KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED
Dean of IT Faculty
_____ HOVORUSHCHENKO T.
"14" 09 2025.

WORKING PROGRAMME OF THE EDUCATIONAL COMPONENT

Software Requirement Analysis and Quality

Field of Study: F Information Technology
Specialty: F2 Software Engineering
Level of Higher Education: First (Bachelor's) Level
Educational and Professional Programme: Software Engineering
Course Load: 7 ECTS credits **Course Code:** CPT.14
Language of Instruction: English
Status of the Educational Component: Compulsory (General Training)
Faculty: Faculty of Information Technology
Department: Department of Software Engineering

Form of Study	Year	Semester	Total Credits		Number of hours						Course project	Coursework	Semester control form	
					Contact Hours					Independent Work (incl. Individual Tasks)				
			ECTS credits	hours	Total	Lectures	Laboratory works	Practical classes	Seminar classes				pass/ fail test	Exam
D	1	1	6	180	66	32	34			114				+

The working programme is based on the Educational and Professional Programme "Software Engineering" within the specialty "Software Engineering".

Program's author _____ O.G. Onyshko

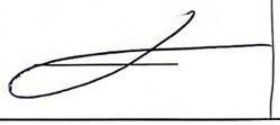
Approved at the meeting of the Department of Software Engineering
Minutes No. 1 dated August 30, 2025
Head of the Department _____ L.P. Bedratyuk

The working programme was reviewed and approved by the Academic Council of the Faculty of Information Technology

Chair of the Academic Council _____ Tetiana HOVORUSHCHENKO

Khmelnytskyi 2025

LETTER OF APPROVAL

Position	Department Name	Signature	First Name, LAST NAME
Head of *D*epartment DSc, Prof.	Software Engineering		Leonid BEDRATIUK
Programme Guarantor *DSc*, Prof.	Software Engineering		Leonid BEDRATIUK

SOFTWARE REQUIREMENT ANALYSIS AND QUALITY

Type of Educational Component	Compulsory
Level of Higher Education	First (Bachelor's) Level
Language of Instruction	English
Semester	1
Number of ECTS Credits Assigned	6
Forms of Study the Course is Designed For	Full-time

Learning Outcomes. In accordance with the Higher Education Standard and the curriculum, the discipline must provide:

competencies: ability to identify, classify and formulate software requirements; ability to formulate and ensure software quality requirements in accordance with customer requirements, terms of reference and standards; ability to comply with specifications, standards, rules and recommendations in the professional field when implementing life cycle processes; ability to accumulate, process and systematize professional knowledge on the creation and maintenance of software and recognize the importance of

program learning outcomes: to analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology; to know the main processes, phases and iterations of the software life cycle; to know and be able to use methods and tools for collecting, formulating and analyzing software requirements; to know approaches to assessing and ensuring software quality

Course Content. Classification of software systems, software requirements, functional and non-functional requirements, validation, verification of requirements, RUP, MSF technology, types of classifiers and relations, design, implementation of subsystems, testing of software systems

Planned Learning Activities. The minimum amount of classroom-based learning activities in one ECTS credit for a course at the first (Bachelor's) level of higher education in full-time study mode is 10 hours per 1 ECTS credit.

Forms (Methods) of Instruction: Lectures (using problem-based learning and visualisation methods), Laboratory works, Independent work

Assessment Methods: Laboratory work defence, Testing

Form of Final Assessment: Exam

Learning Resources:

1. Laplante, P. A., Kassab, M. Requirements Engineering for Software and Systems (4th ed.). Routledge, 2022. – 420 p.
2. Tavana, M., Pick, K. W. Managing Requirements Knowledge. Springer, 2022. – 300 p.
3. Wiegers, K. E., Beatty, J. Software Requirements Essentials: Core Practices for Success. Pearson Education, 2022. – 280 p.
4. O'Regan, G. Introduction to Software Quality (2nd ed.). Springer, 2022. – 440 p.
5. Dalpiaz, F., Spoletini, P. (eds.). Requirements Engineering: Foundation for Software Quality (REFSQ 2023 Proceedings). Springer, 2023. – 385 p.
6. Walkinshaw, N. Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer, 2023. – 350 p.

