

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
COLLEGE OF ENGINEERING  
DEPARTAMENT OF COMPUTER SCIENCE  
ABET COURSE SYLLABI

**IIC3143 SOFTWARE DEVELOPMENT**

**Credits and contact hours:** 10 credits / 10 hours (3 h. Lectures; 1,5 Labs; 5,5 individual work hours per week))

**Instructor's name:** Rodrigo Sandoval

**Course coordinator's name** Rosa Alarcón

**Textbook:**

- Beck, K.; Fowler, M. (2000) Planning extreme programming. Addison-Wesley
- CMMI product team. (2010) CMMI for development, version1.3. CMU/SEI

**Course Catalog Description:** This course is based on knowledge acquired in the software engineering course, where the students learn the complete production of software systems. This course deepens in the development phase. Main topics covered are frameworks and methods, analysis practices, development practices, management practice, results and quality metrics and how to move from theory to practice.

**Prerequisite Courses:** IIC2143 Software Engineering

**Co-requisite Courses:** None

**Status in the Curriculum:** Required

**Course Learning Outcomes:**

1. To find requirements using need analysis techniques, objective analysis and use case analysis.
2. To organize and prioritize requirements.
3. To validate requirements using feasibility criterion, clarity, unambiguity, etc.
4. To represent functional and non functional requirements for different systems, using formal and informal techniques.
5. To specify and measure quality attributes.
6. To negotiate between different stakeholders to agree requirements

**Relation of Course to ABET  
Criteria:**

- b. Design and conduct experiments: analyze and interpret data
- e. Identify, formulate, and solve engineering problems
- i. Recognition of the need for, and an ability to engage in life-long learning
- j. Knowledge of contemporary issues
- k. Techniques, skills, and modern tools for engineering practice.

**Topics covered:**

1. Frameworks and methods: RUP and development process approach. MSF and role-based projects. CMMI and quality; SCRUM, XP and agile.
2. Analysis practices: Requirement discovery techniques, languages and models to represent requirements, need analysis, objectives and use cases; systems engineering context; specification and measures of performance, reliability, safety
3. Development practices: Pair programming, test driven development, code ownership, versioning, refactoring, incremental development.
4. Management practices: Team formation, value for the client versus developed functionality, risk management, change control, client and developer rights, iterations and "daily builds"; resource balance, time, functionality, and quality
5. Results and quality measure: documentation, review instance and strategies, formalization of milestones, repeatability.
6. From theory to practice: study cases analysis and aspects that influenced their success or failure.