ISO 9001 CERTIFIED
SLC. MS.040. LSO9001. 1693

SYLLABUS

INTERNATIONAL EUROPEAN UNIVERSITY

BUSINESS ARS EST

EUROPEAN SCHOOL OF BUSINESS

Software quality and testing

2023

SYLLABUS

Course Name	Software quality and testing
Lecturer (s)	Borys Shevchuk, associate professor at the Department of Information Technology, PhD in Pedagogy, associate professor
Lecturer’s profile	https://business.ieu.edu.ua/kafedry/kafedra-informatsiinykh-tekhnolohii#zzz-063
Consultations	
online consulting	Monday 3:10 p.m. – 4:30 p.m.; Friday 3:10 p.m. – 4:30 p.m. https://meet.google.com/brj-dvft-eqg
offline consulting	–
Contact number	(093) 806 84 58
E-mail	borys_shevchuk@ieu.edu.ua
Discipline page	https://dist.ieu.edu.ua/course/view.php?id=632

Form of final control	test	def. test	exam
	☐	☐	☒

SYLLABUS

1 Brief discipline annotation

The Software quality and testing academic discipline is designed for Bachelors of the knowledge area: 12 Information Technology, specialty: 121 Software Engineering. It is one of the professional disciplines for future software developers.

2 Background for studying discipline

The program is based on the knowledge acquired while studying such disciplines as Software architecture and design, Fundamentals of programming, Object-oriented programming, Software modeling and analysis.

3 Goal and objectives of the discipline

The goal of the Software quality and testing discipline is to develop the modern level of information and programmer culture based on software quality theory and testing, to learn basic methods of test construction and automated testing tools, software quality management processes, terms and basics of verification, to acquire practical skills of independent testing according to software quality standards and creation of quality software for solving various tasks in professional activities.

Key objectives of the discipline:

- to master testing, verification and validation methods;
- to study approaches to problem reporting in software development;
- to explore modern statistical methods and quality control tools.

4 Learning outcomes

PLO 9. To know and be able to use methods and tools for collecting, formulating and analyzing software requirements.

PLO 10. To conduct a pre-project examination of the subject area and the system analysis of the design object.

PLO 19. To know and be able to apply software verification and validation methods.

PLO 20. To know approaches to software quality assessment and assurance.

After learning the discipline, students should

know:

- the concept of quality;
- principles of product quality management;
- methods of quality assurance and verification;
- principles of quality culture, basic international quality standards;
- basic definitions of the testing process, testing techniques and levels of testing;
- testing tools and methods for estimating the size of software systems;
- software testing methods and tools;

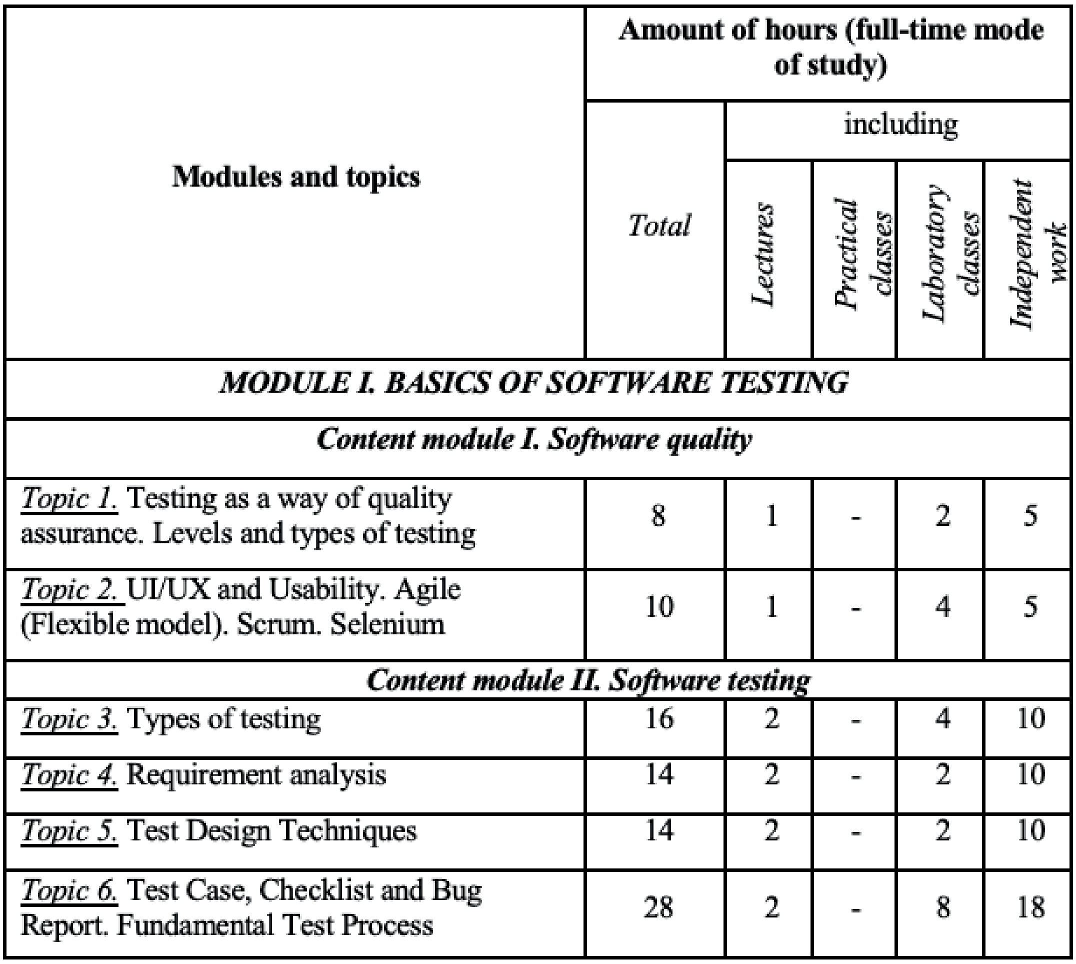
be able to:

- design and implement complex and unit test plans;
- independently or as part of a team inspect the software development process;
- conduct evaluations of tests, defect density and failure probability using metrics;
- determine software quality metrics;
- develop methodologies for software quality assessment, testing, and validation

5 ECTS credits

3

6 Course Content



Modules and topics	Amount of hours (full-time mode of study)				
	Total	including			
		Lectures	*Practical classes*	*Laboratory classes*	*Independent work*
MODULE I. BASICS OF SOFTWARE TESTING					
Content module I. Software quality					
Topic 1. Testing as a way of quality assurance. Levels and types of testing	8	1	-	2	5
Topic 2. UI/UX and Usability. Agile (Flexible model). Scrum. Selenium	10	1	-	4	5
Content module II. Software testing					
Topic 3. Types of testing	16	2	-	4	10
Topic 4. Requirement analysis	14	2	-	2	10
Topic 5. Test Design Techniques	14	2	-	2	10
Topic 6. Test Case, Checklist and Bug Report. Fundamental Test Process	28	2	-	8	18

7 List of obligatory tasks

1. Testing of a common subject.
2. UI Testing.
3. Methodology of Scrum project management. The role of QA and QC in Scrum.
4. Types of testing. Load testing tools (load test with Apache JMeter).
5. Testing of SOAP and REST requests (testing without interface).
6. Analysis of software requirements.
7. Test design techniques in test case development.
8. Creation of checklists.
9. Functional testing. Design and creation of test cases. Creation of test cases using the Testrail tool.
10. Defects and Redmine. Writing Bug Report using Jira tool.
11. Mobile application testing.

8 List of selective tasks

1. Application of UML in programming
2. Technologies for creating electronic glossaries.
3. SOLID principles and patterns of object-oriented programming

9 Discipline features

Period of teaching	**Semester**	**International disciplinary integration**	**Year of study**	**Courses: general training/ professional training/elective**
1 semester	5 semester	Not available	3nd year	Professional training

10 Assessment system and requirements. General system of discipline assessment

Personal computer (PC), Windows OS (7, 8.1, 10, 11), office software packages (Microsoft: Word, Excel Project; Adobe Acrobat), online services.

