UNIVERSITY OF CYPRUS
DEPARTMENT OF COMPUTER SCIENCE

ADVANCED SOFTWARE ENGINEERING

EPL 441 — 7.5 ECTS

Academic Year 2025-2026 — Spring Semester

Instructor:	George Angelos Papadopoulos (Professor)
Prerequisites:	EPL 343 (or equivalent)
Office Hours:	Tuesday & Friday, 3:00-4:30 p.m., (ΘΕΕ 01-118)
E-Mail:	**george@ucy.ac.cy**
URL Μαθήματος:	**http://www.cs.ucy.ac.cy/courses/EPL441**
	Also in Moodle (key epl441)

Purpose

The purpose of this course is to examine some advanced concepts, principles and practices in Software Engineering. In the process, students will have the opportunity to use advanced tools for software development. The course is a natural continuation of the introductory course on Software Engineering (EPL343). The course can also play the role of a "road map" for a number of other undergraduate advanced electives as well as for graduate courses related to Software Engineering.

Teaching Methods and Student Assessment

The course is taught by means of 2 lectures of 90 minutes each and a lab session of 90 minutes per week. The course material is organized in a number of logical modules that essentially follow the contents of [1] which is the primary textbook of the course. The main exception is the material for HCI which follows the contents of [2] which is the main textbook for this part of the course.

During the lectures we will be using the official slides of books [1] and [2] which can be found on the web at the official sites of these two books.

Student assessment will be done by means of a project, a mid-term exam and a final exam. The grade distribution among these three assessment exercises is as follows:

Project	30%
Mid-Term Exam	30%
Final Exam	40%

The mid-term exam will take place on Saturday 28 March 2026, 10:00 – 13:00 in ΧΩΔ02 119.

The students are required to enroll immediately in EPL441 on the Moodle platform (http://moodle.cs.ucy.ac.cy), as all communication and distribution of relevant material will be done through Moodle.

Please note carefully: Failure of a student to accumulate at least 35% of the part of the overall grade of this course that is related to the mid-term and final exam, will automatically result in failure of the course, irrespective of the mark that this student will get from the project. In such a case, the final mark will be the one that results from the student's performance in the mid-term and final exam. As an example, consider the performance of two students A and B. Student A scores 12% in the mid-term exam, 24% in the final exam and 20% in the project. As student A's performance in the two exams accumulates to 12+24=36% (equal or above 35%), student A's final grade will be 36+20=56% which rounds up to the final grade of 5.5/10 (pass). Student B scores 18% in the mid-term exam, 15% in the final exam and 25% in the project. As student B's performance in the two exams accumulates to 18+15=33% (less than 35%), student B's final grade will be 33% which rounds up to 3.5/10 (fail).

Bibiography

[1] Ian Sommerville, 'Software Engineering', Pearson, 10th edition, 2016, ISBN 10: 1-292-09613-4, ISBN 13: 978-1-292-09613-1.

[2] Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, 'Human Computer Interaction', Prentice Hall, 3rd edition, 2004, ISBN 10: 0-13-046109-1, ISBN 13: 978-0-13-046109-4.

Organization and Topics

Advanced Software Engineering

Project management

Risk management. Managing people. Teamwork.

Software reuse

The reuse landscape. Application frameworks. Software product lines. Application system reuse.

Component-based Software Engineering

Components and component models. CBSE processes. Component composition.

Distributed Software Engineering

Distributed systems. Client–server computing. Architectural patterns for distributed systems. Software as a service.

Service-oriented Software Engineering

Service-oriented architectures. RESTful services. Service engineering. Service composition.

Systems of systems

Systems complexity. Systems of systems classification. Reductionism and complex systems. Systems of systems engineering. Systems of systems architecture.

Real-time Software Engineering

Embedded systems design. Architectural patterns for real-time systems. Timing analysis. Real-time operating systems.

Human Computer Interaction for Software Engineering

Interaction design basics

What is design? The process of design. User focus. Scenarios. Navigation design. Screen design and layout. Iteration and prototyping.

HCI in the software process

The software life cycle. Usability engineering. Iterative design and prototyping. Design rationale.

Design rules

Principles to support usability. Standards. Guidelines. Golden rules and heuristics. HCI patterns.

Implementation support

Elements of windowing systems. Programming the application. Using toolkits. User interface management systems.

Evaluation techniques

What is evaluation? Goals of evaluation. Evaluation through expert analysis. Evaluation through user participation. Choosing an evaluation method.

Universal design

Universal design principles. Multi-modal interaction. Designing for diversity.

User support

Requirements of user support. Approaches to user support. Adaptive help systems. Designing user support systems.