San Pablo Catholic University (UCSP) Undergraduate Program in Computer Science SILABO

Universidad Católica San Pablo

CS391. Software Engineering III (Mandatory)

1. General information

1.1 School : Ciencia de la Computación 1.2 Course : CS391. Software Engineering III

1.3 Semester : 7^{mo} Semestre.

1.4 Prerrequisites : CS292. Software Engineering II. (6^{th} Sem)

1.5 Type of course: Mandatory1.6 Learning modality: Face to face1.7 Horas: 2 HT; 2 HP;

1.8 Credits : 3

1.9 Plan : Plan Curricular 2016

2. Professors

Lecturer

• Gustavo Delgado Ugarte <ggdelgado@ucsp.edu.pe>

 MSc in Îngeniería del Software, Escuela Universitaria de Ingeniería Industrial, Informática y Sistemas - UTA, Chile, 2009.

3. Course foundation

Software development requires the use of best development practices, IT project management, equipment management And efficient and rational use of quality assurance frameworks, these elements are key and transversal during the whole productive process. The construction of software contemplates the implementation and use of processes, methods, models and tools that allow to achieve the realization of the quality attributes of a product.

4. Summary

1. Software Evolution 2. Software Project Management 3. Software Project Management 4. Software Processes 5. Estándares ISO/IEC

5. Generales Goals

- Understand and implement the fundamental concepts of project management and software equipment management.
- Understand the fundamentals of project management, including its definition, scope, and need for project management in the modern organization.
- Students have to understand the fundamental concepts of CMMI, PSP, TSP to be adopted in software projects.
- Describe and understand quality assurance models as a key framework for the success of IT projects.

6. Contribution to Outcomes

This discipline contributes to the achievement of the following outcomes:

- 1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 3) Communicate effectively in a variety of professional contexts. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

7. Content

UNIT 1: Software Evolution (12) Competences:	
Content	Generales Goals
 Software development in the context of large, pre-existing code bases Software change Concerns and concernlocation Refactoring Software evolution Characteristics of maintainable software Reengineering systems Software reuse Code segments Libraries and frameworks Components Product lines 	 Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Familiarity] Estimate the impact of a change request to an existing product of medium size [Usage] Use refactoring in the process of modifying a software component [Usage] Discuss the challenges of evolving systems in a changing environment [Familiarity] Outline the process of regression testing and its role in release management [Familiarity] Discuss the advantages and disadvantages of different types of software reuse [Familiarity]

UNIT 2: Software Project Management (10) **Competences:** Content **Generales Goals** • Team participation • Discuss common behaviors that contribute to the effective functioning of a team [Familiarity] - Team processes including responsabilities for task, meeting structure, and work schedule • Create and follow an agenda for a team meeting [Usage - Roles and responsabilities in a software team - Team conflict resolution • Identify and justify necessary roles in a software development team [Usage] - Risks associated with virtual teams (communication, perception, structure) • Understand the sources, hazards, and potential benefits of team conflict [Usage] • Effort estimation (at the personal level) • Apply a conflict resolution strategy in a team setting • Risk [Usage] - The role of risk in the lifecycle • Use an ad hoc method to estimate software develop-- Risk categories including security, safety, marment effort (eg, time) and compare to actual effort ket, financial, technology, people, quality, strucrequired [Usage] ture and process • List several examples of software risks [Familiarity] • Team management • Describe the impact of risk in a software development Team organization and decision-making lifecycle [Familiarity] Role identification and assignment • Describe different categories of risk in software sys- Individual and team performance assessment tems [Familiarity] • Project management • Demonstrate through involvement in a team project the central elements of team building and team man-- Scheduling and tracking agement [Usage] - Project management tools • Describe how the choice of process model affects - Cost/benefit analysis team organizational structures and decision-making processes [Familiarity] • Create a team by identifying appropriate roles and assigning roles to team members [Usage] • Assess and provide feedback to teams and individuals on their performance in a team setting [Usage] • Using a particular software process, describe the aspects of a project that need to be planned and moni-

Readings: Pressman and Maxim (2015), Sommerville (2017)

tored, (eg, estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and manage-

ment) [Familiarity]