Lecturer: Candidate of Pedagogical Sciences, Associate Professor Onyshko O.H.

3. EXPLANATORY NOTE

The course " *Software Requirement Analysis and Quality*" is one of the general training courses and occupies a leading place in the training of students of the first (Bachelor's) level of higher education, full-time mode of study (hereinafter – full-time), who study under the Educational and Professional Programme "Software Engineering" within the specialty F2 "Software Engineering".
Prerequisites – CPT.05 Software Engineering Basics

Postrequisites – CPT.10 Software Modelling and Evaluation; CPT.08 Computer Network Organisation; CPT.18 Qualification Work
competences: GC1. Ability for abstract thinking, analysis and synthesis. PC1. Ability to identify, classify, and formulate software requirements PC4. Ability to formulate and ensure software quality requirements following client requirements, technical specifications, and standards. PC5. Ability to adhere to specifications, standards, rules, and recommendations in the professional field during the implementation of lifecycle processes. PC10. Ability to accumulate, process, and systematise professional knowledge regarding the creation and maintenance of software and recognise the importance of lifelong learning. PC12. Ability to execute the system integration process and apply standards and change management procedures to maintain the integrity, overall functionality, and reliability of the software.
programme learning outcomes: PLO3 To understand the software lifecycle's leading processes, phases, and iterations. PLO4 To know and apply professional standards and other regulatory documents in the field of software engineering.
PLO9 To know and be able to use methods and tools for collecting, formulating, and analysing software requirements. PLO11 To select initial data for design, guided by formal methods of requirement descriptions and modelling. PLO20 To know approaches to software quality evaluation and assurance.
Purpose of the course. 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
Subject of the course. The subject of the discipline " Software Requirement Analysis and Quality" covers theoretical knowledge, tasks, methods and requirements for software, its design and construction processes.

Course objectives. The main tasks of studying the discipline " Software Requirement Analysis and Quality" are knowledge about the development and analysis of requirements for a software product. Classification of requirements is carried out, 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.

4. STRUCTURE OF THE COURSE CREDITS

Topic Title	Number of hours allocated to:		
	Lectures	Lab work	Independent work
Topic 1: Introduction. Classification of software systems	2	6	19
Topic 2. Software requirements	2	6	19
Topic 3: The impact of software requirements analysis on iterative project development cycles.	4	6	19
Topic 4. Software design concepts	4	6	19
Topic 5. Development of requirements. Norms and standards. ESPD	2	6	19
Topic 6: Software verification and testing	2	4	19
Total for the semester	16	34	114

5.1. CONTENT OF THE LECTURE COURSE

Lecture No.	List of Lecture Topics and Annotations	Hours
Topic 1. Introduction. Types of software systems		
1	**:** Stages of software development. The role of requirements in software project planning. Subject and objectives of the discipline "Software Requirements Analysis". Classification of software systems. Overview of standards and technologies for the development of software systems (MRPII, CRM, ERPII, etc.). Methodology of structural analysis and design SADT. RUP software development methodology. Examples of successful and unsuccessful formation of requirements for software projectsRef.: [1] pp. 30–40; [2] pp. 6–9; [3] pp. 18–28	2
Topic 2. Software requirements		
2	**.** Classification of requirements. User requirements. Methods of identifying and processing requirements in MSF, RUP, XP technologies. Types of software requirements. Classification of requirements. Sources of requirements. Quality of requirements. User requirements. Formalization of requirements. Functional and non-functional requirements. Methods of identifying and processing requirements: interviews, questionnaires, active and passive observation, surveys, study of technical documentation. Technologies for reusing design solutions. Identification of requirements using MSF, RUP, XP technologies. User stories, prototyping, use cases, brainstorming.Ref.: [1] pp. 41–42; [2] pp. 14–17, 26–29; [3] pp. 5–18	2
Topic 3. The impact of software requirements analysis on iterative project development cycles		

