SYLLABUS

Name: ***Software Engineering (InfAAu>SI5SE19)***
Name in Polish:
Name in English: ***Software Engineering***

Information on course:

Course offered by department: Faculty of Automatic Control, Electronics and Computer Science
Course for department: Silesian University of Technology

Default type of course examination report:

ZAL

Language:

English

Course homepage:

https://platforma2.polsl.pl/rau2/course/view.php?id=1038

Short description:

The subject has the form of a project. Its essential element is the development of an information system specification. The specification consists of: initial specification, requirements specification, use case model, scenarios, class model, architecture sketch, and behaviour model. The primary learning objective of the subject is to learn how to model an IS using the UML language, as well as to learn how to use the CASE tool.

An additional component of the project is the implementation of the previously designed application. The purpose of this stage is to verify the model and practice the skills of methodical testing of the application.

Description:

ECTS : 3
total hours: 80 (contact 30h, own work 50h)
project - 30h,
own work - 50h
including:
development of artifacts (documents that are part of the project): 20h
program development: 30h.

Classes are in the form of a project of 2 class hours per week.
Students receive 3 ECTS points for completing the following project tasks:
1 - preparation of a preliminary system specification. (6 class hours)
2 - development of a use case model. (6 class hours)
3 - developing a class model (+prototype #1) (6 class hours)
4 - preparing a behaviour model (+prototype #2) (6 class hours)
5 - conducting implementation and testing (6 class hours).

The specific learning objectives are:
1. practising the ability to obtain information and formulate requirements;
2. learning to use notations in practice, mainly the UML language;
3. learning to use CASE tools in the software development process;
4. to develop the habit of drawing up program specifications before starting implementation;
5. to familiarize students with the methodology of software design (in particular, to develop the ability to move between successive artefacts from verbal description, through requirements, etc. - to implementation);
6. practising the ability to implement programs based on documentation;
7. practising the skill of teamwork with design roles;

Bibliography:

• Perdita Stevens, “UML. Inżynieria oprogramowania”, Helion, 2021.
• Grady Booch, Ivar Jacobson, James Rumbaugh, “The Unified Modeling Language User Guide”, Addison Wesley, first edition 1999.
• Koszlajda Adam, “Zarządzanie projektami IT, przewodnik po metodykach”, HELION, Gliwice, 2010.
• Pichler Roman, “Zarządzanie projektami ze Scrumem”, edycja polska, HELION, Gliwice, 2010
• Kshirasagar Nike, Priyadarshi Tripathy, “Software testing and Quality Assurance”, Wiley, 2008, United States.
• Szyjewski Zdzisław, “Metodyki zarządzania projektami informatycznymi”, Wydawnictwo PLACET, Warszawa, 2004.
• Kruchten Philippe, “The rational unified process an introduction”, Parson Education Inc., Upper Saddle River, 2004.
• Sommerville I.: Software Engineering. Addison Wesley Longman, Harlow 2004.
• Alistair Cockburn, “Writing Effective Usecases”, Addison-Wesley, c. 2001.
• Larman C.: Applying UML and patterns: An introduction to object-oriented analysis and design and the Unified Process. Prentice Hall, Upper Saddle River 2001.
• UML Specifications, About the Unified Modeling Language Specification Version 2.5.1 (omg.org), up-to-date version, 2021.

Learning outcomes:

1 The student can solve software problems working in a team - K1_U29 (final evaluation of the project, partial grades of exercises 1-5).
2. The student has a working knowledge of software specification, modelling and design - K1A_W11, K1_U29 (final evaluation from the project, partial grades of exercises 1-4).
3. The student has a working knowledge of the use of UML notation (class diagrams, use case diagrams, component diagrams) K1A_W11 (final project grade, including partial grades of exercises 2-4).
4. The student knows modern CASE software, its functions and applications - K1A_U21 (final evaluation from the project, including partial grades of exercises 2-4).
5. The student can develop software requirements based on textual artefacts describing an example system - K1A_U21 (final grade of the project, including partial grades of exercise 1).
6. The student can use modern CASE software and apply it in practice at the stage of requirement formulation, use case modelling, structure and behaviour modelling - K1A_U21 (final grade from the project, including partial grades from exercises 2-4).
7. The student can verify and validate software - K1A_U15, K1A_U19 (final grade from the project, including partial grade from exercise 5).

Assessment methods and assessment criteria:

Students complete project topics (specification and implementation of an example system). Topics are assigned by the instructor.

Groups are divided into two sections (automatic enrollment in the group based on the password on the RES). Inside the section, students divide into teams. Each team carries out one topic. The project consists of five sections, each of which is graded on a scale of 2-5 in increments of 0.5. The final passing grade, is the arithmetic average of the partial grades rounded to two digits after the decimal point, with X.25, equal to X, and X.75, equal to X+1.

The syllabus is effective from the fall semester / academic year 2025/2026, and its content is not subject to change during the semester.

Course credits in various terms:

<without a specific program>

Type of credits	Number	First term	Last term
European Credit Transfer System (ECTS)	3	2020/2021-Z	

Informatics, full-time first degree engineering studies 7 sem. (InfAAu-SI7)

Type of credits	Number	First term	Last term
European Credit Transfer System (ECTS)	3	2020/2021-Z	