

COURSE DESCRIPTION
Generative AI in Software Engineering
Academic year 2026-2027

1. Programme-related data

1.1. Higher Education Institution	Babeş Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field	Computer Science
1.5. Level of study	Master
1.6. Degree programme / Qualification	Software Engineering
1.7. Form of education	Full time

2. Course-related data

2.1. Course title	Generative AI in Software Engineering			Course code	MME8256
2.2. Course coordinator	Prof.dr. Simona Motogna				
2.3. Seminar coordinator	Prof.dr. Simona Motogna				
2.4. Year of study	2	2.5. Semester	1	2.6. Type of assessment	Exam
2.7. Course status	Optional		2.8. Course type	Specialisation subject	

3. Total estimated time (hours per semester of teaching activities)

3.1. Number of hours per week	4	of which: 3.2. course	2	3.3. seminar/ laboratory/ project	1sem + 1 proj
3.4. Total of hours in the curriculum	56	of which: 3.5. course	28	3.6. seminar/ laboratory	28
Time allocation for individual study (IS) and self-taught activities (ST)					hours
Learning from textbooks, course materials, bibliography, and notes (IS)					28
Additional research in the library, on subject-specific electronic platforms, and on-site					36
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays (greater than or equal to the total number of hours specified in the course calendar for evaluation tasks)					33
Tutoring (professional guidance)					12
Examinations					10
Other activities [i.e.: two-way communication with the course coordinator/tutor]					
3.7. Total hours of individual study (IS) and self-taught activities (ST)				119	
3.8. Total hours per semester				175	
3.9. Number of credits				7	

4. Prerequisites (where applicable)

4.1. curriculum-related	Software Engineering fundamentals
4.2. skills-related	Programming

5. Specific conditions (where applicable)

5.1. course-related	Course room with projector
5.2. seminar/laboratory-related	Lab room

6.1. Competencies resulting from the completion of the degree programme (as referred to in the curriculum)¹

¹ The professional and/or transversal skills targeted by the subject for which the course description is prepared will be copied from the curriculum of the degree programme. For each competency, the complete entry, including the competency code, will be copied with the exact wording that appears in the curriculum, without any changes.

Professional competencies	
Competency code	Competency
PC2	Analysis, design, and implementation of software systems
PC5	Proficient use of methodologies and tools specific to software engineering
PC6	Organization of the software production processes
Transversal competencies	
Competency code	Competency
TC1	Capability of information analysis and synthesis; etic and fair behaviour, commitment to professional deontology
TC2	Team work capabilities, and ability to fulfil different roles

6.2. Learning outcomes relevant to the degree programme (as referred to in the curriculum)²

Learning outcomes targeted by the subject		
Competency code	Knowledge and comprehension	Specific academic skills
SC1	1. The graduate has the necessary knowledge to devise, model and design of complex software applications in the field of software engineering	1. The graduate is able to carry on activities for education and training on different topics related to software systems
SC2	2. The graduate has the ability of interdisciplinary vision between computer science subdomains in order to combine them in a software system	2. The graduate is able to analyse concrete educational situation in terms of general ethical principles and rules
SC3	3. The graduate proves advance programming skills which will allow to learn and comprehend modern technologies	3. The graduate knows and respects the ethical and legal principles and rules in scientific research
SC5	4. The graduate possesses the fundamental knowledge for modelling, being able to analyse real life problems and to translate them in concrete requirements and to design a corresponding software model	4. The graduate proves knowledge related to specifying the requirements of research activities in the domain of computer science in general and software engineering in particular and he/she understands the role of research in promoting progress

7. Subject-specific learning outcomes (referred to by each subject coordinator across the range of competencies and learning outcomes at the level of the degree programme)

Knowledge and comprehension
1. The graduate has the fundamental knowledge in AI Literacy and tooling such that she/he has the ability to use Ai-augmented IDEs for code generation, debugging and documentation
2. The graduate is capable of performing verification engineering including validating generated code, detecting defects and security vulnerabilities
3. The graduate has the ability to design hybrid systems, incorporating software architecture using LLMs
4. The graduate has the skills required for Responsible GenAI, such that she/he is able to detect biases, developer responsibilities, and sustainability issues
Specific academic skills
1. The graduate has to ability to carry on educational tasks related to incorporating GenAI in Software Engineering

If no competency is copied from either of the two categories, the row corresponding to that category is deleted from the table.

² The learning outcomes relevant for the degree programme and targeted by the subject for which the course description is prepared will be listed. The entries, copied without any changes from the Curriculum by subject type (Core Subject/Specialisation Subject/Complementary Subject), are listed under the corresponding competency.

