



University of Engineering and Technology
School of Computer Science
Syllabus of Course – Academic Period 2017-I

1. Code and Name: CS391. Software Engineering III

2. Credits: 4

3. Hours of theory and Lab: 2 HT; 4 HP;

4. Professor(s)

Meetings after coordination with the professor

5. Bibliography

[PM14] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2014.

[Som10] Ian Sommerville. *Software Engineering*. 9th. Addison-Wesley, Mar. 2010.

6. Information about the course

(a) **Brief description about the course** 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.

(b) **Prerequisites:** CS292. Ingeniería de Software II. (7^{mo} Sem)

(c) **Type of Course:** Elective

7. Competences

- 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.

8. Contribution to Outcomes

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

m) Transform your knowledge of the area of Computer Science into technological enterprises. (**Assessment**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Usage**)

9. Competences (IEEE)

- C7.** Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome c**
- C11.** Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).⇒ **Outcome i,k**
- C12.** Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).⇒ **Outcome j,m**
- C13.** Understanding the relationship between quality and lifecycle management⇒ **Outcome c,i,m**
- C18.** Ability to participate actively and as a member of a team. .⇒ **Outcome d**
- C19.** Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.⇒ **Outcome j**
- CS6.** Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.⇒ **Outcome c,i,m**
- CS7.** Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.⇒ **Outcome d,i,o**
- CS9.** Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context. ⇒ **Outcome c,d,m**

10. List of topics

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

11. Methodology and Evaluation

Methodology:

Theory Sessions:

The development of the theoretical sessions is focused on the student, through his active participation, solving problems related to the course with the individual contributions and discussing real cases of the industry. The students will develop throughout the course a project of application of the tools received in a company.

Lab Sessions:

Practical sessions are held in the laboratory. Laboratory practices are performed in teams to strengthen their communication. At the beginning of each laboratory the development of the practice is explained and at the end the main conclusions of the activity in group form are highlighted.

Oral Presentations :

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:

Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:

12. Content

Unit 1: Software Evolution (12)	
Competences Expected: C7, C11, C12, CS6	
Learning Outcomes	Topics
<ul style="list-style-type: none"> • 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] 	<ul style="list-style-type: none"> • Software development in the context of large, pre-existing code bases <ul style="list-style-type: none"> – Software change – Concerns and concernlocation – Refactoring • Software evolution • Characteristics of maintainable software • Reengineering systems • Software reuse <ul style="list-style-type: none"> – Code segments – Libraries and frameworks – Components – Product lines
Readings : [PM14], [Som10]	

Unit 2: Software Project Management (10)	
Competences Expected: C18, C19, CS7, CS9	
Learning Outcomes	Topics
<ul style="list-style-type: none"> • Discuss common behaviors that contribute to the effective functioning of a team [Familiarity] • Create and follow an agenda for a team meeting [Usage] • Identify and justify necessary roles in a software development team [Usage] • Understand the sources, hazards, and potential benefits of team conflict [Usage] • Apply a conflict resolution strategy in a team setting [Usage] • Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage] • List several examples of software risks [Familiarity] • Describe the impact of risk in a software development lifecycle [Familiarity] • Describe different categories of risk in software systems [Familiarity] • Demonstrate through involvement in a team project the central elements of team building and team management [Usage] • Describe how the choice of process model affects 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 monitored, (eg, estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management) [Familiarity] 	<ul style="list-style-type: none"> • Team participation <ul style="list-style-type: none"> – Team processes including responsibilities for task, meeting structure, and work schedule – Roles and responsibilities in a software team – Team conflict resolution – Risks associated with virtual teams (communication, perception, structure) • Effort estimation (at the personal level) • Risk <ul style="list-style-type: none"> – The role of risk in the lifecycle – Risk categories including security, safety, market, financial, technology, people, quality, structure and process • Team management <ul style="list-style-type: none"> – Team organization and decision-making – Role identification and assignment – Individual and team performance assessment • Project management <ul style="list-style-type: none"> – Scheduling and tracking – Project management tools – Cost/benefit analysis
Readings : [PM14], [Som10]	

Unit 3: Software Project Management (8)	
Competences Expected: C18, C19, CS7, CS9	
Learning Outcomes	Topics
<ul style="list-style-type: none"> • Track the progress of some stage in a project using appropriate project metrics [Usage] • Compare simple software size and cost estimation techniques [Usage] • Use a project management tool to assist in the assignment and tracking of tasks in a software development project [Usage] • Describe the impact of risk tolerance on the software development process [Assessment] • Identify risks and describe approaches to managing risk (avoidance, acceptance, transference, mitigation), and characterize the strengths and shortcomings 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 system that arise from aspects other than the software [Usage] 	<ul style="list-style-type: none"> • Software measurement and estimation techniques • Software quality assurance and the role of measurements • Risk <ul style="list-style-type: none"> – 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
Readings : [PM14], [Som10]	

Unit 4: Software Processes (12)	
Competences Expected: C7, C13, C19, CS6, CS7	
Learning Outcomes	Topics
<ul style="list-style-type: none"> • Describe how software can interact with and participate in various systems including information management, embedded, process control, and communications systems [Usage] • Describe the relative advantages and disadvantages among several major process models (eg, waterfall, iterative, and agile) [Usage] • Describe the different practices that are key components of various process models [Usage] • Differentiate among the phases of software development [Usage] • Describe how programming in the large differs from individual efforts with respect to understanding a large code base, code reading, understanding builds, 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 non-functional 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] 	<ul style="list-style-type: none"> • System level considerations, i.e., the interaction of software with its intended environment • Introduction to software process models (e.g., waterfall, incremental, agile) <ul style="list-style-type: none"> – Activities with software lifecycles • Programming in the large vs. individual programming • Evaluation of software process models • Software quality concepts • Process improvement • Software process capability maturity models • Software process measurements
Readings : [PM14], [Som10]	

Unit 5: Estándares ISO/IEC (6)**Competences Expected: C7, C13, C19, CS6, CS7****Learning Outcomes**

- Learn and apply correctly standards and international standards . [Usage]

Topics

- ISO 9001:2001.
- ISO 9000-3.
- ISO/IEC 9126.
- ISO/IEC 12207.
- ISO/IEC 15939.
- ISO/IEC 14598.
- ISO/IEC 15504-SPICE.
- IT Mark.
- SCRUM.
- SQuaRE.
- CISQ.

Readings : [Som10], [PM14]