University of Gdańsk

Subject card

Subject name and code	High Quality Software Development, PG_00209937		
Field of study	Informatics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2025/2026
Education level	Master's studies	Subject group	Obligatory subject group in the field of study Optional subject group
Mode of study	part-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	4	ECTS credits	6.0
Learning profile	academic	Assessment form	credit
Conducting unit	Division of Software Engineering -> Institute of Informatics -> Faculty of Mathematics, Physics and Informatics -> Rector		
Name and surname of lecturer (lecturers)	Subject supervisor	dr Tomasz Borzyszkowski	
	Teachers	dr Tomasz Borzyszkowski	

Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	0.0	0.0	0.0	0.0	20.0	20
	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	20	0.0	80.0	100

Subject objectives	Guidance and support for students in the process of preparing their Master's thesis in the field of software engineering. The aim of the seminar is to develop students' ability to formulate research problems independently, conduct a critical review of the literature, and apply advanced methods and tools for ensuring high-quality software (including in the areas of architecture, reliability and testing). The course also prepares students for the public presentation of their own work and the defence of the theses contained in their dissertation.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFMU2_K04] understands and appreciates the importance of intellectual honesty in his own and others' actions; acts ethically	The student demonstrates intellectual integrity and strictly adheres to the principles of academic ethics when preparing their dissertation, as evidenced by the accurate citation of sources, the correct attribution of authorship for concepts, algorithms and code snippets used, and the objective, unbiased reporting of their own research findings.	[SK2] presentation/project/paper/ report [SK3] text preparation/written work
[INFMU2_K01] knows the limits of his own knowledge and understands the need for further learning	Students critically assess their level of knowledge and skills in software engineering, recognising that rapid technological developments (e.g. new architectural patterns, automation tools, cybersecurity) and the growing complexity of IT systems necessitate continuous, self-directed professional development.	[SK8] observation of student's independent or team work
[INFMU2_K03] is able and ready to formulate opinions on basic IT issues	The student is able to formulate critical and well-founded opinions on the current state of knowledge, market standards and the quality of existing IT solutions, as demonstrated by their own conclusions, summaries and comparative analyses included in the text of their thesis.	[SK2] presentation/project/paper/ report
[INFMU2_U08] is able to acquire information from professional literature, databases, the Internet and other sources, integrate them, assess their reliability, make interpretations and draw conclusions and formulate opinions	The student is able to conduct a critical review of Polish- and English-language academic literature and technical documentation on methodologies, standards and tools for ensuring high-quality software, and then integrate this information in order to define the research area for their own Master’s thesis.	[SU2] presentation/project/paper/ report
[INFMU2_U10] is able to determine the directions of further learning and implement the process of self-education	The student is able to identify gaps in their technical and methodological skills required to address the research problem at hand, and then plan and carry out the process of independently mastering the new technologies, tools or methodologies used in their dissertation.	[SU2] presentation/project/paper/ report [SU3] text preparation/written work
[INFMU2_W07] has an in-depth knowledge of the current legislation on the activities of a computer scientist (teaching, scientific and professional activities) and intellectual property	The student demonstrates in-depth knowledge of intellectual property protection and copyright law in academic work, with particular emphasis on the principles of academic integrity, correct citation of sources, avoidance of plagiarism, and the economic and moral rights relating to the software developed and the text of the Master’s thesis.	[SW2] presentation/project/paper/ report [SW3] text preparation/written work
[INFMU2_W01] has in-depth knowledge of the branches of mathematics necessary for the study of computer science; has a good understanding of the role and importance of the construction of mathematical reasoning	The student has in-depth knowledge of the mathematical foundations of software engineering (including formal logic in test design, statistical reliability analysis, and graph theory-based complexity metrics) and understands their significance in the process of objectively verifying and measuring the quality of IT systems.	[SW2] presentation/project/paper/ report

Course outcome	Subject outcome	Method of verification
[INFMU2_U09] is able to present the results of research in the form of an independently prepared dissertation (paper) containing a description and justification of the purpose of the work, the methodology adopted, the results and their significance in comparison with other similar studies	The student is able to independently write a complete Master's thesis on high-quality software development, in which they describe, in a logical and linguistically correct manner, the research objective, the chosen engineering methodology, and the results obtained (e.g. code metrics, performance test results) and subject them to critical discussion against the background of existing solutions found in the literature or on the market.	[SU2] presentation/project/paper/ report
[INFMU2_U02] is able to formulate questions with precision, serving to deepen his/her own reasoning on a given topic or to find missing elements of reasoning	The student is able to identify gaps in their knowledge or in existing technological solutions, and then precisely formulate a research problem and the main and specific research questions for their own Master's thesis in the field of high-quality software development.	[SU2] presentation/project/paper/ report
[INFMU2_W02] has in-depth knowledge of formal languages, models of computation, and issues of computational complexity; is familiar with the formal apparatus for formulating and studying the properties of computer objects	The student demonstrates in-depth knowledge of computational complexity theory and is able to apply this conceptual framework to the theoretical and practical analysis of the performance, scalability and optimisation of algorithms (e.g. Big O notation) used in their own research or engineering project.	[SW2] presentation/project/paper/ report

Subject contents

1. Issues related to software architecture:
   - refactoring existing code
   - evolutionary architecture and the use of metrics
   - the use of AI to improve code quality and analysis
2. Software development technologies:
   - compilation of programming languages
   - design of hybrid programmes
   - algorithmic support (including AI) for software development
3. Quality assurance and reliability:

- specifying, verifying and testing smart contracts (in blockchain technology) and conventional programmes (object-oriented, functional and hybrid)
   - proving code correctness
   - testing software vulnerability to various types of attacks

Prerequisites and co-requisites

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
Assessment of the Master's thesis	51.0%	80.0%
Class presentations	51.0%	20.0%

Recommended reading

Basic literature	1. Fowler M., Refaktoryzacja. Ulepszanie struktury istniejącego kodu. Wydanie II, Wydawnictwo Helion, Gliwice 2019. 2. Feathers M. C., Praca z zastanym kodem. Najlepsze techniki, Wydawnictwo Helion, Gliwice 2007 (lub nowsze dodruki). 3. Ford N., Parsons R., Kua P., Budowanie architektur ewolucyjnych. Wsparcie dla ciągłych zmian, Wydawnictwo Helion, Gliwice 2018. 4. Khorikov V., Testy jednostkowe. Zasady, praktyki i wzorce, Wydawnictwo Helion, Gliwice 2021. 5. Antonopoulos A. M., Wood G., Mastering Ethereum. Tworzenie inteligentnych kontraktów i aplikacji rozproszonych, Wydawnictwo Helion, Gliwice 2019.
Supplementary literature	1. Parr T., Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages, The Pragmatic Bookshelf, Raleigh 2010. 2. Janca T., Alice and Bob Learn Application Security, John Wiley & Sons, Indianapolis 2020. 3. Huth M., Ryan M., Logic in Computer Science: Modelling and Reasoning about Systems (2nd Edition), Cambridge University Press, Cambridge 2004. 4. Pierce B. C. i in., Software Foundations, publikacja elektroniczna (Materiały z University of Pennsylvania), udostępniana na licencji open-source i aktualizowana na bieżąco.

	eResources addresses	
Example issues/ example questions/ tasks being completed		
Work placement	Not applicable	

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