SYLLABUS

Quality Aspects of Security 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	**Department of Computer Science**
1.4. Field of study	**Computer Science**
1.5. Study cycle	**Master**
1.6. Study programme/Qualification	**Cyber Security**
1.7. Form of education	**Full time**

2. Information regarding the discipline

2.1. Name of the discipline	**Quality Aspects of Security in Software Testing**					Discipline code	**MME8195**
2.2. Course coordinator	**Lecturer PhD Maria-Camelia Chisăliță-Crețu**						
2.3. Seminar coordinator	**Lecturer PhD Maria-Camelia Chisăliță-Crețu**						
2.4. Year of study	**1**	2.5. Semester	**1**	2.6. Type of evaluation	**C**	2.7. Discipline regime	**Compulsory**

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**

Time allotment for individual study (ID) and self-study activities (SA)	**hours**
Learning using manual, course support, bibliography, course notes (SA)	10
Additional documentation (in libraries, on electronic platforms, field documentation)	20
Preparation for seminars/labs, homework, papers, portfolios and essays	40
Tutorship	12
Evaluations	8
Other activities: *communication with the course lecturer*	4
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	• OOP, Programming Fundamentals, Advanced Programming Methods, Software Systems Verification and Validation
4.2. competencies	• Good programming skills in at least one of the programming languages Java, C#

5. Conditions (if necessary)

5.1. for the course	• Course hall with projector
5.2. for the seminar /lab activities	• Computers and use of a programming language environment for the seminar and project activities

6.1. Specific competencies acquired [1]

[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.

Professional/essential competencies	• Know and understand the main paradigms related to data protection: confidentiality, integrity and data availability. • Knowledge of all security aspects that can impact the processes and IT&C assets of an organization. • Proficient use of verification, validation, and evaluation criteria and methods in order to ensure software security. • Demonstrate advanced skills to analysis, design, and construction of secure software systems, using a wide range of hardware / software platforms, programming languages and environments, and modeling, verification and validation tools.
Transversal competencies	• Professional communication skills; concise and precise description, both oral and written, of professional results. • Ethic and fair behaviour, commitment to professional deontology. • Applying the norms of organized and efficient work, responsibility and reliability of the work performed both individually and within a team. • Entrepreneurial skills; working with economical knowledge; continuous learning.

6.2. Learning outcomes

Knowledge	The student knows the basic mechanisms that define the security of the system and the software environment in which an application runs, such as: access permissions, security policies, interaction with the external environment. The student knows the most popular types of cyber attacks that can take place on the Internet and has knowledge about how to prevent such attacks. The student learns effective techniques for studying and evaluating a source code from the security perspective and has the ability to identify possible vulnerabilities.
Skills	The student is able to identify possible security issues in software systems. The student is able to use security techniques in the software systems development process. The student is able to coordinate project management activities, using decision-making skills, critical and innovative thinking, as well as digital skills.
Responsibility and autonomy:	The student has the ability to evaluate the security features of software applications at the source code level. The student has the ability to work with various tools in the testing process to identify software vulnerabilities. The student has the ability to identify the worst types of vulnerabilities in the field, as well as about measures to prevent these vulnerabilities.

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

7.1 General objective of the discipline	• Enhance the students understanding of security testing and particular offensive security testing and defensive security testing. • Provide the students with an environment in which they can explore, practice, and deepen security testing in various business scenarios. • Induce a realistic and industry driven view of software security testing concepts and their inherent benefits.

7.2 Specific objective of the discipline	• Give students the ability to explore various test design techniques applied to different security aspects pertaining to offensive security testing and defensive security testing. • Improve the students' abilities to tackle on goal driven testing. • Enhance the students' understanding types of vulnerabilities and effective methods to detect them. • Students will be able to use various tools for the security testing process. • Students will be able to design test cases according to an established testing goal and using specific test design technique to investigate the software for specific vulnerabilities.

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	Interactive exposure Explanation. Conversation Didactical demonstration	
2. Cyber Security Concepts 2.1. Terminology. Vulnerabilities 2.2. Attacks 2.3. Frameworks	Interactive exposure Explanation. Conversation Didactical demonstration	
3. Coverage-based Techniques 3.1. Focus. Objectives 3.2. Tours. Logical Expressions 3.3. Specification-based Testing 3.4. Requirements-based Testing	Interactive exposure Explanation. Conversation Didactical demonstration	
4. Offensive Security Testing 4.1. Definitions. Vulnerabilities 4.2. Approaches 4.3. Tools	Interactive exposure Explanation. Conversation Didactical demonstration	
5. Risk-based Techniques I 5.1. Focus. Objectives 5.2. HTSM 5.3. Risk catalogues	Interactive exposure Explanation. Conversation Didactical demonstration	
6. Risk-based Techniques II 6.1. Quick-tests. History-based Testing 6.2. Boundary testing. Usability Testing	Interactive exposure Explanation Conversation Didactical demonstration	
7. Activity-based Techniques 7.1. Focus. Objectives 7.2. Guerilla Testing. All-pairs Testing 7.3. Use Cases Testing 7.4. Scenario Testing	Interactive exposure Explanation. Conversation Didactical demonstration	
8. Defensive Security Testing 8.1. Definitions. Vulnerabilities 8.2. Approaches 8.3. Tools	Interactive exposure Explanation. Conversation Didactical demonstration	
9. Evaluation-based Techniques 9.1. Focus. Objectives 9.2. Function Equivalence Testing 9.3. Self-verifying data	Interactive exposure Explanation. Conversation Didactical demonstration	
10. Desired result-based Techniques	Interactive exposure	

