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

Department of computer systems software

Components of Software Engineering. Course work

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 Retrieval Systems*
Discipline status	*Normative*
Form of education	*Daytime*
Year of training, semester	*3rd year of training, 5th semester*
Scope of the discipline	*Independent work: 30 hours.*
Semester control/ control measures	*Credit*
Lessons schedule	*Not provided*
Language of teaching	*English*
Information about the head of the course/teachers	*Lecturer: PhD. , assistant, Pogorelov Volodymyr volodymyr.pogorelov@gmail.com* *Computer Workshop: PhD, assistant, Pogorelov V.V.* *Assistant Burchak Pavlo*
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

The purpose *of studying the discipline "Components of Software Engineering. Course work" is the formation of students' abilities to:*

- *consolidation of theoretical knowledge of the discipline;*
- *students gain experience in software testing methods;*
- *determine and analyse software quality metrics;*
- *ensure quality inspection of software development artifacts;*
- *provide modular and integration testing of software;*
- *determine and analyse software quality metrics;*
- *ensure high-quality reformatting of the existing software code.*

The subject *of the discipline "Components of Software Engineering. Course work" is a mathematical and algorithmic support for the processes of testing and evaluating the quality of software.*

Studying the discipline "Components of Software Engineering. Course work" helps students to develop the competencies necessary for solving practical tasks of ***general (GC) and professional competencies (PC)*** *related to testing, improvement and operation of software systems of various purposes:*

GC02 *Ability to apply knowledge in practical situations.*

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.*

Studying the discipline "Components of Software Engineering. Course work" contributes to students' formation of the following ***program learning outcomes (PLO)*** *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.*
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.*
PLO05 *To know and apply relevant mathematical concepts, domain methods, system and object-oriented analysis and mathematical modeling for software development.*
PLO06 *Ability to select and use the appropriate task of software development methodology.*
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.*
PLO12 *To apply effective approaches to software design in practice.*
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.*

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

To the successful study of the discipline "Components of Software Engineering. Course work" precedes the study of the disciplines "Fundamentals of Programming", "Programming", "Components of Software Engineering" of the curriculum of bachelor's training in the specialty 121 Software engineering.
Received during the assimilation of the discipline "Components of Software Engineering. Course work" theoretical knowledge and practical skills ensure the successful implementation of course projects and bachelor's theses in the specialty 121 Software engineering.

3. **Content of the academic discipline**

The task of the course work is issued by the teacher - the supervisor of the work and must provide the conditions for achieving the educational goal with the possibility of a meaningful solution to the set tasks.

The content of the task is a simple (from the point of view of software development) data processing task, for the solution of which it is necessary to develop software that will become the object of testing.

The student can propose his own version of the task for the course work, including from the point of view of the possibility of using its results in the future bachelor's thesis.

The main tasks for course work are:

- *development of tests for modular testing of software with a structural architecture using the "black box" methodology.*
- *development of tests for modular testing of software with a structural architecture according to the "white box" methodology;*
- *drawing up a plan for modular testing and conducting modular testing of software with a structural architecture;*
- *development of tests for modular software testing with object-oriented architecture using the "black box" methodology.*
- *development of tests for modular testing of software with object-oriented architecture according to the "white box" methodology;*
- *drawing up a plan for modular testing and conducting modular testing of software with an object-oriented architecture;*
- *development of the test program and methodology;*
- *conducting software tests.*

4. **Requirements for the structure, content and design of the explanatory note**

The results of the course work must be presented in an explanatory note in the form necessary for the evaluation of the author's qualifications.
The explanatory note of the coursework should contain the following sections:

- Table of Contents;

1. Statement of the problem;

2. Software specifications (with presentation of problem solving methods)

3. Software testing (with structural or object-oriented architecture).

3.1. Software structure

3.2. Unit testing plan;

3.3. Black box and white box tests (for modules or for classes).

4. *Test program and methodology.*
5. *Test protocol.*
6. *Conclusions.*

5. Course work schedule

- *Analysis of the subject area and available technologies, development of conceptual architecture - by March 15.*
- *Analysis and description of requirements, development of client-server protocol - by April 1.*
- *Development of test scenarios and detailed architecture, start of development - by April 15.*
- *Demonstration of the beta version of the product - until May 15.*
- *Protection of course work - until May 30.*

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

Basic literature:

1. *Educational materials from the discipline "Components of Software Engineering. Course work". Use to master practical skills in the discipline. The materials are in Google classroom. Access is granted to registered students.*

Policy and control

7. The policy of academic discipline (educational component)

- *Adherence to the policy of academic integrity.*
- *Rules for protecting the work of the computer workshop: the work must be done according to the student's option, which is determined by his number in the group list.*
- *The rules for assigning incentive and penalty points are as follows.*

Penalty points are calculated for:

- plagiarism (the program code does not correspond to the task variant, the identity of the program code among different works): -15 points

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

The maximum number of points for the course work: 100 points.
Criteria for evaluating the quality of a software product:
24-25 points – the development is done qualitatively, in full;
20-23 points – the development is done qualitatively, in full, but has minor flaws;
6-19 points – the development is carried out to a sufficient extent, but contains shortcomings;
0-5 points – the development is not completed in full or contains significant shortcomings.

Criteria for evaluating compliance with planning processes according to the software development methodology:
24-25 points – all tasks are planned according to the selected software development methodology, plans were adjusted according to changes;
20-23 points – all tasks are planned according to the selected software development methodology, plans were not adjusted according to changes;
6-19 points – some tasks are planned according to the selected software development methodology, plans were not adjusted according to changes;
0-5 points – tasks were not planned according to the selected software development methodology, plans were not adjusted according to changes.

Criteria for evaluating software quality assurance measures:
24-25 points – measures were taken to maintain the proper level of quality product and prevent risks;
20-23 points – measures were taken to maintain the proper level of quality product or prevent risks;

6-19 points – only product testing was conducted;
0-5 points – no measures were taken to ensure proper product quality.

Criteria for evaluating the quality and completeness of documentation:
10 points – the documentation is done at a high level, there are no comments;
6-9 points – the documentation is done qualitatively, but has shortcomings;
1-5 points – the documentation is completed at an acceptable level, but has significant shortcomings;
0 points - the documentation is done poorly.

Criteria for evaluating the quality and completeness of the presentation and demonstration of the software product:
10 points – the presentation and demonstration were performed at a high level, there are no comments;
6-9 points – the presentation and demonstration are done well, but there are shortcomings;
1-5 points – the presentation and demonstration are performed at an acceptable level, but there are significant shortcomings;
0 points – the presentation and demonstration were performed poorly.

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

The maximum number of points for completing and defending the course work: 25 points + 25 points + 25 points + 10 points + 10 points + 5 points = 100 points.

Semester control: assessment.

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

The working program of the academic discipline (syllabus):

Compiled by PhD, assistant Pogorelov V.V.;

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)