

## SYLLABUS

### 1. Information regarding the programme

1.1 Higher education institution	<b>Babes-Bolyai University</b>
1.2 Faculty	<b>Faculty of Mathematics and Computer Science</b>
1.3 Department	<b>Department of Computer Science</b>
1.4 Field of study	<b>Computers and Information Technology</b>
1.5 Study cycle	<b>Bachelor</b>
1.6 Study programme / Qualification	<b>Information Engineering</b>

### 2. Information regarding the discipline

2.1 Name of the discipline	<b>Test Design Techniques</b>						
2.2 Course coordinator	<b>Lecturer PhD Camelia Chisăliță-Crețu</b>						
2.3 Seminar coordinator	<b>Lecturer PhD Camelia Chisăliță-Crețu</b>						
2.4. Year of study	<b>4</b>	2.5 Semester	<b>8</b>	2.6. Type of evaluation	<b>C</b>	2.7 Type of discipline	<b>Optional DS</b>
2.8 Discipline Code	<b>MLE5110</b>						

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

3.1 Hours per week	5	Of which: 3.2 course	2	3.3 seminar/laboratory	1 LP + 2 P
3.4 Total hours in the curriculum	70	Of which: 3.5 course	28	3.6 seminar/laboratory	42
Time allotment:					Hours
Learning using manual, course support, bibliography, course notes					20
Additional documentation (in libraries, on electronic platforms, field documentation)					8
Preparation for seminars/labs, homework, papers, portfolios and essays					20
Tutorship					2
Evaluations					5
Other activities: .....					-
3.7 Total individual study hours	55				
3.8 Total hours per semester	125				
3.9 Number of ECTS credits	5				

### 4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> <li>• OOP, Programming Fundamentals, Advanced Programming Methods</li> </ul>
4.2. competencies	<ul style="list-style-type: none"> <li>• Good programming skills in at least one of the programming languages Java, C#</li> </ul>

### 5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> <li>• Course hall with projector</li> </ul>
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> <li>• Laboratory: computers and use of a programming language environment</li> </ul>

## 6. Specific competencies acquired

<b>Professional competencies</b>	<ul style="list-style-type: none"> <li>• <b>C3.2</b> Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results</li> <li>• <b>C3.3</b> Applying solution patterns using specific engineering tools and methods</li> <li>• <b>C3.4</b> Comparatively and experimentally evaluation of the alternative solutions for performance optimization</li> <li>• <b>C4.4</b> Managing the life cycle of hardware, software and communications systems based on performance evaluation</li> <li>• <b>C4.5</b> Developing, implementing and integrating software solutions</li> </ul>
<b>Transversal competencies</b>	<ul style="list-style-type: none"> <li>• <b>CT1</b> Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation</li> <li>• <b>CT3</b> Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge</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>• Enhance the students understanding of testing and test design techniques.</li> <li>• Provide the students with an environment in which they can explore the usage and usefulness of software testing and test design concepts in various business scenarios.</li> <li>• Induce a realistic and industry driven view of software testing concepts and their inherent benefits.</li> </ul>
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> <li>• Give students the ability to explore various test design techniques applied to different levels of testing.</li> <li>• Improve the students' abilities to tackle on goal driven testing.</li> <li>• Enhance the students understanding of test design techniques value in business.</li> <li>• Students will be able to use various tools for the testing process (i.e., test management, test running, test reporting and bug reporting).</li> <li>• Students will be able to design test cases according to an established testing goal and using specific test design technique in order to investigate the software.</li> </ul>

## 8. Content

8.1 Course	Teaching methods	Remarks
<b>1. Software Testing. Test Design Techniques</b> 1.1. Software Testing. Goals. Scope 1.2. Test Design Technique. Attributes 1.3. Taxonomy of Test Design Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>2. Coverage-based Techniques I</b> 2.1. Focus. Objectives 2.2. Tours. Logical Expressions	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>3. Coverage-based Techniques II</b> 3.1. Specification-based Testing; 3.2. Requirements-based Testing;	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>4. Risk-based Techniques I</b> 4.1. Focus. Objectives 4.2. HTSM. Failure modes	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>5. Risk-based Techniques II</b> 5.1. Risk Management-based Techniques 5.2. Quick-tests. History-based Testing.	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	

