

COURSE SYLLABUS AND COURSE REQUIREMENTS

ACADEMIC YEAR 2023/2024 SEMESTER II.

| | |
|------------------------------|----------------------------------|
| <i>Course title</i> | <i>Software Engineering</i> |
| <i>Course Code</i> | IVB064AN |
| <i>Hours/Week: le/pr/lab</i> | 2/0/0 |
| <i>Credits</i> | 4 |
| <i>Degree Programme</i> | Computer Science Engineering BSC |
| <i>Study Mode</i> | <i>full-time</i> |
| <i>Requirements</i> | Exam |
| <i>Teaching Period</i> | 4. semester, Spring |
| <i>Prerequisites</i> | Databases I |
| <i>Department(s)</i> | System and Software Technology |
| <i>Course Director</i> | Dr. Etelka Szendrői |
| <i>Teaching Staff</i> | <i>Dr. Etelka Szendrői</i> |

COURSE DESCRIPTION

The course introduces the engineering of complex software systems. The focus is on software engineering principles and the methods and tools that support the principles. Particular attention will be paid to object-oriented development techniques, UML modelling. Students learn methods used in large scale software development projects, emphasizing requirements analysis, design, implementation, and testing.

SYLLABUS

1. GOALS AND OBJECTIVES

Course goals:

- To provide an understanding of principal concepts, major methods, tools, and basic approaches in software engineering processes.
- To provide theoretical and practical level experience in UML modelling.
- To provide the ability to understand and analyse real life problem solving.

Learning Objectives:

Students who successfully complete this course will have a comprehensive overview of software engineering methods and technics. Furthermore, students will have hands on experiences in UML modelling.

2. COURSE CONTENT

TOPICS

LECTURE

1. *What is Software Engineering, software crisis*
2. *The software processes, software life cycle*
3. *Waterfall model, iterative model, spiral model*
3. *Business requirements, requirement analysis, requirement gathering*
4. *CASE tools. Architectural Design*
5. *Object oriented design, UML model language*
6. *Design patterns. Planning user interface*
7. *Software Testing, software maintenance.*
8. *Documenting requirements, effective verification and validation strategies*
9. *Software evolution*
10. *RAD development. Agile methods.*

DETAILED SYLLABUS AND COURSE SCHEDULE

LECTURE

| week | Topic | Compulsory reading; page number (from ... to ...) | Required tasks (assignments, tests, etc.) | Completion date, due date |
|------|--|--|---|------------------------------|
| 1. | Object of Software Engineering. Software crises. Basic principles of software engineering | [1] Chapter 1 and 2 https://www.youtube.com/watch?v=W3YJeoYgozw | ... | ... |
| 2. | Software engineering process. Software Lifecycle. Software engineering projects. Phases of software development. | [1] Chapter 2 | | |
| 3. | Software development methodologies. RUP | [1] Chapter 2 | | |
| 4. | Requirements gathering technics. Requirement specification. | [1] Chapter 4 | | |
| 5. | Object -oriented modelling. UML modelling language. Use Cases | [1] Chapter 5 - 7 | | |
| 6. | UML diagrams (Sequential, activity) | [1] Chapter 5 and 7 | | |
| 7. | Test 1 | | Test 1 | 17 th March |
| 8. | UML modelling, diagrams (State charts). Data modelling | [1] Chapter 5 and 7 | | |
| 9. | Academic HOLIDAY | | | |
| 10. | User interface design. Menu structure | [2] Chapter 11 | | |
| 11. | Validation and verification. Testing. Documentation. Deployment | [1] Chapter 6,8 | | |
| 12. | Test 2 | | Test 2 | 23 th April |
| 13. | RAD technics. Agile methods | [1] Chapter 3 | | |
| 14. | Retake test | | | |

4. ASSESSMENT AND EVALUATION

ATTENDANCE

In accordance with the Code of Studies and Examinations of the University of Pécs, Article 45 (2) and Annex 9. (Article 3) a student may be refused a grade or qualification in the given full-time course if the number of class absences exceeds 30% of the contact hours stipulated in the course description.

Method for monitoring attendance (e.g.: attendance sheet / online test/ register, etc.)

Attendance sheet.

ASSESSMENT

Course-unit with final examination

Mid-term assessments, performance evaluation and their weighting as a pre-requisite for taking the final exam

| Type | Assessment | Weighting as a proportion of the pre-requisite for taking the exam |
|-----------|------------|--|
| 1. TEST 1 | max. 100% | 40 % |
| 2. TEST 2 | max. 100% | 60 % |

Requirements for the end-of-semester signature

The weighted test results must be equal or greater than 40%. The formula: $0,4 * \text{TEST1\%} + 0,6 * \text{TEST2\%} \geq 40$

Re-takes for the end-of-semester signature (PTE TVSz 50§(2))

The specific regulations for grade betterment and re-take must be read and applied according to the general Code of Studies and Examinations. E.g.: all the tests and the records to be submitted can be repeated/improved each at least once every semester, and the tests and home assignments can be repeated/improved at least once in the first two weeks of the examination period.

Retake test is scheduled to the 14th week.

Type of examination (written, oral): **written**.

The exam is successful if the result is minimum 40 %.

Calculation of the grade (TVSz 47§ (3))

The mid-term performance accounts for **40 %**, the performance at the exam accounts for **60 %** in the calculation of the final grade. The final grade formula: $0,4 * \text{Semester\%} + 0,6 * \text{Exam\%}$

Calculation of the final grade based on aggregate performance in percentage.

| Course grade | Performance in % |
|------------------|------------------|
| excellent (5) | 85 % ... |
| good (4) | 70 % ... 85 % |
| satisfactory (3) | 55 % ... 70 % |
| pass (2) | 40 % ... 55 % |
| fail (1) | below 40 % |

The lower limit given at each grade belongs to that grade.

5. SPECIFIED LITERATURE

COMPULSORY READING AND AVAILABILITY

[1.] Ian Sommerville, *Software Engineering*, 10th Edition, Pearson, 2015

RECOMMENDED LITERATURE AND AVAILABILITY

[2.] R.S. Pressmann: *Software Engineering, a Practitioner's approach*, 7th Edition, McGraw-Hill Higher education, 2010

[3.] The lectures' materials (pptx, video)