11 Admission to final control

The form of final control is a Pass/Fail test. Each module includes the grade for students' current work at laboratory classes. Modular control activities are carried out after completion of training materials in this module. If the student agrees, the final grade can be put into the student's record based on the current rating points.

The results of the students' current work during the semester are graded as follows:

Current performance (maximum 100 points) = *Final grade*

Students desiring to improve their level of learning achievements in the discipline can take the Pass/Fail test additionally (in the form of tests)

Current performance + *Tests* = *Final grade*

12 Discipline policy

Teaching of the discipline is based on cutting-edge educational technologies aimed at increasing the level of students' interest in the course, providing theoretical and practical knowledge of the discipline.

To activate the learning and cognitive activity of students, the discipline includes the consolidation of knowledge obtained at the lecture and acquisition of practical skills in lecture topics during laboratory classes and independent work.

13 Policy of absence and late task performance

Scores are not given for missed lectures. If students miss a laboratory work, they should perform all tasks of the missed laboratory work before the next laboratory work and present the results to the lecturer.

Students who have missed classes without valid reasons and have not participated in current control activities are not admitted to the final semester control. In this case, a mark 'non-admission' is put in the exam record on the day of the exam.

Repeated taking of the exam of the discipline is appointed in case of accomplishing all types of educational, individual work stipulated by the working program of the academic discipline and is carried out according to the approved schedule of academic failure liquidation.

14 Academic integrity policy

Tasks and laboratory works submitted later are assessed with a lower grade. The grade is reduced by one point for each week of lateness.

15 Recommended sources of information

PRIMARY (BASIC) LITERATURE:

1. Bandura V.V. Software quality and testing: Laboratory workshop / V.V. Bandura. – Ivano-Frankivsk: IFNTUOG, 2018, – 69 p. – electronic version.
2. Bilas O. Software quality and testing: study guide. – Lviv: Lviv Polytechnic University Publishing House, 2011. – 216 p.
3. V.V. Bandura, V.I. Sheketa, M.M. Pikh. Software quality and testing: lecture notes. – Ivano-Frankivsk: IFNTUOG, 2022. – 199 p.
4. Information technology. Software measurement. Functional size measurement. Part 3. Verification of functional size measurement methods (ISO/IEC TR 14143-3:2003, ITD): DSTU ISO/IEC TR 14143-3:2013. – [Valid from July 1, 2014]. – K.: MINISTRY OF ECONOMY of Ukraine, 2014. – 16 p. – (National Standard of Ukraine).
5. Kuz M.V. Documentation and qualimetry of software products: Study guide / M.V. Kuz, S.I. Melnychuk, L.M. Zamikhovskyi // Second edition revised and supplemented. – Ivano-Frankivsk: IFNTUOG, 2014. – 116 p. (marked by the Ministry of Education and Science).
6. Software engineering. Product quality. Part 1. Quality pattern (ISO/IEC 9126-1:2001, IDT): DSTU ISO/IEC 9126-1:2013 – [Valid from July 1, 2014]. – K.: MINISTRY OF ECONOMY of Ukraine, 2014. – 20 p. – (National Standard of Ukraine).
7. Software engineering. Product quality. Part 2. External metrics (ISO/IEC 9126-2:2003, IDT): DSTU ISO/IEC 9126-2:2008 – [Valid from July 1, 2010]. – K.: State Committee of Ukraine for Technical Regulation and Consumer Policy, 2011. – 85 p. – (National Standard of Ukraine).
8. Software engineering. Product quality. Part 3. Internal metrics (ISO/IEC 9126-3:2003, IDT): DSTU ISO/IEC 9126-3:2012 – [Valid from May 1, 2013]. – K.: MINISTRY OF ECONOMY of Ukraine, 2013. – 46 p. – (National Standard of Ukraine).
9. Software engineering. Product quality. Part 4. Quality metrics during usage (ISO/IEC 9126-4:2004, IDT): DSTU ISO/IEC 9126-4:2012 – [Valid from May 1, 2013]. – K.: MINISTRY OF ECONOMY of Ukraine, 2013. – 49 p. – (National Standard of Ukraine).

