SUBJECT PROGRAM

I. IDENTIFICATION OF THE SUBJECT

Subject: **Software Engineering**		Acronym: **INF-225**	Approval date 09/08/2015 (CC. DD. Agreement 18/2015)		
UTFSM Credits: **3**	Prerequisites: **INF-236**	Exam: **Does not have**	Faculty.		
SCT Credits: **5**			**Computer Science Department**		
Lecture Hours Weekly: **3**	Hours Assistantship Weekly: **1.5**	Weekly Laboratory Hours:	Semester in which it is taught		
			Odd _	Pair	Both
Formative axis: **Applied Engineering Sciences**					
Total time dedicated to the subject: **148 chronological hours.**					

Subject Description

The student acquires strategies for the systematic, effective and efficient construction of effective and efficient software systems. The student develops the ability to use engineering and management techniques for all disciplines of software production, carrying out partial team development of a specific application.

Entry requirements

- Analyze, model and design information systems.
- Understand technical texts in English.

Contribution to the graduation profile

- P1. Conceive, model, design, evaluate and implement alternative computer technology solutions, based on the analysis of specific problems in any business area.
- Q5. Act with autonomy, flexibility and initiative in their work.
- Q6. Incorporate a dynamic of permanent updating of their skills, typical of a rigorous, effective, and efficient task, based on their determination and tenacity.

Specific Competition.

- CE4. Develop, implement and maintain reliable, efficient and feasible software systems. **Elements of the competition.**
- EC42. Applies the fundamental concepts of processes and life cycles, quality, verification and validation, evolution and software architecture.
- EC43. Records and applies software requirements monitoring and verification techniques.

Transversal Competencies.

- B. Communicate oral and written information effectively within the organizations in which one works, as well as with entities in the environment.
- F. Develop their work with solid criteria that allow you to ensure quality from a systemic perspective.

Learning outcomes expected to be achieved in this subject

- **Analyzes** key software development processes , **applying them** in specific organizational contexts.
- **Plan** software development projects, **proposing** risk mitigation measures.
- **Make** decisions about software analysis and design techniques, **comparing** their suitability to specific systems, projects, and organizations.
- **Evaluates** software systems, **applying** verification and validation techniques.
- **Estimates** efforts, deadlines and complexity of the development of software systems, **applying** various techniques.

Thematic contents

- Software engineering: systematic, effective and efficient software construction.
- Software products: software quality, standards, verification and validation.

- Advanced development practices: requirements engineering; software design and architecture.
- Development projects: risk management; estimates of effort, time, complexity; evolution and transition; ALM (Application-Lifecycle Management).
- Development processes: meta-models; traditional processes; agile processes.
- Organizations: Software quality assurance (SQA), process improvement.
- Advanced topics: empirical verification, global software development, others.

Teaching and learning methodology.

- Expository classes supported by visual media.
- Learning based on tasks associated with a project.
- Study of cases.
- Discussion in classes about the state of the state in Software Engineering and Architecture.

Evaluation and grading of the subject. (Adjusted to Institutional Regulations-Regulation No. 1)

Approval requirements and qualification

Evaluation and qualification process:

Evaluation instruments.	Amount	%
Competition (NC)	**2**	**Four. Five**
Average biweekly reading checks (NL)	**around of 5 and 6**	**10**
Average number of tasks associated with a project (NP)	**1**	**Four. Five**

Final note:
NF = 0.45*NC+0.1*NL+0.45*NP

Resources for learning. Bibliography:

Guide Text	• Shari, L. Pfleeger. (2009). Software Engineering: Theory and Practice (4th Ed.), Prentice Hall.
Complementary or Optional	• Roger, S. Pressman. (2010). Software Engineering: a Practical Approach (7th Ed.), McGraw-Hill • Ian, Summerville. (2012). Software Engineering (9th Ed.), Pearson. • Virtual platform.

CALCULATION OF NUMBER OF HOURS OF DEDICATION - (SCT-Chile) - SUBJECT SUMMARY TABLE.

ACTIVITY	Number of hours of dedication		
	Number of hours per week	Number of weeks	Total number of hours
PRESENCE			
Lecture or theoretical classes	3	17	51
Assistantship/Exercises	1.5	17	25.5
Industrial visits (from Field)			
Laboratories / Workshop			
Evaluations (exams, others)	1.5	2	3
Others (specify)			
NO PRESENCE			
Assistantship			
Mandatory tasks	2	17	3. 4
Personal Study (Individual or group)	2	17	3. 4
Others (specify)			
TOTAL (HOURS)			**148**
Total number of TRANSFERABLE CREDITS			**5**