Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of computer systems software

COMPONENTS OF SOFTWARE ENGINEERING. PART 4. SOFTWARE QUALITY AND TESTING

The working program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	***First (Bachelor)***
Branch of knowledge	*12 Information Technologies*
Speciality	*121 Software engineering*
Educational program	*Software engineering of multimedia and information-search systems*
Discipline status	*Normative*
Form of education	*Daytime*
Year of training, semester	*3rd year of training, 5th semester*
Scope of the discipline	*Lectures: 36 hours, computer workshop: 18 hours, self-study: 66 hours.*
Semester control/ control measures	*Examination, calendar control*
Lessons schedule	*According to the schedule for the autumn semester of the current academic year (http://roz.kpi.ua/)*
Language of teaching	*Ukrainian*
Information about the head of the course/teachers	*Lecturer: PhD. , assistant, Pogorelov Volodymyr volodymyr.pogorelov@gmail.com* *Computer Workshop: PhD, assistant, Pogorelov V.V.*
Placement of the course	Google Classroom access is granted to registered students.

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

Studying the discipline "Components of Software Engineering. Part 4. Software Quality and Testing" allows students to develop the competencies necessary for solving practical problems of professional activity related to software quality assessment and testing.

The purpose of *studying the discipline "Components of Software Engineering. Part 4. Software Quality and Testing" is the formation of students' abilities to independently assess the quality and conduct testing of the developed software.*

The subject of *the discipline is "Components of Software Engineering. Part 4. Software Quality and Testing" are the processes, principles and techniques used for software quality analysis and testing.*

Studying the discipline "Components of Software Engineering. Part 4. Software Quality and Testing" forms ***speciality professional competencies (PC)*** *necessary for solving practical tasks of professional activity related to quality assessment and testing of developed software:*
PC01 *Ability to identify, classify and formulate software requirements.*
PC02 *Ability to participate in software design, including its structure, behavior and functioning processes modeling (formal description).*
PC03 *Ability to develop software systems architectures, modules and components.*

PC04 *Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards.*
PC05*. Ability to follow specifications, standards, rules and recommendations in the professional field during the life cycle processes implementation.*
PC07 *Knowledge of information data models, the ability to create software for data storage, retrieval and processing.*
PC08 *Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.*
PC10 *Ability to accumulate, process and systematize professional knowledge about software creation and maintenance, and determination of the importance of lifelong learning.*
PC11*. Ability to implement phases and iterations of the life cycle of the software systems and information technology based on appropriate models and approaches to software development.*
PC12 *Ability to carry out the system integration process, apply change management standards and procedures to maintain software integrity, overall functionality and reliability.*
PC13 *Ability to reasonably select and master software development and maintenance tools.*
PC21*. Ability to identify, analyze and document software requirements for multimedia and information retrieval systems*
PC22*. Ability to create innovative startup projects, calculate basic technical and economic indicators and develop business models of multimedia software and information retrieval systems innovative startup projects that have commercial potential for investment.*

Program learning outcomes *(PLO) of the discipline "Components of Software Engineering. Part 4. Software Quality and Testing" according to the educational program:*
PLO01 *To analyze, purposefully search and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.*
PLO02 *To know the professional ethics code, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.*
PLO03 *To know the software life cycle basic processes, phases and iterations.*
PLO04 *To know and apply professional standards and other regulatory documents in the field of software engineering.*
PLO06 *To know and apply relevant mathematical concepts, domain methods, system and object-oriented analysis and mathematical modeling for software development.*
PLO07 *To know and to apply in practice the fundamental concepts, paradigms and basic principles of the functioning of language, instrumental and computational tools of software engineering.*
PLO08 *To know and to be able to develop a human-machine interface.*
PLO09 *To be able to use collecting, formulating and analyzing software requirements methods and tools.*
PLO10 *To conduct a pre-project survey of the subject area, system analysis of the design object.*
PLO11 *To select initial data for design, guided by formal methods of describing requirements and modeling.*
PLO13 *To know and apply methods of developing algorithms, designing software and data and knowledge structures.*
PLO14 *To apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and documentation of software.*
PLO15 *To choose programming languages and development technologies to solve the problems of creating and maintaining software.*
PLO16 *To have the software development, design approval and all types of software documentation release skills.*
PLO17 *To be able to apply methods of component software development.*
PLO18 *To know and be able to apply information technology of processing, storage and transmission of data.*
PLO19 *To know and be able to apply software verification and validation methods.*

__PLO20__ To know approaches to evaluation and quality assurance of software.
__PLO23__ To be able to document and present the software development results.
__PLO31__ To be able to identify, analyze and document software requirements for multimedia and information retrieval systems
__PLO32__ To be able to develop and analyze full cycle models for multimedia and information retrieval systems software creation.
__PLO33__ To be able to organize a software product management complete cycle.
__PLO34__ To be able to create innovative startup projects of designing multimedia and information-search systems software that have commercial potential for investment.
__PLO35__ To be able to develop and analyze business models of innovative startup projects of developing multimedia and information retrieval systems software that have commercial potential for investment.
__PLO36__ To be able to manage the creation and implementation of software projects in accordance with international standards.
__PLO38__ To be able to apply programming technologies for multimedia and information retrieval systems software development.