Lecture No.	List of Lecture Topics and Annotations	Hours
3	Requirements management. Quality management. Decision management. The impact of software requirements analysis on iterative project development cycles. Requirement Management. Quality management of software development. The ISO 9000 standard. Decision management. COBIT standard. Business decisions and decision management. ITIL information technology infrastructure library.Ref.: [1] pp. 62–83; [2] pp. 33–37; [4] pp. 28–47	2
4	RUP technology. Unified modeling language UML 2.0. Three groups of diagrams for modeling software systems. Design templates. The Rational Rose software environment. Ref.: [2] pp. 14–23; [5] pp. 18–21	2
Topic 4. Concepts of software design		
5	RUP technology. The process of software development. Methods of identifying and documenting requirements. Unified Modeling Language UML, history of creation and basic concepts. Three groups of diagrams for modeling software systems in UML 2.0. Types of classifiers and relationships.Ref.: [1] pp. 83–85; [2] pp. 50–59; [3] pp. 47–62	2
6	Tools for automation of software design by Rational Software. Rational Rose software environment. Design patterns: abstract-concrete, composite, player-role, singleton, observer, delegation, facade, adapter. Ref.: [2] pp. 7–9; [4] pp. 18–25	2
Topic 5. Development of requirements. Norms and standards. ESPD		
7	Stages of developing requirements for software systems. Specification of requirements. Standards and methodologies for business process design. Unified system of program documentation (USPD). Stages and stages of development. Design of text program documents. Technical specifications. Design of operational documents. Ref.: [1] pp. 85–112; [3] pp. 148–251	2
Topic 6. Software verification and testing		
8	Software development through testing (test-first programming). Implementation of software systems. Documentation developed for the implementation of a software project. Ref.: [7] pp. 5–7; [8] pp. 18–28	2
Total for the semester		**16**

5.2. CONTENT OF LABORATORY WORKS

Topic No.	Laboratory Session Topic	Hours
1	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.	4
2	Business Process Modeling Notation (BPMN). Using them in modeling and analyzing business processes.	4
3	Using UML to describe, visualize, and document various artifacts of a software-intensive system.	4
4	Visual representation of business process stages using the flowchart technique.	4
5	Represent the data flow in the system through a data flow diagram (DFD) and identify potential problems and opportunities in the system.	4
6	Identify the roles and activities of stakeholders in the system through role activity diagrams (RADs).	4
7	Planning and tracking project progress through Gantt charts.	4
8	Modeling and analyzing functional requirements using IDEF diagrams.	6
Total for the semester		**34**

5.3. CONTENT OF INDEPENDENT WORK

The volume of independent work in the discipline " Software Requirement Analysis and Quality " is 128 hours. They include the study of lecture material, theoretical and laboratory tasks, preparation for laboratory work, their defense, and current testing..

Week No.	Type of Independent Work	Hours
1	Study of theoretical material from T1, preparation for Laboratory Work No. 1 Ref. [1] pp. 30–40; [2] pp. 6–9	7
2	Study of theoretical material from T1, preparation for Laboratory Work No. 1 Ref: [1] pp. 45–50; [2] pp. 16–19	7
3	Study of theoretical material from T2, preparation for Laboratory Work No. 2 Ref.: [4] pp. 5–7; [5] pp. 48–51	7
4	Study of theoretical material from T2, preparation for Laboratory Work No. 2 Ref.: [3] pp. 14–27; [4] pp. 8–15	7
5	Study of theoretical material from T2, preparation for Laboratory Work No. 3 Ref.: [5] pp. 20–24; [6] pp. 10–23	7
6	Study of theoretical material from T3, preparation for Laboratory Work No. 3	7

