

# Course Policies

## Lecture 1a

# Agenda

- Course Outline
- Marking Scheme
- Project Considerations
- Suggestions for Written Communications

# Course Outline

# Course Description

- This course addresses the importance of software engineering, the development life-cycle of a software product, and the application of a software development methodology.
  - Specifically, it makes emphasis on planning, management, requirements analysis and design of a software system of medium complexity.

# Course Description

- Technical and non-technical skills are developed or improved during the course.
  - Among the technical skills we have: the application of methods and tools to plan a software system, the analysis of the needs of a real customer, and the design of a software system: both architectural and detailed design.
  - The non-technical skills include teamwork; ethics in the software engineering profession; and oral and written communication

# Course Goals

- To identify a software development process model that adds value to a specific project by analysing the characteristics of existing models and the specificities of the project.
- To organize the development of a software system through the creation of work teams considering roles and responsibilities.
- To apply standards and best practices in requirements engineering for the generation of software requirements specifications for a system of medium complexity.
- To create design models, based on software requirements specifications, that represent the structure and behaviour of a software system.

# Course Content

- For the first evaluation:

<b>1. Procesos de software</b>
1.1. La programación individual vs. la programación de grandes sistemas.
1.2. Ingeniería de software: ¿Qué es? ¿Cuál es su importancia?
1.3. El ciclo de vida de desarrollo de software.
1.4. Introducción a los modelos de proceso de software (cascada, incremental, espiral, iterativos, ágiles).
1.5. Evaluación de modelos de proceso de software.
<b>4. Ingeniería de requerimientos</b>
4.1. Conceptos básicos de ingeniería de requerimientos de software: propiedades e importancia de requerimientos de software, tanto funcionales como no funcionales.
4.2. Técnicas para el levantamiento y análisis de requerimientos de software.
4.3. Descripción y documentación de requerimientos de software.
4.4. Prototipado de requerimientos funcionales.
4.5. Verificación de especificaciones de requerimientos funcionales de software.
<b>2. Administración de proyectos de software</b>
2.1. Roles y responsabilidades en un equipo de desarrollo de software.
2.2. Gestión del equipo de desarrollo de software: organización, asignación de recursos y responsabilidades.
2.3. Medición de software y estimación de esfuerzo.

# Course Content

- For the second evaluation:

<b>2. Administración de proyectos de software</b>
2.4. Análisis de riesgos.
2.5. Gestión del proyecto de software: planificación de actividades, monitoreo.
<b>3. Herramientas y ambientes de desarrollo de software</b>
3.1. Herramientas para la planificación y monitoreo de proyectos de desarrollo de software.
3.2. Herramientas para el modelamiento de prototipos o "mockups" de requerimientos funcionales de software.
3.3. Herramientas para el diseño de software.
<b>5. Diseño de software</b>
5.1. Revisión de principios y criterios de diseño de software.
5.2. Conceptos de arquitectura de software y estilos arquitectónicos (cliente-servidor, capas, tubos y filtros, etc.).
5.3. Modelos de diseño de software estructurales y de comportamiento usando el Lenguaje de Modelamiento Unificado (UML).
<b>6. Ética en la profesión</b>
6.1. El impacto de mal software en la sociedad.
6.2. Los principios éticos de la ingeniería de software (códigos de ética).
6.3. Dilemas éticos: argumentación ética, análisis de responsabilidades y consecuencias.



# Bibliography

- Sommerville, Ian. (2015). Software Engineering, Global Edition. (Tenth Edition).
- Pressman Roger, Maxim Bruce. (2019). Software Engineering: A Practitioner's Approach. (Ninth Edition).
- Pilone Dan, Miles, Russ. (2008). Head First Software Development: A Learner's Companion to Software Development. (1st Edition).

# Key Dates

- Project Deliveries:
  - 21 June 2023, First Project Delivery
  - 14 and 16 August 2023, Second Project Delivery
  - 6 September 2023, Third Project Delivery

# Marking Scheme

# Marking Scheme

- Self-registration in teams for workshops and the project.
- Dishonesty (including plagiarism) produces a mark of zero.
- Aula virtual records and reports all the individual activities of a student that is taking an evaluation.

