

SYLLABUS

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	Department of Computer Science
1.4 Field of study	Computer Science
1.5 Study cycle	Master
1.6 Study programme / Qualification	Cyber Security (in English)

2. Information regarding the discipline

2.1 Name of the discipline	Quality Aspects of Security in Software Testing (Aspecte calitative ale securității în testarea software)						
2.2 Course coordinator	Lecturer PhD Camelia Chisăliță-Crețu						
2.3 Seminar coordinator	Lecturer PhD Camelia Chisăliță-Crețu						
2.4. Year of study	1	2.5 Semester	1	2.6. Type of evaluation	C	2.7 Type of discipline	Mandatory
2.8 Discipline Code	MME8195						

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					10
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					44
Tutorship					12
Evaluations					8
Other activities:					-
3.7 Total individual study hours	94				
3.8 Total hours per semester	150				
3.9 Number of ECTS credits	6				

4. Prerequisites (if necessary)

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

5. Conditions (if necessary)

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

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • C3.2 Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results • C3.3 Applying solution patterns using specific engineering tools and methods • C3.4 Comparatively and experimentally evaluation of the alternative solutions for performance optimization • C4.4 Managing the life cycle of hardware, software and communications systems based on performance evaluation • C4.5 Developing, implementing and integrating software solutions
Transversal competencies	<ul style="list-style-type: none"> • CT1 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation • CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

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

7.1 General objective of the discipline	<ul style="list-style-type: none"> • Enhance the students understanding of testing and test design techniques. • 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. • Induce a realistic and industry driven view of software testing concepts and their inherent benefits.
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • Give students the ability to explore various test design techniques applied to different levels of testing. • Improve the students' abilities to tackle on goal driven testing. • Enhance the students understanding of test design techniques value in business. • Students will be able to use various tools for the testing process (i.e., test management, test running, test reporting and bug reporting). • 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.

8. Content

8.1 Course	Teaching methods	Remarks
1. Software Testing. Test Design Techniques 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"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
2. Coverage-based Techniques I 2.1. Focus. Objectives 2.2. Tours. Logical Expressions	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
3. Coverage-based Techniques II 3.1. Specification-based Testing; 3.2. Requirements-based Testing;	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
4. Risk-based Techniques I 4.1. Focus. Objectives 4.2. Quick-tests. History-based Testing	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	

4.3. Boundary testing. Usability Testing		
5. Risk-based Techniques II 5.1. HTSM 5.2. Risk catalogues	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
6. Activity-based Techniques 6.1. Focus. Objectives 6.2. Guerilla Testing. All-pairs Testing 6.3. Use Cases Testing. Scenario Testing Coverage-based Techniques vs Activity-based Techniques	<ul style="list-style-type: none"> • Interactive exposure • Explanation • Conversation • Didactical demonstration 	
7. Evaluation-based Techniques 7.1. Focus. Objectives 7.2. Function Equivalence Testing. Self-verifying data	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
8. Desired result-based Techniques 8.1. Focus. Objectives 8.2. Confirmation Testing. User Acceptance Testing 8.3. Desired-based Techniques vs Evaluation-based Techniques	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
9. Tester-based Techniques 9.1. Focus. Objectives 9.2. User Testing. Alpha Testing. Beta Testing 9.3. Bug Bashes. Paired Testing. 9.4. Coverage-based Techniques vs Tester-based Techniques	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
10. Test Design Techniques Analysis 10.1. Tester-based Techniques vs Activity-based Techniques 10.2. Risk-based Techniques vs Coverage-based Techniques 10.3. Desired result-based Techniques vs Risk-based Techniques	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
11. Security Testing 11.1. Terminology 11.2. Types 11.3. Advanced Techniques	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
12. Bug Reporting 12.1. Challenges 12.2. RIMGEA Strategy	<ul style="list-style-type: none"> • Interactive exposure • Explanation. Conversation • Didactical demonstration 	
13. Project Preparation	<ul style="list-style-type: none"> • Interactive exposure • Conversation 	
14. Project Presentations	<ul style="list-style-type: none"> • Interactive exposure • Conversation 	

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. Grecory, *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
- [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
- [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
1. Laboratory 1 Testing Project Setup	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
2. Laboratory 2 Coverage-based Techniques	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
3. Laboratory 3 Risk-based Techniques	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
4. Laboratory 4 Test Automation Tools	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
5. Laboratory 5 Activity-based Techniques	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
6. Laboratory 6 Desired result-based Techniques	Presentation, Conversation, Problematisations, Discovery, Other methods – individual study, exercises	
7. Laboratory 7 Project turn-in	Evaluation	

References:

- [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.testineducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf>
- [BBST2010] BBST – *Fundamentals of Testing*, Cem Kaner,
<http://www.testineducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf>.
- [BBST2011] BBST – *Test Design*, Cem Kaner,
<http://www.testineducation.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
- [NRVR2014] Ana Filipa Nogueira, José Carlos Ribeiro, Francisco Fernández de Vega, Mário Alberto Zenha-Reia, *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
- [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 (%)
10.1 Lecture	Two workshops on the offensive and defensive security on applications. Students will be graded based on the engagement in the activities. The arithmetic average of the grades is	Oral evaluation	20%

	denoted by W .		
10.2 Seminar	Seminar report on a security topic will be graded. The grade is denoted by S .	Oral evaluation	30%
10.3 Project	Design and develop a testing solution (project) for a software product with focus various types of security vulnerabilities. The corresponding grade is denoted by P .	Oral Examination	50%
Remark: <ul style="list-style-type: none"> • Seminar reports will be achieved in groups of 3-4 students. • Security testing projects will be achieved in groups of 3-4 students. 			
10.6 Minimum performance standards			
<ul style="list-style-type: none"> • The final grade (M) is computed as follows: $M = 30\%S + 20\%W + 50\%P$. • The seminar report and the project should be turned in order to pass the exam. • At least $M \geq 5.00$ is favourable to pass this course exam. 			

Date	Signature of course coordinator	Signature of seminar coordinator
16.04.2024	Lect. PhD. Camelia Chisăliță-Crețu,	Lect. PhD. Camelia Chisăliță-Crețu,

Date of approval	Signature of the head of department
	Assoc. Prof. PhD. Adrian Sterca