

DESCRIPTION OF STUDY MODULE*

Study programme

Applied Informatics and Programming

Study module

SOFTWARE ENGINEERING

Credits in total

4

Learning outcomes

- Ability to use the main concepts, standards of programs engineering and definition of software process.
- Assimilation of the creation stages of software and systems, lifetime cycles and its models, perform the network graphic analysis, applying the methods of requirements specification and validation during the development of program systems.
- Ability to use effectively the CASE programming measures of automatic construction and documentation.
- Understanding the processes of requirements specification and analysis, ability to model dynamics and static structure of the subject field, assimilation of object oriented UML modelling language.
- Ability to communicate with other people in the working group, organize its own work and perform the planned activities in time gaining the ability to work in a team.
- Applying the methods of requirements specification and validation during the development of program systems.

Aims of study module

The **aim** of the course – introduce the students to the methods and measures of Program Engineering and its application in the creation of program products.

Annotation of a study module

The course provides students with fundamental knowledge of program engineering methods, tools and their application in the design and development process of engineering systems. Students are taught to design a system from the projects vision and requirement documentation to full implementation and support, i.e. during this module project proposal, requirements document, system architecture document are being prepared, system testing and evaluation are done.

Topics of the subject

1. Introduction to the Course.
2. Definition of program engineering (PE).
3. Management of software projects.
4. Analysis and design stages of a programming product: processes, methods and means.
5. Object-oriented modelling.
6. Object-oriented design – UML language: use case diagram.
7. Purpose of requirements engineering. Conversation with the customer: dialogue, interview, etc. CASE tools.
8. Object-oriented design – UML language: classes' diagram.
9. Design of program architecture.
10. Modelling of UML interface: sequences and communication diagrams.
11. Object-oriented design – UML language: realization diagrams: components and composition models.
12. Stage of programming realizations (of coding and code generation) and works integration.
13. Techniques of software testing and evaluation.
14. The concept of project management.
15. Agile methods in Software Engineering.
16. Improvement of a project. New technologies of program engineering.

Procedure for assessment of knowledge and competences

Knowledge and abilities are evaluated on the basis of a criteria-based ten-point scoring system and accumulative assessment: individual task (5%), control work (5%), average assessment of laboratory works (15%), individual work (project) (25%) and assessment of the examination (50%). $G = II*0.05+KD*0.05+LD*0.15+PD*0.25+E*0.5$.

Main literature

1. NEMURAITĖ, Lina (2008). *Informacinių sistemų programinės įrangos projektavimas / Design of Information System Software Tools*. Klaipėda: Publishing House of Klaipėda University, 381 p. ISBN 978-9955-18-347-1.
2. PICHLER, Roman (2010). *Agile Product Management with Scrum: Creating Products that Customers Love*. New York: Addison Wesley, 160 p. ISBN 9780321605788.
3. PRESSMAN, Roger S.; and MAXIM, Bruce R. (2014). *Software Engineering: A Practitioner's Approach*. Boston, Mass: McGraw-Hill, 8th edition, 976 p. ISBN 978-0078022128.
4. SOMMERVILLE, Ian (2010). *Software Engineering*. Harlow: Pearson Education, 9th edition, 792 p. ISBN 978-0137035151.

* Short form