

# SYLLABUS

## *Advanced topics in software testing*

University year 2025-2026

### 1. Information regarding the programme

1.1. Higher education institution	Babes-Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	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		Advanced topics in software testing					Discipline code		MME8150		
2.2. Course coordinator					Assoc. Prof. dr. Vescan Andreea						
2.3. Seminar coordinator					. Prof. dr. Vescan Andreea						
2.4. Year of study		2	2.5. Semester		3	2.6. Type of evaluation		E	2.7. Discipline regime		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/project	2
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	28
<b>Time allotment for individual study (ID) and self-study activities (SA)</b>					<b>hours</b>
Learning using manual, course support, bibliography, course notes (SA)					84
Additional documentation (in libraries, on electronic platforms, field documentation)					14
Preparation for seminars/labs, homework, papers, portfolios and essays					14
Tutorship					3
Evaluations					4
Other activities:					0
<b>3.7. Total individual study hours</b>	<b>119</b>				
<b>3.8. Total hours per semester</b>	<b>175</b>				
<b>3.9. Number of ECTS credits</b>	<b>7</b>				

### 4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	

### 5. Conditions (if necessary)

5.1. for the course	Video projector, Internet access
5.2. for the seminar /lab activities	Video projector, Internet access

### 6.1. Specific competencies acquired <sup>1</sup>

<sup>1</sup> 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"> <li>advanced programming skills in high-level programming languages</li> <li>development and maintenance of software systems</li> </ul>
Transversal competencies	<ul style="list-style-type: none"> <li>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</li> <li>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</li> </ul>

## 6.2. Learning outcomes

Knowledge	<p>The student knows:</p> <ul style="list-style-type: none"> <li>The graduate is able to present and explain methods, algorithms, paradigms and techniques used in various branches of computer science.</li> </ul>
Skills	<p>The student is able to:</p> <ul style="list-style-type: none"> <li>The graduate is familiar with methods for testing and verifying software systems.</li> <li>The graduate is familiar with project management tools, version control systems, and continuous integration/continuous delivery (CI/CD) concepts, methods, tools.</li> </ul>
Responsibility and autonomy:	<p>The student has the ability to work independently to obtain:</p> <ul style="list-style-type: none"> <li>The graduate has the ability to understand and communicate information effectively.</li> <li>The graduate has the ability to observe and obtain information from various sources.</li> </ul>

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

7.1 General objective of the discipline	<ul style="list-style-type: none"> <li>Definitions of common concepts and terms in the field</li> <li>Gain familiarity with a variety of test techniques and compare them</li> <li>To learn the methods of program verification and validation.</li> <li>Team work abilities, assuming different execution and leading roles, performing professional tasks with considerable amounts of autonomy and responsibility</li> </ul>
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> <li>Students will know how to use tools for the management of testing process.</li> <li>Demonstrate advanced skills to analysis and design test cases</li> <li>Understand that there are different missions for testing effort (selection of mission depends on contextual factors)</li> <li>Understand the concept of oracles</li> </ul>

## 8. Content

8.1 Course	Teaching methods	Remarks
<b>Lecture 1. Introduction in Testing and Web Internals</b> <ul style="list-style-type: none"> <li>• General notions</li> <li>• Web Internals explained</li> <li>• (HTTP) Request Structure</li> <li>• Web System in test (Structure of web application, clients etc)</li> <li>• Application Sample and what we will test</li> <li>• Different Tools and frameworks - awareness on tools on market</li> <li>• Project Lifecycle - Scrum Agile methodologies</li> </ul> <b>System Architecture Analysis - DB</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>Lecture 2. Web GUI Automation - Selenium</b> <ul style="list-style-type: none"> <li>• Programming languages testing context- C#, Javascript, Java, Ruby</li> <li>• Selenium IDE - WebDriver (1-2)</li> <li>• CSS selectors</li> <li>• XPath</li> </ul> <b>Page Object patterns</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>Lecture 3 – 4. Web GUI Automation - Selenium</b> <ul style="list-style-type: none"> <li>• Maven Config</li> <li>• Data Driven tests</li> <li>• GUI Automation FWKs - Serenity, Cucumber, Specflow</li> <li>• BDD pros and cons</li> <li>• DB NoSql + Big Decimal</li> </ul> <b>Use of Oracles</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>Lecture 5. JS Unit Tests</b> <ul style="list-style-type: none"> <li>• Data generation</li> <li>• Coverage - examples</li> <li>• Javascript + unit tests</li> <li>• Leverage between GUI - UNIT - API tests</li> </ul> (maybe moved in a web context course)	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>Lecture 6. API Testing - REST, SOAP</b> <ul style="list-style-type: none"> <li>• What you want to test</li> <li>• Execution types</li> <li>• SOAP and REST</li> <li>• Pros and cons in API testing</li> </ul>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>Lecture 7. API Testing - REST, SOAP</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> </ul>	