10.1. Focus. Objectives 10.2. Confirmation Testing 10.3. User Acceptance Testing	Explanation. Conversation Didactical demonstration	
11. Test Design Techniques Analysis 11.1. Coverage-based Techniques vs Risk-based Techniques 11.2. Coverage-based Techniques vs Activity-based Techniques 11.3. Risk-based Techniques vs Desired result-based Techniques 11.4. Desired result-based Techniques vs Evaluation-based Techniques	Interactive exposure Explanation. Conversation Didactical demonstration	
12. Vulnerability Reporting 12.1.Terminology. Challenges 12.2. RIMGEA Strategy	Interactive exposure Explanation. Conversation Didactical demonstration	
13. Project Preparation	Presentation, Conversation, Problematizations, Discovery, Evaluation	
14. Project Presentations	Presentation, Conversation, Problematizations, Discovery, 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. Grecory, 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
[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 Heuritics, 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.
[CISA2024] Cybersecurity and Infrastructure Security Agency (CISA).
[IBM2024] International Business Machine (IBM), IBM Technologies.
[NIST2024] National Institute of Standards and Technology, Glossary.
[Abbas2017] Siddiqui, Abbas. (2017). Framework for Supervised & Secure Distributed Simulations of Critical Infrastructures.

[SentinelOne2024] SentinelOne, What Is A Threat Actor? – Types & Examples.
[Anand2022] Anand, P., Singh, Y., Singh, H. et al. SALT: transfer learning-based threat model for attack detection in smart home. Sci Rep 12, 12247 (2022). https://doi.org/10.1038/s41598-022-16261-9.
[Businesstech2021] Business Tech, Vulnerability Assessments: 4 Crucial Steps to Identify Vulnerabilities in Your Business.
[NCSC2016] NCSC, Common cyber attacks: reducing the impact.
[Baker2023] Kurt Baker, 10 Most Common Types of Cyber Attacks.
[Bergmans2023] Bart Lenaerts-Bergmans, 10 Types of social engineering attacks and how to prevent them.
[Zhuang] Rui Zhuang, Alexandru G. Bardas, Scott A. DeLoach, and Xinming Ou. 2015. A Theory of Cyber Attacks: A Step Towards Analyzing MTD Systems. In Proceedings of the Second ACM Workshop on Moving Target Defense (MTD '15). Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/2808475.2808478.
[MTechSystems2022] MTechSystems, The Most Common Cyber Attacks.
[Cyber2024] Cybersecurity Basics. NIST.
[Li2021] Yuchong Li, Qinghui Liu, A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments, Energy Reports, Volume 7, 2021, Pages 8176-8186, ISSN 2352-4847, https://doi.org/10.1016/j.egyr.2021.08.126.

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Seminar 1 Security Concepts Seminar Report Requirements	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
2. Seminar 2 Application Security	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
3. Seminar 3 Vulnerabilities and Tools	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
4. Seminar 4 Seminar Report Presentations	Presentation, Conversation, Problematizations, Discovery, Evaluation	
5. Seminar 5 Seminar Report Presentations	Presentation, Conversation, Problematizations, Discovery, Other methods – individual study, exercises	
6. Seminar 6 Seminar Report Presentations	Presentation, Conversation, Problematizations, Discovery, Evaluation	
7. Seminar 7 Project Presentations	Presentation, Conversation, Problematizations, Discovery, Evaluation	
Bibliography *Similar to the 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 follows the IEEE and ACM Curriculla Recommendations for Computer Science studies.
- The course content exists in the studying programs of all major universities in Romania and abroad.
- The course content is considered relevant by software companies that are focused security and security testing approaches.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade

10.4 Course	Design and develop a testing solution (project) for a software product with focus on security issues and using various test design techniques. The corresponding grade is denoted by **P**.	Oral Examination	70%
10.5 Seminar/laboratory	Seminar report will be graded. The grade is denoted by **S**.	Oral Examination	30%
Remarks: • Seminar reports will pe achieved in groups of 2-3 students. • Security testing projects will pe achieved in groups of 4-5 students.			
10.6 Minimum standard of performance			
• Students will be able to identify vulnerabilities in software accoring to the security testing perfomed, i.e., offensive security testing, defensive security testing. • Students will be able to unstandand the differences between types of vulnerabilities, identify types of attacks and fix vulnerabilities. • The final grade (M) is computed as follows: M = 30%S+70%P. • At least M >= 5.00 is favourable to pass this course exam.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:

15 April 2025

Signature of course coordinator

Lect. PhD. Maria-Camelia CHISĂLIȚĂ-CREȚU

Signature of seminar coordinator

Lect. PhD. Maria-Camelia CHISĂLIȚĂ-CREȚU

Date of approval:

...

Signature of the head of department

Assoc. Prof. PhD. Adrian STERCA

[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*.”.