

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ЗАТВЕРДЖЕНО

на засіданні кафедри
інформаційних систем.
Протокол № 7 від 19.01.2026 р.

ПОГОДЖЕНО

Проректор з навчально-методичної роботи
_____ Каріна НЕМАШКАЛО

**МЕТОДИ ТЕСТУВАННЯ ТА ОЦІНКИ
ЯКОСТІ ПРОГРАМНИХ СИСТЕМ**

робоча програма навчальної дисципліни (РПНД)

Галузь знань	12 "Інформаційні технології"
Спеціальність	122 "Комп'ютерні науки"
Освітній рівень	другий (магістерський)
Освітня програма	Комп'ютерні науки

Статус дисципліни
Мова викладання, навчання та оцінювання

**обов'язкова
англійська**

Розробник:
Кандидат технічних
наук, доцент

підписано КЕП

Юрій СКОРІН

Завідувач кафедри
інформаційних
систем

Дмитро БОНДАРЕНКО

Гарант програми

підписано КЕП

Сергій МІНУХІН

**Харків
2026**

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS**

APPROVED

at the meeting
of the Information Systems Department
Protocol № 1 of 22.08.2024

AGREED

Vice-rector for educational and methodical
work

_____ Karina NEMASHKALO

**METHODS OF TESTING AND QUALITY ASSESSMENT OF
SOFTWARE SYSTEMS**

Programme of the course

Field of knowledge	12 “Information Technologies”
Specialty	122 “Computer Sciences”
Study cycle	second (master)
Study programme	Computer Sciences

Course status	mandatory
Language	English

Developers:		
PhD, associate professor	digitally signed	Yury SKORIN

Head of Information Systems Department	_____	Dmytro BONDARENKO
---	-------	-------------------

Head of Study Programme	digitally signed	Sergii MINUKHIN
-------------------------	------------------	-----------------

**Kharkiv
2024**

INTRODUCTION

The course "METHODS OF TESTING AND QUALITY ASSESSMENT OF SOFTWARE SYSTEMS" will be useful to future managers and executors of projects on the development of information systems, primarily during the organization of the process of testing software tools of information systems based on modern information technologies.

The purpose of the course is to expand and deepen theoretical knowledge and applied skills and abilities regarding the main concepts and definitions in the field of quality assurance and software testing, test selection criteria, an overview of testing types, analysis of the features of the process and technology of industrial testing, acquisition of skills in the use of modern information technologies for analysis and testing of information system software, creation of reportable test documentation.

The task of the discipline consists in the formation of a systematized idea among applicants regarding the methods and means of solving the problem of quality assurance, as well as concepts, models and principles of the organization of the process of testing software tools of information systems, namely:

- mastering basic terms and definitions from the field of testing software tools of information systems;

- phases and testing technology software tools of information systems;

- analysis of the main problems and tasks of testing software tools of information systems;

- study of structural, functional, stochastic, mutational test selection criteria, assessment of project coverage;

- modular, integration, system, regression testing;

- testing automation, testing costs;

- test planning, approaches to test development, features of manual test development and generation, test cycle automation;

- testing documentation, reviews and metrics, analysis of test selection methods, evaluation of their effectiveness.

The subject of the course is the basics of the theory of ensuring the quality of software products using modern methods and means of testing, modern information technologies for testing information systems, methods of developing and creating test documentation for evaluating the quality of a software product.

The object of study of the discipline are processes that reflect various aspects of ensuring the quality of software products using modern methods and means of testing.

The learning outcomes and competencies formed by the course are defined in table 1.

Learning outcomes and competences formed by the educational discipline

Learning outcomes	Competencies
LO1	SC01, SC07, SC11
LO2	SC08, SC10, SC11
LO4	GC02, SC02
LO5	SC04
LO6	GC05, SC04
LO9	SC07, SC08, SC10
LO10	GC01, SC02
LO11	GC01, GC03, SC02
LO13	GC01, GC02, GC03, GC05, GC07, SC02, SC07, SC10, SC11
LO14	GC01, GC02, GC03, GC05, GC07, SC07, SC08, SC09, SC10, SC11, SC12
LO15	GC05, GC07, SC01, SC02, SC07, SC08, SC11
LO16	SC07, SC08
LO17	GC01, GC02, GC03, GC05, GC06, GC07, SC01, SC05, SC10, SC11
LO18	GC02, GC03, GC05, GC06, GC07, SC02, SC11
LO19	SC07, SC11
LO20	SC07, SC10, SC11

where, LO1. Have specialized conceptual knowledge that includes modern scientific achievements in the field of computer science and is the basis for original thinking and conducting research, critical understanding of problems in the field of computer science and at the border of fields of knowledge.