Usability Testing		
<b>6. Activity-based Techniques I</b> 6.1. Focus. Objectives 6.2. Use Cases Testing. Scenario Testing	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>7. Activity-based Techniques II</b> 7.1. Guerilla Testing. All-pairs Testing 7.2. Coverage-based Techniques vs Activity-based Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>8. Evaluation-based Techniques</b> 8.1. Focus. Objectives 8.2. Function Equivalence Testing. Self-verifying data	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation</li> <li>• Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>9. Desired result-based Techniques</b> 9.1. Focus. Objectives 9.2. Confirmation Testing. User Acceptance Testing 9.3. Desired-based Techniques vs Evaluation-based Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>10. Tester-based Techniques</b> 10.1.Focus. Objectives 10.2.User Testing. Alpha Testing. Beta Testing 10.3.Bug Bashes. Paired Testing. 10.4.Coverage-based Techniques vs Tester-based Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>11. Test Design Techniques Analysis</b> 11.1.Tester-based Techniques vs Activity-based Techniques 11.2.Risk-based Techniques vs Coverage-based Techniques 11.3.Desired result-based Techniques vs Risk-based Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>12. Security Testing</b> 12.1.Terminology 12.2.Types 12.3.Advanced Techniques	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Explanation. Conversation</li> <li>• Didactical demonstration</li> </ul>	
<b>13. Bug Reporting</b> 13.1.Challenges 13.2.RIMGEA Strategy	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Conversation</li> </ul>	
<b>14. Project Presentations</b>	<ul style="list-style-type: none"> <li>• Interactive exposure</li> <li>• Conversation</li> </ul>	

## Bibliography

- [Kaner99] C. Kaner, J. Falk, H.Q. Nguyen, *Testing Computer Software*, Wiley, 1999.
- [Brn02] I. Burnstein, *Practical Software Testing*, Springer, 2002.
- [Kaner02] C. Kaner, J. Bach, B. Pettichord, *Lesson Learned in Software Testing*, Wiley, 2002.
- [Mye04] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004.
- [Nai08] K. Naik, P. Tripathy, *Software testing and quality assurance. Theory and Practice*, A John Wiley & Sons, Inc., 2008.
- [Crs09] L. Crispin, J. Gregory, *Agile testing: a practical guide for testers and agile teams*, Addison-Wesley, 2009.
- [Pres10] R. S. Pressman, *Software engineering: a practinioner's approach*, seventh edition, Higher Education, 2010.
- [BBST2008] BBST – *Bug Advocacy*,  
<http://www.testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf>

- [BBST2010] BBST – *Fundamentals of Testing*, Cem Kaner, <http://www.testingeducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf>.
- [BBST2011] BBST – *Test Design*, Cem Kaner, <http://www.testingeducation.org/BBST/testdesign/BBSTTestDesign2011pfinal.pdf>
- [Whitt2012] J. Whittaker, J. Arbon J. Carollo, *How Google Tests Software*, Google, Pearson Education, 2012.
- [OWASP2014] QWASP, *Testing guide 4.0*, 2014, [https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP\\_Testing\\_Guide\\_v4.pdf](https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf)
- [NRVR2014] Ana Filipa Nogueira, José Carlos Ribeiro, Francisco Fernández de Vega, Mário Alberto Zenha-Rela, *Object-Oriented Evolutionary Testing: A Review of Evolutionary Approaches to the Generation of Test Data for Object-Oriented Software*, International Journal of Natural Computing Research 4(4):15-35, October, 2014.
- [KMS2014] Kaur, Manpreet and Rupinder Singh. *A Review of Software Testing Techniques*, 2014.
- [Meer2014] Joris Meerts, *Functional Testing Heuristics*, [https://www.testingreferences.com/docs/Functional\\_Testing\\_Heuristics.pdf](https://www.testingreferences.com/docs/Functional_Testing_Heuristics.pdf)
- [Draghia2019] Claudiu Draghia, *Gamificarea in software testing. Testing Challenges*, <http://testingchallenges.thetestingmap.org/>, 2019.
- [ForK2019] István Forgács, Attila Kovács, *Practical Test Design Selection of traditional and automated test design techniques*, BCS, 2019.
- [BSR2021] F. A. Bhuiyan, M. B. Sharif and A. Rahman, *Security Bug Report Usage for Software Vulnerability Research: A Systematic Mapping Study*, IEEE Access, vol. 9, pp. 28471-28495, 2021, doi: 10.1109/ACCESS.2021.3058067.
- [AIW2021] Samah W.G. AbuSalim, Rosziati Ibrahim, Jahari Abdul Wahab, *Comparative Analysis of Software Testing Techniques for Mobile Applications*, Journal of Physics: Conference Series, vol 1793, 2021.
- [PLGM2022] Sheena Panthaplackel, Junyi Jessy Li, Milos Gligoric, Raymond J. Mooney, *Learning to Describe Solutions for Bug Reports Based on Developer Discussions*, ACL 2022, pp. 2935 – 2952.

