

SYLLABUS

Software Quality

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		Software Quality				Discipline code		MME8023
2.2. Course coordinator				Prof.PhD. Simona Motogna				
2.3. Seminar coordinator				Prof.PhD. Simona Motogna				
2.4. Year of study	2	2.5. Semester	2	2.6. Type of evaluation	E	2.7. Discipline regime		Mandatory

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	1/0/1
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)					20
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					50
Tutorship					15
Evaluations					14
Other activities:					
3.7. Total individual study hours			119		
3.8. Total hours per semester			175		
3.9. Number of ECTS credits			7		

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	Basic software development skills

5. Conditions (if necessary)

5.1. for the course	Room with projector
5.2. for the seminar /lab activities	Access to computer/laptop

6.1. Specific competencies acquired ¹

Professional/essential competencies	• analysis, design, and implementation of software systems; • proficient use of methodologies and tools specific to programming languages and software systems; • organization of software production processes.
Transversal competencies	• team work capabilities; able to fulfill different roles; • professional communication skills; concise and precise description, both oral and written, of professional results, negotiation abilities; • entrepreneurial skills; working with economical knowledge; continuous learning;

6.2. Learning outcomes

Knowledge	- The graduate knows the software processes and can integrate them in the organisational culture of a software company - The graduate proves advance programming skills which will allow to learn and comprehend modern technologies
Skills	- The graduate has the ability to follow the entire life cycle of software system development - The graduate proves the capacity to reflect over own learning resources
Responsibility and autonomy:	- The graduate has the ability to communicate and develop relation and partnerships with industrial partners and with all actors involved in the software development process - The graduate can apply advanced information system knowledge starting from a high level of abstraction and being able to offer implementation solutions for complex software system

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

7.1 General objective of the discipline	• Know and understand fundamental concepts of software quality. • To be able to apply basic methods for software analysis and software quality assurance.
7.2 Specific objective of the discipline	At the end of the course, students • will acquire theoretical aspects regarding software quality, • will be able to define a software quality assurance plan and • will be able to apply quality assurance techniques.

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.

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction; characteristics, facts and statistics	Exposure,description, explanation, debate and dialogue, discussion of case studies	
2. Testing, inspection, walkthrough	explanation, debate and dialogue, discussion of case studies	
3. Software quality assurance and SQ Models	Exposure,description, explanation	
4. SQ factors – reliability	Exposure,description, explanation	
5. SQ factors – integrity, security, safety	Exposure,description, explanation	
6. SQ factors – efficiency, maintainability, flexibility	Exposure,description, explanation	
7. SQ factors – portability, reusability, interoperability	Exposure,description, explanation, discussion of case studies	
8. SQ metrics and tools	Exposure,description, explanation, discussion of case studies	
9. SQ standards	Exposure,description, explanation, discussion of case studies	
10. SQ standards – cont.	Exposure,description, explanation, discussion of case studies	
11. CMMI	Exposure,description, explanation, discussion of case studies	
12. SQ assurance vs. SQ control	Exposure,description, explanation, discussion of case studies	
13. SQ and software development phases	Exposure,description, explanation, discussion of case studies	
14. Reserved topic		Usually dedicated to an invited guest from a software company
Bibliography 1. D. Galin – Software quality assurance – From theory to implementation, Addison Wesley, 2003 2. S.H. Kan –Metrics and models in Software Quality Engineering. Addison Wesley, 2nd ed., 2003 3. R.A. Khan, K. Mustafe, S.I. Ahson – Software Quality: Concepts and Practice, Alpha Science, 2006 4. G. Schulmeyer - Handbook of Software Quality Assurance , Artech House, 2007 5. D. Spinellis. <i>Code Quality: The Open Source Perspective</i> . Addison Wesley, 2006 S. McConnell – Code Complete, 2 nd Edition, Microsoft Press, 2004		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Apply and evaluate a Code review tool	Conversation, debate, case studies	Seminar is organized as a total of 7 hours – 2 hours every second week
2. Apply and evaluate a Metrics tool	Conversation, debate, case studies, examples	
3. Establish theme project	Conversation, debate, case studies	
4. Establish SQ moel	Evaluation	
5. Establish SQ factors to be followed and associated metrics	Conversation, debate, case studies	
6. Discuss results and refine metrics	Conversation, debate, case studies, examples	
7. Project presentation	Evaluation	

Bibliography
Students will search and use SQ tools suitable for their project

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 Software Engineering studies;
- The course exists at the major universities in Romania offering similar study programs;
- The content of the course is considered by the software companies as important for average software development skills and quality assurance 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 principle of the domain; - understand and apply the course concepts - problem solving	Oral exam	30%
	SWOT analysis, risk analysis	Workshop active participation	10%
10.5 Seminar/laboratory	- be able to implement course concepts - use tools for different SQ aspects - evaluate quality factors for an application	-Practical examination -documentation -portfolio -continuous observations Laboratory assignments Project	20% 40%
10.6 Minimum standard of performance			
➤ At least grade 5 (from a scale of 1 to 10) at both written exam and laboratory work ➤ Understand and apply software quality attributes in software development ➤ Evaluate software quality of applications			

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

Date:

12.04.2025

Signature of course coordinator

Prof.PhD. Simona Motogna

Signature of seminar coordinator

Prof.PhD. Simona Motogna

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 of approval:

...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA