

syllabus

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	Component Based Programming

2. Information regarding the discipline

2.1 Name of the discipline		Requirements Engineering					
2.2 Course coordinator		Assoc. Prof. PhD. Grigoreta Cojocar					
2.3 Seminar coordinator		Assoc. Prof. PhD. Grigoreta Cojocar					
2.4. Year of study	2	2.5 Semester	3	2.6. Type of evaluation	E	2.7 Type of discipline	Optional

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	1 sem + 1 project
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/ laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					22
Additional documentation (in libraries, on electronic platforms, field documentation)					30
Preparation for seminars/labs, homework, papers, portfolios and essays					60
Tutorship					7
Evaluations					25
Other activities:					-
3.7 Total individual study hours		144			
3.8 Total hours per semester		200			
3.9 Number of ECTS credits		8			

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	<ul style="list-style-type: none">· Average design and programming skills in a programming language

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none">· Videoprojector
5.2. for the seminar /lab activities	<ul style="list-style-type: none">· Laboratory with computers;

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none">· C2.1 Identification of suitable methodologies for developing software systems.· C2.2 Identification and explanation of suitable mechanism for software systems specification· C2.3 Usage of methodologies, specification mechanisms and development environments for software systems development· C2.5 Development of specific software systems.
Transversal competencies	<ul style="list-style-type: none">· CT2 Efficient development of activities organized in a inter-disciplinary group and the development of emphatic abilities of inter-human communication, relationships and collaboration with different groups.· CT3 Usage of efficient learning, information, research and development methods and techniques for knowledge revaluation abilities, for adaptation to the requirements of a dynamic society, and for communication in romanian language and another foreign language.

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 requirements engineering.	Exposure: description, 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, debate, dialogue	
4. SysML. Behaviour Driven Development	Exposure: description, explanation, examples, discussion of case studies	
5. Business Motivation Model	Exposure: description, explanation, examples, proofs	
6. Requirements Prioritization	Exposure: description, explanation, examples, proofs, debate, dialogue	
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	
9. Projects presentation	Exposure: description, explanation, examples, discussion of case studies	
10. BMM to SOA	Exposure: description, explanation, examples, debate	
11. Requirements Management	Exposure: description, explanation, examples, discussion of case studies	
12. Agile Methods and RE	Exposure: description, explanation, examples, discussion of case studies	
13. Tools for Requirements Engineering		
14. Projects presentation		

Bibliography

1. A. Aurum, C. Wohlin –Engineering and Managing Software Requirements, Springer, 2005
2. B. Berenbach, D. Paulish a.o. – Software & Systems requirements Engineering: In practice, McGraww Hill, 2009
3. E.Hull, K. Jackson, J. Dick – Requirements Engineering, Springer, 2005
4. R. Young – The requirement engineering handbook, Artech House, 2004
5. 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
6. Dan North, Introducing BDD, <http://dannorth.net/introducing-bdd/>
7. Suzanne Robertson, James Robertson, Mastering the Requirements Process: Getting Requirements Right (3rd Edition), Addison-Wesley Professional, 2012
8. Karl Wieggers, Joy Beatty, Software Requirements (3rd Edition), Microsoft Press, 2013

8.2 Seminar	Teaching methods	Remarks
1. Requirements characteristics analysis	Explanation, dialogue	The seminar is structured as 2 hours classes every second week
2. Requirements elicitation	Explanation, dialogue, case studies	
3. Behavior Driven Development	Explanation, dialogue, case studies	
4. Requirements prioritization	Explanation, dialogue, case studies	
5. Business Motivation Model	Explanation, dialogue, case studies	
6. System design based on requirements specification	Explanation, dialogue, case studies	
7. Project evaluation	Expose, evaluation	

Bibliography

7. A. Aurum, C. Wohlin –Engineering and Managing Software Requirements, Springer, 2005
8. B. Berenbach, D. Paulish a.o. – Software & Systems requirements Engineering: In practice, McGraww Hill, 2009
9. E.Hull, K. Jackson, J. Dick – Requirements Engineering, Springer, 2005
1. 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

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Know the basic concepts of requirements engineering	Written exam	60%
10.5 Seminar/lab activities	Be able to gather and analyze requirements for a new software	Running software	40%
10.6 Minimum performance standards			
At least grade 5 (from a scale of 1 to 10) at both written exam and project.			

Date

Signature of course coordinator

Signature of seminar coordinator

28.04.2016

Assoc. Prof. PhD. Grigoreta Cojocar

Assoc. Prof. PhD. Grigoreta Cojocar

Date of approval

Signature of the head of department

.....

.....