8.2 Seminar / laboratory	Teaching methods	Remarks
<b>1. Laboratory 1</b> Testing Project Setup	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>2. Laboratory 2</b> Coverage-based Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>3. Laboratory 3</b> Risk-based Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>4. Laboratory 4</b> Test Automation Tools	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>5. Laboratory 5</b> Activity-based Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>6. Laboratory 6</b> Desired result-based Techniques	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>7. Laboratory 7</b> Project turn-in	Evaluation	

### Bibliography

- [Kaner99] C. Kaner, J. Falk, H.Q. Nguyen, *Testing Computer Software*, Wiley, 1999.
- [Brn02] I. Burnstein, *Practical Software Testing*, Springer, 2002.
- [Kaner02] C. Kaner, J. Bach, B. Pettichord, *Lesson Learned in Software Testing*, Wiley, 2002.
- [Mye04] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004.
- [Nai08] K. Naik, P. Tripathy, *Software testing and quality assurance. Theory and Practice*, A John Wiley & Sons, Inc., 2008.

- [Crs09] L. Crispin, J. Gregory, *Agile testing: a practical guide for testers and agile teams*, Addison-Wesley, 2009.
- [Pres10] R. S. Pressman, *Software engineering: a practitioner's approach*, seventh edition, Higher Education, 2010.
- [BBST2008] BBST – *Bug Advocacy*,  
<http://www.testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf>
- [BBST2010] BBST – *Fundamentals of Testing*, Cem Kaner,  
<http://www.testingeducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf>.
- [BBST2011] BBST – *Test Design*, Cem Kaner,  
<http://www.testingeducation.org/BBST/testdesign/BBSTTestDesign2011pfinal.pdf>
- [Whitt2012] J. Whittaker, J. Arbon J. Carollo, *How Google Tests Software*, Google, Pearson Education, 2012.
- [OWASP2014] QWASP, *Testing guide 4.0*, 2014, [https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP\\_Testing\\_Guide\\_v4.pdf](https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf)
- [NRVR2014] Ana Filipa Nogueira, José Carlos Ribeiro, Francisco Fernández de Vega, Mário Alberto Zenha-Rela, *Object-Oriented Evolutionary Testing: A Review of Evolutionary Approaches to the Generation of Test Data for Object-Oriented Software*, International Journal of Natural Computing Research 4(4):15-35, October, 2014.
- [KMS2014] Kaur, Manpreet and Rupinder Singh. *A Review of Software Testing Techniques*, 2014.
- [Meer2014] Joris Meerts, *Functional Testing Heuristics*,  
[https://www.testingreferences.com/docs/Functional\\_Testing\\_Heuristics.pdf](https://www.testingreferences.com/docs/Functional_Testing_Heuristics.pdf)
- [Draghia2019] Claudiu Draghia, *Gamificarea in software testing. Testing Challenges*,  
<http://testingchallenges.thetestingmap.org/>, 2019.
- [ForK2019] István Forgács, Attila Kovács, *Practical Test Design Selection of traditional and automated test design techniques*, BCS, 2019.
- [BSR2021] F. A. Bhuiyan, M. B. Sharif and A. Rahman, *Security Bug Report Usage for Software Vulnerability Research: A Systematic Mapping Study*, IEEE Access, vol. 9, pp. 28471-28495, 2021, doi: 10.1109/ACCESS.2021.3058067.
- [AIW2021] Samah W.G. AbuSalim, Rosziati Ibrahim, Jahari Abdul Wahab, *Comparative Analysis of Software Testing Techniques for Mobile Applications*, Journal of Physics: Conference Series, vol 1793, 2021.
- [PLGM2022] Sheena Panthaplackel, Junyi Jessy Li, Milos Gligoric, Raymond J. Mooney, *Learning to Describe Solutions for Bug Reports Based on Developer Discussions*, ACL 2022, pp. 2935 – 2952.

8.3 Project	Teaching methods	Remarks
<b>1. Week 01</b> Software Project Presentation Software project goals Team organization	Presentation, Conversation, Problematizations,	
<b>2. Week 02</b> Application Context Information Objectives (Part I and Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>3. Week 03</b> Testing Mission (Part I and Part II) Testing Strategy (Part I and Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>4. Week 04</b> Test Design Technique Selection (part I and Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>5. Week 05</b> Test Design (Part I)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study,	

	exercises	
<b>6. Week 06</b> Test Design (Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>7. Week 07</b> Test Implementation (Part I)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>8. Week 08</b> Test Implementation (Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>9. Week 09</b> Testing Report (Part I)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>10. Week 10</b> Testing Report (Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>11. Week 11</b> Bug Reporting (Part I)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>12. Week 12</b> Bug Reporting (Part II)	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>13. Week 13</b> Testing project presentation preparation	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
<b>14. Week 14</b> Project turn-in	Evaluation	

