



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Faculty of Electrical and Computer Engineering**  
**COURSE SYLLABUS**  
*Software Engineering I*

**1. CODE AND NUMBER OF CREDITS**

<b>CODE</b>	FIEC03046	
<b>NUMBER OF CREDITS: 5</b>	<b>Theoretical: 5</b>	<b>Practical: 0</b>

**2. COURSE DESCRIPTION**

This course covers various Software Engineering concepts, their applications, and their impact on the real world. In addition, this course allows the student to design a real software application, and to document a development proposal for a real customer. During this course the student should interact with the customer in order to elicit the requirements. The student should also validate and estimate the requirements, elaborate the project proposal, design the software applications to be developed and document all the activities and results of the development process. Standards de jure and de facto are used at each stage depending on the type of activity to be performed.

**3. PRE-REQUISITES AND CO-REQUISITES**

<b>PRE-REQUISITES</b>	FIEC05553 Database Systems I
<b>CO-REQUISITES</b>	FIEC05884 Web Applications Development

**4. CORE TEXT AND OTHER REQUIRED REFERENCES FOR THE TEACHING OF THE COURSE**

<b>CORE TEXT</b>	<ol style="list-style-type: none"> <li>1. Ian Sommerville, Software Engineering. 9th Edition, 2011, Addison-Wesley</li> <li>2. Dan Pilone and Russell Miles, Head First Software Development, 2008, O'Reilly Media</li> </ol>
<b>REFERENCES</b>	<ol style="list-style-type: none"> <li>1. Roger Pressman, Ingeniería de Software: Un Enfoque Práctico. Sexta Edición, 2005, McGraw-Hill</li> <li>2. E. Freeman, E. Robson, K. Sierra and B. Bates, Head First Design Patterns, 2004, O'Reilly Media</li> <li>3. Armando Fox and David Patterson, Engineering Long-Lasting Software, 2012, CreateSpace</li> <li>4. Craig Larman, UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado. Segunda Edición, 2003, Prentice Hall</li> <li>5. Mike Field and Laurie Keller, Project Management, 1997, Cengage Learning EMEA</li> <li>6. S. Robertson and J.C. Robertson, Mastering the Requirements Process. 2nd Edition, 2006, Addison Wesley</li> <li>7. ISO/IEC TR 19759:2005, Software Engineering - Guide to the Software Engineering Body of Knowledge, 2005, International Organization for Standardization</li> </ol>

**5. COURSE LEARNING OUTCOMES**

At the end of the course, the student will be able to:

1. Discuss the importance of software engineering and the impact of poor quality software on society.
2. Compare various existing software development models and their applicability.
3. Interact with a real customer in order to elicit software requirements, propose a work plan, and communicate progress.
4. Apply effort estimation techniques
5. Elaborate a budget and a schedule for the execution of the project
6. Apply in the development of a software product: techniques, tools, standards and methodologies that are used by practitioners
7. Document a software project (planning, requirements, design) following international guidelines
8. Analyze user requirements in order to design the software application to be developed
9. Evaluate if a software design follows principles of a good design
10. Use team-working skills such as: conducting meetings, organizing, time management, negotiation, resources allocation.



## 6. COURSE PROGRAM

- I. Software development seen as an engineering discipline and its importance (2 sessions - 5 hours).
  - Impact of poor quality software on society.
  - Quality criteria for software development.
  - Software development seen as an engineering discipline.
  - Software engineering code of ethics.
- II. Software development models (3 sessions – 7.5 hours).
  - The software development life cycle.
  - The water fall model.
  - The spiral model.
  - The incremental model.
  - Iterative development model.
- III. Vision and Project management (6 sessions – 15 hours).
  - Software project planning fundamentals.
  - Initial requirements gathering.
  - Effort estimation.
  - Introduction to software metrics.
  - Project scheduling.
  - Project team and resources allocation.
  - Software Project monitoring and control.
  - Risks management.
  - The vision document.
- IV. Software engineering tools (3 sessions – 7.5 hours).
  - Integrated development environment.
  - Tools for software project hosting and management.
  - Version control system.
- V. Requirements (6 sessions – 15 hours).
  - Requirements fundamentals.
  - Requirements elicitation.
  - Requirements specifications: best practices and documentation.
  - Requirements analysis: guidelines and techniques.
  - Requirements verification and validation.
- VI. Software design (8 sessions – 20 hours).
  - Software design fundamentals.
  - Software architecture.
  - Software design standards: UML.
  - Good design principles.
  - Refactoring.
  - Software design patterns.
  - Tools for software design and reverse engineering.

## 7. WORKLOAD: THEORY/PRACTICE

This course is held twice per week. Each of the two class sessions is two hours and a half long. Depending on the topic to be covered (i.e. Software engineering tools) classes might have a practical orientation. If a topic includes a practical component, then one of the two weekly sessions will be held in a computer lab of the department.