LO2. Have specialized computer science problem-solving skills necessary for conducting research and/or carrying out innovative activities in order to develop new knowledge and procedures.

LO4. Manage work processes in the field of information technology, which are complex, unpredictable and require new strategic approaches.

LO5. Evaluate the results of teams and collectives in the field of information technologies and ensure the effectiveness of their activities.

LO6. Develop a conceptual model of an information or computer system.

LO9. Develop algorithmic and software for data analysis (including large data).

LO10. To design architectural solutions of information and computer systems for various purposes.

LO11. Create new algorithms for solving problems in the field of computer science, evaluate their effectiveness and limitations on their application.

LO13. Assess and ensure the quality of information and computer systems for various purposes.

LO14. Test the software.

LO15. Identify the needs of potential customers regarding the automation of information processing.

LO16. Conduct research in the field of computer science.

LO17. Identify and eliminate problematic situations during software operation, formulate tasks for its modification or reengineering.

LO18. Collect, formalize, systematize and analyze the needs and requirements for the information or

computer system being developed, operated or supported.

LO19. To analyze the current state and global trends in the development of computer sciences and information technologies.

LO20. Develop algorithms and software components of computer information systems for high-performance big data processing systems (including distributed and parallel computing) and cloud platform services.

GC01. Ability to abstract thinking, analysis and synthesis.

GC02. Ability to apply knowledge in practical situations.

GC03. Ability to communicate in the national language both orally and in writing.

GC05. Ability to learn and master modern knowledge.

GC06. Ability to be critical and self-critical.

GC07. Ability to generate new ideas (creativity).

SC01. Awareness of the theoretical foundations of computer science.

SC02. The ability to formalize the subject area of a certain project in the form of an appropriate information model.

SC04. The ability to collect and analyze data (including large data) to ensure the quality of project decision-making.

SC05. Ability to develop, describe, analyze and optimize architectural solutions of information and computer systems for various purposes.

SC07. Ability to develop software according to formulated requirements, taking into account available resources and constraints.

SC08. The ability to develop and implement software development projects, including in unpredictable conditions, with unclear requirements and the need to apply new strategic approaches, use software tools to organize teamwork on the project.

SC09. Ability to develop and administer databases and knowledge bases.

SC10. The ability to evaluate and ensure the quality of IT projects, information and computer systems of various purposes, to apply international standards for assessing the quality of software of information and computer systems, models for assessing the maturity of information and computer systems development processes.

SC11. Ability to initiate, plan and implement the development processes of information and computer systems and software, including its development, analysis, testing, system integration, implementation and support.

SC12. Ability to develop, apply and integrate data processing and analysis technologies in high-performance systems and cloud platforms to ensure efficient use of computing resources of computer systems.

COURSE CONTENT

Content module 1. Fundamentals of software testing

Topic 1. Testing as a way to ensure software quality

1.1. Introduction. Course requirements. The main topics of the lecture course. The main topics of the workshop.

1.2. Motivation to study testing technologies. The history of the development of assurance testing. Software quality. Software quality in the context of international standards.

Topic 2. Basic concepts of software testing

2.1. Concept of testing. Basic software testing terminology. Organization of software testing. Application specification.

2.2. Development of tests. Analysis of test cases. Execution of test cases. Evaluation of the results of the program execution on the tests. Stages of testing. The managing graph of the program. The main problems of testing.

Topic 3. Classification of types of testing

3.1. By object of testing. Functional testing. Performance testing. Load testing. Stress testing. Stability testing. Configuration testing. Usability testing. User interface testing. Security testing. Localization testing. Compatibility testing.