 Competences: Content Generales Goals Software measurement and estimation techniques Software quality assurance and the role of measurements Risk Risk Risk identification and management Risk analysis and evaluation Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) Risk planning System-wide approach to risk including hazards associated with tools System-wide approach to risk including hazards associated with tools Explain how risk affects decisions in the software development process [Usage] Explain how risk affects decisions in the software development process [Usage] Demonstrate a systematic approach to the task of identifying hazards and risks in a particular situation [Usage] Apply the basic principles of risk management in a variety of simple scenarios including a security situation [Usage] Conduct a cost/benefit analysis for a risk mitigation approach [Usage] Identify and analyze some of the risks for an entire system that arise from aspects other than the soft-
 Software measurement and estimation techniques Software quality assurance and the role of measurements Risk Risk identification and management Risk analysis and evaluation Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) Risk planning System-wide approach to risk including hazards associated with tools Explain how risk affects decisions in the software development process [Usage] Explain how risk affects decisions in the software development process [Usage] Identify security risks for a software system [Usage] Demonstrate a systematic approach to the task of identifying hazards and risks in a particular situation [Usage] Apply the basic principles of risk management in a variety of simple scenarios including a security situation [Usage] Conduct a cost/benefit analysis for a risk mitigation approach [Usage] Identify and analyze some of the risks for an entire
 Software quality assurance and the role of measurements Risk Risk identification and management Risk analysis and evaluation Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) Risk planning System-wide approach to risk including hazards associated with tools System-wide approach to risk including hazards associated with tools Explain how risk affects decisions in the software development process [Usage] Identify risks and describe approaches to managing risk (avoidance, acceptance, transference, mitigation), and characterize the strengths and short-comings of each [Familiarity] Explain how risk affects decisions in the software development process [Usage] Identify security risks for a software system [Usage] Demonstrate a systematic approach to the task of identifying hazards and risks in a particular situation [Usage] Apply the basic principles of risk management in a variety of simple scenarios including a security situation [Usage] Conduct a cost/benefit analysis for a risk mitigation approach [Usage] Identify and analyze some of the risks for an entire
ware [Usage]

UNIT 4: Software Processes (12) Competences: Generales Goals Content • System level considerations, i.e., the iteraction of • Describe how software can interact with and particsoftware with its intended environment ipate in various systems including information management, embedded, process control, and communi-• Introduction to software process models (e.g., watercations systems [Usage] fall, incremental, agile) • Describe the relative advantages and disadvantages - Activities with software lifecycles among several major process models (eg, waterfall, iterative, and agile) [Usage] • Programming in the large vs. individual program-• Describe the different practices that are key components of various process models [Usage] • Evaluation of software process models • Differentiate among the phases of software develop-• Software quality concepts ment [Usage] • Process improvement • Describe how programming in the large differs from • Software process capability maturity models individual efforts with respect to understanding a large code base, code reading, understanding builds, • Software process measurements and understanding context of changes [Usage] • Explain the concept of a software lifecycle and provide an example, illustrating its phases including the deliverables that are produced [Usage] • Compare several common process models with respect to their value for development of particular classes of software systems taking into account issues such as requirement stability, size, and nonfunctional characteristics [Usage] • Define software quality and describe the role of quality assurance activities in the software process [Usage • Describe the intent and fundamental similarities among process improvement approaches [Usage] • Compare several process improvement models such as CMM, CMMI, CQI, Plan-Do-Check-Act, or ISO9000 [Usage] • Assess a development effort and recommend potential changes by participating in process improvement (using a model such as PSP) or engaging in a project retrospective [Usage] • Explain the role of process maturity models in process improvement [Usage] • Describe several process metrics for assessing and controlling a project [Usage] • Use project metrics to describe the current state of a project [Usage]

Readings: Pressman and Maxim (2015), Sommerville (2017)

Competences:	
Content	Generales Goals
ISO 9001:2001.ISO 9000-3.	• Learn and apply correctly standards and international standards . [Usage]
• ISO/IEC 9126.	
 ISO/IEC 12207. ISO/IEC 15939.	
 ISO/IEC 14598. ISO/IEC 15504-SPICE.	
• IT Mark.	
SCRUM.SQuaRE.	
• CISQ.	

8. Methodology

- 1. El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.
- 2. El profesor del curso presentará demostraciones para fundamentar clases teóricas.
- 3. El profesor y los alumnos realizarán prácticas
- 4. Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. Assessment

Continuous Assessment 1 : 20 %

Partial Exam : 30 %

Continuous Assessment 2 : 20 %

Final exam : 30 %

References

Pressman, Roger S. and Bruce Maxim (Jan. 2015). Software Engineering: A Practitioner's Approach. 8th. McGraw-Hill. Sommerville, Ian (Mar. 2017). Software Engineering. 10th. Pearson.