Week No.	Type of Independent Work	Hours
	Ref.: [6] pp. 5–7; [8] pp. 48–51	
7	Study of theoretical material from T3, preparation for Laboratory Work No. 4 Ref.: [3] pp. 45–50; [6] pp. 24–35	7
8	Study of theoretical material from T4, preparation for Laboratory Work No. 4. Preparation for TC No. 1 Ref.: [6] pp. 35–47	7
9	Study of theoretical material from T4, preparation for Laboratory Work No. 5 Ref.: [7] pp. 17–35; [8] pp. 38–41	7
10	Study of theoretical material from T4, preparation for Laboratory Work No. 5 Ref.: [4] pp. 125–137; [5] pp.148–151	7
11	Study of theoretical material from T4, preparation for Laboratory Work No. 6 Ref.: [1] pp. 55–67; [2] pp. 48–51	7
12	Study of theoretical material from T5, preparation for Laboratory Work No. 6 Ref.: [4] pp. 59–67; [6] pp. 25–56	7
13	Study of theoretical material from T5, preparation for Laboratory Work No. 7 Ref.: [3] pp.35–49; [5] pp. 56–71	7
14	Study of theoretical material from T5, preparation for Laboratory Work No. 7 Ref.: [7] pp.124–153	7
15	Study of theoretical material from T6, preparation for Laboratory Work No. 8 Ref.: [4] pp. 5–7; [6] pp. 48–51	7
16	Study of theoretical material from T6, preparation for Laboratory Work No. 8. Preparation for TC No. 2 Ref.: [7] pp. 55–67; [8] pp. 38–41	7
17	Study of theoretical material from T6, preparation for the final exam	2
Total:		114

Notes: TC – Test Control, T1–T6 – Topics of the course.

6. TECHNOLOGIES AND TEACHING METHODS

The learning process for the course is based on the use of both traditional and modern teaching technologies and methods, in particular: lectures (using visualisation methods, problem-based and interactive learning, motivational techniques, and information and communication technologies); laboratory works (using training exercises, problem situation analysis, explanation, discussions, etc.); independent work (study of theoretical material, preparation for laboratory works, ongoing and final assessment), with the use of information and computer technologies and distance learning technologies.

7. METHODS OF ASSESSMENT

Ongoing assessment is carried out during practical classes, as well as on the days of control activities established by the working programme and the academic schedule.
The following methods of ongoing assessment are used:
– test-based assessment of theoretical material;
– evaluation of the results of laboratory work defence.

When determining the final semester grade, the results of both ongoing assessment and final assessment are taken into account. The final assessment is conducted on all the material of the course according to examination papers prepared in advance and approved at the meeting of the department. A student who has scored less than 60 percent of the maximum score for any type of academic work is not allowed to undergo the semester assessment until the amount of work stipulated by the Working Programme is completed. A student who has achieved a positive weighted average score (60 percent or more of the maximum score) for all types of ongoing assessment but has failed the examination is considered to have an academic debt.
Elimination of academic debt for the semester assessment is carried out during the examination session or according to the schedule set by the dean's office in accordance with the *Regulation on Control and Assessment of Learning Outcomes of Students at Khmelnytskyi National University*.

8. COURSE POLICY

The policy of the academic course is generally determined by the system of requirements for the student as stipulated by the current University regulations on the organisation and teaching and learning support of the educational process. In particular, this includes completing the safety briefing; attendance at course classes is compulsory. For valid reasons (documentarily confirmed), theoretical training may, with the lecturer's approval, take place online. Successful completion of the course and the formation of professional competences and programme learning outcomes require preparation for each laboratory work (studying the theoretical material for the topic of the work), active participation during the class, thorough preparation of the report, defence of the results, participation in discussions regarding the constructive decisions made during the laboratory works, etc.
Students must meet the established deadlines for completing all types of academic work in accordance with the Working Programme of the course. A missed laboratory class must be completed within the deadline set by the lecturer, but no later than two weeks before the end of the theoretical classes in the semester.
The student's mastery of the theoretical material of the course is assessed through testing.
When performing laboratory work, the student must comply with the policy of academic integrity (cheating, plagiarism — including with the use of mobile devices — is prohibited). If a violation of academic integrity is detected in any type of academic work, the student receives an unsatisfactory grade and must re-do the task on the relevant topic (type of work) as stipulated by the Working Programme. Any form of academic dishonesty is unacceptable.
Within the framework of studying the course, students are provided with recognition and crediting of learning outcomes acquired through non-formal education, available on accessible platforms (https://prometheus.org.ua/, https://www.coursera.org/), which contribute to the formation of competences and the deepening of learning outcomes defined in the Working Programme of the course, or ensure the study of a relevant topic and/or type of work from the course syllabus (for more details, see the *Regulation on the Procedure for Recognition and Crediting of Learning Outcomes of Students at Khmelnytskyi National University*).

