VIDZEMES AUGSTSKOLA

FACULTY OF ENGINEERING
STUDY COURSE DESCRIPTION

Course Title:	**Software development principles from operations perspective**					
Course code (LAIS):						
Study programme:	**Information technologies**					
Level of Study programme:	☐	1st level professional higher education				
	X	Professional Bachelor				
	☐	Professional Master				
	☐	Academic Master				
	☐	PhD level				
Type of Study programme:	☐	Compulsory course (Part A)				
	X	Professional specialization courses (Part B, compulsory)				
	☐	Professional specialization optional courses (Part B, optional)				
	☐	Elective courses (Part C)				
Course Workload:	**Credits**		**ECTS**	**Academic hours**	**Contact hours**	**Independent work hours**
	4		6	160	64	96
Course Author/ Tutor:	**Kaspars Urbāns**					
	Guest lecturer, Bc.sc.ing.					
	e-mail: kaspars.urbans@va.lv					
	Consultation: according to the schedule for each semester					
Study Form:	Full time studies					
Study year, semester:	2 nd year, 1st semester					
Language:	Latvian, English					
Prerequisites for the Course:	Skills working with UNIX type terminal (can be learned during the course); Basic knowledge working with at least one of the following programming languages: Pyth PHP, C++ (one of the mentioned or any other that is not mentioned here); Understanding of SQL principles					
Course Summary:	The aim of this course is to provide students with practical and theoretical knowledge about managing software development life cycle independent of the chosen programming language. Main topics of the course: 1. Software source code version control and management; 2. Software development basic principles; 3. Software testing, deployment and maintenance process automation; 4. Software monitoring; 5. Software and generated data backup creation and restoring. For each course topic students will be provided with knowledge about basic principles, potential tools to be used and real life examples based on success and failure stories. Students are encouraged to seek for solutions themselves, cirticize the solutions provided by the course and to figure out solutions suitable for real life working applications. During the course, students will be improving their skills working with various tools to achieve the objectives of each topic. The work during the course is carried out idividually. Each of the student is working on its own project from the beginning till the end of the course adopting principles learned during all the course topics.					
Assessment:	Exam					
Requirements for Credits:	The course final score is determined by: 1. All practical exercises scores - 40% 2. All topic test scores - 20 % 3. Attendance of all practical lectures - 10% 4. Exam score - 30% Students who have completed all topic practical exercises requirements are allowed to do the final exam.					

VIDZEMES AUGSTSKOLA

	To pass the exam, the student needs to demonstrate knowledge about one of the topics covered in the course. The topic is chosen randomly chosen before the exam. Each topic exercise contains a problematic situation description. Students task is to think of a solution for the situation described. Students have 30 minutes to prepare. During preparation, students are allowed to use their project developed during the course.
Abiding by the Academic Ethics	Students must abide by the academic and research ethics, Vidzeme University of Applied Sciences Ethics Regulations, incl.: – study papers must be independently developed; – the study work should reference all statements, ideas and data used that have been authored by someone else; – appropriate data acquisition methods should be used in the acquisition of data, the research ethics must be respected, empirical data must be collected independently and cannot be distorted or falsified; – the examination must be carried out by the student independently, without the use of supporting materials and/or consultations with other students, unless the lecturer states otherwise. In the event of non-compliance with the academic and research ethics, punishment is imposed in accordance with the ViA Ethics Regulations and the study course must be re-taken, unless the punishment is exmatriculation.

Learning Outcomes; the evaluation methods and criteria	**Learning Outcomes**	**The evaluation methods and criteria**
	Knowledge	
	Knowledge about software source code version management and managing process with "git-flow", about information to be stored in source code version control repositories	Online test about corresponding topic.
	Knowledge about software project static and dynamical configuration management.	Online test about corresponding topic.
	Knowledge about principles when storing data in relational data bases. Structuring the data, structure change automation, principles when storing data in files.	Online test about corresponding topic.
	Knowledge about principles when developing automated software deployment and management scripts, including automating software version roll back to previous versions quickly	Online test about corresponding topic.
	Skills	
	Skills to use source code version control tool "git".	Score for a completed practical implementation of the corresponding topic objective into students project.
	Implement static or/and dynamic configuration outside software source code according to the project requirements and structure.	Score for a completed practical implementation of the corresponding topic objective into students project.
	Implement event logging functionality, separating messages into various log levels. Create data base structure and data automated migration scripts, create software project event logging and configuration management tactics for various work environments.	Score for a completed practical implementation of the corresponding topic objective into students project.
	Create automated software deploy scripts for various work environments	Score for a completed practical implementation of the corresponding topic objective into students project.

	Competency	
	Manage software version control for software projects with a large development team.	Score for a completed practical implementation of the corresponding topic objective into students project, exam score.
	Plan software project configuration tactics to deliver continuous development, deployment and operations processes	Score for a completed practical implementation of the corresponding topic objective into students project, exam score.
	Analyze software events and incidents based on the logged events data.	Score for a completed practical implementation of the corresponding topic objective into students project, exam score.
	Plan software project data base structure to deliver deliver continuous development, deployment and operations processes. Optimize software deployment and operations processes utilizing automated scripts. Create software operations monitoring tactics and determine key metrics and indicators for uninterrupted operation of software.	Score for a completed practical implementation of the corresponding topic objective into students project, exam score.
Course Compulsory literature:	Scott Chacon, Ben Straub. Pro Git, 2020. (https://git-scm.com/book/en/v2)	
Course additional literature:	Anton A. Chuvakin. Logging and Log Management, 2012. Mikael Krief. Learning DevOps, 2019.	
Course confirmation date:		
Date of course description update:		

Study Course Plan:

Date	Theme	Academic hours		Study Form/ Organization of independent work of students and task description
		Contact hours	**Independent work hours**	
The date is specified before the implementation of the course	Intro about the course and main topics. Insight in to the course topic interaction in each of the development life cycle phases.	**2**	**6**	Theoretical lecture
	Software source code version control and management	**4**	**6**	Theoretical lecture
	Practical lecture working with sorce code version control tools (GIT, git-flow). Creating of source code repository, development environment preparation, work with version control system	4	8	Practical lecture, in some sections group activities.
	Software development principles (code formatting, comments, static/dynamic configuration, event logging, database structure management, file storing principles, scaleability).	6	8	Theoretical lecture

	Practical lecture implementing configuration management, event logging, database structure automated management into student software projects.	14	14	Practical lecture, individual activities on students software project.
	Software testing, deployment and operations process automation.	4	6	Theoretical lecture
	Practical lecture implementing operations services, automated testing and deployment procedures into student software projects	8	12	Practical lecture, individual activities on students software project
	Software monitoring	4	6	Theoretical lecture
	Practical lecture implementing external monitoring of critical parameters to deliver uninterrupted operation of software into student software project.	6	8	Practical lecture, individual activities on students software project.
	Software and created data backing up and restoring.	4	6	Theoretical lecture
	Practical lecture creating data back up mechanism for student software projects.	4	6	Theoretical lecture
	Exam	4	10	Individual exam in spoken form.
	Hours total:	***64***	***96***	