15 Recommended sources of information

10. Quality management systems. Requirements (ISO 9001:2015, IDT): DSTU ISO 9001:2015 – [Valid from July 1, 2016]. – K.: SE “UkrNDNC”. – 30 p. – (National Standard of Ukraine).

11. Quality management systems. Genera provisions and glossary. (ISO 9000:2015, IDT): DSTU ISO 9000:2015 – [Valid from July 1, 2016]. – K.: SE “UkrNDNC”, 2016. – 49 p. – (National Standard of Ukraine).

12. Software quality and testing: basic course. Study guide / Edited by Krepych S.Y., Spivak I.Y. / for Bachelors in the knowledge area: 12 Information Technology, specialty: 121 Software Engineering. – Ternopil: PE Palianytsia V.A., 2020. – 478 p.

1. Gojko Adzic. Fifty Quick Ideas to Improve Your Tests / Gojko Adzic, David Evans, Tom Roden// Neuri Consulting LLP. – 2015. – 198p.

2. ISO/IEC 9126-1: 2001. Software engineering. Product quality. Part 1: Quality Model.

3. Jonathan Rasmusson. The Way of the Web Tester. A Beginner’s Guide to Automating Tests / J. Rasmusson// Pragmatic Bookshelf. – 2016. – 258p.

Additional:

1. Borodkina, I.L. Software engineering: study guide / I.L. Borodkina, H.L. Borodkin. – Kyiv: Center for Educational Literature, 2019. – 204 p.

2. Designers of mood. Golden Section in Web Design. [Electronic resource]. – Available at: http://jarlex.com/article/zolo toe-sechenie-v-veb-designe/.

3. Software life cycle: study guide / Y.V. Levus, T.A. Marusenkova, O.O. Nytrebych. – Lviv: Lviv Polytechnic University Publishing House, 2017. – 208 p.

4. Kucherov D.P. Software engineering: study guide / D.P. Kucherov, Y.B. Artamonov. – Kyiv: NAU, 2017. – 386 p.

5. Lavrishcheva K.M. Software engineering. Textbook. K.: Akademperiodyka, 2008. – 319 p.

6. Y.V. Levus. Introduction to Software engineering: study guide / Y.V. Levus, N.B. Melnyk. – Lviv: Lviv Polytechnic University Publishing House, 2018. – 246 p.

7. Object-oriented modeling in the design of embedded systems and time systems: study guide: approved by the Ministry of Education and Science of Ukraine / V.V. Lytvynov, S.V. Holub, K.M. Hryhoriev, V.Y. Zhyhulska. – Kyiv-Cherkasy: IntroligaTOR, 2011. – 511 p.

8. Bugs Catcher. Thinking about high quality testing [Electronic resource] – Access mode: http://bugscatcher.net/archives/3307

9. Lars Lundberg. Software quality attributes and trade-offs. / Lars Lundberg, Michael Mattson, Claes Wohlin. – Blekinge Institute of Technology, 2005.

Internet resources:

1. Knowledge Portal. [Electronic resource]. – Available at: http://www.znannya.org/?view=software-testing-testing .

2. E-learning system of International European University. Software quality and testing course. resource: https://dist.ieu.edu.ua/course/view.php?id=631.

3. Software Testing Help. Entries Tagged 'Cookie Testing. Website Cookie Testing, Test cases for testing web application cookies? – [Electronic resource]. – Available at: http://www.softwaretestinghelp.com/category/cookie-testing/

4. Web testing. [Electronic resource]. – Available at: http://www.edb.utexas.edu/minliu/multimedia/PDFfolder/WebTestingPadolina.pdf

16 Tips on successful study during the course

Everyone has a unique goal, although the proper setting of the goal highly affects the results and the learning process. For example, learning materials of the theoretical component (lectures) of the discipline will provide the understanding and knowledge of the software development process and architecture, while the practical component will provide practical skills in applying software design methods and tools. Thus, any kind of study based on the accurate plan and serious approach to materials will always be successful.