9. ASSESSMENT OF STUDENTS' LEARNING OUTCOMES DURING THE SEMESTER

Assessment of a student's academic achievements is carried out in accordance with the *Regulation on the Control and Assessment of Students' Learning Outcomes at Khmelnytskyi National University*.
During the ongoing assessment of the work performed by the student for each structural unit and the results obtained, the lecturer awards a certain number of points as set out in the Working Programme for that type of work.
Each structural unit of academic work may be credited only if the student has scored at least 60 percent (the minimum level for a positive grade) of the maximum possible points assigned to that structural unit.

When assessing students' learning outcomes for any type of academic work (structural unit), it is recommended to use the generalised criteria provided below:

Table – Assessment Criteria for Student Learning Outcomes

Grade and Level of Achievement of Intended Learning Outcomes and Competences	General Description of Assessment Criteria
Excellent *(High)*	The student has deeply and fully mastered the course content, confidently navigates it, and skilfully uses the conceptual framework; demonstrates the ability to connect theory with practice, solve practical problems, and clearly express and justify their reasoning. An excellent grade implies a logical presentation of the answer in the language of instruction (oral or written), high-quality formatting of the work, and proficiency in using specialised tools, instruments, or application software. The student demonstrates confidence when answering reformulated questions, is capable of making detailed and summarised conclusions, and shows practical skills in solving professional tasks. The answer may contain two or three minor inaccuracies.
Good *(Average)*	The student has shown full understanding of the course content, possesses the conceptual framework, and navigates the material well; applies theoretical knowledge consciously to solve practical tasks. The answer is generally well-articulated, although some minor inaccuracies or vague formulations of rules or principles may occur. The student's answer is based on independent thinking. Two or three minor mistakes are acceptable.
Satisfactory *(Sufficient)*	The student demonstrates knowledge of the basic course material sufficient for continued learning and practical activity in the profession; is able to complete the practical tasks foreseen by the programme. The answer is usually based on reproductive thinking. The student has limited knowledge of the structure of the discipline, makes inaccuracies and significant errors in the answer, and hesitates when answering reformulated questions. Nevertheless, they possess basic skills to complete simple practical tasks that meet the minimum assessment criteria and, under the lecturer's guidance, can correct their mistakes.
Unsatisfactory *(Insufficient)*	The student demonstrates fragmented, unstructured knowledge, cannot distinguish between main and secondary ideas, makes conceptual errors, misinterprets definitions, presents material in a chaotic and unconfident manner, and cannot apply knowledge to solve practical problems. An unsatisfactory grade is typically given to a

Grade and Level of Achievement of Intended Learning Outcomes and Competences	General Description of Assessment Criteria
	student who is unable to continue learning the subject without additional study.

Structuring of the Course by Types of Academic Work and Assessment of Student Learning Outcomes

In-Class Work								***Assessment Activities***		***Semester Final Assessment***	
Laboratory Work №:								Test control:		Exam	Total
1	2	3	4	5	6	7	8	T 1-3	T 4-6		
Number of points per type of academic work (min–max)											
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	**60-100***
24-40								**12-20**		**24-40**	

Notes: If the number of points earned for any type of academic work in the course is below the established minimum, the student receives a failing grade and must retake the work within the deadline set by the lecturer (or dean). The institutional grade is determined in accordance with the table "**Correspondence between the Institutional Grading Scale and the ECTS Grading Scale**".

Assessment of Laboratory Work Defence Results

A laboratory work completed and formatted in accordance with the requirements established in the Methodological Guidelines is comprehensively assessed by the lecturer during its defence based on the following criteria:

– independence and accuracy of execution;

– completeness of the answer and understanding of the principles of building machine learning models;

– ability to justify the choice of algorithm or method;

– correctness of model implementation in the Python programming environment using appropriate libraries;

– ability to interpret the results of modelling and evaluate their suitability for solving the given task.

When assessing a laboratory session, the lecturer uses the generalised criteria outlined in the table "Assessment Criteria for Student Learning Outcomes" (minimum passing score – 3 points, maximum – 5 points).

If the student demonstrates a knowledge level below 60 percent of the maximum score established in the Working Programme for each structural unit, the laboratory work is not credited. In such a case, the student must study the topic more thoroughly, review the methodology, correct major mistakes, and re-defend the work at the time set by the lecturer.

