University of Gdańsk

Subject card

Subject name and code	Software Testing, PG_00178493		
Field of study	Informatics and Econometrics		
Date of commencement of studies	October 2026	Academic year of realisation of subject	2028/2029
Education level	Bachelor's studies	Subject group	Obligatory subject group in the field of study Optional subject group Subject group related to scientific research in the field of study
Mode of study	part-time studies	Mode of delivery	at the university
Year of study	3	Language of instruction	Polish
Semester of study	5	ECTS credits	5.0
Learning profile	academic	Assessment form	credit
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr Dariusz Kralewski	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	8.0	0.0	24.0	0.0	0.0	32

E-learning hours included: 0.0

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	32	2.0	91.0	125

Subject objectives

- Familiarisation with application testing techniques, tools and processes
- Practical implementation of tests
- Ability to assess the selection of testing tools/techniques for specific needs and their appropriate application
- Practical learning of the test documentation creation process

Learning outcomes

Course outcome	Subject outcome	Method of verification
[IiEL3_U12] The student can design and implement IT systems to enhance business operations and effectively utilize modern ICT technologies for management and business communication.	Student: - is able to identify the objectives and needs of testing for a selected application - is able to conduct an exploratory testing session - is able to conduct a usability testing session - is able to create test documentation - plans elements of the testing process - analyses individual test cases in terms of selecting appropriate actions in the area of application testing	[SU5] implementation of a problem task [SU6] demonstration of practical skills
[IiEL3_W06] To an advanced degree, the student knows and understands the processes and methods of creating, developing, and providing appropriate conditions for using informatics or statistics tools, particularly those that improve human and organizational functioning.	Student: - distinguishes between different levels of testing - characterises different types of testing - learns about good practices in application testing.	[SW4] test/exam - oral or written

Subject contents	Lecture topics 1. Fundamentals of testing • Why testing is necessary • What is testing? • General principles of testing • Basic testing process • Psychology of testing 2. Testing in the software life cycle • Software development models • Test levels • Types of tests • Maintenance testing 3. Static testing techniques • Static techniques and the testing process • Review process • Static analysis using tools 4. Test design techniques • Test development process • Categories of test design techniques • Specification-based or black-box techniques • Structure-based or white-box techniques • Experience-based techniques • Selection of testing techniques 5. Test management • Test organisation • Test planning and estimation • Test progress monitoring and supervision • Configuration management • Risk and testing • Incident management 6. Tool-assisted testing • Types of testing tools • Effective use of tools, potential benefits and risks • Implementing tools in an organisation Issues covered in exercises / seminars / laboratories 1. Unit tests • Test-driven development • Manual tests • Testing costs • Test pyramid • Unit test frameworks • NUnit in Visual Studio • Characteristics of good unit tests • Test naming and organisation • Basic unit testing techniques • Creating reliable tests • Testing methods that return a value • Testing non-returning methods • Testing methods that return an exception • Testing methods that trigger events • Testing private methods 2. Integration tests • Dependency Injection • Ways to loosen code • DI frameworks • Dummies • Mock frameworks 3. Test process automation. • Selenium • Cypress 4. Creating test documentation. 5. Conducting exploratory testing sessions. 6. Conducting usability testing sessions. 7. Familiarisation with the work of a software tester in the following contexts: agile teams, security testing, automated test design.
Prerequisites and co-requisites	Basic knowledge of programming and computer software
Assessment methods and criteria	

Subject passing criteria	Passing threshold	Percentage of the final grade
Examination in the form of a test	50.0%	40.0%
Completion of a final project – independent solution of a practical problem in a given field (e.g. business) using the software and methods learned during the course.	50.0%	60.0%

Recommended reading	Basic literature	A.1. used during classes 1. Wrycza S., Maślankowski J. (red.), Informatyka ekonomiczna. Teoria i zastosowania, PWN Warszawa 2019 2. Smilgin R., Praktyka testowania, Zeszyt ćwiczeń, Wydawnictwo Naukowe PWN, 2020 3. Roman A., Testowanie i jakość oprogramowania. Modele, techniki, narzędzia. Wydawnictwo Naukowe PWN, 2015 4. Rafał Pawlak, Testowanie oprogramowania. Podręcznik dla początkujących, Helion, 2014 A.2. studied independently by the student 1. Renu Rajani, Testowanie kodu w praktyce, Helion, 2018
	Supplementary literature	1. Roman A., Stapp L., Certyfikowany tester ISTQB. Poziom podstawowy, Helion, 2020 2. Axelrod A., Automatyzacja testów, Wydawnictwo Naukowe PWN, 2020. 3. Zmitrowicz K., Jakość projektów informatycznych. Rozwój i testowanie oprogramowania. Helion, 2015
	eResources addresses	
Example issues/ example questions/ tasks being completed		
Work placement	Not applicable	

Document generated electronically. Does not require a seal or signature.