eKHMELNYTSKY NATIONAL UNIVERSITY

Faculty of Information Systems

Departments of software engineering

I APPROVE
Dean of the Faculty of IT
Hovorushchenko T.O
"03" 09 2024 p.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ • ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ • ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

SILABUS

Academic discipline **Requirements analysis and software quality**
Educational and scientific program **Software Engineering**

Level of higher education **First (bachelor's degree)**

Position.	**Content of information**
Teacher(s)	**Onyshko Oksana Hryhorivna**
Teacher profile	http://ipz.khnu.km.ua/онишко-о-г/
Teacher's e-mail	fpcts.khnu@gmail.com
Contact phone number	Filled in by agreement
Discipline page in the ISU	https://msn.khnu.km.ua/course/view.php?id=6688#section-0
Academic year	2024-2025
Consultations	Intramural: Tuesday 6 p.m., 1-210 Online: if necessary and by prior arrangement

General characteristics of the discipline

Status of the discipline	Form of education	Course.	Semester	Total volume		Number of hours				Course project	Course work	The form of semitic control	
				European credit	Hours.	Audit classes			Independent work			Credit	Examination
						Total	Lectures	Laboratory work					
3	D/o	2	3	7	210	85	34	51	125				+

Summary of the subject

The discipline "Software Requirements Analysis" is a discipline from the cycle of professional and practical training in the field of software engineering. For successful completion of this course, the student must have programming skills in high-level languages, be able to work with databases and be able to summarize information obtained from various sources, be able to present the results of their research.

The discipline is taught for applicants of the first (bachelor's) level of higher education of full-time education in the field of information technology. In teaching the discipline, active and creative forms of teaching are used, in particular, methods of problem-based learning.

Aims and objectives of the discipline

The purpose of the ***discipline*** is to provide theoretical and practical training of students, which should ensure that students acquire basic knowledge in the field of modern design technologies, software requirements engineering, gain practical skills in the implementation of software systems, the basics of modeling and analysis of software systems, development analysis, specification and requirements management

Objectives of the discipline. The main objectives of studying the discipline "Software Requirements Analysis" is knowledge about the development and analysis of requirements for a software product. The requirements are classified, the properties of requirements are analyzed, methodologies, standards, notations for working with requirements are considered. The components of requirements analysis are analyzed: identification, specification and documentation, verification. The role of models, tools, and requirements management processes is considered.

Expected learning outcomes

According to the Standard of Higher Education in the specified specialty and educational program, the discipline must provide:

competencies: Ability to solve complex specialized problems or practical problems of software engineering, characterized by complexity and uncertainty of conditions, using theories and methods of information technology. PC1. Ability to identify, classify and formulate software requirements. PC4. Ability to formulate and ensure software quality requirements in accordance with customer requirements, terms of reference, standards. PC5. Ability to comply with specifications, standards, rules and guidelines in the professional field when implementing life cycle processes. PC10. Ability to accumulate, process and systematize professional knowledge of software development and maintenance and recognize the importance of lifelong learning. PC11. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate models and approaches to software development. PC12. Ability to carry out the process of system integration, apply change management standards and procedures to maintain the integrity, overall functionality and reliability of the software.

program learning outcomes:. PLO1. Analyze, purposefully search and select the necessary information and reference resources and knowledge for solving professional problems, taking into account modern advances in science and technology. PLO3. Know the basic processes, phases and iterations of the software life cycle. PLO9. To know and be able to use methods and tools for collecting, formulating and analyzing software requirements. PLO20. Know the approaches to assessing and ensuring software quality.

Thematic and calendar plan for studying the discipline

No. of the week	**Lecture topic**	**Topic of the laboratory session**	**Independent work**		
			Contents.	Year .	Literature
1	**2**	**3**	**4**	**5**	**6**

1	**Lecture 1:** Stages of software development. The role of requirements in software project planning. Subject and objectives of the discipline	Analysis of the problem. Formulation. Work with real customers, identification of stakeholders and interviews with them, analysis of the material received, formulation of the problem, its relevance and interest needs	Study of lecture material. Preparation for LR1.	18	1-5; 7; 11
2	**Lecture 2.** Classification of requirements. User requirements		Study of lecture material. Preparation for the defense of LR1.	7	4; 7; 11
3	**Lecture 3**. . Functional and nonfunctional requirements, features of implementation in software projects....	Business Process Modeling Notation (BPMN). Using them in modeling and analyzing business processes	Study of lecture material. Preparation for LR2.	7	2; 5; 12
4	**Lecture 4.** Checking the requirements. Changing and modifying the requirements for a software project.		Study of lecture material. Preparation for the defense of LR2.	8	4; 7; 11
5	**Lecture 5.** Requirements management.	Using UML to describe, visualize, and document various artifacts of a software-intensive system	Preparation for TK T 1 -4. Study of lecture material. Preparation for LC 3.	11	4; 7; 11
6	**Lecture 6:** RUP technology. Unified modeling language UML		Study of lecture material. Preparation for the defense of the LR 3.	4	1; 4; 12

