

SYLLABUS

Requirements Engineering

University year 2025-2026

1. Information regarding the programme

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 of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Software engineering
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline		Requirements Engineering					Discipline code		MME8025		
2.2. Course coordinator				Assoc. Prof. PhD. Grigoreta Cojocar							
2.3. Seminar coordinator				Assoc. Prof. PhD. Grigoreta Cojocar							
2.4. Year of study		1	2.5. Semester		1	2.6. Type of evaluation		E	2.7. Discipline regime		Compulsory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	4	of which: 3.2 course	2	3.3 seminar/laboratory/project	1 sem+1 project
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	28
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					22
Additional documentation (in libraries, on electronic platforms, field documentation)					15
Preparation for seminars/labs, homework, papers, portfolios and essays					49
Tutorship					3
Evaluations					5
Other activities:					-
3.7. Total individual study hours	94				
3.8. Total hours per semester	150				
3.9. Number of ECTS credits	6				

4. Prerequisites (if necessary)

4.1. curriculum	-
4.2. competencies	Average design and programming skills in a programming language

5. Conditions (if necessary)

5.1. for the course	Videoprojector
5.2. for the seminar /lab activities	-

6.1. Specific competencies acquired ¹

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	<ul style="list-style-type: none"> • use of theoretical foundations of computer science as well as of formal models • development and maintenance of software systems
Transversal competencies	<ul style="list-style-type: none"> • efficient development of organized activities in an interdisciplinary group and the development of empathetic abilities for interpersonal communications, to relate to and cooperate with various groups • use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language

6.2. Learning outcomes

Knowledge	<p>The student knows:</p> <ul style="list-style-type: none"> • The graduate is familiar with the concepts related to software modelling and is able to implement functional and non-functional requirements described in specific documents for the analysis and design of software systems. • The graduate has the knowledge to apply model-based software development techniques. • The graduate has the necessary knowledge related to the UML language as well as the ability to use CASE tools to understand, document and implement software systems.
Skills	<p>The student is able to ...</p> <ul style="list-style-type: none"> • The graduate has the ability to observe and obtain information from various sources. • The graduate has the ability to choose and use existing modules and environments for application development.
Responsibility and autonomy:	<p>The student has the ability to work independently to obtain ...</p> <ul style="list-style-type: none"> • The graduate is able to combine diverse information to formulate solutions and generate ideas for developing new products and applications. • The graduate has the ability to apply general rules to specific problems and produce relevant solutions.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	<ul style="list-style-type: none"> • Be able to understand requirements engineering concepts and techniques • Average requirements engineering skills
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • To understand the role of requirements engineering for software engineering • To understand the basic concepts of software engineering. • To apply the methods for requirements gathering and analysis.

8. Content

8.1 Course	Teaching methods	Remarks
1.Introduction. Basic concepts. Role of	Exposure: description,	

requirements engineering.	explanation, examples, discussion of case studies	
2. Requirements Elicitation	Exposure: description, explanation, examples, discussion of case studies	
3. Types of Requirements. Structure of System Specification Document.	Exposure: description, explanation, examples, discussion of case studies	
4. SysML. Behaviour Driven Development	Exposure: description, explanation, examples, discussion of case studies	
5. Requirements Prioritization	Exposure: description, explanation, examples, discussion of case studies	
6. Projects presentation	Discussions	
7. Requirements traceability and interdependencies. Impact Analysis	Exposure: description, explanation, examples, discussion of case studies	
8. Quality Assurance for Requirements. Requirements Negotiation	Exposure: description, explanation, examples, discussion of case studies	
9. Business Motivation Model (BMM)	Exposure: description, explanation, examples, discussion of case studies	
10. BMM to SOA	Exposure: description, explanation, examples, discussion of case studies	
11. Projects presentation	Discussions	
12. Security requirements engineering	Exposure: description, explanation, examples, discussion of case studies	
13. Agile Methods and RE	Exposure: description, explanation, examples, discussion of case studies	
14. Projects presentation	Discussions	

Bibliography

- A. Aurum, C. Wohlin –Engineering and Managing Software Requirements, Springer, 2005
- B. Berenbach, D. Paulish a.o. – Software & Systems requirements Engineering: In practice, McGraww Hill, 2009
- E.Hull, K. Jackson, J. Dick – Requirements Engineering, Springer, 2005
- R. Young – The requirement engineering handbook, Artech House, 2004
- C. Williams, M. Kaplan, T. Klinger, A. Paradkar, “Toward Engineered, Useful Use Cases”, in Journal of Object Technology, Vol. 4, No. 6, Special Issue: Use Case Modeling at UML-2004, 2005 , pp. 45-57
- Dan North, Introducing BDD, <http://dannorth.net/introducing-bdd/>
- Suzanne Robertson, James Robertson, Mastering the Requirements Process: Getting Requirements Right (3rd Edition), Addison-Wesley Professional, 2012
- Karl Wiegers, Joy Beatty, Software Requirements (3rd Edition), Microsoft Press, 2013

8.2 Seminar / laboratory	Teaching methods	Remarks
S1. Project Requirements	Discussion	The seminar is structured as 2 hours classes every second week
S2. Requirements elicitation	Dialogue, case studies	
S3. Projects presentations	Discussion	
S4. Behavior Driven Development	Dialogue, case studies	

S5. Requirements prioritization	Dialogue, case studies	
S6. Prototype development	Dialogue, case studies	
S7. Project evaluation	Discussion	
Bibliography		
<ul style="list-style-type: none"> • A. Aurum, C. Wohlin –Engineering and Managing Software Requirements, Springer, 2005 • B. Berenbach, D. Paulish a.o. – Software & Systems requirements Engineering: In practice, McGraww Hill, 2009 • E.Hull, K. Jackson, J. Dick – Requirements Engineering, Springer, 2005 • R. Young – The requirement engineering handbook, Artech House, 2004 		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
- The content of the course is considered the software companies as important for advanced requirements engineering and designing skills

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Know the basic concepts of requirements engineering	Written exam	60%
10.5 Seminar/laboratory	Be able to gather and analyze requirements for a new software	Project (presentation, review, running software)	40%
10.6 Minimum standard of performance			
At least grade 5 (from a scale of 1 to 10) at both written exam and project.			
Project evaluation: 30% for the requirements elicitation, 40% for requirements specification and 30% for design and prototype. Each missed deadline will be penalized with 3 points. The project can be delivered only during the semester. The students who do not obtain at least grade 5 for the project during the semester, may deliver the project only during the re-exam session, and the grade will be at most 5.			

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

² Keep only the labels that, according to the [Procedure for applying ODD labels in the academic process](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.

Date:

...

Signature of course coordinator

Assoc. Prof. PhD. Grigoreta Cojocar

Signature of seminar coordinator

Assoc. Prof. PhD. Grigoreta Cojocar

Date of approval:

...

Signature of the head of department

Assoc. Prof. PhD. Adrian STERCA