



SOFTWARE ENGINEERING COMPONENTS 2. SOFTWARE DESIGN. ANALYSIS OF SOFTWARE REQUIREMENTS

Syllabus

Requisites of the Course

Cycle of Higher Education	<i>First cycle of higher education (Bachelor's degree)</i>
Field of Study	<i>12 Information Technologies</i>
Speciality	<i>121 Software engineering</i>
Education Program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Type of Course	<i>Normative</i>
Mode of Studies	<i>full-time</i>
Year of studies, semester	<i>2 year (3 semester)</i>
ECTS workload	<i>4 credits (ECTS). Time allocation: 36 hours for lectures, 18 hours for practicals, 54 hours for self-study.</i>
Testing and assessment	<i>Exam</i>
Course Schedule	<i>According to rozklad.kpi.ua</i>
Language of Instruction	<i>English</i>
Course Instructors	<i>Senior lecturer, Olga Sulema, PhD olga.sulema@pzks.fpm.kpi.ua</i>
Access to the course	<i>Google classroom at https://classroom.google.com/u/0/c/NDA4NTY1MjAyMzMx</i>

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The study of the Software Engineering Components 2. Software Design. Analysis of Software Requirements course allows students acquire competencies necessary for solving practical problems related to the designing software and analyzing software requirements.

The purpose of studying the Software Engineering Components 2. Software Design. Analysis of Software Requirements course is to build capacity to design software, formulate and analyze software requirements.

The subject of the Software Engineering Components 2. Software Design. Analysis of Software Requirements course is the process of software design and requirements analysis.

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

The Software Engineering Components 2. Software Design. Analysis of Software Requirements course is a normative discipline and students do not need any specific initial knowledge for its study.

Theoretical knowledge and practical skills acquired in the Software Engineering Components 2. Software Design. Analysis of Software Requirements course provide the necessary background for studying other disciplines in Bachelor and Master programs of 121 Software Engineering speciality.

3. Content of the course

Topic 1. Introduction to requirements analysis

Topic 2. Software development life cycle

Topic 3. Types of software requirements

Topic 4. Requirement analysis techniques

4. Coursebooks and teaching resources

Main literature:

1. Maciaszek L. *Requirements analysis and system design*, 2007.
2. Aurum A., Wohlin C. *Engineering and managing software requirements*, 2005.
3. Hay D.C. *Requirements analysis: from business views to architecture*, 2003.
4. Wiegers K. *More about software requirements: thorny issues and practical advice*, 2005.
5. Murch R. *The Software Development Lifecycle. A Complete Guide*, 2012.
6. Shore J., Warden S. *The art of agile development*, 2021.

Educational content

5. Methodology

No	Type of a class	Materials for self-studying
<i>Topic 1. Introduction to requirements analysis</i>		
1.	<i>Lecture 1. Introduction to requirement analysis</i>	6, №1
2.	<i>Lecture 2. Requirement analysis process. Part 1</i>	6, №2
3.	<i>Lecture 3. Requirement analysis process. Part 2</i>	6, №3
4.	<i>Practical 1. Requirement analysis process</i>	6, №4, 29
<i>Topic 2. Software development life cycle</i>		
5.	<i>Lecture 4. Software development life cycle</i>	6, №5
6.	<i>Lecture 5. Software development life cycle models. Part 1</i>	6, №6
7.	<i>Practical 2. Waterfall and Iterative SDLC models</i>	6, №7, 31
8.	<i>Lecture 6. Software development life cycle models. Part 2</i>	6, №8
9.	<i>Practical 3. Spiral and V-shaped SDLC models</i>	6, №9, 31
10.	<i>Lecture 7. Software development life cycle models. Part 3</i>	6, №10
11.	<i>Practical 4. Agile SDLC model</i>	6, №11, 31
<i>Topic 3. Types of software requirements</i>		
12.	<i>Lecture 8. Functional requirements</i>	6, №12
13.	<i>Practical 5. Functional requirements in practice</i>	6, №13, 32
14.	<i>Lecture 9. Technical requirements</i>	6, №14
15.	<i>Lecture 10. Transitional requirements</i>	6, №15
16.	<i>Lecture 11. Operational requirements</i>	6, №16

17.	<i>Practical 6. Non-functional requirements</i>	6, №17, 32
<i>Topic 4. Requirement analysis techniques</i>		
18.	<i>Lecture 12. Introduction to requirement analysis techniques</i>	6, №18
19.	<i>Lecture 13. BPMN and UML</i>	6, №19
20.	<i>Practical 7. BPMN and UML in practice</i>	6, №20, 33
21.	<i>Lecture 14. Data flow diagram and RAD</i>	6, №21
22.	<i>Lecture 15. Gantt chart</i>	6, №22
23.	<i>Practical 8. Gantt chart in practice</i>	6, №23, 33
24.	<i>Lecture 16. IDEF and Gap analysis</i>	6, №24
25.	<i>Practical 9. IDEF and Gap analysis in practice</i>	6, №25, 33
26.	<i>Lecture 17. Real-life examples of requirement analysis</i>	6, №26
27.	<i>Midterm test</i>	6, №27, 30, 32