7	**Lecture 7.** RUP technology. The process of software development.	Visual representation of business process stages through flowcharting technique	Study of lecture material. Preparation for the LR 4.	3	4; 7; 11
8	**Lecture 8.** Software design automation tools		Study of lecture material. Preparation for the defense. ЛР4.	3	1; 4; 12
9	**Lecture 9.** Usercentered design.	Represent the data flow in the system through a data flow diagram (DFD) and identify potential problems and opportunities in the system	Study of lecture material. Preparation for LR5.	3	2; 4-6; 9
10	**Lecture 10.** Implementation of subsystems. Diagrams of structures. Development of software system architecture.		Study of lecture material. Preparation for the defense. ЛР5.	3	4; 6; 13
11	**Lecture 11.** Developing business class and class diagrams.	Identify the roles and activities of stakeholders in the system through role activity diagrams (RADs).	Study of lecture material. Preparation for LR6.	3	1-3; 7; 11
12	**Lecture 12.** Principles of software development. Interaction diagrams.		Study of lecture material. Preparation for the defense. ЛР6.	3	2; 13
13	**Lecture 13.** Interaction diagrams in modeling software systems	Planning and tracking project progress through Gantt charts	Study of lecture material. Preparation for LR7.	3	4-5; 9-12

14	**Lecture 14.** Stages of developing requirements for software systems		Study of lecture material. Preparation for the defense. ЛР7.	10	4; 7; 13
15	**Lecture 15:** Terms of reference for a software project	Modeling and analyzing functional requirements using IDEF diagrams.	Study of lecture material. Preparation for LR8.	10	2; 5; 11
16	**Lecture 16.** Software development through testing		Study of lecture material. Preparation for the defense of LR8.	10	4; 7; 13
17	**Lecture 17.** Software development through testing (test-first programming). Implementation of software systems.	Final lesson	Preparation for TC T5-8 Testing	11	3-4; 6; 13

Note: * The sequence of classes is determined by the schedule (may not correspond to the numbered weeks)

Discipline policy.

The organization of the educational process in the discipline meets the requirements of the provisions on organizational and educational and methodological support of the educational process, the educational program and the curriculum. The student is obliged to attend lectures and laboratory classes according to the schedule, not to be late for classes, and to complete assignments in accordance with the schedule. The student is obliged to work out the missed laboratory lesson independently in full and report to the teacher no later than one week before the next certification. The student must prepare for laboratory classes on the relevant topic and be active. The knowledge acquired by a person in the discipline or its individual sections in non-formal education is credited in accordance with the Regulations on the procedure for re-crediting learning outcomes at KhNU (http://khnu.km.ua/root/files/01/06/03/006.pdf).

Criteria for assessing learning outcomes.

Each type of work in the discipline is evaluated on a four-point scale. The semester final grade is determined as a weighted average of all types of academic work completed and passed positively, taking into account the weighting factor. The weighting factors vary depending on the structure of the discipline and the importance of its individual types of work. When assessing students' knowledge, various means of control are used, in particular: an oral survey before admission to the laboratory work is carried out at the beginning of the work; mastering theoretical material on topics is checked by test control; the quality of performance, acquisition of theoretical knowledge and practical skills is checked by defending each laboratory work in accordance with the work program of the discipline and the work curriculum.

Structuring the discipline by types of work and evaluating the results of students' learning in the semester by weighting coefficients

Audit work									Semester control, exam (I)	Final score
Laboratory work (LW)								Test.		
1	2	3	4	5	6	7	8	T	0,4	LR*0.3+T*0.3+I*0.4
VC = 0.3								VC = 0.3		

Symbols: WK - weighting coefficient, LW - laboratory work, T - test, E - exam.

Evaluation of test tasks. The thematic test for each applicant consists of twenty test tasks, each of which is evaluated by one point. The maximum amount of points that an applicant can score is 20.