## Bibliography

- [Kaner99] C. Kaner, J. Falk, H.Q. Nguyen, *Testing Computer Software*, Wiley, 1999.
- [Brn02] I. Burnstein, *Practical Software Testing*, Springer, 2002.
- [Kaner02] C. Kaner, J. Bach, B. Pettichord, *Lesson Learned in Software Testing*, Wiley, 2002.
- [Mye04] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004.
- [Nai08] K. Naik, P. Tripathy, *Software testing and quality assurance. Theory and Practice*, A John Wiley & Sons, Inc., 2008.
- [Crs09] L. Crispin, J. Gregory, *Agile testing: a practical guide for testers and agile teams*, Addison-Wesley, 2009.
- [Pres10] R. S. Pressman, *Software engineering: a practitioner's approach*, seventh edition, Higher Education, 2010.
- [BBST2008] BBST – *Bug Advocacy*,  
<http://www.testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf>
- [BBST2010] BBST – *Fundamentals of Testing*, Cem Kaner,

<http://www.testingeducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf>.

[BBST2011] BBST – *Test Design*, Cem Kaner,

<http://www.testingeducation.org/BBST/testdesign/BBSTTestDesign2011pfinal.pdf>

[Whitt2012] J. Whittaker, J. Arbon J. Carollo, *How Google Tests Software*, Google, Pearson Education, 2012.

[OWASP2014] QWASP, *Testing guide 4.0*, 2014, [https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP\\_Testing\\_Guide\\_v4.pdf](https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf)

[NRVR2014] Ana Filipa Nogueira, José Carlos Ribeiro, Francisco Fernández de Vega, Mário Alberto Zenha-Rela, *Object-Oriented Evolutionary Testing: A Review of Evolutionary Approaches to the Generation of Test Data for Object-Oriented Software*, International Journal of Natural Computing Research 4(4):15-35, October, 2014.

[KMS2014] Kaur, Manpreet and Rupinder Singh. *A Review of Software Testing Techniques*, 2014.

[Meer2014] Joris Meerts, *Functional Testing Heuristics*,

[https://www.testingreferences.com/docs/Functional\\_Testing\\_Heuristics.pdf](https://www.testingreferences.com/docs/Functional_Testing_Heuristics.pdf)

[Draghia2019] Claudiu Draghia, *Gamificarea in software testing. Testing Challenges*,

<http://testingchallenges.thetestingmap.org/>, 2019.

[ForK2019] István Forgács, Attila Kovács, *Practical Test Design Selection of traditional and automated test design techniques*, BCS, 2019.

[BSR2021] F. A. Bhuiyan, M. B. Sharif and A. Rahman, *Security Bug Report Usage for Software Vulnerability Research: A Systematic Mapping Study*, IEEE Access, vol. 9, pp. 28471-28495, 2021, doi: 10.1109/ACCESS.2021.3058067.

[AIW2021] Samah W.G. AbuSalim, Rosziati Ibrahim, Jahari Abdul Wahab, *Comparative Analysis of Software Testing Techniques for Mobile Applications*, Journal of Physics: Conference Series, vol 1793, 2021.

[PLGM2022] Sheena Panthaplackel, Junyi Jessy Li, Milos Gligoric, Raymond J. Mooney, *Learning to Describe Solutions for Bug Reports Based on Developer Discussions*, ACL 2022, pp. 2935 – 2952.

## 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

- Students will know how to apply test design techniques for a software product, in a similar way they are used in industry.
- Students will be able to understand the differences between the goals and scope of the various test techniques applied to a software system.

## 10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Project	Design and develop a testing solution (project) for a software product with focus on test design techniques. The corresponding grade is denoted by <b>P</b> .	Oral Examination	70%
10.5 Seminar/laboratory activities	Each lab activity will be graded. The arithmetic average of the grades is denoted by <b>L</b> .	Laboratory Activity	30%

### Remark:

- Laboratory assignments will be achieved in groups of 2-3 students.
- Testing project will be achieved in groups of 4-5 students.

### 10.6 Minimum performance standards

- Students will be able to apply test design techniques according to established goals for a software system.

- Students will be able to understand the differences between software testing goal, scope, and test design technique concepts.
- The final grade (M) is computed as follows:  $M = 30\%L + 70\%P$ .
- At least  $M \geq 5.00$  is favourable to pass this course exam.

Date

Signature of course coordinator

Signature of seminar coordinator

17.05.2022

Lect. PhD. Camelia Chisăliță-Crețu,

Lect. PhD. Camelia Chisăliță-Crețu,



Date of approval

Signature of the head of department

**Prof. PhD. Laura Dioșan**

24.05.2022

