

## COURSE DESCRIPTION

### Search-based software engineering

University year 2026-2027

#### 1. Programme-related data

1.1. Higher education institution	Babeş-Bolyai University of Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Doctoral School	Mathematics and Computer Science
1.4 Field of study	Computer Science
1.5. Level of study	Doctoral

#### 2. Course-related data

2.1. Course title	<b>Search-based software engineering</b>			Course code	<b>MDE8166</b>
2.2. Course coordinator	Prof. PhD Czibula Istvan Gergely				
2.3. Seminar coordinator	Prof. PhD Czibula Istvan Gergely				
2.4. Year of study	1	2.5. Semester	1	2.6. Type of assessment	Exam
2.7. Course status	Optional			2.8. Course type	Core subject

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

3.1. Number of hours per week	3	of which: 3.2. course	2	3.3. seminar/laboratory/project	1 sem
3.4. Total hours in the curriculum	36	of which: 3.5. course	24	3.6. seminar/laboratory/project	12
<b>Time allocation for individual study (IS) and self-taught activities (ST)</b>					<b>hours</b>
Learning from textbooks, course materials, bibliography, and notes (IS)					36
Additional research in the library, on subject-specific electronic platforms, and on-site					64
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays					50
Tutoring (professional guidance)					33
Examinations					31
Other activities					
<b>3.7. Total hours of individual study (IS) and self-taught activities (ST)</b>					<b>214</b>
<b>3.8. Total hours per semester</b>					<b>250</b>
<b>3.9. Number of credits</b>					<b>10</b>

#### 4. Prerequisites (if necessary)

4.1. curriculum-related	Artificial Intelligence
4.2. skills-related	Programming skills

#### 5. Specific conditions (if necessary)

5.1. course-related	Classroom with a projector
5.2. seminar /laboratory-related	Laboratory with computers; high-level programming language environment (.NET or any Java environment, etc.)

#### 6. Subject-specific learning outcomes

<b>Knowledge</b>
1. The student knows the ethical and legal principles and rules in scientific research
2. The student knows modelling methods, being able to analyse real-life problems and to translate them in concrete requirements and to design a corresponding software model.

3. The student has knowledge related to specifying the requirements of research activities in the domain of computer science in general and computational intelligence (CI) in particular, and he/she understands the role of research in promoting progress.
4. The student knows to conduct original research in the field of Search-based software engineering (SBSE).
<b>Skills</b>
1. The student can use specific language and terminology for the field of search-based software engineering, being able to communicate and interact with members of a team.
2. The student can use advanced search-based software engineering knowledge, starting from a high level of abstraction and being able to offer implementation solutions for complex software systems.
3. The student can apply search-based software engineering techniques to solve real-world problems.
<b>Responsibility and autonomy</b>
1. The student can work independently to obtain the knowledge necessary for designing, managing, and evaluating research activities in the field of search-based software engineering.
2. The student can devise, model, and design software applications in the field of Search-based software engineering.

## 7. Contents

7.1 Course	Teaching and learning methods	Remarks <sup>1</sup>
<b>1. Introduction</b> - Search-Based Software Engineering - Main concepts and problems	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>2. Machine learning in Software Engineering</b> - Machine learning techniques - Applications	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Program Comprehension</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Refactoring</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Defect Detection and prediction</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Software Testing</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Software Visualization</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Effort prediction and Cost estimation</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>CI techniques for Software Reuse</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	

<sup>1</sup> For example, organisational aspects, recommendations for students, specific aspects relating to the course/seminar, such as inviting experts in the field, etc.

