



National University of Engineering (UNI)

School of Computer Science
Syllabus 2024-II

1. COURSE

CS391. Software Engineering III (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS391. Software Engineering III
2.2 Semester	: 7 th Semester.
2.3 Credits	: 3
2.4 Horas	: 2 HT; 2 HP;
2.5 Duration of the period	: 16 weeks
2.6 Type of course	: Mandatory
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS292. Software Engineering II. (6 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO 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.

5. 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. COMPETENCES

- 1) Analyze a complex computing problem and 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. TOPICS

Unit 1: Evolución de Software (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Desarrollo de Software en el contexto de código grande pre existente – Cambios de software – Preocupaciones y ubicación de preocupaciones – <i>Refactoring</i> • Evolución de Software. • Características de Software mantenible. • Sistemas de Reingeniería. • Reuso de Software. – Segmentos de código – Bibliotecas y <i>frameworks</i> – Componentes – Líneas de Producto	• Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Familiarizarse] • Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar] • Usar refactorización en el proceso de modificación de un componente de software [Usar] • Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Familiarizarse] • Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Familiarizarse] • Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Familiarizarse]
Readings : [PM15], [Som17]	

Unit 2: Gestión de Proyectos de Software (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• La participación del equipo: – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) • Estimación de esfuerzo (a nivel personal) • Riesgo. – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • Gestión de equipos: – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación • Gestión de proyectos: – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio	• Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Familiarizarse] • Crear y seguir un programa para una reunión del equipo [Usar] • Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] • Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] • Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] • Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] • Listar varios ejemplos de los riesgos del software [Familiarizarse] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Familiarizarse] • Describir las diferentes categorías de riesgo en los sistemas de software [Familiarizarse] • Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar] • Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Familiarizarse] • Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar] • Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar] • Usando un software particular procesar, describir los aspectos de un proyecto que necesita ser planeado y monitoreado, (ejemplo, estimar el tamaño y esfuerzo, un horario, reasignación de recursos, control de configuración, gestión de cambios, identificación de riesgos en un proyecto y gestión) [Familiarizarse]
Readings : [PM15], [Som17]	

Unit 3: Gestión de Proyectos de Software (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Software de medición y técnicas de estimación. • Aseguramiento de la calidad del software y el rol de las mediciones. • Riesgo. – Identificación de riesgos y gestión. – Análisis riesgo y evaluación. – La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo) – Planificación de Riesgo • En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.	• Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar] • Comparar las técnicas simples de tamaño de software y estimación de costos [Usar] • Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Evaluar] • Identificar riesgos y describir enfoques para manejar riesgos (evitar, aceptar, transferir, mitigar) y caracterizar fortalezas y defectos para cada uno [Familiarizarse] • Explicar cómo el riesgo afecta las decisiones en el proceso de desarrollo de software [Usar] • Identificar los riesgos de seguridad para un sistema de software [Usar] • Demostrar un enfoque sistemático para la tarea de identificar los peligros y riesgos en una situación particular [Usar] • Aplicar los principios básicos del manejo de riesgos en una variedad de escenarios simples incluyendo una situación de seguridad [Usar] • Dirigir un análisis de costo/beneficio para el enfoque de mitigación de riesgos [Usar] • Identificar y analizar alguno de los riesgos para un sistema entero que surgen de aspectos distintos del software [Usar]
Readings : [PM15], [Som17]	

Unit 4: Procesos de Software (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno. • Introducción a modelos del proceso de software (e.g., cascada, incremental, ágil): – Actividades con ciclos de vida de software. • Programación a gran escala versus programación individual. • Evaluación de modelos de proceso de software. • Conceptos de calidad de software. • Mejoramiento de procesos. • Modelos de madurez de procesos de software. • Mediciones del proceso de software.	• Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Usar] • Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar] • Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar] • Diferenciar entre las fases de desarrollo de software [Usar] • Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Usar] • Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar] • Comparar varios modelos comunes de procesos con respecto a su valor para el desarrollo de las clases particulares de sistemas de software, teniendo en cuenta diferentes aspectos tales como, estabilidad de los requisitos, tamaño y características no funcionales [Usar] • Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Usar] • Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Usar] • Comparar varios modelos comunes de procesos con respecto a su valor para el desarrollo de las clases particulares de sistemas de software, teniendo en cuenta diferentes aspectos tales como, estabilidad de los requisitos, tamaño y características no funcionales [Usar] • Evaluar un esfuerzo de desarrollo y recomendar cambios potenciales al participar en la mejora de procesos (usando un modelo como PSP) o involucración en una retrospectiva de un proyecto [Usar] • Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar] • Describir varias métricas de procesos para la evaluación y el control de un proyecto [Usar] • Usar las medidas en proyecto para describir el estado actual de un proyecto [Usar]
Readings : [PM15], [Som17]	

Unit 5: Estándares ISO/IEC (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• 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.	• Learn and apply correctly standards and international standards . [Usar]
Readings : [Som17], [PM15]	

8. WORKPLAN

8.1 Methodology

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

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

***** EVALUATION MISSING *****

10. BASIC BIBLIOGRAPHY

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.