



CS417

APPLIED SOFTWARE TECHNOLOGIES

Updated: protocol № 7/30.01.2019

Lecturer: Assoc.Prof. Veselina Zhecheva, PhD

ANNOTATION

The course provides knowledge of the theoretical aspects of modeling the overall process of software products development and its individual elements - lifecycle, quality, development approach, pricing, management and organization. Traditional and modern software design and development methods are examined in detail, as well as software assessment methods.

BASIC PURPOSES

Learning about existing models for describing the overall process of the software product development, methods and tools for planning and managing project work and applying them to a specific example project. Students will acquire knowledge and practical skills for analyzing user requirements in a specific subject area, building and step-by-step detailing of a model of the developed system, developing a software project, as well as additional development activities such as testing and quality assurance, support, documentation, etc.

PREREQUISITES

The students need basic knowledge of programming, object-oriented programming, databases and software application development.

STATUS AND STRUCTURE

PROGRAMME	status	ECTS	Full time				Part time			
			л	с	у	общ	л	с	у	Общ
Computer Science	Mandatory	6	30	30		60	15	15	30	
Software Engineering	Mandatory	6	30	30		60	15	15	30	

COURSE CONTENT

Topic 1. Introduction to software technologies. Nature and concept of programs and software products. Features of the software. Software Technologies - Definitions, Terminology and Objectives.

Topic 2. Lifecycle of a product and a software product. Lifecycle models. Classification of models.

Topic 3. Gunter's model for the life cycle of a software product. Phases. Functions.

Topic 4. Software development technologies. Basic paradigms. Structural (traditional) and object-oriented technologies. Main stages. Project and team organization.

Topic 5. Modern software development techniques. Agile techniques. Extreme programming. Specific and providing activities. Design.

Topic 6. Software quality assessment. General concepts. Quality models. Measurement in software production. Software metrics.

Topic 7. Cost of software development. Costing methods. Boehm model COCOMO. Method of function points.

Topic 8. Testing and documentation. Error detection and correction. Strategies and types of testing. Tools to automate testing. Types of documentation.

Topic 9. Support of the software. Configuration management.

Topic 10. Concept of project quality. SEI CMM / CMMI methodology. BOOTSTRAP Methodology. Software quality standards.

Topic 11. Unified Modeling Language (UML). Types of views and charts. Application in the description of software.

SEMINARS

Topic 1. Work organization and project management. Project Management Rules. Elements of a sample project. Project plan.

Topic 2. Initialization of a software project. Technical and economic terms of reference (TEC) and development contract.

Topic 3. Requirements Specification (SR) and Preliminary Development Assignment. Control points and parameters.

Topic 4. Project planning. External and internal project. Individual plans. Budget. Structure of a development assignment.

Topic 5. Network schedule. Decomposing the activities of a sample project.

Topic 6. Tools for testing automation. Types of funds.

Topic 7. Software products for project planning and management. MS Project. Basic settings.

Topic 8. Activities and deadlines in MS Project. Task settings and time scheduling. Network schedule for a sample project.

Topic 9. MS Project - Resources and Pricing. Project budget. Project implementation tracking.

Topic 10. Modeling language UML. Types of diagrams and their application in software description.

PLANNED LEARNING ACTIVITIES AND TEACHING METHODS

Training methods:

Face-to-face lectures and seminars

Visual learning

Practical Education

Interactive learning

E-learning through the Moodle platform

Teaching tools:

Self-paced work

Educational video materials incl. video presentations

Practical tasks

Programming tasks using application software

Use of electronic resources in the Moodle platform: theoretical materials, presentations, sample programs, tests and tasks for self-paced work on each topic

COURSEWORK

Course work is a task assigned to a group of students (two or three), which aims to summarize and systematize students' knowledge in the field of design and development of applied software. It is a complete application development project on a topic chosen by the students or assigned by the teacher and is presented and defended by the exam team. The task includes a description of the activities of a sample project and an illustration of the accompanying actions and documents (administrative and for the needs of project management):

- Development contract;

- Technical and economic assignment;
- Development assignment;
- External design;
- Individual work plan for each participant;
- Development plan implemented on Microsoft Project:
 - o Gantt chart containing a schedule of activities along with corresponding resources
 - o Project budget containing the total cost of the project by task
- At least 2 UML diagrams describing various aspects of the designed software.

ASSESSMENT METHODS

Final exam and possible oral interview. The final assessment includes assessment (current control) of the exercises, assessment of the coursework and assessment of the exam.

To form the assessment, the student collects points, the maximum value of which is 100. The distribution of points by evaluated activities is as follows:

- Students develop a course assignment in teams of two or three, corresponding to software development teams in an area chosen by the students and consulted with the instructor. The assignment includes a description of the activities of a sample project and an illustration of the accompanying actions and documents (administrative and for the needs of project management) - contract, technical and economic assignment, internal project, network schedule, etc. The implementation and protection of the development are evaluated - up to 30 points. The criteria for evaluating the development are: originality, thoroughness of the research, complexity of the topic, presentation of the development.
- For presentation of own ideas - up to 10 items.
- Up to 6 points are awarded for attendance and participation in the exercises.
- The exam is written and is a test with open questions, which is evaluated with a maximum of 54 points. The final evaluation is complex and includes ongoing control of the seminar exercises, an evaluation of the course work and an evaluation of the written exam. To form the grade, the student collects points, the maximum value of which is 100. The final grade is formed by distributing the points on the scale:
 - from 54 to 60 points - Medium (3);
 - from 61 to 70 points - Good (4);
 - from 71 to 80 points - Very good (5);
 - from 81 to 100 points - Excellent (6).