Software requirements engineering

COURSE SYLLABUS

Code and name of specialty	121 – Software Engineering	**Institute / faculty**	Faculty of Computer Science and Software Engineering
Program name	«Software Engineering»	**Department**	Software Engineering and Management Information Technologies
Type of program	**Educational and Professional**	**Language of instruction**	**Ukrainian/English**

LECTURER

Moskalenko Valentyna Volodymyrivna Valentyna.Moskalenko@khpi.edu.ua

Doctor of Technical Sciences, Associate Professor, Professor of SEMIT Department. Number of scientific and educational publications is more than 90. (https://publons.com/researcher/1588564/valentyna-moskalenko/; Web of Science Researcher ID R-9960-2018; https://scholar.google.com.ua/citations?user=eUidJHIAAAAJ&hl=ru; https://www.scopus.com/authid/detail.uri?authorId=36021571200; https://orcid.org/0000-0002-9994-5404).

Courses taught: "Probability Theory and Mathematical Statistics", "Fundamentals of Computer Science and Artificial Intelligence Methods", "Fundamentals of Information Systems and Technologies", "Software Requirements Engineering", "Fundamentals of Business Analysis", "Computational Intelligence Methods", "Methods of computational intelligence and intellectual analysis", "Machine learning", "Business systems analytics"

GENERAL DESCRIPTION OF THE COURSE

Summary	The course “Software requirements engineering” is a discipline in the cycle of special mandatory training in the specialty 121 – Software Engineering. It is taught in the fourth semester in the amount of 90 hours (3 ECTS credits), in particular: lectures - 32 hours, workshops - 16 hours, independent work – 42 hours. The course provides two content modules and two module tests. The discipline ends with the test. The subject of the discipline is the main aspects of software requirements management throughout the software development cycle. The aim of the course is to gain knowledge about the types of user requirements, features of their formation and methods of formulating requirements by business analysts, methods of analysis and certification of requirements for different types of software systems, skills in resolving conflicts between requirements of different types; gaining knowledge of software requirements management techniques; formation of skills of application of tools for development of user requirements; gaining knowledge of the potential risks associated with the requirements.
Course objectives	The purpose of the discipline is theoretical and practical training of students in the field of software requirements engineering, students gain practical skills in analysis and modeling of the problem area, development of requirements specifications, as well as skills management skills throughout the software life cycle
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment. Final assessment – the test.
Term	4

Student workload (credits) / Type of course	3 / Mandatory	**Lectures (hours)**	32	**Workshops (hours)**	16	**Self-study (hours)**	42

Program competences	GC 05. Ability to learn and master modern knowledge. GC 06. Ability to search, process and analyze information from various sources. GC1. Ability to abstract thinking, analysis and synthesis.

	PC13. Ability to identify, classify and formulate software requirements. PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes. PC16. Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards. PC17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes. PC18. Ability to analyze, select and apply methods and tools to ensure information security (including cybersecurity). PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development. PC24. Ability to carry out the system integration process, apply change management standards and procedures to maintain the integrity, overall functionality and reliability of the software. PC25. Ability to reasonably select and master software development and maintenance tools. PC26. Ability to algorithmic and logical thinking.

Learning outcomes	**Teaching and learning methods**	**Forms of assessment (continuous assessment CAS, final assessment FAS)**
PLO 1. Apply knowledge of the fundamental forms and laws of abstract-logical thinking, the basics of the methodology of scientific knowledge, forms and methods of extraction, analysis, processing, and synthesis of information in the subject area of computer science. PLO 3. Use knowledge of the laws of random phenomena, their properties and operations with them, models of random processes, and modern software environments to solve problems of statistical data processing and construction of predictive models. PLO 9. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements PLO10. Conduct a pre-project survey of the subject area, systematic analysis of the design object. PLO11. Choose source data for design, guided by formal methods of describing requirements and modelling. PLO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software. PLO19. Know and be able to apply methods of software verification and validation. PLO23. Be able to document and present the results of software development	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester test, according to the schedule of the educational process (FAS)

ASSESSMENT AND GRADING

Ranges of points corresponding to grades

Total score (points) for all types of learning activities	**ECTS grading scale**	**The national grading scale**
90-100	A	excellent
82-89	B	good
74-81	C	
64-73	D	satisfactory
60-63	E	
35-59	FX	Unsatisfactory (with the exam retake option)

Allocation of grade points

100% Final assessment as a result of Final test (10%) and Continuous assessment (90%).
10% Final test
90% Continuous assessment:
Test №1 (5%)
Test №2 (5%)
Laboratory works (80%)
Laboratory work №1 (20%)
Laboratory work №2 (20%)

	0-34	F	Unsatisfactory (with mandatory repetition of the course)		Laboratory work №3 (20%) Laboratory work №4 (20%)

Course policy	Students must attend all classes according to the study schedule and adhere to the norms of academic ethics. To study the course, students need to have their personal computer and (or) use computers of the computer center at the department. Students must work with compulsory and recommended reading, including Internet resources. Students must complete and submit all laboratory works during the semester in which the course is taught, before the examination session. The final assessment is not carried out without the personal presence of students.