2. The graduate is able to apply ethical principles related to the use of GenAI in software development tasks
3. The graduate proves skills professional teams an interdisciplinary in order to efficiently implement AI enabled software systems

8. Contents

8.1. Course	Teaching and learning methods	Remarks ³
1. AI based Software Development Intro: genAI stack, risks and ethics	Exposure: description, explanation, examples, debate, dialogue	
2. AI based Software Development Intro: Human in the Loop	Exposure: description, explanation, examples, debate, dialogue	
3. AI Assisted Agile methodology	Exposure: description, explanation, examples, debate, dialogue	
4. Using AI for requirement analysis; Prompt engineering for specs	Exposure: description, explanation, examples, debate, dialogue	
5. AI-Assisted Architecture & Design – part 1: design scalable software architectures using AI tools	Exposure: description, explanation, examples, debate, dialogue	
6. AI-Assisted Architecture & Design – part 2: AI to support complex engineering decisions in system design	Exposure: description, explanation, examples, debate, dialogue	
7. Backend Development: AI-assisted coding and review	Exposure: description, explanation, examples, debate, dialogue	
8. Frontend Development: AI-assisted coding and review	Exposure: description, explanation, examples, debate, dialogue	
9. AI-driven Testing: automated test generation, fuzz testing	Exposure: description, explanation, examples, debate, dialogue	
10. CI/CD pipelines, DevOps	Exposure: description, explanation, examples, debate, dialogue	
11. Agentic Workflows	Exposure: description, explanation, examples, debate, dialogue	
12. AI-driven Maintenance and Evolution	Exposure: description, explanation, examples, debate, dialogue	
13. Capstone project presentations	Presentation and evaluation	
14. Capstone project presentations	Presentation and evaluation	
Bibliography		
<ol style="list-style-type: none"> 1. Tom Taulli - AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment, O'Reilly, 2024, ISBN 1098164563 2. Mark Winteringham - Software Testing with Generative AI, Manning Publications, 2024, ISBN-10 1633437361 3. Gene Kim, J. Humble, P. Debois, J. Willis – The DevOps Handbook, IT Revolution, 2016, 4. A. Burkov, Machine Learning Engineering, True Positive Inc., 2020, ISBN-10 1999579577 5. P. Jalote - A Concise Introduction to Software Engineering: With Open Source and GenAI, Springer Nature, 2025 6. Anh Nguyen-Duc, Pekka Abrahamsson, Foutse Khomh - Generative AI for Effective Software Development, Springer, 2024, ISBN-10 3031556410 		
8.2. Seminar	Teaching and learning methods	Remarks
1. Project initial phase: establish student teams, project statement, planning	Use cases, explanation, examples	Students will work in teams on a software project

³ For example, organisational aspects, recommendations for students, specific aspects relating to the course/seminar, such as inviting experts in the field, etc.

2. From client requirements produce: Software Requirement Specification (SRS) and database schema diagram using only AI tools	Use cases, explanation, examples	
3. Use GenAI tools to assist in API design and architecture – 2 seminars	Use cases, explanation, examples	
4. Use GenAI tools for code generation – 2 seminars	Use cases, explanation, examples	
5. Testing and human review of code	Use cases, explanation, examples	
6. Deployment and documentation	Use cases, explanation, examples	
Bibliography: GenAI tools documentation AWS Documentation GitHub Copilot documentation FaaS APIs documentation LangChain AutoGPT		

9. Evaluation

Type of activity	9.1 Evaluation criteria ⁴	9.2 Evaluation methods ⁵	9.3 Percentage in the final grade
9.4. Course	Fundamental knowledge about GenAI in SE Apply basic concepts	Ongoing evaluation	10%
9.5. Seminar/ laboratory	Apply course principles Teamwork	Ongoing evaluation	30%
	Capstone project	Final evaluation	60%
9.6 Minimum standard for passing			
<ul style="list-style-type: none"> - Obtain minimal grade 5 at each evaluated aspect - Understand and apply basic GenAI principles in SE 			



10. SDG labels (Sustainable Development Goals)⁶

		Sustainable Development Generic Label
---	---	---------------------------------------

⁴ The evaluation criteria must directly reflect the learning outcomes targeted at the level of the degree programme respectively at the level of the subject. More specifically, the learning outcomes set out in the expected learning outcomes are assessed.

⁵ Both final evaluation methods and ongoing evaluation strategies should be established.

⁶ Select a single label which, according to the [Implementation of SDG labels in the academic process](#), best matches the subject. If the subject addresses sustainable development in a generic manner (i.e. by presenting/introducing the general framework of sustainable development, etc.), then the Sustainable Development generic label may be applied. If none of the labels describe the subject, select the last option: "No label applies."

								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
								No label applies
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Date of entry:
30.04.2026

Signature of course coordinator
Prof.dr. Simona Motogna

Signature of seminar coordinator
Prof.dr. Simona Motogna

Date of approval in the department:
...

Signature of the head of department
.....