GDAŃSK UNIVERSITY
OF TECHNOLOGY

Subject card

<table>
<tr><td>Subject name and code</td><td colspan="7">Software Engineering, PG_00063885</td></tr>
<tr><td>Field of study</td><td colspan="7">Informatics</td></tr>
<tr><td>Date of commencement of studies</td><td colspan="2">October 2025</td><td colspan="3">Academic year of realisation of subject</td><td colspan="2">2026/2027</td></tr>
<tr><td>Education level</td><td colspan="2">first-cycle studies</td><td colspan="3">Subject group</td><td colspan="2">Obligatory subject group in the field of study
Subject group related to scientific research in the field of study</td></tr>
<tr><td>Mode of study</td><td colspan="2">Full-time studies</td><td colspan="3">Mode of delivery</td><td colspan="2">at the university</td></tr>
<tr><td>Year of study</td><td colspan="2">2</td><td colspan="3">Language of instruction</td><td colspan="2">Polish</td></tr>
<tr><td>Semester of study</td><td colspan="2">4</td><td colspan="3">ECTS credits</td><td colspan="2">3.0</td></tr>
<tr><td>Learning profile</td><td colspan="2">general academic profile</td><td colspan="3">Assessment form</td><td colspan="2">exam</td></tr>
<tr><td>Conducting unit</td><td colspan="7">Department Of Software Engineering -> Faculty Of Electronics Telecommunications And Informatics -> Wydziały Politechniki Gdańskiej</td></tr>
<tr><td rowspan="2">Name and surname of lecturer (lecturers)</td><td colspan="2">Subject supervisor</td><td colspan="5">dr inż. Aleksander Jarzębowicz</td></tr>
<tr><td colspan="2">Teachers</td><td colspan="5">dr inż. Aleksander Jarzębowicz</td></tr>
<tr><td rowspan="3">Lesson types and methods of instruction</td><td>Lesson type</td><td>Lecture</td><td>Tutorial</td><td>Laboratory</td><td>Project</td><td>Seminar</td><td>SUM</td></tr>
<tr><td>Number of study hours</td><td>30.0</td><td>0.0</td><td>30.0</td><td>0.0</td><td>0.0</td><td>60</td></tr>
<tr><td colspan="7">E-learning hours included: 0.0</td></tr>
<tr><td rowspan="2">Learning activity and number of study hours</td><td>Learning activity</td><td colspan="2">Participation in didactic classes included in study plan</td><td colspan="2">Participation in consultation hours</td><td>Self-study</td><td>SUM</td></tr>
<tr><td>Number of study hours</td><td colspan="2">60</td><td colspan="2">2.0</td><td>13.0</td><td>75</td></tr>
<tr><td>Subject objectives</td><td colspan="7">The course is focused on introducing to students the aspects of industrial software development: large systems, compliant to requirements of a specific customer, supporting business goals, providing a required level of quality characteristics, produced and maintained by large developers teams.</td></tr>
</table>

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_W44] knows and understands, to an advanced extent, architecture, design principles and methods of hardware and software support for local and distributed information systems, including computing systems, databases, computer networks and information applications, as well as the principles of human-computer interaction, the operation and evaluation criteria of data processing, storage and transfer methods, including computational algorithms, artificial intelligence and data mining as well as standards and methods of IT systems administration, monitoring of processes and robustness to undesirable phenomena and activities	The student understands what IT system's architecture is, knows what issues are essential during architectural design.	[SW1] Assessment of factual knowledge
[K6_U03] can design, according to required specifications, and make a simple device, facility, system or carry out a process, specific to the field of study, using suitable methods, techniques, tools and materials, following engineering standards and norms, applying technologies specific to the field of study and experience gained in the professional engineering environment	The student develops analytical and design models of IT system using CASE (Computer Aided Software Engineering) software supporting tools.	[SU1] Assessment of task fulfilment
[K6_U09] can carry out a critical analysis of the functioning of existing technical solutions and assess these solutions, as well as apply experience related to the maintenance of technical systems, devices and facilities typical for the field of studies, gained in the professional engineering environment	The student develops “Vision of IT system” document that includes a critical analysis of the present state of the customer organization as well as basic requirements and restrictions of the planned IT system.	[SU1] Assessment of task fulfilment

Subject contents

1. Introduction
2. Scope and subject of software engineering. Essential motivations and concepts.
3. Planning and defining scope of software project. Rich Picture.
4. Areas of software engineering - an overview
5. Conceptual modelling. Languages for modelling and specification.
6. Use cases
7. Object-oriented analysis using UML
8. Modelling of logical system structure: class diagrams
9. Modelling of system structure: other structural diagrams
10. Modelling system dynamics: sequence and communication diagrams
11. Modelling system dynamics: representing object"s state
12. System design: system architecture
13. System design: high-level design
14. System design: class design (low level)
15. Foundations of software quality. Metrics of object-oriented design.
16. Software reuse
17. Classical design patterns
18. Other patterns (Internet Applications patterns, analysis patterns, architectural patterns, management patterns)
19. Risk and social responsibility related to IT systems
20. Requirements engineering: requirements determination
21. Requirements engineering: requirements specification
22. User interface design: motivations, terms, techniques
23. User interface design: Nielsen"s heuristics and examples
24. Software testing: terms, place in software development process
25. Software testing: techniques (black/white box), levels of testing, managing tests
26. Software reviews and inspections
27. Software deployment
28. Software usage and maintenance
29. Configuration management and software evolution
30. Classical (waterfall) software lifecycle model
31. Non-classical software lifecycles and development processes
32. Adjusting development process to particular software project context
33. Outline of software project management
34. Software development and management methodologies
35. Properties of plan-driven and agile development
36. CASE tools
37. Other tools supporting software engineering

<table>
<tr><td>Prerequisites
and co-requisites</td><td colspan="2">Presence during laboratory courses is mandatory. Delivery of all laboratory exercises and positive verification by tutor is required to pass the lab. Delays in delivering exercises affects the assessments. Only students who pass the lab are entitled to write the exam.</td></tr>
<tr><td>Assessment methods
and criteria</td><td colspan="2"><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>Lab (assignments & tests)</td><td>50.0%</td><td>50.0%</td></tr>
<tr><td>Written exam</td><td>50.0%</td><td>50.0%</td></tr>
</table></td></tr>
<tr><td rowspan="3">Recommended reading</td><td>Basic literature</td><td>1. Pressman R., Software Engineering: a Practitioners Approach, 8th edition, McGraw-Hill, 2014
2. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010
3. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007
4. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, 2005
5. Fowler M., UML distilled, 3rd edition, Addison-Wesley, 2003</td></tr>
<tr><td>Supplementary literature</td><td>No requirements</td></tr>
<tr><td>eResources addresses</td><td>Adresy na platformie eNauczanie:</td></tr>
<tr><td>Example issues/
example questions/
tasks being completed</td><td colspan="2"></td></tr>
<tr><td>Work placement</td><td colspan="2">Not applicable</td></tr>
</table>

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