual Source Safe, Visual Studio .NET.

Textbook:

Software Engineering: A Practitioner's Approach, 6th edition, Roger Pressman, McGraw-HILL, 2004

References:

1. The Deadline: A Novel about Project Management, Tom deMarco, Dorset House Publishing Company, 1997
2. Software Engineering, 8th ed.; Ian Sommerville; Addison Wesley, 2006
3. Design Patterns: Elements of Reusable Object-Oriented Software, 1st ed; Erich Gamma et al; Addison-Wesley, 1995
4. Software Engineering, 3rd ed.; Shari Lawrence Pfleger; Prentice Hall, 2005
5. Schaum's Outline of Software Engineering, 1st ed; David Gustafson; McGraw-Hill, 2002
6. The Mythical Man-Month, 1st ed; Frederick Brooks; Addison-Wesley, 1995
7. Object-Oriented Software Engineering: Using UML, Patterns and Java, 2nd ed; Bernd Bruegge; Prentice Hall, 2003
8. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed; Craig Larman; Prentice Hall, 2004

Contribution to Professional Component:

1. Engineering Science Topics – 0.0 credits
2. Engineering Design Topics – 3.0 credits

Contribution to Program Outcomes and IEEE Program Criteria:

This is a **key course** to measure outcomes (**in bold**): **a, c, d, e, f, g, j, k** (the course instructor has a commitment to measure the underlined part of these outcomes on a regular basis, and to show documented evidence to prove its achievement).

- **Outcome a:** Ability to apply knowledge of mathematics (**a.1**), science (**a.2**) and engineering (**a.3**)
- **Outcome c:** Ability to design a system (**c.1**), component (**c.2**) or process (**c.3**) to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability
- **Outcome d:** Ability to function on multidisciplinary teams – Team work is performed in the course (**d.1**), Students work on multidisciplinary teams (**d.2**)
- **Outcome e:** Ability to identify (**e.1**), formulate (**e.2**) and solve (**e.3**) engineering problems
- **Outcome f:** Understanding of professional (**f.1**) and ethical (**f.2**) responsibility
- **Outcome g:** Ability to communicate effectively – written (**g.1**) and oral (**g.2**) communication
- **Outcome j:** Knowledge of Contemporary Issues (**j**)
- **Outcome k:** Ability to use techniques (**k.1**), skills (**k.2**), and modern engineering tools (**k.3**) necessary for engineering practice

Course Objectives (Related to Program Outcomes and the IEEE Program Criteria):

1. Learns to select the software development model that best fits a given application development scenario (**c1-3, k1-3**)
2. Understands the importance and appropriateness of artifact generation and communication (**f1, g1, g2**)
3. Can design the principal artifacts of the process such as User Requirements, Requirements Specifications, Functional Specifications, Detailed Conceptual Design, Test Specifications, and others (**a1-3, c1-3**)
4. Is able to fulfill one cycle in software development process (**a1-3, c1-3, d1, e1-3, j**)
5. Is able to carry-out the principal umbrella activities related to Software Project Management, such as Software Quality Assurance planning, Risk Analysis, Project Scheduling, Software Configuration Management and Change Control (**e1-3**)
6. Knows the rights and duties of the roles, including the business client, involved in project (**d1, f1**)

**POLYTECHNIC UNIVERSITY OF PUERTO RICO
COMPUTER ENGINEERING CURRICULUM
COURSE SYLLABUS**

Topics Covered:

1. The Software Process.
 - a. General processes
 - b. Rational Unified Process in detail
 - c. Agile development methods overview
2. Software Engineering Practice
 - a. Requirements Engineering
 - b. Analysis and Design Phase using UML
 - c. Implementation and Documentation
 - d. Configuration Management
 - e. Web-application building
 - f. Rudimentary testing
3. Managing Software Engineering
 - a. Project Management
 - b. Metrics
 - c. Risks Management
 - d. Change Management
 - e. Artifact Management
 - f. Project Estimation

Evaluation Criteria:

Instruments for course evaluation will be used to measure established course objectives. These instruments will be associated with the Program Outcomes and IEEE Program Criteria that they can support. These are:

Daily Quizzes – (c1-3, f1, k1-3)

Project Presentations– (f1, g2, e1-3)

Project Documentation – (f1, g1, , a1-3, c1-3)

Project Implementation - (a1-3, c1-3, d1, e1-3, j, f1)

The course instructor will be responsible to show the course coordinator how these instruments will determine the final grade. The same person will also choose the most adequate instrument(s) from the above list to gather data and analyze it, and display valuable evidence of the Program Outcomes Assessment for which this is a **key course**.

Revised by:

Prof. Kay Berkling

Date:

June 27, 2007