Assessment of Test-Based Control Results

Each test included in the Working Programme consists of 30 test items, each carrying equal weight. According to the table for structuring types of academic work, the student may receive between 3 and 5 points depending on the number of correct answers.

Distribution of points depending on the number of correct answers to test items:

The test duration is 30 minutes. Students complete the test online in the Modular Learning

Environment.
If a failing grade is received, the test must be retaken before the next scheduled assessment.

Distribution of points depending on correct answers to test questions

Number of Correct Answers	1-17	18-23	24-26	27-30
Percentage of Correct Answers	0-59	60-79	80-89	90-100
Number of Points	-	3	4	5

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is provided in the table "Correspondence" below.

Assessment of the Final Semester Control (Exam)

The educational programme provides for a final semester control in the form of an examination, the purpose of which is to systematically and objectively assess both the theoretical and practical preparation of the student in the course. The examination is conducted according to examination papers prepared in advance and approved at the meeting of the department. In accordance with this, the examination paper contains a combination of both theoretical questions (including in test form) and practical tasks.

Table – Assessment of Final Semester Examination Results *for full-time students (40 points allocated for final control)*

Type of Task	**For each individual type of task**		
	Minimum (Satisfactory) Score	***Potential Positive Score (Good)****	**Maximum (Excellent) Score**
Theoretical Question № 1	3	4	5
Theoretical Question № 2	3	4	5
Practical Tasks *(6 tasks worth 3 points each)*	18	24	30
Total:	**24**	**32**	**40**

Note. ***A passing score for the exam, different from the minimum (24 points) and the maximum (40 points), falls within the range of 25–39 points and is calculated as the sum of points for all structural elements (tasks) of the exam.***

For each individual type of task in the final semester assessment, the assessment criteria for student learning outcomes provided above (see **Table – Assessment Criteria for Student Learning Outcomes**) are applied.

The final semester grade according to the institutional grading scale and the ECTS grading scale is determined automatically after the lecturer enters the assessment results in points for all types of academic work into the electronic gradebook. The correspondence between the institutional grading scale and the ECTS grading scale is shown below in the **Correspondence Table**.

The final examination grade is recorded if the total number of points accumulated by the student in the course as a result of ongoing assessment falls within the range of 60 to 100 points. In this case, a grade of *Excellent/Good/Satisfactory* is assigned according to the institutional scale, and a letter grade is assigned according to the ECTS scale, corresponding to the total number of points earned by the student as specified in the **Correspondence Table**.

Table – Correspondence between the Institutional Grading Scale and the ECTS Grading Scale

<table>
<tr><th rowspan="2">ECTS Grade</th><th rowspan="2">Rating Scale (Points)</th><th colspan="2">Institutional Grade(Level of Achievement of the Intended Learning Outcomes in the Course)</th></tr>
<tr><th>Pass/ Fail</th><th>Exam / Graded Credit</th></tr>
<tr><td>A</td><td>90-100</td><td rowspan="5">Pass</td><td>Excellent – a high level of achievement of the intended learning outcomes in the course, indicating the learner’s full readiness for further study and/or professional activity in the field.</td></tr>
<tr><td>B</td><td>83-89</td><td rowspan="4">Good – an average (maximally sufficient) level of achievement of the intended learning outcomes in the course and readiness for further study and/or professional activity in the field.

Satisfactory – the student has demonstrated a minimally sufficient level of achievement of the learning outcomes required for further study and/or professional activity in the field.</td></tr>
<tr><td>C</td><td>73-82</td></tr>
<tr><td>D</td><td>66-72</td></tr>
<tr><td>E</td><td>60-65</td></tr>
<tr><td>FX</td><td>40-59</td><td rowspan="2">Fail</td><td>Fail – several intended learning outcomes in the course have not been achieved. The level of acquired learning outcomes is insufficient for further study and/or professional activity in the field.</td></tr>
<tr><td>F</td><td>0-39</td><td>Fail – no learning outcomes have been achieved.</td></tr>
</table>