3.2. According to the completeness of the information about the object of testing. Black box testing. White box testing.

3.3. By the degree of automation of the testing process. Manual testing. Automated testing.

3.4. According to the degree of isolation of the components. Component (module) testing. Integration testing. System testing.

3.5. By the time of testing. Alpha testing. Beta testing. Release

3.6. According to the testing strategy. Testing parts versus testing the whole. Pseudo-debugging and mutation testing.

Content module 2. Organization of the software testing process

Topic 4. Automation of testing

4.1. Manual testing

4.2. Automated testing

4.3. Testing costs

Topic 5. Features of industrial testing

5.1. Industrial approach. Features of industrial testing.

5.2. Testing process. Phases of the testing process. Test cycle. Testing planning. Test plan.

5.3. Approaches to test development. Testing specifications. Testing scenarios. Manual development of tests. Test generation. Performing tests.

5.3. Documentation and test support. Test procedures. Description of tests. Test procedures. Test report. Evaluation of the quality of tests. Test metrics. Test reviews and strategies.

Topic 6. Documentation and assessment of industrial testing

6.1. Performing tests.

6.2. Documentation and support of tests.

6.3. Evaluation of the quality of tests.

Topic 7. Software quality problems

7.1. Software quality issues and practical recommendations. Software quality as a necessity. End-to-end software quality assurance. Software development standards and their use.

7.2. Software quality assurance methods within the entire development life cycle. Requirements analysis. Analysis and end-to-end control of the code. Session testing. Risk-based testing.

The list of laboratory studies in the course is given in table 2.

Table 2

List of laboratory studies

Name of the topic and / or task	Content
Topic 1, 2, 3. Laboratory work 1	Creating a software requirements specification
Topic 4. Laboratory work 2	Automated testing using TestLog tools.
Topic 4. Laboratory work 3	Automated testing using Katalon Studio tools.
Topic 5. Laboratory work 4	Creation of test scenarios
Topic 6. Laboratory work 5	Development of a test plan
Topic 7. Laboratory work 6	Error reporting

The list of self-studies in the course is given in table 3.

Table 3

List of self-studies

Topic name	Content
Topic 1.	Processing of lecture material. Studying the methodology of creating a software requirements specification. Preparation for the laboratory session.
Topic 2.	Processing of lecture material. Studying the methodology of creating a specification of software requirements.
Topic 3.	Processing of lecture material. Studying the methodology of creating a specification of software requirements.
Topic 4.	Processing of lecture material. Acquiring skills to work with using TestLog and Katalon Studio. Preparation for the laboratory session.
Topic 5.	Processing of lecture material. Acquisition of skills creation of test scenarios. Preparation for the laboratory session.
Topic 6.	Processing of lecture material. Acquisition of skills development of a test plan. Preparation for the laboratory session.

Topic 7.	Processing of lecture material. Analysis of software quality problems. Preparation for the laboratory session.
----------	--

The number of hours of lectures, laboratory classes and hours of independent work is given in the work plan (technological map) for the academic discipline.

TEACHING METHODS

In the process of teaching an educational discipline, in order to acquire certain learning outcomes, to activate the educational process, it is envisaged to use such teaching methods as:

Verbal (lecture (Topic 1, 3, 4, 5, 6, 7), problem lecture (Topic 2, 7)).

In person (demonstration (Topic 1 – 7)).

Practical (laboratory work (Topic 1 - 7)).

FORMS AND METHODS OF ASSESSMENT

The university uses a 100-point cumulative system for assessing the learning outcomes of students..

Current control carried out during lecture and laboratory classes and is aimed at checking the level of readiness of the student of higher education to perform specific work and is evaluated by the sum of points scored:

– for courses with a form of semester control as grading: maximum amount is 100 points; minimum amount required is 60 points.

The final control includes current control and assessment of the student.

Semester control is carried out in the form of a grading.

The final grade in the course is determined:

– for disciplines with a form of grading, the final grade is the amount of all points received during the current control.

During the teaching of the course, the following control measures are used:

Current control: defense of laboratory work (60 points), written control work (testing) (40 points).

Semester control: Grading.