## 8. CONTRIBUTION OF THE COURSE TO THE EDUCATION OF THE STUDENT

This course introduces the students to software development models, effort estimation techniques and other basic software engineering concepts. Students have the opportunity to apply, in the modeling and design of real software systems, various concepts and techniques previously learnt in other courses such as: object oriented programming, and web applications development. The course reviews topics related to software project management that are useful for the IS project evaluation and formulation course. In addition, students should formally interview real customers in order to elicit software requirements. Then, the requirements are analyzed in order to model and design a software application of varying complexity. As part of the planning activities of the software project, students need to estimate the required effort to develop the software application. Students



also learn that mathematical models and regression analysis can be used to perform effort estimation in a software development project. The requirements are also analyzed to establish compromises with the customer; these compromises should consider time and resource constraints as well as the trade-offs of those compromises. Students prepare a vision document for their project. They also document the requirements, compromises, and design of their software projects, following software documentation standards. Students must develop an architectural and detailed design of their software projects. Students apply design principles to analyze the trade-offs of various design possibilities in order to propose an initial good software design that should be continuously improved during the course project. Students must respect the software engineering code of ethics as well as ESPOL's code of ethics. Finally, students work as part of a team for the course project.

BASIC TRAINING	PROFESSIONAL TRAINING	SOCIAL SKILLS DEVELOPMENT
	x	

**9. THE RELATIONSHIP BETWEEN THE LEARNING OUTCOMES OF THE COURSE AND THE LEARNING OUTCOMES OF THE DEGREE PROGRAM**

LEARNING OUTCOMES OF THE DEGREE PROGRAM	CONTRIBUTIO N (High, Medium, Low)	LEARNING OUTCOMES OF THE COURSE	THE STUDENT MUST:
a) An ability to apply knowledge of computing and mathematics appropriate to the discipline	.....	.....	.....
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	High	3,4,5,6,7,8,10	Document and validate customer's software requirements, defining their priorities and estimating the required effort.
c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	High	6,7,8,9	Create an architectural and detailed system designs.
d) An ability to function effectively on teams to accomplish a common goal	Medium	3,5,6,7,10	Be part of a project team to prepare and deliver a solution proposal that satisfies the customer's needs and to conduct a successful software development process.
e) An understanding of professional, ethical, legal, security and social issues and responsibilities	High	1,2,3,5,7,10	Work ethically when gathering confidential information from the customer. Demonstrate an understanding of the software engineering code of ethics.
f) An ability to communicate effectively with a range of audiences	High	3,4,5,7,8,10	Interview a real customer and present him the solution proposal. Present to the other students the project outcomes.
g) An ability to analyze the local and global impact of computing on individuals, organizations, and society	Medium	1,2,3,5,7,9	Propose a good software design that satisfies the customer needs.
h) Recognition of the need for and an ability to engage in continuing professional development	Low	1,6,9	Learn additional software design patterns besides those covered in class.
i) An ability to use current techniques, skills, and tools necessary for computing practice	Low	3,4,5,6,7,8,10	Estimate the time required for developing the project's software application using one of the methods covered in class.
j) An ability to lead others, manage or undertake projects.	Medium	3,4,5,6,7,10	Prepare a Project Schedule, execute and monitor it.



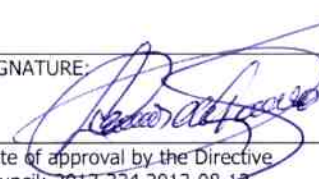
**10. EVALUATION IN THE COURSE**

Evaluation activities	
Exams	x
Tests	x
Homework/tasks	
Projects	x
Laboratory/Experiments	
Class participation	x
Visits	
Other	

**11. PERSON RESPONSIBLE FOR THE CREATION OF THE SYLLABUS AND THE DATE OF ITS CREATION**

<b>Created by</b>	Carlos Monsalve
<b>Date</b>	May 6, 2013

**12. APPROVAL**

ACADEMIC SECRETARY OF THE ACADEMIC DEPARTMENT	DIRECTOR OF TECHNICAL ACADEMIC SECRETARY
NAME: Sra. Leonor Caicedo G.	NAME: Ing. Marcos Mendoza V.
SIGNATURE: 	 ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL <hr style="border-top: 1px dashed black;"/> <b>Ing. Marcos Mendoza V.</b> DIRECTOR DE LA SECRETARIA TÉCNICA ACADÉMICA
Date of approval by the Directive Council: 2013-334 2013-08-12	

**13. VALIDITY OF THE SYLLABUS**

RESOLUTION OF THE POLYTECHNIC BOARD:	13-10-269
DATE:	2013-10-17