# COURSE SYLLABUS

## 1. Program information

| | |
|---|---|
| Faculty | Faculty of Electrical Engineering and Computer Science |
| Department | Computer Science |
| Field of study | Computers and Information Technology |
| Cycle of studies | Master's Degree |
| Study Program | Artificial Intelligence & Social Robotics Applied to Global Health |

## 2. Course information

| | | | | | | |
|---|---|---|---|---|---|---|
| Course name | | **ADVANCED SOFTWARE ENGINEERING** | | | | |
| Year of study | 1 | Semester | 1 | Type of assessment | Ongoing assessment | |
| Course regime | | Course category by educational content:<br>DF – Fundamental, DS – Specialization, DC – Complementary | | | | DF |
| | | Course category by optionality:<br>DOB – Compulsory, DOP – Optional, DFA – Elective | | | | DOB |

## 3. Total estimated time (hours allocated to teaching activities)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| I a) Number of hours per week | 3 | Lecture | 2 | Seminar | - | Laboratory / practical work | 1 | Project | - |
| I b) Total number of hours per semester according to curriculum | 42 | Lecture | 28 | Seminar | - | Laboratory / practical work | 14 | Project | - |

| Distribution of time allocation per semester (hours) | No. hours |
|---|---|
| II. Individual study | 103 |
| III. Examinations | 5 |
| IV. Other activities (specify): | - |

| | |
|---|---|
| Total individual study hours (II+III) | 108 |
| Total hours per semester (I.b+II+III+IV) | 150 |
| Number of credits | 6 |

## 4. Specific competencies acquired

| | |
|---|---|
| Transversal competencies | CT1 – work in teams |
| Professional / general competencies | CP2 – develop product design<br>CP3 – design prototypes<br>CP4 – develop data processing applications<br>CP12 – perform project management |

## 5. Learning outcomes

| **Knowledge** | **Skills** | **Responsibility and autonomy** |
|---|---|---|
| At completion of the course, the student will be able to:<br>K1. Critically explain advanced software engineering principles, methodologies, architectures, and research approaches, including their applicability to AI-intensive, agentic, and robotic systems.<br>K2. Analyze requirements engineering, software design, system evolution, and technical debt, considering the specific challenges of generative AI, autonomous | The student will be able to:<br>S1. Elicit, analyze, validate, and document requirements for AI-enabled software products, translating stakeholder needs into specifications, development plans, and measurable acceptance criteria.<br>S2. Design and justify software architectures and technical solutions for AI-intensive, generative-AI, agentic, and robotic applications, using appropriate modeling techniques, design patterns, and risk assessments.<br>S3. Develop, integrate, and iteratively improve functional prototypes of AI-enabled applications, applying software design | At completion of the course, the student will be able to:<br>RA1. Collaborate effectively in small engineering teams, autonomously organize and monitor project activities, negotiate requirements, manage risks, and assume responsibility for assigned deliverables and technical decisions.<br>RA2. Exercise independent professional judgment in selecting and using AI-assisted engineering tools, critically validating AI-generated outputs and taking responsibility for the ethical, safe, |

| | | |
|---|---|---|
| agents, and data-driven applications.<br>K3. Critically evaluate approaches to software quality, testing, verification, validation, safety, security, and regulatory compliance in AI-enabled and robotic systems. | principles, version control, configuration management, and CI/CD practices.<br>S4. Design and execute systematic testing and evaluation procedures, including automated, adversarial, and cross-team testing, to assess software quality, AI behavior, reliability, safety, and security.<br>S5. Apply software engineering research methods to investigate technical problems, critically analyze scientific literature, evaluate experimental results, and communicate evidence-based findings. | secure, and legally compliant development of AI-enabled software. |

6. **Course objectives** (derived from the grid of specific competencies acquired)

| | |
|---|---|
| General objective of the course | To develop advanced software engineering and research competencies for the systematic specification, design, implementation, testing, evaluation, and management of complex software systems, with an emphasis on AI-intensive, agentic, and robotic applications for global health. The course integrates contemporary software engineering methods and AI-assisted development practices, addressing software quality, reliability, safety, security, and responsible development in healthcare and social robotics contexts. |