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

The successful study of the discipline "Components of Software Engineering. Part 4. Software Quality and Testing" precedes the study of the disciplines "Fundamentals of Programming", "Components of Software Engineering. Part 1. Introduction to Software Engineering", "Components of Software Engineering. Part 2. Software Modeling. Analysis of Software Requirements", "Components of Software Engineering. Part 3. Software Architecture" of the curriculum for bachelor's training in the speciality 121 Software engineering.

Received during the assimilation of the discipline "Software engineering components. Part 4. Software quality and testing" theoretical knowledge and practical skills ensure successful completion of pre-diploma practice, completion of course projects and diploma projects in the speciality 121 Software Engineering.

3. Content of the academic discipline

Discipline "Components of Software Engineering. Part 4. Software Quality and Testing" involves the study of the following topics:
Topic 1. Fundamentals of software testing
Topic 2. Types of software testing
Topic 3. Organization of the software testing process
Topic 4. Evaluation of software quality
Topic 5. Means of optimization of the testing and quality control process
Exam

4. Educational materials and resources

Primary literature:

1. Pogorelov Volodymyr. Components of Software Engineering. Part 4. Software Quality and Testing. Google Classroom, access is granted to registered students.

2. Software Testing and Analysis: Process, Principles and Techniques, by Mauro Pezze and Michal Young, John Wiley & Sons

3. The Art of Software Testing, Second Edition by Glenford J. Myers et al. Digital copy available in DePaul library.

4. Software Engineering: A Practitioner's Approach, Roger S Pressman, McGraw-Hill. Chapters 13 and 14.

Use to master the practical skills of the discipline. The materials are freely available on the Internet.

Educational content

5. Methods of mastering an educational discipline (educational component)

No	*Type of training session*	*Description of the training session*
		Topic 1. Introduction to search engines and services
1	*Lecture 1. Introduction to software testing*	*Definition of software testing. Purpose and goals of software testing. Types of software testing. The software development life cycle and the role of testing at each stage. The importance of software testing. Advantages of software testing.* *Task on self-study No. 1.*
2	*Lecture 2. Inspection in testing. Basic concepts and characteristics of inspection.*	*Definition of Software Inspection. Benefits of Software Inspection. Inspection process and methodology. Roles and responsibilities in Software Inspection. Types of software inspection. Tools and techniques used for Software Inspection* *Task on self-study No. 2.*
3	*Lecture 3. Static analysis. Basic concepts and characteristics, means of static analysis.*	*Definition of static analysis. Advantages of static analysis. Types of static analysis (source code analysis, data flow analysis, etc.). Tools and methods used in static analysis. Integration of static analysis into the software development life cycle. Limitations of static analysis.* *Task on self-study No. 3.*
		Topic 2. Types of software testing
4	*Lecture 4. Modular testing. Tasks, input data, main testing steps, unit testing completion conditions.*	*Definition of unit testing. Advantages of unit testing. Module testing process and methodology. Writing practical unit tests. Test Driven Development (TDD). Tools and techniques used in unit testing.* *Task on self-study No. 4.*
5	*Lecture 5. Integration testing and system testing.*	*Definition of integration and system testing. Advantages of integration and system testing. Process and methodology of integration testing. System testing process and procedure. Types of integration and system testing. Tools and methods used in integration and system testing.* *Task on self-study No. 5.*
6	*Lecture 6. Regression testing, basic concepts and characteristics, key stages.*	*Definition of regression testing. Advantages of regression testing. Process and methodology of regression testing. Types of regression testing (manual and automatic regression testing). Strategies for effective regression testing. Tools and techniques used in regression testing.* *Task on self-study No. 6.*
7	*Lecture 7. Functional testing.*	*Definition of functional testing. Advantages of functional testing. Process and methodology of functional testing. Types of functional*