More detailed information about the evaluation system is given in the work plan (technological map) for the course.

RECOMMENDED LITERATURE

Main

1. Шабельник Т. В. Економічна інформатика : навч. посібник / Т. В. Шабельник, С. В. Кривенко, О. Ф. Дяченко. – Маріуполь : МДУ, 2020. – 102 с.
Режим доступу : <http://www.repository.hneu.edu.ua/handle/123456789/28089>

2. Інформатика в сфері комунікацій [Електронний ресурс]: навчально-практичний посібник : у 3-х частинах. Частина 2. Обробка та аналіз даних / С. Г. Удовенко, О. В. Тесленко, Н. О. Бринза [та ін.]; за заг. ред. С. Г. Удовенка; Харківський національний економічний університет ім. С. Кузнеця. – Електрон. текстові дан. (14,3 МБ). – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 249 с. Режим доступу : <http://repository.hneu.edu.ua/handle/123456789/23347>

3. Інформатика в сфері комунікацій [Електронний ресурс] : навч.-практ. посіб. : у 3-х ч. Ч. 3 : Використання web-технологій у сфері комунікацій / С. Г. Удовенко, В. А. Затхей, О. В. Гороховатський [та ін.] ; за заг. ред. С. Г. Удовенка; Харківський національний економічний університет ім. С. Кузнеця. – Електрон. текстові дан. (10.5 МБ). – Харків : ХНЕУ ім. С. Кузнеця, 2020. – 154 с. Режим доступу : <http://repository.hneu.edu.ua/handle/123456789/24506>

4. Крепич С.Я. Якість програмного забезпечення та тестування: базовий курс. Навчальний посібник / За ред. Крепич С.Я., Співак І.Я. / для бакалаврів галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення». – Тернопіль: ФОП Паляниця В.А., 2020. – 478 с. Режим доступу :

<http://dspace.wunu.edu.ua/bitstream/316497/39773/1/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9%20%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA%20%D0%B7%20%D1%8F%D0%BA%D0%BE%D1%81%D1%82%D1%96%20%D0%9F%D0%97%20%D1%82%D0%B0%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%281%29.pdf>

5. Paul C. Jorgensen. Software Testing. A Craftsman's Approach. Fourth Edition / C. Jorgensen Paul. – CRC Press. Taylor & Francis Group. New York, 2021. – 438 p. – Access mode : <https://malenezi.github.io/malenezi/SE401/Books/Software-Testing-A-Craftsman-s-Approach-Fourth-Edition-Paul-C-Jorgensen.pdf>

6. James A. Whittaker. Exploratory Software Testing / A. Whittaker James. – Corporate and Government Sales. – Boston, 2019. – 224 p. – Access mode : <https://www.pearson.de/exploratory-software-testing-tips-tricks-tours-and-techniques-to-guide-test-design-9780321647856>

Additional

7. Дідковська М.В., Тимошенко Ю.О. Тестування. Основні визначення, аксіоми та принципи. Текст лекцій. Частина I [Електронний ресурс]. – Режим доступу :

http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska_mv_testing_definition_part1.pdf

8. Університетський комп'ютер – підручник, орієнтований на проект (win7+office2016), версія для мікроуроку / Гоу Кванденг, Сяоронг Ху; Юлі Юань; Яньціонг Чжан; Ци Чжун; Ліцян Чжан / За загальною редакцією Гоу Кванденг. - Ченду, провінція Сичуань: Університет електронних наук і

технологій Китаю, 2022. – 266 с. Режим доступу : <http://www.repository.hneu.edu.ua/handle/123456789/30140>

9. Текст лекцій до курсу «Технології розробки і тестування програм» / Дідковська М.В. [Електронний ресурс]. – Режим доступу : <http://mmsa.kpi.ua/disciplines/sac/14c>

10. Тестування програмного забезпечення [Електронний ресурс] : методичні рекомендації до самостійної роботи студентів спеціальності 122 "Комп'ютерні науки" першого (бакалаврського) рівня / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2022. – 47 с. – Режим доступу : <http://repository.hneu.edu.ua/bitstream/123456789/26832/1/2022%20-%20%D0%A1%D0%BA%D0%BE%D1%80%D1%96%D0%BD%20%D0%AE%20%D0%86.pdf>