6. Self-study

No	Topic for self-studying	Hours	Literature
1.	<i>Preparation to a lecture 1</i>	1	1, 4
2.	<i>Preparation to a lecture 2</i>	1	1, 3
3.	<i>Preparation to a lecture 3</i>	1	1, 3
4.	<i>Preparation to a practical 1</i>	1,5	1, 3, 4
5.	<i>Preparation to a lecture 4</i>	1	5, 6
6.	<i>Preparation to a lecture 5</i>	1	5
7.	<i>Preparation to a practical 2</i>	1,5	5
8.	<i>Preparation to a lecture 6</i>	1	5
9.	<i>Preparation to a practical 3</i>	1,5	5
10.	<i>Preparation to a lecture 7</i>	1	5, 6
11.	<i>Preparation to a practical 4</i>	1,5	5, 6
12.	<i>Preparation to a lecture 8</i>	1	1, 2, 4
13.	<i>Preparation to a practical 5</i>	1,5	1, 2, 4
14.	<i>Preparation to a lecture 9</i>	1	2, 4
15.	<i>Preparation to a lecture 10</i>	1	2, 4
16.	<i>Preparation to a lecture 11</i>	1	1, 2, 4
17.	<i>Preparation to a practical 6</i>	1,5	1, 2, 4
18.	<i>Preparation to a lecture 12</i>	1	1, 4

19.	<i>Preparation to a lecture 13</i>	1	1, 4
20.	<i>Preparation to a practical 7</i>	1,5	1, 4
21.	<i>Preparation to a lecture 14</i>	1	4
22.	<i>Preparation to a lecture 15</i>	1	4
23.	<i>Preparation to a practical 8</i>	1,5	4
24.	<i>Preparation to a lecture 16</i>	1	1, 2
25.	<i>Preparation to a practical 9</i>	1,5	1, 2
26.	<i>Preparation to a lecture 17</i>	1	2
27.	<i>Preparation to a midterm test</i>	4	1, 4, 5, 6
28.	<i>Preparation to an exam</i>	6	1, 4, 5, 6
29.	<i>Requirement analysis process</i>	4	1
30.	<i>Software development life cycle</i>	4	5
31.	<i>Software development life cycle models</i>	4	5, 6
32.	<i>Software requirements</i>	4	2, 3
33.	<i>Software development techniques</i>	4	1, 2, 3

Policy and Assessment

7. Course policy

- *Attending lectures is mandatory.*
- *Attending seminars is mandatory.*
- *Rules of conduct: activity, taking part in discussions, respect to teacher and groupmates, muting cellphones.*
- *Compliance with the policy of academic integrity.*
- *Rules of fulfilling seminar tasks: assignment has to be completed according to the student's team number or the student's variant defined with a generator of pseudo-random numbers (hereafter – randomizer).*
- *Rules of assigning bonus points and penalty points:*

Bonus points are being assigned for:

- *taking part in discussion during lecture classes;*
- *answering other students' questions on lecture material;*
- *creative approach in team tasks.*

Max bonus points during the semester: 5 points.

Penalty points might be assigned because of:

- *plagiarism: -5 points for every attempt;*
- *turning homework in untimely: -0,5 points for every week after the deadline (max penalty points for one homework: -5 points).*

8. Monitoring and grading policy

In the first lecture, the students are being acquainted with the grading policy which is based on Regulations on the System of Learning Outcomes Assessment (https://document.kpi.ua/files/2020_1-273.pdf).

The student's rating in the course consists of points that they receive for homework (R_1), a midterm test (R_2) and an exam (R_3).

$$R_5 = R_1 + R_2 + R_3 = 40 + 20 + 40 = 100 \text{ points}$$

According to the university regulations on the monitoring of students' academic progress (https://kpi.ua/document_control), there are two assessment weeks, usually during 7th/8th and 14th/15th week of the semester, when students take the Progress and Module tests respectively, to check their progress against the criteria of the course assessment policy.

The table of compliance between overall points and the final grade:

Points	Grade
95-100	Excellent
85-94	Very good
75-84	Good
64-74	Satisfactory
60-64	Fair
Less than 60	Unsatisfactory
Course requirements are not met	Not Graded

9. Additional information about the course

The list of questions for an exam are adduced in Appendix 1.

Syllabus of the course

Is designed by PhD, senior lecturer, Olga Sulema

Adopted by Computer Systems Software Department (protocol № 12, 26 April 2023)

Approved by the Faculty Board of Methodology (protocol № 10, 26 May 2023)