# Marking Scheme

- The weighted sum of the marks is as follows:

Activity		First Evaluation	Second Evaluation	Third Evaluation	
EHD/Theoretical	Examination	35%	35%	35%	70%**
EHTA/Autonomous	Project <sup>δ</sup>	30% *	30% *	35% *	--
	Tests	05%	05%	--	--
EHP/Practical	Workshops	30%		--	

- \* An acceptance form signed by the client is compulsory.
- \*\* If student passed with the first and second evaluation.

# Project Considerations

# Project Considerations

- A software system must be developed for a real customer.
- The customer (company) cannot be (belong to) a member of your family.
- The software system must have a Web and mobile components.
- The project should be developed in the city of Guayaquil. The company/organization must have the infrastructure or resources required to put in operation the software system (Software Engineering 2).

# Project Considerations and Teamwork

- Teams composed by 4 people.
- Fill the form of project acceptance in Aula Virtual.
- The project must be completed in two academic terms (Software Engineering 1 and 2).
- The customer of your project must be committed to attend to scheduled meetings with the team members.
- Co-evaluation between team members.



# Project Considerations

- The learning of development tools are part of a self-learning process of the student that depends on the specific needs of their customer.
- You are expected to live a real experience and therefore you should satisfy the requirements of your customer.

# Co-Evaluation of Teamwork

- Individual grades on group projects will be determined based on a rubric.
  - For example, consider an individual assignment that result from group work or a group assignment (e.g., poster presentation) that earns a grade of 85%. A student who achieves a Level 5 rating will receive the full points (i.e., 85%) earned for the assignment. A student who achieves a Level 2 rating will receive 25% of the points earned for the assignment (i.e., 21.25%).

# Co-Evaluation of Teamwork

- **Level 5: 100% of the assignment grade**

- The team member participated fully in all discussions, contributing your own ideas and suggestions in each phase of the project.
- The team member participated fully in project activities (planning, data or information collection, data analysis, writing, presenting, etc.).
- The team member attended all group meetings.
- The team member completed all tasks that you agreed to conduct for the project/team.

# Co-Evaluation of Teamwork

- **Level 4: 75% of the assignment grade**

- The team member participated in each phase of the project but showed less initiative in contributing your ideas and suggestions.
- The team member usually participated in project activities, but not always.
- The team member usually attended team meetings.
- The team member usually completed tasks that you agreed to conduct for the project/team.

# Co-Evaluation of Teamwork

- **Level 3: 50% of the assignment grade**
  - The team member occasionally participated in the various phases of the project, generally as a “follower” rather than contributing to project development
  - The team member occasionally attended team meetings
  - The team member completed one or two minor tasks that you agreed to conduct for the project/team

# Co-Evaluation of Teamwork

- **Level 2: 25% of the assignment grade**

- The team member contributed very little to the project, showing no initiative
- The team member missed the vast majority of the team meetings
- The team member failed to complete the majority of tasks that you agreed to conduct for the project/team

- **Level 1: No credit**

- The team member did not participate in developing the project/presentation

# Project Reviews

- Reviews per sprint.
- Monitoring of deployments in production.
- Deployments in production are only accepted once the first sprint is completed.
- Each completed sprint must include:
  - An acceptance form signed by the client.
  - A record of each sprint review.
  - Software product deployed in production.

# Suggestions for Written Communications



# How to write an email appropriately?

- <http://es.wikihow.com/escribir-un-e-mail-formal>
- <http://www.ecoescritura.com/%C2%BFcomo-escribir-correos-electronicos/>
- [http://www.ehowenespanol.com/reglas-saludo-correo-electronico-lista\\_43535/](http://www.ehowenespanol.com/reglas-saludo-correo-electronico-lista_43535/)

# Equipos de trabajo

- Cada equipo deberá estar conformado de 4 personas.
- El equipo debe auto-agruparse en Aula Virtual.
- El líder del grupo debe crear un canal privado en Teams con el mismo del nombre del equipo creado en Aula Virtual.
- En dicho canal, el líder deberá agregar a las siguientes personas:
- profesora: Dra.Mónica Villavicencio C. [mvillavi@espol...](mailto:mvillavi@espol...)
- Técnico Docente: Ing. MercedesMawyin [mnmawyin@espol..](mailto:mnmawyin@espol..)
- Ayudante docente: Srta. Sandy Intriago [saintria@espol...](mailto:saintria@espol...)