		testing (black box testing, white box testing, etc.). Writing practical functional tests. Tools and techniques used in functional testing. *Task on self-study No. 7.*
8	*Lecture 8 . Structural testing*	*Definition of structural testing. Advantages of structural testing. Process and methodology of structural testing. Types of structural tests (branch testing, state testing, etc.). Writing practical structural tests. Tools and methods used in structural testing.* *Task on self-study No. 8.*
9	*Lecture 9 . Testing object-oriented software*	*Definition of object-oriented software. Strategies for testing object-oriented software (imitation, polymorphism, encapsulation). Writing practical tests for object-oriented software. Tools and methods used in testing object-oriented software.* *Task on self-study No. 9.*
10	*Computer workshop 1. Testing object-oriented software*	*Task: Using software to implement a module for testing, the types of which are discussed in Topic 2.* *Task on self-study No. 10.*
Topic 3. Organization of the software testing process		
11	*Lecture 10. Choosing a test case*	*Determination of test case selection. The process and methodology of selecting test cases. Factors to consider when selecting test cases (coverage, risk, etc.). Types of test case selection methods (equivalence distribution, analysis of boundary values, etc.). Tools and techniques used to select test cases.* *Task on self-study No. 11.*
12	*Lecture 11. Performance testing*	*Definition of performance testing. Benefits of performance testing. Performance testing process and methodology. Types of performance testing (load testing, stress testing, etc.). Performance testing criteria (response time, throughput, etc.). Tools and techniques used in performance testing* *Task on self-study No. 12.*
13	*Lecture 12. Security testing*	*Definition of security testing. Benefits of security testing. Security testing process and methodology. Types of security testing (penetration testing, vulnerability scanning, etc.). Security check criteria (confidentiality, integrity, availability). Tools and techniques used in security testing.* *Task on self-study No. 13.*
14	*Lecture 13. Testing web applications*	*Definition of web application testing. Benefits of web application testing. The process and methodology of testing web applications. Writing practical tests for web applications. Tools and techniques used to test web applications.* *Task on self-study No. 14.*
15	*Computer workshop 2. Organization of the software testing process*	*Task: Using software to implement a module for testing, the types of which are discussed in Topic 3.* *Task on self-study No. 15.*

Topic 4. Evaluation of software quality		
16	*Lecture 14. Graphical User Interface (GUI) Testing*	*Definition of GUI testing. GUI testing process and methodology. GUI testing criteria (usability, accessibility, consistency, etc.). Tools and techniques used for GUI testing.* *Task on self-study No. 16.*
1 7	*Lecture 15. Usability testing*	*Definition, benefits, process and methodology of usability testing. Types (expert review, heuristic evaluation, etc.). Usability testing criteria. Tools and methods used for usability testing.* *Task on self-study No. 17.*
18	*Lecture 16. Error-based testing*	*Definition of error-based testing. Advantages of error-based testing. Error-based testing process and methodology. Types of error-based testing (fault injection, mutation testing, etc.). Fault-based testing criteria (detection rate, false alarm rate, etc.). Tools and techniques used in error-based testing.* *Task on self-study No. 18.*
Topic 5. Means of optimization of the testing and quality control process		
19	*Lecture 17. Automation and testing tools*	*Defining test automation and tools. Advantages of test automation. Test automation process and methodology. Types of test automation. Criteria for choosing test automation tools (coverage, reliability, etc.). Means and methods used for test automation.* *Task on self-study No. 19.*
20	*Computer workshop 3. Using JUnit for software testing.*	*Task: Using software to implement a software testing module.* *Task on self-study No. 20.*
21	*Lecture 18. Planning and monitoring of the software quality process*	*Defining the process of software quality planning and monitoring. Planning and monitoring process and methodology. Types of software quality indicators (reliability, efficiency, etc.). Methods of improving the software quality process (Six Sigma, Total Quality Management, etc.). Tools and techniques used to plan and monitor the software quality process.* *Task on self-study No. 21.*
Modular control work		

6. Independent work of a student/graduate student

The discipline "Components of Software Engineering. Part 4. Software Quality and Testing" is based on independent preparations for classroom classes on theoretical and practical topics.

No. z/p	*The name of the topic submitted for independent processing*	*Number of hours*	*Literature*

1	*Preparation for lecture 1*	*1*	*1-4*
2	*Preparation for lecture 2*	*1*	*1-4*
3	*Preparation for lecture 3*	*1*	*1-4*
4	*Preparation for lecture 4*	*1*	*1-4*
5	*Preparation for lecture 5*	*1*	*1-4*
6	*Preparation for lecture 6*	*1*	*1-4*
7	*Preparation for the computer workshop 1*	*3*	*1-4*
8	*Preparation for lecture 7*	*1*	*1-4*
9	*Preparation for lecture 8*	*1*	*1-4*
10	*Preparation for lecture 9*	*1*	*1-4*
11	*Preparation for the computer workshop 2*	*3*	*1-4*
12	*Preparation for lecture 10*	*1*	*1-4*
13	*Preparation for lecture 11*	*1*	*1-4*
14	*Preparation for lecture 12*	*1*	*1-4*
15	*Preparation for lecture 13*	*1*	*1-4*
16	*Preparation for lecture 14*	*1*	*1-4*
17	*Preparation for lecture 15*	*1*	*1-4*
18	*Preparation for lecture 16*	*1*	*1-4*
19	*Preparation for lecture 17*	*1*	*1-4*
20	*Preparation for the computer workshop 3*	*1*	*1-4*
21	*Preparation for lecture 18*	*1*	*1-4*
22	*Preparation for modular control work*	*9*	*1-4*
23	*Preparation for the exam*	*36*	*1-4*

