Course title: SOFTWARE ENGINEERING

Lecturers	Full Prof. Vjeran Strahonja, Ph.D., Assoc. Prof. Zlatko Stapić, Ph.D. Snježana Križanić, M.Inf. Marko Mijač, Ph.D.
Language of instruction	Croatian and English
Study level	Bachelor
Study programme	Information and Business Systems
Semester	4th (summer)
ECTS	6
Goal	The goal of Software Engineering course is to give students the insights into the most important phases, activities and the best practices of software product development, management of development project, tools to support this process, and associated technologies. The discipline of Software Engineering is a young discipline of science and the profession, but it is being highly intensively developed and is constantly undergoing numerous changes. By having insights into the most important stages of the development process, as well as understanding the mentioned process, students will gain fundamental knowledge about this complex area, which will give them a solid ground for their further development in these areas of development of software, applications for mobile or smart devices, web applications, and other systems like internet of things, embedded systems and alike.
General and specific learning outcomes	
Content	**1. Software Engineering as a Discipline (2)** Definition of discipline of software engineering and software. Positioning software engineering in relation to other disciplines. Professional software development in response to the software crisis. Engineering approach to development. Professional ethics. **2. Software Engineering Methodology (3)** General structure of software engineering methodologies. Object-oriented and agile methodologies for the development of software systems. Software product development process. Primary and secondary activities in program development (modeling, programming, documentation, testing, verification, validation, management ...). Program development methods and techniques. roles in program development. Development and work environments. **3. Models and Modeling in Software Engineering (3)** Modeling as the basis of engineering design. Object-oriented development of software systems. Historical development, sources and role of UML in software development. Modeling of structure, behavior and interactions. Model hierarchy and metamodels. 4. **Analysis and specification of user requirements (2)** Engineering requirements. Functional and non-functional requirements. Requirement engineering processes. Claim Elicitation. Request specification. Requirement validation. Manage user request changes. User requirements specification document. Good practice examples.

	5. Software product structure and behavior design (8) Object-oriented design. UML diagrams of structure, behavior and collaboration. Develop use cases and / or user stories. Diagram of classes, objects and packages. Behavior modeling of activity diagrams and state machines. Modeling of interaction with sequence diagrams, communication and timing. Components and deployment diagrams. Software product specification document. **6. Software Product Development (8)** Integrated development environments. Implementation of object-oriented principles in the selected programming language and development environment. Prototyping user interfaces. Implementation of UI concepts and experience in selected development environment and UI design tools. Implementation of working with data. Development frameworks. Organization of code. Code versioning. **7. Software System Development Management (2)** Project planning and management. Risk management. Cost of development and implementation. Versioning and quality management. Tools for team-based development and project management of software system development. **8. Testing and delivery of software product (2)** Testing and checking the quality of the software product. DevOps basics. Preparing a software product for delivery. Delivery and maintenance of software. Tools and technologies for checking and delivering software.
Exercises	Laboratory exercises are following the content of lectures and practically represent all stages of software development process that are theoretically addressed in lectures based on the practical part of exercises presented by the teacher, and on materials available in learning management system, students are required to implement additional assignments.
Realization and examination	Classes: lectures and exercises Exam: three project-related assignments (analysis, design, implementation), final exam or oral exam
Related courses	1. Introduction to Software Engineering, Carnegie Mellon University, https://www.cs.cmu.edu/~aldrich/courses/413/ 2. Introduction to Software Engineering, University of Adelaide, https://www.adelaide.edu.au/course-outlines/108366/1/sem-2/ 3. Introduction to Software Engineering, Douglas College, https://www.douglascollege.ca/course/cmpt-2276
Literature	Basic Sommerville I., Software Engineering, 8th edition or newer, Addison Wesley, 2007 or newer Optional Teaching and other course materials available in the learning management system, Authorized sources, web materials, and books on topics that the course addresses, which due to frequent and major changes in technologies and tools are to be defined for each generation of students separately.