Introduction to the LETI-ESOFT Course Unit

2025 – 2026

Alexandre Gouveia
School of Engineering
Polytechnic of Porto
aas@isep.ipp.pt

Context

- 3rd curricular semester (1st semester / 2nd year) of the Bachelor on Telecommunications and Informatics Engineering (LETI)

- Previous related course units (1st year)
  - APROG – Algorithms and Programming (1st semester)
  - FSOFT – Fundamentals of Software Development (2nd semester)

- Other related course units (2nd year)
  - BDAMD – Databases and Data Warehouses (3rd semester)
  - DSSMV – Software for Mobile Systems (3rd semester)
  - PSOFT – Software Project (4th semester)

Previous Knowledge Assumed

- Fundamental notions of algorithms (APROG)
  - Decompose a problem into smaller problems
  - Algorithms implementation in C through a procedural paradigm

- Basic notions of software development (FSOFT)
  - Knowledge and practice of the major software development activities
  - Adoption of the object-oriented (OO) paradigm
  - Usage of UML diagrams to elaborate some software artifacts
  - Software implementation in C++

ESOFT – General Goals

- To consolidate software development competencies previously acquired (APROG, FSOFT)

- To deepen and extend these competencies, namely regarding:
  - Iterative and Incremental (I&I) Software Development Process (SDP)
  - Requirements Engineering
  - Development guided by automatic regression tests
  - System design and coding

ESOFT – Specific Goals

At the end of the course unit, the student must be able to:

1. **Apply** an I&I SDP, in a deterministic way
2. Apply **requirements** engineering and **analysis** methods enhancing the understanding and specification of the application domain
3. Apply and justify the adoption of software **design** methods, principles, patterns and architectures
4. Apply a **test-driven** development (TDD) approach for system construction
5. **Build** the system in accordance with the design, using C++

Some Tips and Rules

- **Classes (theoretical and lab) are important**
  - Faculty provided documents and presentations do not cover all content in-depth
  - Students' competencies will not be acquired just by reading the bibliography
  - Practice is required → Put your hands on the job
  - Both lectures (T) and lab classes (PL) are in person (cf. the ISEP room in your timetable)
- Repository: https://moodle.isep.ipp.pt
  - Documents and presentations
  - Projects/Resolutions
  - Does not contain all the content of the classes
- Basic rules:
  - Do not use cell phones during classes (both T and PL)
  - Do not use a computer in T classes
  - During classes, do not use software other than the one used in the Course Unit (e.g. chat)
  - Punctuality: 5 minutes tolerance (both T and PL classes)

Pedagogical Approach

- Iterative and Incremental
  - 4 Sprints → 4 Deliveries/Submissions in Moodle at the end of each sprint
  - 2 Feedback moments for sprints 1 and 3
  - Assessment
    - **How:** Mandatory and individual presentation/defense
    - **When:** At the end of sprints 2 and 4
    - **Where**: In lab classes and, if necessary, also out of classes
- Project – Practical teamwork
  - Groups of 4 students (exceptionally 3) from the same lab class
  - Goals
    - Formative
    - Evaluative
  - To be developed
    - In lab classes
    - **Outside of classes**

Semester Overview

Iteration	Week	Starting	Ending	Project Goals	Lab/Autonomous Work
	1	15/09/2025	19/09/2025	n/a	1. Setting up and getting used to required tools 2. Studying/Recalling SW Engineering concepts, principles and patterns presented
	2	22/09/2025	26/09/2025		
	3	29/09/2025	03/10/2025		
Sprint 1	4	06/10/2025	10/10/2025	Focus on Requirements Elicitation/Specification and Analysis	1. Studying the introduced topics 2. Sprint 1 development
	5	13/10/2025	17/10/2025		
	6	20/10/2025	24/10/2025		
Sprint 2	7	27/10/2025	31/10/2025	Focus on Analysis and Design activities	1. Studying the introduced topics 2. Sprint 1 improvement (if needed) 3. Sprint 2 development
	8	03/11/2025	07/11/2025		
	9	10/11/2025	14/11/2025		
Sprint 3	10	17/11/2025	21/11/2025	Focus on all SW activities Applying more SW patterns/principles Development of a software prototype	1. Studying the introduced topics 2. Sprint 3 development
	11	24/11/2025	28/11/2025		
	12	01/12/2025	05/12/2025		
Sprint 4	13	08/12/2025	12/12/2025	Focus on all SW activities Development of a software prototype	1. Studying the introduced topics 2. Sprint 3 improvement (if needed)
	14	15/12/2025	19/12/2025		
	X-mas	22/12/2025	26/12/2025	Holidays	
		29/12/2025	02/01/2026		
	15	05/01/2026	09/01/2026	Exposing an HTTP/REST API	3. Sprint 4 development
	16	12/01/2026	16/01/2026	Final Project Assessment	

Assessment / Evaluation

- There is **No Final Exam**
  - This is valid for any period of exams (regular, appealing, special)
- 100% by frequency (project-based)
- Each sprint starts on the first Monday and ends at the last Sunday of the sprint development period

Iteration	Week																Weight (%)
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Sprint 1																	40
Sprint 2																	
Sprint 3																	60
Sprint 4																	

development period
improvement period
feedback
presentation/defense

About ESOFT Workload

- ESOFT worth 5 ECTS (European Credit Transfer System) credits
  - ECTS is used to measure student workload
  - 1 ECTS credit ≅ 25 to 30 working hours (considering an average student)

- ESOFT workload = 5 ECTS credits x 28 hours = **140 hours** (throughout the semester)
  - Classes workload = 15 weeks x (1h + 2h) = **45 hours**
  - Autonomous workload = 140 hours – 45 hours = **95 hours**
  - **Autonomous workload per week (avg.)** ≅ **6 hours/per week**

Recommended Tools

- Moodle ISEP
- C++ IDE (e.g. CLion, Visual Studio, Eclipse)
- Google C++ Testing Framework
- UML editor: PlantUML, Visual Paradigm
- Markdown editor/reader
- HTTP/REST client (e.g. Postman)
- GitHub

Recommended Bibliography

- Most important
  - Applying UML and Patterns, Craig Larman (3rd Edition), Prentice Hall, 2004
  - Clean Architecture, Robert C. Martin, Prentice Hall, 2017
  - Faculty-provided documents and links

- Supplementary
  - Hands-On Design Patterns with C++, Fedor G. Pikus, 2019
  - C++ Programming: From Problem Analysis to Program Design, D.S. Malik, 2018