The evaluation is based on a four-point scale.

The correspondence of the scores for the test task to the grade assigned to the applicant is shown in the table below.

Sum of points for the test task	1-11	12-14	15-18	19-20
Assessment.	2	3	4	5

The testing period is 20 minutes. Testing is conducted using the MOODLE modular learning environment. The applicant registers the correct answers online in the MOODLE modular environment. After 30 minutes, students complete the test and send their answers to the server. The teacher announces the results of the test according to the gradebook of the MOODLE modular environment.

If the applicant receives a negative grade, he or she must retake it in the prescribed manner, but always before the next control.

The final semester grade according to the national and ECTS scales is set in an automated mode after all grades are entered into the electronic journal. The ratio of the national grading scale and the ECTS grading scale is shown in the following table.

Correlation of the national grading scale and the ECTS grading scale

ECTS assessment	Institutional interval scoring scale	Domestic assessment, criteria		
A	4,75-5,00	5	***Excellent*** **-** deep and complete mastery of the educational material and identification of relevant skills	Enrolled
B	4,25-4,74	4	***Good*** **-** complete knowledge of the study material with a few minor mistakes	
C	3,75-4,24	4	***Good*** **-** generally correct answer with two or three significant errors	
D	3,25-3,74	3	***Satisfactory*** - incomplete mastery of the program material, but sufficient for practical work in the profession	
E	3,00-3,24	3	***Satisfactory*** - incomplete mastery of the program material that meets the minimum assessment criteria	
FX	2,00-2,99	2	Unsatisfactory - unsystematic knowledge and inability to continue studying without additional knowledge of the discipline	Unaccounted
F	0,00-1,99	2	***Unsatisfactory*** **-** serious further work and re-study of the discipline is required	

A credit is awarded if the weighted average score received by the applicant in the discipline is in the range from 3.00 to 5.00 points. In this case, the grade "passed" is assigned according to the national scale, and the ECTS scale is the letter grade corresponding to the number of points scored by the applicant in accordance with the Correlation table.

Questions for the final control of the discipline "Software modeling and evaluation"

1. Name the components of the object-oriented approach
2. Describe the classes and relationships between them
3. What are the categories of classes
4. What are the signs of a complex system
5. Describe the structure and design of complex systems.
6. Define the concept of system decomposition
7. System structure
8. System structure
9. What are the specifications
10. Give a description of the object model
11. What state and transition diagrams consist of
12. What are the components of an object diagram
13. System architecture of the software
14. Object-oriented analysis, design and programming
15. Give a description of the interaction diagram
16. Classification of systems
17. Project management
18. Identification of system components
19. Abstraction mechanisms
20. System architecture of the software system
21. Usage options
22. UML as a software modeling language
23. UML class diagrams
24. Universal approach to software modeling
25. System decomposition
26. UML diagrams of objects
27. UML action diagrams
28. Components of the object-oriented approach
29. UML diagrams of implementation scenarios
30. Object model
31. Components of the object-oriented approach
32. Object-oriented analysis, design, and programming
33. Object model
34. Classification of systems
35. Object-oriented analysis, design, and programming
36. UML diagramsof use cases
37. Object model of the system.
38. Components of the object approach.
39. The use of object models.
40. Classification of models and their identification.

for

Methodological support

The educational process in the discipline is provided with the necessary educational and methodological developments in a modular environment.

Recommended reading

1. Methodology for assessing the sufficiency of information for determining the quality of software: a monograph. Khmelnytskyi: Khmelnytsky National University, 2017. 310 c.

2. Tabunshchyk, T.I., Kaplienko, G.V., Petrova, O.A. (2016) Design and modeling of software of modern information systems in Zaporizhzhia: Wild Field,

3. Introduction to Software Engineering and Software Life Cycle Management Guide to Software Engineering Base of Knowledge (SWEBOK): Trans. from English by S. Orlik Retrieved from:sorlik.blogspot.com/

4. Tabunshchik G.V., Kaplienko, T.I. Petrova. (2016) Design and modeling of software of modern information systems. Zaporizhzhia.

7. 2. Petryk MR, Petryk OY Software modeling: scientific and methodological manual. Ternopil: Ternopil National Technical University, 2015. 200 c.

.

Developer: Onyshko O.G.
Agreed:
Head of the Department. OF THE DEPARTMENT OF INDUSTRIAL ENGINEERING Bedratiuk L.P.
Guarantor of the ONP Bedratiuk L.P.