



COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Agile Development with Ruby	

Lecturer(s)	Department where the course unit is delivered
Coordinator: Saulius Grigaitis Other lecturers: -	Department of Computer Science Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
First	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	5, 7 semesters	Lithuanian

Prerequisites
Prerequisites: Object-oriented Programming

Number of credits allocated	Student's workload	Contact hours	Individual work
5	130	68	62

Purpose of the course unit: programme competences to be developed		
<p>Purpose of the course unit: to introduce students to advanced software development technologies by teaching Ruby programming language, introducing related tools and applying AGILE software development methods.</p> <p><i>Generic competences:</i></p> <ul style="list-style-type: none"> Life-long learning (BK2) <p><i>Specific competences:</i></p> <ul style="list-style-type: none"> Knowledge and skills of underlying conceptual basis (DK4) Technological and methodological knowledge and skills, professional competence (DK6) 		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
<ul style="list-style-type: none"> Understand Ruby programming language syntax and principles of Ruby interpreter. Apply Test Driven Development and Behaviour Driven Development methods. Write unit, functional and acceptance tests driven code, isolate modules and simulate objects in testing environment. Understand, modify and develop not complex web systems with <i>Ruby on Rails</i> framework. Apply agile development method for small and medium size Ruby software projects. 	Lectures, problem-oriented teaching, case studies, literary reading, individual work, tutorials, laboratory work.	Laboratory works and results presentation, written exam (open, semi-open and close-ended questions and tasks).

Course content: breakdown of the topics	Contact hours						Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Contact hours	Individual work	Assignments
Introduction to Ruby programming language: classes, objects, variables, types, containers, blocks, iterators, expression, input/output, threads, modules.	8				8	16	16	Individual reading. Laboratory works. Self-control tasks.
Test Driven Development and Behaviour Driven Development principles and live demo.	4				4	8	8	
Pair Programming principles.	2				2	4	4	
Automated unit testing.	2				2	4	4	
Automated functional testing	2				2	4	4	
Acceptance tests automation.	2				2	4	4	
Stubbing.	2				2	4	4	
Mocking.	2				2	4	4	
Introduction to <i>Ruby on Rails</i> framework. Models, Controllers, Views, Router.	6				6	12	10	
Automated testing with Ruby on Rails	4				4	4	4	
Tutorials during the semester		2				2		
Final exam (written)						2		
Total	32	2			32	8	68	62

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory works	60	Deadlines for 5 tasks are 3rd, 5th, 8th, 11th and 14th weeks respectively.	During the semester, a student is required to complete five tasks (implement Ruby project at selected subject). First task consists of defining project subject, defining and prioritizing main functionality. Student needs to implement main functionality implementation for second task. Student implements entire application functionality and command line interface for third task. Fourth task is previous Ruby application refactoring to ActiveRecord library. Fifth task is previous Ruby application refactoring to Ruby on Rails application with web interface Successfully completed first task is valued 0.5 points, second task is valued 1 point, third, fourth and fifth tasks are valued 1.5 points. Student can collect 6 point in total. That's 60% of final score. Completed tasks are evaluated based on quantitative and qualitative criteria, student answers to related questions. Student must collect at least 2 points to be allowed to take the exam.
Exam (written)	40	Exam session	During the exam, it is possible to get at most 4 points, which are equivalent to 40% of the final score. During exam student must answer various questions of diverse complexity (0-4 points).

Author	Publis hing year	Title	Number or volume	Publisher or URL
Required reading				
Dave Thomas, Andy Hunt, Chad Fowler	2013	Programming Ruby 1.9 & 2.0: The Pragmatic Programmers' Guide		https://pragprog.com/book/ruby4/ programming-ruby-1-9-2-0 . http://ruby- doc.com/docs/ProgrammingRuby
Michael Hartl	2015	Ruby on Rails Tutorial		https://www.railstutorial.org/book
Recommended reading				
Aaron Sumner	2015	Everyday Rails Testing with RSpec		<a href="https://leanpub.com/everydayrailsr
spec">https://leanpub.com/everydayrailsr spec
Paolo Perrotta	2014	Metaprogramming Ruby 2: Program Like the Ruby Pros		<a href="https://pragprog.com/book/ppmetr
2/metaprogramming-ruby-2">https://pragprog.com/book/ppmetr 2/metaprogramming-ruby-2