

SOFTWARE QUALITY, TESTING AND SUPPORT

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

Name, E-mail	Sergey Orekhov, Serhii.Oriekhov@khpi.edu.ua
---------------------	---



Ph.D., Associate Professor of the Department of Software Engineering and Management Information Technologies, NTU «KhPI». Prepared and published more than 30 research papers, 5 articles in publications indexed in Scopus) (ORCID: <https://orcid.org/0000-0002-5040-5861>).

Leading lecturer of courses: *Internet of Things Fundamentals, Information Retrieving and Semantic Web (in Ukrainian and English)*

GENERAL DESCRIPTION OF THE COURSE

Summary	The course "Software quality, testing and support" is a course in the cycle of professional compulsory training of the specialty 121 "Software Engineering". It is taught in the six semester in the amount of 90 hours (5 ECTS credits), in particular: lectures – 32 hours, laboratory classes – 32 hours, self-study work – 26 hours. The course includes two modules and one modular test. The study of the discipline ends with the test.
Course objectives	Teaching students the methodology of analysis and evaluation of software quality, which allows at the stage of software development (software) to solve the following main tasks: choose a model of software quality at the level of IT company, IT product or IT processes; prepare a test strategy and test plan implemented in the test automation system; choose a specific set of testing tools and support to implement a testing strategy or plan.
Types of classes and control	Lectures, workshops, self-study work (and individual calculation work). Final assessment – test.
Term	6

Student workload (credits) / Type of course	3 / Mandatory	Lectures (hours)	32	Workshops (hours)	32	Self-study (hours)	90
--	---------------	-------------------------	----	--------------------------	----	---------------------------	----

Program competences	<p>GC 05. Ability to learn and master modern knowledge.</p> <p>GC 06. Ability to search, process and analyze information from various sources.</p> <p>PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes.</p> <p>PC16. Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards.</p> <p>PC17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes.</p> <p>PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing.</p> <p>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.</p>
----------------------------	---

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)
PO03. Know the basic processes, phases and iterations of the software life cycle.	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)
PO09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)
PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)
PO19. Know and be able to apply methods of software verification and validation.	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)
PO20. Know the approaches to evaluating and ensuring the quality of software	Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning	Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)

PO23. Be able to document and present the results of software development.

Interactive lectures with presentations, discussions, workshops, teamwork, case method, feedback method from students, problem learning

Written individual assignments for workshops (CAS), assessment of knowledge in workshops (CAS), express - survey(CAS), online tests (CAS), final / semester control in the form of a semester test, in accordance with the schedule of the educational process (FAS)

ASSESSMENT AND GRADING

Ranges of poin corresponding grades	Total score (points) all types of learnir activities	ECTS grading scale	The national grading scale	Allocation of grade points
	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 retak option)	
	0-34	F	Unsatisfactory (with mandatory repetition of the course)	

100% Final assessment as a result of Final test (30%) and Continuous assessment (70%).
30% Final test
70% Continuous assessment:
 Workshop (40%)
 One module test (30%)

Course policy Students are required to attend classes as scheduled and comply with ethical conduct. If absent, students will need to complete all tasks to compensate for the missed classes. Participation in workshops requires preliminary preparation and advance processing of all the necessary materials for productive discussions during the session. Written assignments must be submitted on time.

COURSE STRUCTURE AND CONTENT

COURSE STRUCTURE AND CONTENT					
Topic 1	The concept of software quality and reliability	Workshop 1	Create a test software system for further experimentation with testing in Java, C #, Javascript, or PHP.	Individual work	Study of software reliability models and analysis of the possibility of their implementation.
Topic 2	Software quality models at the level of the company's IT, process and IT product	Workshop 2	Preparation of documentation for creating a test plan in the form of software requirements in UML and using the IDEF methodology		Overview of existing IT product quality models
Topic 3	Software quality characteristics	Workshop 3	Calculations of values of typical quality metrics on the basis of the developed software artifacts		Analysis of software quality metrics according to the SWEBOK model
Topic 4	The place of verification and testing processes in the software life cycle	Workshop 4	Use of metric theory of programs for conducting test experiments		Study of existing methods of testing "white box"
Topic 5	Testing criteria (structural, functional, mutational, random and others)	Workshop 5	Application of structural testing criteria		Exploring existing black box testing methods
Topic 6	Classification of	Workshop 6	Features of application of testing criteria within		Analysis of features and implementation of

	software errors		OOP paradigm		an experiment on testing a web project online
Topic 7	Test plans. Structure and purpose. Construction algorithm	Workshop 7	Use components to automate testing in Java, C #, Javascript, or PHP programming environments		Review of testing automation systems
Topic 8	Software verification and testing support systems	Workshop 8	Creating test plans. Control of elimination of software errors by means of the special software.		Study of elimination control systems errors online

RECOMMENDED READING

Compulsory	<p>1. Mili, A., Tchier, F. (2015). Software Testing Concepts and Operations. John Wiley & Sons, Inc.</p> <p>2. Singh, Y. (2012). SOFTWARE TESTING. Cambridge University Press.</p> <p>3. P. Ammann, J. Offutt. (2008). INTRODUCTION TO SOFTWARE TESTING. Cambridge University Press.</p> <p>4. D. Graham, E. Veenendaal, I. Evans, R. Black. (2018). Foundations of software testing. istqb certification. Thomson.</p> <p>5. M. Pezzè, M. Young. (2008). Software Testing and Analysis: Process, Principles, and Techniques.</p> <p>6. K. NAIK, P. Tripathy. (2008). SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice. John Wiley & Sons, Inc.</p> <p>7. Hutcheson, M. (2003). Software Testing Fundamentals. Methods and Metrics. Wiley Publishing Inc.</p> <p>8 Standard for Software Verification and Validation Plans.(1986). (ANSI / IEEE standard 1012-1986).</p>	Recommended	<p>9. Куликов, С. (2021). Тестирование программного обеспечения. Базовый курс. EPAM Systems.</p> <p>10. Введення в програмну інженерію і управління життєвим циклом програмного забезпечення Guide to Software Engineering Base of Knowledge (SWEBOOK): Пер. з англ. С.Орлик: [Електронний ресурс]. Retrieved from: sorlik.blogspot.com/.</p> <p>11. Майерс, Г. (1980). Надежность программного обеспечения. М.: МИР, 1980.</p> <p>12. Майерс, Г. (1982). Искусство тестирования программ. Москва: Финансы и статистика.</p> <p>13. Бек, К. (2003). Экстремальное программирование: разработка через тестирование. – Санкт-Петербург: Питер.</p> <p>14. Бейзер, Б. (2004). Тестирование черного ящика. Технологии функционального тестирования ПО и систем. Санкт-Петербург.</p> <p>15. McCabe, T. J. (1976). A Complexity Measure // IEEE Transactions on Software Engineering, 2(4), 308–320.</p> <p>16. Канер, С. (2001). Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес приложений. Киев: Диасофт.</p> <p>17. Тамре, Л. (2003). Введение в тестирование программного обеспечения. Москва: Вильямс.</p>
-------------------	--	--------------------	---

Academic integrity

Graduate 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.