11. Дуднік А.С. Методичні рекомендації дисципліни «Якість програмного забезпечення та тестування» (для освітньо-кваліфікаційного рівня «бакалавр»). – К.: МАУП, 2019. – 19 с. [Електронний ресурс]. – Режим доступу : <https://maup.com.ua/assets/files/lib/metod/9387.pdf>

12. Ушакова І. О. Підходи до забезпечення якості програмного забезпечення / І. О. Ушакова // Сучасні інформаційні технології і системи : монографія / за заг. ред . В. С. Пономаренка. - Харків : «Стильіздат», 2021. – С. 125-140. Режим доступу : <http://repository.hneu.edu.ua/handle/123456789/25612>

13. Бурдаєв В. П. Сучасні інформаційні технології і системи: монографія / В. П. Бурдаєв, Н. Г. Аксак, М. В. Кушнар'єв та ін.; за заг. ред . В. С. Пономаренка. - Харків : Вид. «Стиль-іздат», 2021. – 182 с. Режим доступу : <http://repository.hneu.edu.ua/handle/123456789/25920>

14. Pradeep Soundararajan. Buddha in Testing: Finding Peace in Chaos / Soundararajan Pradeep. – McNichols Road, Chetpet. – 2020. – 136 p. – Access mode : <https://www.practitest.com/resource-center/blog/top-testing-books/>

15. Brooks F.P. No silver bullet—Essence and accident in software engineering, Proceedings of the IFIP Tenth World Computing Conference / F.P. Brooks. – IEEE Computer, Vol. 20, No. 4, 2019. – pp. 1069–1076. – Access mode : https://worrydream.com/refs/Brooks_1986_-_No_Silver_Bullet.pdf

Information resources

16. Якість програмного забезпечення. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/%D0%AF%D0%BA%D1%96%D1%81%D1%82%D1%8C_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F

17. Тестування програмного забезпечення: типи, види та застосування [Електронний ресурс]. – Режим доступу : <https://foxminded.ua/testuvannia-prohramnoho-zabezpechennia/>

18. EPAM University Online: Java, .NET, QA, DevOps и Front-End відтепер доступні з будь-якого куточку планети програміста [Електронний ресурс]. – Режим доступу : <https://training.epam.ua/#!/News/329?lang=ua>

19. Software testing [Electronic resource]. – Access mode : https://en.wikipedia.org/wiki/Software_testing

20. S. Kulikov. Software Testing. Base Course. (3rd edition), book (PDF, updated version, in English) / Kulikov S. — [Electronic resource]. – Access mode : https://svyatoslav.biz/software_testing_book/

21. Glenford J. Myers. The Art of Software Testing / Myers Glenford J., Sandler Corey, Badgett Tom. — 3rd ed.p. cm. — [Electronic resource]. – Access mode : <https://t.me/qarocks/170>

22. Software Testing Tutorial [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/software-testing-tutorial/>

23. Thomas Hamilton. Software Testing Tutorial / Hamilton Thomas. — [Electronic resource]. – Access mode : <https://www.guru99.com/software-testing.html>

24. Lakshay Sharma. Software Testing Tutorial / Sharma Lakshay. — [Electronic resource]. – Access mode : <https://www.toolsqa.com/software-testing/software-testing-tutorial/>

25. Testing the Web Layer. — [Electronic resource]. – Access mode : <https://spring.io/guides/gs/testing-web>

26. Kristin Jackvony. The Complete Software Tester: Concepts, Skills, and Strategies for High-Quality Testing /Jackvony Kristin. — [Electronic resource]. – Access mode : https://www.amazon.com/Complete-Software-Tester-Strategies-High-Quality-ebook/dp/B09NGVVCJ9/ref=sr_1_6?crid=2P1LJPQSFU0LJ&keywords=software+testing&qid=1665297283&qu=eyJxc2MiOiI1LjMwIiwicXNhIjojNC45NCIsInFzcCI6IjQuOTEifQ%3D%3D&s=digital-text&sprefix=software+testing%2Cdigital-text%2C166&sr=1-6