7. **Teaching and learning content**

| **Lectures** | **No. of hours** | **Teaching methods** | **Remarks/ Notes** |
|---|---|---|---|
| **1. Introduction, basic principles, consolidation of core concepts**<br>1.1. Overview. Evolution and contemporary role of software engineering. Discussions<br>1.2. Landmark failures in health software<br>1.3. Concepts and terminology. Review of software development methodologies | 2h | Lecture, discussion, exemplification, demonstration | |
| **2. Software Engineering Research Methods** | 2h | | |
| **3. Product Discovery, Requirements and Specification for AI Systems** | 2h | | |
| **4. Architecture of AI-Intensive, Agentic and Robotic Systems** | 2h | | |
| **5. Advanced Software Design, Evolution and Technical Debt** | 2h | | |
| **6. Engineering Generative-AI Applications** | 2h | | |
| **7. Agentic Systems: Architectures, Patterns and Protocols** | 2h | | |
| **8. AI-Assisted and Agentic Software Engineering** | 2h | | |
| **9. Advanced Software Testing and AI System Evaluation** | 2h | | |
| **10. Verification, Validation, Safety and Security of AI/Agent Systems** | 2h | | |
| **11. Engineering Real-Time, Edge and Social Robotic Systems** | 2h | | |
| **12. Managing AI-Native Software Projects and Human–Agent Teams** | 2h | | |
| **13. AI Governance, Ethics, Legal and Regulatory Engineering** | 2h | | |
| **14. Research Frontiers and Open Challenges in AI-Native Software Engineering** | 2h | | |
| | | | |

**Recommended minimum bibliography**


1. Otoum, N., & Elkhalili, N. (2026). Methods and Techniques of Agentic Software Engineering: A Systematic Literature Review. IEEE Access, 14, 7443-7465.
2. Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., ... & Abrahamsson, P. (2025). Generative artificial intelligence for software engineering—A research agenda. *Software: Practice and Experience*, *55*(11), Wiley Online Library, 1806-1843, DOI 10.1002/spe.70005.
3. Stray, V., Hanssen, G. K., Barbala, A., Šmite, D., & Stol, K. J. (2025). What is Generative AI good for? Introduction to the special issue on Generative AI in software engineering. Information and Software Technology, 107857. https://www.sciencedirect.com/science/article/abs/pii/S095058492500196X
4. Bazzan, T., Olojo, B., Majda, P., Kelly, T., Yilmaz, M., Marks, G., & Clarke, P. M. (2024, September). Analysing the Role of Generative AI in Software Engineering-Results from an MLR. In European Conference on Software

Process Improvement (pp. 163-180). Cham: Springer Nature Switzerland. https://link.springer.com/chapter/10.1007/978-3-031-71139-8_11
5. Huang, Yuan, Yinan Chen, Xiangping Chen, Junqi Chen, Rui Peng, Zhicao Tang, Jinbo Huang, Furen Xu, and Zibin Zheng. "Generative Software Engineering." arXiv, April 3, 2024. https://doi.org/10.48550/arXiv.2403.02583.
6. Bourque, P.; Fairley, R.E. Guide to the Software Engineering Body of Knowledge (SWEBOK) V4; IEEE Computer Society, 2024; ISBN 978-0-7695-5166-1. https://www.computer.org/education/bodies-of-knowledge/software-engineering/v4
7. Sommerville, I. Software Engineering; 10th edition.; Pearson: Boston, 2016; ISBN 978-0-13-394303-0.
8. Scientific articles published in peer-reviewed journals indexed in Web of Science

| Applications (Seminar / Laboratory / Practical work / Project) | No. of hours | Teaching methods | Remarks / Notes |
|---|---|---|---|
| The practical activities consist of the iterative design, analysis, and evaluation of a complex software system, explicitly incorporating Generative AI and/or agentic AI techniques as part of the software engineering workflow. Throughout the sessions, students incrementally develop as part of a team and refine the system by introducing new components, engineering artifacts, and lifecycle controls, rather than focusing solely on feature implementation. Each laboratory builds upon previous work, requiring students to critically evaluate the benefits, limitations, risks, and governance implications of using GenAI and agentic AI, and reflecting realistic software evolution in advanced and safety-sensitive application domains. | | | |
| **Product, team and engineering environment**<br>Organizational elements: general occupational health and safety regulations. Presentation of the internal rules and regulations.<br>Presentation of platforms and tools (MCP Inspector, LLM API access, project board).<br>Forming 2-person teams and the beneficiary/contractor pairing. Rules for logging AI use. | 2 | Practical work, experiment, discussion, case study | |
| **Requirements, specification and planning**<br>Requirement elicitation, analysis and validation. Kanban board and Gantt plan. Using AI to draft and to critique requirements (and checking for hallucinated requirements). | 2 | | |
| **Architecture, data and risk**<br>UML diagrams, design patterns, architecture decision records. Risk register, with safety and guardrails for app's AI behavior (approval gates, retry limits, failure modes). | 2 | | |
| **Prototype + ADLC**<br>Iterative build, integration of components, first working prototype (MVP). Research report begins. | 2 | | |
| **Engineering quality, AgentOps and CI/CD**<br>Refactoring and SOLID principles applied to agent and workflow code, documentation, configuration and version management (exported workflows, environment setup, one-command Docker run), basic observability such as logs and metrics, including Github Actions. | 2 | | |
| **V&V + adversarial testing**<br>Presentation of own test plan and automated tests. Cross-team testing: each team tests the opposing team's application and writes a defect report. Maintenance plan and quality assessment. | 2 | | |
| **Integrated review and defense**<br>Project presentation and evaluation. Demo, oral defense of design and AI-use choices. Submission of the research report. | 2 | Demonstration | |
| Attendance at compulsory activities is regulated by USV's "Framework Regulation on Student Assessment". Make-up for mandatory practical activities scheduled during the semester is carried out in accordance with the provisions of this regulation. Practical activities scheduled throughout the semester may be replaced by the completion and presentation of high-complexity projects from the course syllabus, with the approval of the course instructor. | | | |