10. SELF-ASSESSMENT QUESTIONS ON LEARNING OUTCOMES

1. Give a definition of the software
2. Define the life cycle
3. Name the main processes and stages of the life cycle
4. What is the purpose of software development.
5. What types of projects are divided into according to the success of The Standish Group terminology
6. What are the main reasons for the failure and success of projects according to The Standish Group?
7. Define the levels of software requirements.
8. What levels of requirements do you know?
9. What are the stages of requirements development?
10. What actions are included in requirements management?
11. Name the risks in software development
12. What are the benefits of a high-quality requirements development process?
13. What are the characteristics of the requirements specification provisions?
14. What sources of requirements do you know?
15. Describe the requirements strategies?
16. What are the ways to represent requirements?

17. Describe the actors and use cases for requirements.
18. Name the main styles of describing use case specifications
19. What does the template for a full description of the use case by A. Coburn consist of?
20. Describe the sections of the RUP use case template
21. Specification of non-functional requirements
22. What UML models explain the system functionality?
23. Describe the use case diagram
24. What are the main components of an action diagram?
25. When is a state diagram used?
26. Which UML diagrams explain the internal structure of the system?
27. What product characteristics are considered software quality attributes?
28. What is meant by software requirements management?
29. What are the basic versions of the requirements?
30. What are the requirements management techniques?
31. Name the attributes of software requirements
32. Define the quality of the software
33. Define testing for software
34. What types of testing are there?
35. What types of tests do you know?
36. Describe combined testing methods
37. What does program verification mean?
38. Describe agile software development based on Agile.
39. What standards do you know that regulate the software development process?
40. Describe the international ISO standards

11. EDUCATIONAL AND METHODOLOGICAL SUPPORT

The educational process for the course *"*Software Requirement Analysis and Quality*"* is supported with all necessary instructional and methodological materials, which are available in the Modular Learning Environment MOODLE:

1. Course *"*Software Requirement Analysis and Quality*"*: Курс: Аналіз вимог та якість програмного забезпечення

Primary

1. Laplante, P. A., Kassab, M. Requirements Engineering for Software and Systems (4th ed.). Routledge, 2022. – 420 p.
2. Tavana, M., Pick, K. W. Managing Requirements Knowledge. Springer, 2022. – 300 p.
3. Wiegers, K. E., Beatty, J. Software Requirements Essentials: Core Practices for Success. Pearson Education, 2022. – 280 p.
4. O'Regan, G. Introduction to Software Quality (2nd ed.). Springer, 2022. – 440 p.
5. Dalpiaz, F., Spoletini, P. (eds.). Requirements Engineering: Foundation for Software Quality (REFSQ 2023 Proceedings). Springer, 2023. – 385 p.
6. Walkinshaw, N. Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer, 2023. – 350 p.
7. ISO/IEC/IEEE 29148:2022. Systems and Software Engineering — Life Cycle Processes — Requirements Engineering. IEEE Standards Association.

8. ISO/IEC/IEEE 25010:2023. Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models. IEEE Standards Association.

Supplementary

9. Suryn, W. Software Quality Engineering: A Practitioner's Approach (2nd ed.). Wiley, 2022. – 410 p.
10. Dalpiaz, F., Spoletini, P., Méndez Benavides, A. (eds.). Requirements Engineering: Foundation for Software Quality (REFSQ 2022 Proceedings). Springer, 2022. – 370 p.
11. O'Connor, R. V., El Emam, K. Software Process and Product Quality Assurance. *Springer, 2022. – 295 p.*
12. SWQD 2023. Software Quality: Higher Software Quality through Zero Waste Development. *Springer, 202*3. – 315 p.
13. Onyshko, O. H. Software Requirements Analysis and Quality Methodology in Engineering Education. *Khmelnytskyi National University Press*, 2023. – 210 p.

13. INFORMATION RESOURCES

1. Electronic Library of the University. [Electronic resource]. – Access: http://library.khmnu.edu.ua/
2. Institutional Repository of Khmelnytskyi National University. [Electronic resource]. – Access: http://elar.khmnu.edu.ua/jspui/?locale=uk
3. Modular Learning Environment. [Electronic resource]. – Access: https://msn.khmnu.edu.ua/