COURSE STRUCTURE AND CONTENT

Lecture 1	Basic software engineering requirements			**Self-study**	Areas of knowledge for SWEBOK
Lecture 2	Defining requirements as a stage of software development				Software life cycle
Lecture 3	Classification of requirements. Levels of requirements according to K. Wiggers; classification within the SWEBOK concept; RUP classification (FURPS +), etc.				
Lecture 4	The process of forming requirements				
Lecture 5	Problems of requirements formation and approaches to their solution				
Lecture 6	Traditional requirements detection methods and requirements detection technologies	**Laboratory work 1**	Identification of business requirements for software (Formation of Business Requirements Specification)		Brainstorming and other methods of generating ideas
Lecture 7	Modern methods of identifying requirements				Types of software prototyping
Lecture 8	Requirements analysis process	**Laboratory work 2**	Formation of functional and non-functional requirements for software		
Lecture 9	Advanced requirements analysis.				Business process approach to enterprise management
Lecture 10	Using modeling methods to analyze requirements.	**Laboratory work 3**	Forming Software Requirements Specification		Development of UML diagrams
Lecture 11					SADT modeling methodology and IDEF standards
Lecture 12	Software requirements management	**Laboratory work 4**	Software requirements change management		
Lecture 13	Software requirements change management				

Lecture 14	Software quality attributes				
Lecture 15	Software requirements testing				
Lecture 16	Features of requirements management for different models of software development				Features of software development models

RECOMMENDED READING

Compulsory

1 A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide). International Institute of Business Analysis, Toronto, Ontario, Canada. Version 3.0 published 2015.

2 Guide to Software Engineering Body of Knowledge (SWEBOK). IEEE Computer Society, 2004

3 K.E. Wiegers, Software Requirements, 3nd ed., Microsoft Press, 2013

4 Грицюк Ю. І. Аналіз вимог до програмного забезпечення. Навчальний посібник. 2018. – 456 с.

5 Klaus Pohl Requirements Engineering Fundamentals, 2nd Edition: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant. Rocky Nook, Inc. 2016 p.184.

6 Berenbach, B., Paulish, D., Katzmeier, J., & Rudorfer, A. (2009). Software & Systems Requirements Engineering: In Practice. New York: McGraw-Hill Professional.

7 Лавріщева К.М. Програмна інженерія.–К.– 2008.–319 с.

8 IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998.

9 Dick, J., Hull, E., & Jackson, K. (2017). Requirements Engineering. (4rd ed.) Springer. https://doi.org/10.1007/978-3-319-61073-3

10 Avramenko AS, Avramenko VS, Kosenyuk GV Software testing. Tutorial. - Cherkasy: Bohdan Khmelnytsky National University, 2017. - 284 p.

Додаткова

1. Klaus Pohl Requirements Engineering: Fundamentals, Principles, and Techniques. pringer Publishing Company, Incorporated, 2010. - P. 813

2. Leffingwell D., Widrig D. Managing software requirements: a unified approach Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©20001.

3. Cohn Mike. User Stories Applied. For Agile Software Development . Addison-Wesley; Pearson Education, 2004. — 291 p.

4. Rumpe, Bernhard Modeling with UML. Language, Concepts, Methods Springer. - 2016. https://nibmehub.com/opac-service/pdf/read/Modeling%20with%20UML_%20Language-%20Concepts-%20Methods.pdf

5. Change Management Scenario // https://oberemokii.com/uk/additional-scenarios/managing-the-content-and-timing/the-scenario-of-change-management

6. Training resource for software testing. Features of software requirements. Testing methods. Testing phases. Equivalence classes.//https://qalearning.com.ua/theory/lectures/material/requirements-testing-methods-equivalence/

7. Software Requirements (3rd Edition): Karl Weigers and Joy Beatty. Microsoft Press, 2013.

8. Mall R. (2014) Fundamentals Of Software Engineering, 4Th Ed., Phi, 2014

ACADEMIC INTEGRITY

Students are expected to adhere to the Code of Ethics of Academic Relations and Integrity of NTU “KhPI”.

The content of this syllabus is consistent with the course program.