**Recommended minimum bibliography**

1. Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., ... & Abrahamsson, P. (2025). Generative artificial intelligence for software engineering—A research agenda. *Software: Practice and Experience*, *55*(11), Wiley Online Library, 1806-1843, DOI 10.1002/spe.70005.

2. Huang, Yuan, Yinan Chen, Xiangping Chen, Junqi Chen, Rui Peng, Zhicao Tang, Jinbo Huang, Furen Xu, and Zibin Zheng. "Generative Software Engineering." arXiv, April 3, 2024. https://doi.org/10.48550/arXiv.2403.02583.
3. Bourque, P.; Fairley, R.E. Guide to the Software Engineering Body of Knowledge (SWEBOK) V4; IEEE Computer Society, 2024; ISBN 978-0-7695-5166-1. https://www.computer.org/education/bodies-of-knowledge/software-engineering/v4
4. Sommerville, I. Software Engineering; 10th edition. Pearson: Boston, 2016; ISBN 978-0-13-394303-0.

## 8. Evaluation

| Type of activity | Evaluation criteria | Evaluation methods | Weight in final grade |
|---|---|---|---|
| Final evaluation | Correctness and depth of theoretical knowledge; ability to critically analyse advanced software engineering concepts and AI-assisted/agentic development approaches;<br>Quality of the research-oriented technical report; coherence and justification of architectural, lifecycle, quality, and governance decisions; quality and maturity of the developed software system; clarity of presentation and correctness of responses to questions | Individual multiple-choice test with conceptual and scenario-based items;<br><br>Research-oriented technical report documenting and critically analysing the development of the application; final presentation of the team project, including system demonstration and individual questioning | 30%<br><br>20% |
| Evaluation of ongoing practical activities | Correct design, implementation and evaluation of an application; participation in various practical activities related to the specific objectives of the course. | Continuous project-based assessment through iterative laboratory deliverables, milestone reviews, design and code reviews, repository and project-management evidence, automated testing and CI/CD results, AI/agentic workflow documentation, demonstrations, retrospectives, peer assessment, and individual oral verification. | 50% |

The course syllabus includes, where applicable, adaptations for persons with disabilities, depending on their type and degree. **Physical accessibility:** activities are conducted in classrooms and laboratories accessible to wheelchair users via ramps/lifts (USV buildings C/D); **Accessibility of teaching materials:** documents are available in digital PDF format with selectable text and slides with large fonts; **Evaluation accessibility:** ongoing evaluation through projects; possibility of extended time for the written tests and oral verification.

| Date | Academic title, Name, Surname, Signature of Course Instructor | Academic title, Name, Surname, Signature of Practical Work Instructor |
|---|---|---|
| 23.09.2026 | Prof. univ. dr. ing. Cristina Elena TURCU | Șef de lucrări dr. ing. Ovidiu GHERMAN |

| Date of approval | Academic title, Name, Surname, Signature of Program Coordinator |
|---|---|
| 24.09.2026 | Prof. univ. dr. ing. Radu-Daniel VATAVU |

| Date of departmental approval | Academic title, Name, Surname, Signature of Head of Department |
|---|---|
| 24.09.2026 | Prof. univ. dr. ing. Ovidiu-Andrei SCHIPOR |

| Date of Faculty Council Approval | Academic title, Name, Surname, Signature of Dean |
|---|---|
| 25.09.2026 | Prof. univ. dr. ing. Laurentiu-Dan MILICI |