<ul style="list-style-type: none"> <li>• SOAP + JMeter (JAVA + jMeter    jMeter + jar)</li> <li>• Use of APIs for Test Data setup</li> <li>• Test using IMAP, POP3, SMTP, FTP, HTTP Client</li> <li>• Proxy Tools - fiddler etc</li> <li>• Libraries</li> </ul>	<ul style="list-style-type: none"> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Lecture 8. Performance Testing <ul style="list-style-type: none"> <li>• Request Analysis - yslow, page speed</li> <li>• Load testing</li> <li>• User experience - practices</li> <li>• Browser tools</li> </ul>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Lecture 9. Performance Testing <ul style="list-style-type: none"> <li>• Report analysis</li> <li>• Stress, Volume, Spyke</li> </ul>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Lecture 10- 11 Security Testing	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Lecture 12-13 Mobile Testing <ul style="list-style-type: none"> <li>• Issues</li> <li>• Native + Web + Embedded (hybrid)</li> <li>• API clients</li> <li>• Responsiveness</li> <li>• Segmentation</li> </ul> Analytics	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Lecture 14. Continuous Integration <ul style="list-style-type: none"> <li>• Jenkins, TeamCity, Bamboo</li> <li>• Master - Slave Setup</li> <li>• Parallel test execution</li> <li>• Selenium Grid</li> <li>• Continuous Delivery</li> </ul>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
Bibliography <b>Bibliography</b>  <b>Books</b> [Eri15] Bayo Erinle, Performance testing with JMeter, 2015 [Eri14] Bayo Erinle, JMeter CookBook, Packt Publishing, 2014 [Ava14] S. Avasarala, SeleniumWebDriver Practical Guide, 2014 [Kov14] Dima Kovalenko, Selenium Design Patterns ad Best Practices, Packt Publishing , 2014 [Bur12] David Burns, Selenium 2 Testing Tools: Beginner's guide, 2012 [Unm12] G. Unmesh, Selenium Testing CookBook, 2012 [Gra12] D. Graham, M. Fewster, Experiences of test automation: Case studies of Software Test Automation, 2012		

[Pres10] R. S. Pressman, Software engineering: a practitioner's approach, seventh edition, Higher Education, 2010

[Kan99] C. Kaner, J. Falk, H. Nguyen, Testing Computer Software, 1999

[Crs09] L. Crispin, J. Gregory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009

[You08] M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008

[Nai08] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008

[Pat05] R. Patton, Software Testing, Sams Publishing, 2005

[Mye04] Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004

[Brn02] I. Brnstein, Practical software testing, Springer, 2002

### Articles

[1] Meszaros, G., Smith, S., Andrea, J, The test automation manifesto, LNCS vol 2753, pp. 73-81, 2003

### Internet resources

- Serenity, <http://thucydides.info/docs/serenity-staging/>
- Selenium, <http://www.seleniumhq.org/>
- CSS Selector, [http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)
- Selenium tutorial, <http://software-testing-tutorials-automation.blogspot.ro/2014/01/selenium-webdriver-tutorials-basic.html>

### Tutorials

During lectures/seminars/laboratories tutorials will be given for each assignment.

8.2 Seminar / laboratory	Teaching methods	Remarks
<b>Seminar 1:</b> Selenium IDE, CSS Selectors, Webdriver project setup (Page Object)	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
<b>Seminar 2</b> Webdriver project DDT BDT Page Object Reporting (FWKs) DB NoSql + BigDecimal – Optional (Bonus)	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
<b>Seminar 3</b> API testing in Java/JMeter	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
<b>Seminar 4</b> Performance test using JMeter	Presentation, Conversation, Problematisations, Discovery, Other methods –	

	individual study, exercises	
<b>Seminar 5</b> Security testing	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>Seminar 6</b> Mobile testing Appium	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>Bibliography</b>		
See from Course bibliography		

**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 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 average programming skills.

**10. Evaluation**

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.1 Course	- know the basic principle of the domain;	Written examination	25%
	- apply the course concepts problem solving	Practical examination	25%
10.2 Seminar/laboratory			
10.3 Minimum standard of performance			

**Remark .**

- Seminar/Laboratory assignments work may not be redone in the retake session.
- Written and practical exams can be taken during the retake session.
- Students from Previous Years to the current academic year
  - All the above rules apply to students from previous years.
  - Seminar/Laboratory assignments must be redone during didactic activity time (in the 14 weeks before normal session).

At least grade 5 (from a scale of 1 to 10) at written exam. The final grade computed with the given formula must be at least 5 in order to pass the exam. At least grade 5 (from a scale of 1 to 10) at written and practical exams and laboratory/seminar activity.

## 11. Labels ODD (Sustainable Development Goals)<sup>2</sup>

*Not applicable.*

Date:	Signature of course coordinator	Signature of seminar coordinator
...	Assoc. Prof. dr. Vescan Andreea	Assoc. Prof. dr. Vescan Andreea
Date of approval:	Signature of the head of department	
...	Assoc.prof.phd. Adrian STERCA	

---

<sup>2</sup> 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.*”.