Policy and control

7. The policy of academic discipline (educational component)

The policy and principles of academic integrity are defined in Chapter 3 of the Code of Honor of the National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code.

8. Types of control and rating system for evaluating learning outcomes (RSO)

During the semester, students perform three computer practicals. ***The maximum number of points*** *for each computer workshop: 10 points.*

Points are awarded for:
- the quality of performance of the computer workshop: 0-6 points;
- answer during the defence of the computer workshop: 0-2 points;
- timely presentation of work for defence: 0-2 points.

Performance evaluation criteria:

6 points – the work is done qualitatively, in full;
3-5 points – the work is done qualitatively, in full, but has shortcomings;
1-2 points – the work is completed in full but contains significant errors;
0 points - the work is not completed in full.

Answer evaluation criteria:
2 points – the answer is complete, and well-argued;
1 point – there are significant errors in the answer;
0 points - there is no answer, or the answer is incorrect.

Criteria for evaluating the timeliness of work submission for defence:
2 points – the work is presented for defence no later than the specified deadline;
0 points – the work is submitted for defence later than the specified deadline.

The maximum number of points for performing and defending computer practicals:
10 points × 3 comp. practice = 30 points.

During the semester, students complete a test. The task consists of 3 theoretical and 2 practical questions. The answer to each question is evaluated by 4 points.

Evaluation criteria for each test question:
4 points – the answer is correct, complete, and well-argued;
2-3 points - in general, the answer is correct but has flaws;
1 point – there are significant errors in the answer;
0 points - there is no answer, or the answer is incorrect.

The maximum number of points for a modular control work:
4 points × 5 questions = 20 points.

The rating scale for the discipline is equal to the following:
$R = R_C = R_{com.practice} + R_{MKR} + R_{exam}$ *= 30 points + 20 points + 50 points = 100 points.*

Calendar control: is conducted twice a semester as a monitoring of the current state of fulfilment of the syllabus requirements.
At the first certification (8th week), the student receives "credited" if his current rating is at least 10 points (50% of the maximum number of points a student can receive before the first certification).
At the second certification (14th week), the student receives "passed" if his current rating is at least 20 points (50% of the maximum number of points a student can receive before the second certification).

Semester control: exam
Conditions for admission to semester control:
With a semester rating (R_C) of at least 30 points and the admission of all the work of the computer workshop, the student is admitted to the exam. After passing the exam, a grade is assigned according to the table (Table of correspondence of rating points to grades on the university scale).

Completion and defence of a computer workshop is a necessary condition for admission to the exam.

Table of correspondence of rating points to grades on the university scale :

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

9. Additional information on the discipline (educational component)

The list of questions submitted for semester control is given in Appendix 1.

The working program of the academic discipline (syllabus):

Compiled by PhD, assistant Pogorelov V.V.; graduate student Ilyin M.O.

Adopted by Computer Systems Software Department (protocol № 12 from 26.04.23)

Approved by the Faculty Board of Methodology (protocol № 10 from 26.05.23)

Appendix 1. List of issues that are submitted to the semester control

1. *What is software testing and why is it an important stage of software development?*
2. *What is the difference between Inspection testing and static testing?*
3. *What is unit testing, and how does it differ from integration and system testing?*
4. *What is regression testing, and why is it performed?*
5. *What is functional testing, and how is it performed?*
6. *What is structural testing, and what methods are used?*
7. *Selection of test cases for the software system.*
8. *How does object-oriented software testing differ from other types of software?*
9. *What is performance testing, and how is it performed?*
10. *What is security testing, and what methods are used?*
11. *What is web application testing, and what methods are used?*
12. *What is Graphical User Interface (GUI) testing, and what methods are used?*
13. *What is usability testing, and what methods are used?*
14. *What is error-based testing, and what methods are used?*
15. *What is test automation, and what tools are used?*
16. *What is the role of planning and monitoring in the software quality process?*
17. *Software quality indicators.*
18. *What is the difference between unit testing and integration testing?*
19. *What is the purpose of regression testing, and how is it performed?*
20. *What is the difference between security testing and performance testing?*