<b>CI techniques for Design Patterns identification</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>SBSE research reports presentation</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Conversation</li> <li>• Oral assessment</li> </ul>	
Bibliography 1. Czibula, I., G., Use of search techniques to software development, Editura Risoprint, ISBN 978-973-53- 0119-4, 2009 2. Mark Harman and Bryan F. Jones. Search-based software engineering. Information & Software Technology, 43(14):833-839, 2001. 3. Olaf Seng, Johannes Stammel, and David Burkhart. Search-based determination of refactorings for improving the class structure of object-oriented systems. In GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1909{1916, New York, NY, USA, 2006. ACM Press. 4. Frank Simon, Frank Steinbruckner, and Claus Lewerentz. Metrics based refactoring. In CSMR '01: Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, pages 30- 38, Washington, DC, USA, 2001. IEEE Computer Society.		
<b>7.2 Seminar / laboratory</b>	<b>Teaching and learning methods</b>	<b>Remarks</b>
1. Administration of seminars. Survey of the sources of information available on Internet and Intranet	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
2. Survey of the sources of information available on Internet and Intranet; chosing the paper topic and scheduling the presentation.	<ul style="list-style-type: none"> <li>• Documentation</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
<i>A software project on a SBSE topic (Project 1) will be developed using an open source ML development environment. The second project (Project 2) will be realized from scratch and documented. The software will have to demonstrate the use of CI techniques for some specific SE task.</i>		
3. Problem definition and specification for Project 2	<ul style="list-style-type: none"> <li>• Lab assignment</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
4. Comments about the solution (problem analysis) and search-based modeling of the problem (Project 2). Demonstration of Project 1	<ul style="list-style-type: none"> <li>• Lab assignment</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
5. Design documentation for Project 2	<ul style="list-style-type: none"> <li>• Lab assignment</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
6. The electronic version of the source code, test files, and any other files required to test Project 2. Project 2 demonstration	<ul style="list-style-type: none"> <li>• Lab assignment</li> <li>• Explanation</li> <li>• Conversation</li> </ul>	
Bibliography 1. Czibula, I., G., Use of search techniques to software development, Editura Risoprint, ISBN 978-973-53- 0119-4, 2009 2. Mark Harman and Bryan F. Jones. Search-based software engineering. Information & Software Technology, 43(14):833-839, 2001. 3. Olaf Seng, Johannes Stammel, and David Burkhart. Search-based determination of refactorings for improving the class structure of object-oriented systems. In GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1909{1916, New York, NY, USA, 2006. ACM Press. 4. Frank Simon, Frank Steinbruckner, and Claus Lewerentz. Metrics based refactoring. In CSMR '01: Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, pages 30- 38, Washington, DC, USA, 2001. IEEE Computer Society.		

## 8. Evaluation

Type of activity	8.1 Evaluation criteria <sup>2</sup>	8.2 Evaluation methods <sup>3</sup>	8.3 Percentage in the final grade
8.4 Course	A theoretical research report on a learning technique, based on some recent research papers should be prepared and presented	Evaluation of the research report (a written paper of about 10 pages and an oral presentation)	20%
	The correctness and completeness of the accumulated knowledge.	Oral assessment	40%
	Class attendance	4 unmotivated absences are accepted, but each unmotivated absence other than those specified above are penalised	10%
8.5 Seminar/laboratory	A software project on a SBSE topic will be fully implemented, without using existing libraries.	Evaluation of the project (software implementation, documentation and demonstration)	30%
8.6 Minimum standard for passing			
<ul style="list-style-type: none"><li>• Each student has to prove that (s)he acquired an acceptable level of knowledge and understanding of the SBSE field, that (s)he is capable of stating this knowledge in a coherent form, and that (s)he can establish certain connections and to use the knowledge in solving different problems.</li><li>• Successful passing of the exam is conditioned by the final grade that has to be at least 5.</li></ul>			

## 9. SDG labels (Sustainable Development Goals)<sup>4</sup>

*Not applicable.*

Date of entry:

10.02.2026

Signature of course coordinator

Prof. PhD Czibula Istvan Gergely



Signature of seminar coordinator

Prof. PhD Czibula Istvan Gergely



Date of approval in the department:

Signature of the head of department

<sup>2</sup> 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.

<sup>3</sup> Both final evaluation methods and ongoing evaluation strategies should be established.

<sup>4</sup> 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."