



SOFTWARE ENGINEERING METHODOLOGY.

COURSE PROJECT

Working program of the academic discipline (Syllabus)

Course Requisites

Cycle of Higher Education	<i>Second (master)</i>
Study area	<i>12 Information technologies</i>
Speciality	<i>121 Software engineering</i>
Educational program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Discipline status	<i>Normative</i>
Study form	<i>Daytime</i>
Year of study, semester	<i>1 year of training, 2 semester</i>
Discipline volume	<i>Self work: 45 hours.</i>
Semester control/ control measures	<i>Final test</i>
Course schedule	<i>According to the schedule for the spring semester of the current academic year (rozklad.kpi.ua)</i>
Language	<i>English</i>
Information about head of the course / teachers	<i>commission for accepting course project defenses: PhD, senior lecturer, Iana Khitsko, iana.khitsko@gmail.com</i>
Access to course	Google classroom: https://classroom.google.com/c/NTU00DU5MjM1NzY2?cjc=e7ug4pz

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The purpose of studying the discipline "Software Engineering Methodology. Course Project" is the formation of students' abilities to:

- analyze requirements for software systems and their design conditions;*
- choose software systems development methodology in accordance with defined requirements and software design and construction environment;*
- determine and analyze software quality metrics;*
- ensure quality inspection of software development artifacts;*
- provide unit and integration software testing;*
- determine and analyze software quality metrics;*
- ensure high-quality refactoring of the existing software code.*

The subject of the discipline "Software Engineering Methodology. Course Project" is the mathematical and algorithmic support of the processes of analysis, design, source code construction and refactoring.

The study of the discipline "Software Engineering Methodology. Course Project" contributes to the formation of students of general (SK) and professional (FC) competencies necessary for solving practical tasks of professional activity related to the development, improvement and operation of software systems of various purposes:

GC01 - ability to abstract thinking, analysis and synthesis;

PC01 - ability to analyze subject areas, form, classify software requirements;

PC03 - ability to design software architecture, model the operation of individual subsystems and modules;
PC05 - ability to develop, analyze and apply specifications, standards, rules and guidelines in the field of software engineering;
PC06 - ability to effectively manage financial, human, technical and other project resources in the field of software engineering;
PC07 - ability to critically comprehend problems in the field of information technology and at the frontiers of knowledge, to integrate relevant knowledge and solve complex problems in broad or multidisciplinary contexts;
PC08 - ability to develop and coordinate processes, stages and iterations of the software life cycle based on the application of modern models, methods and technologies of software development;
PC09 - ability to ensure software quality;
PC17 - ability to apply software engineering methodologies in practice.

Studying the discipline "Software Engineering Methodology. Course Project" contributes to students' formation of the following **program learning outcomes (PLO)** according to the educational program:

PLO01 - know and apply modern professional standards and regulations on software engineering;
PLO02 - evaluate and choose effective methods and models of software development, implementation, support and relevant processes management at all stages of the life cycle;
PLO03 - build and research models of information processes in the application field;
PLO04 - identify information needs and classify data for software design;
PLO05 - develop, analyze, justify and systematize software requirements;
PLO06 - develop and evaluate software design strategies; substantiate, analyze and evaluate options for design solutions in terms of the final software product quality, resource constraints and other factors;
PLO07 - analyze, evaluate and apply at the system level modern software and hardware platforms to solve complex problems of software engineering;
PLO08 - develop and modify software architecture to meet customer requirements;
PLO09 - choose reasonable paradigms and programming languages for software development; apply modern software development tools in practice;
PLO10 - modify existing and develop new algorithmic solutions for detailed software design;
PLO11 - ensure quality at all stages of the software life cycle, including the use of relevant models and assessment methods, as well as automated software testing and verification tools;
PLO13 - configure software, manage its changes and develop software documentation at all stages of the life cycle;
PLO14 - predict the development of software systems and information technology;
PLO15 - carry out software reengineering in accordance with customer requirements;
PLO16 - plan, organize and perform software testing, verification and validation;
PLO17 - collect, analyze, evaluate the information needed to solve scientific and applied problems, using scientific and technical literature, databases and other sources;
PLO21 - know the theoretical foundations underlying research methods of information systems and software, research methodologies and computational experiments.

2. Discipline prerequisites and postrequisites (place in the structural and logical education scheme according to the relevant educational program)

The successful study of the discipline "Software Engineering Methodology. Course Project" is preceded by the study of the disciplines "Programming", "Object-Oriented Programming", "Software Quality", "Software Requirements" of the curriculum for bachelor's training in the specialty 121 Software Engineering.

The theoretical knowledge and practical skills obtained during the mastering of the discipline "Software Engineering Methodology. Course Project" ensure the successful completion of course projects and master's theses in the specialty 121 Software Engineering.

3. Content of the course

Online game development with a client and server part with the division of students into three or four teams and support of software engineering methodologies during development.

Basic requirements for the program:

1. Client-server exchange protocol availability for effective interaction between parts of the system at the initial stage of development.
2. Clearly expressed interface part presence for training the skills of graphical interface verification.
3. Unit and integration tests creation for verification of the server part.

4. Organization of the defense and implementation of the course project

The course project is carried out by a group of five to six students. A typical distribution of roles between performers:

1. Requirements analysis.
 2. Analysis of the subject area, conceptual architecture development, available technologies analysis.
 3. Client-server protocol (for the team developing the server part) and the graphical interface (for the team developing the client part) development.
 4. Detailed architecture development.
 5. Coding and creation of unit tests.
 6. Creation of test scenarios.
- The course project consists of a software system and its documentation.
 - The course project is defended in the form of a presentation with a demonstration of the developed software product.
 - Tasks, their planning and distribution among teams takes place in the Trello web application.

5. Course project implementation schedule

- Division into teams and distribution of roles - until March 1.
- Analysis of the subject area and available technologies, development of conceptual architecture - by March 15.
- Requirements analysis, client-server protocol development - 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.
- Defense of the course project - until May 30.

6. Coursebooks and teaching resources

Basis reference:

1. Educational materials from the discipline "Software engineering methodology. Course project". Use to master practical skills in the discipline. The materials are in Google classroom. Access is granted to registered students.

Policy and Assessment

7. Course policy

- Adherence to the policy of academic integrity.
- Rules for protecting the works of the computer workshop: the works must be done according to the option of the student, 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 version, the identity of the program code among different works): -15 points.

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

Maximum points value for course project: 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 – 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: final test.

Points	Grade
100-95	Excellent
94-85	Very good
84-75	Good

74-65	Satisfactorily
64-60	Enough
< 60	Unsatisfactorily
Admission conditions are not met	Not admitted

Course syllabus:

Is created by PhD, senior lecturer Iana Khitsko.

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)