

UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2017/18)							
Predmet:		Sodobne metode razvoja programske opreme					
Course title:		Modern software development methods					
Študijski program in stopnja Study programme and level		Študijska smer Study field			Letnik Academic year		Semester Semester
Interdisciplinarni magistrski študijski program Računalništvo in matematika		ni smeri			1 ali 2		drugi
Interdisciplinary Master's study programme Computer Science and Mathematics		none			1 or 2		second
Vrsta predmeta / Course type					izbirni / elective		
Univerzitetna koda predmeta / University course code:					63515		
Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Klinične vaje work	Druge oblike študija	Samost. delo Individ. work	ECTS	
45	10	20			105	6	
Nosilec predmeta / Lecturer:		prof. dr. Viljan Mahnič					
Jeziki / Languages:		Predavanja / Lectures:		slovenski / Slovene, angleški / English			
		Vaje / Tutorial:		slovenski / Slovene, angleški / English			
Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:				Prerequisites:			
Vpis v letnik študija.				Enrolment in the programme.			
Vsebina:				Content (Syllabus outline):			

<p>Predavanja:</p> <p>Vsebina predmeta se prilagaja trenutno aktualnim trendom na področju razvoja programske opreme. Trenutno je prilagojena obravnavi agilnih metod in empiričnemu ovrednotenju njihove učinkovitosti:</p> <ol style="list-style-type: none"> 1. Pregled značilnosti agilnih metod za razvoj programske opreme 2. Vodenje agilnih projektov razvoja programske opreme <ul style="list-style-type: none"> - iterativen in inkrementalen razvoj - metoda Scrum 3. Tipične prakse ekstremnega programiranja <ul style="list-style-type: none"> - Programiranje v parih - Testno voden razvoj programske opreme - Sprotno preoblikovanje kode (refactoring) - Orodja za podporo testno vodenemu razvoju in preoblikovanju kode 4. Pomanjkljivosti ekstremnega programiranja 5. Metrike v programski opremi in merjenje učinkovitosti razvojnega procesa 	<p>Lectures:</p> <p>The contents adapts to current trends in software development. At the moment the focus is on agile methods for software development and empirical evaluation of these methods:</p> <ol style="list-style-type: none"> 1. Overview of agile methods for software development and their characteristics 2. Agile software project management <ul style="list-style-type: none"> - Iterative and incremental development - Scrum 3. Typical Extreme Programming practices <ul style="list-style-type: none"> - Pair programming - Test driven development (TDD) - Refactoring - Software tools that support TDD and refactoring 4. Weaknesses of Extreme Programming 5. Metrics in Software Engineering and measuring the effectiveness of the software development process 6. Case study: Empirical evaluation of the
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>6. Študija primera: empirično ovrednotenje posameznih praks iz točk 2 in 3</p> <p>7. Primerjava agilnega pristopa z modeli za discipliniran razvoj programske opreme (Personal Software Process, Team Software Process, Capability Maturity Model)</p> <p>Vaje:</p> <p>Namen vaj je dvojen:</p> <ol style="list-style-type: none"> 1. seznanjanje s sodobnimi orodji za razvoj programske opreme, 2. empirično ovrednotenje posameznih pristopov k razvoju programske opreme na podlagi praktičnega dela na projektih, ki so čim bolj podobni realnim. <p>Delo izven kontaktnih ur:</p> <p>Študenti razvijajo programe, ki so sestavni del projekta, ki služi kot osnova za študijo primera.</p>	<p>aforementioned practices</p> <p>7. Comparison of agile approach to traditional disciplined software development processes (Personal Software Process, Team Software Process, Capability Maturity Model)</p> <p>Lab practice:</p> <p>The purpose of lab practice is twofold:</p> <ol style="list-style-type: none"> 1. to acquaint students with modern software development tools, 2. to empirically evaluate different approaches to software development through practical work on (almost) real software projects. <p>Individual work outside of contact hours:</p> <p>Students develop programs that are part of the project that serves as a case study.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Temeljni literatura in viri / Readings:

1. K. Schwaber: Agile Project Management with Scrum, Microsoft Press, 2004.
2. K. Beck: Extreme Programming Explained, Addison-Wesley, več izdaj.
3. L. Williams, R. Kessler: Pair Programming Illuminated, Addison-Wesley, 2003.
4. K. Beck: Test-Driven Development: By Example, Addison-Wesley, 2003.
5. M. Cohn: User stories applied, Addison-Wesley, 2004.

Dodatna literatura:

1. B. Boehm, R. Turner: Balancing Agility and Discipline – A Guide for the Perplexed, Pearson Education, 2004.
2. CMMI for Development (CMMI-DEV), Version 1.2. CMU/SEI-2006-TR-008, Software Engineering Institute, Carnegie Mellon University, 2006.

Cilji in kompetence:

Cilj predmeta je poglobljena obravnava sodobnih (trenutno aktualnih) metod razvoja programske opreme in empirično ovrednotenje njihove učinkovitost v primerjavi z že uveljavljenimi pristopi. Študenti na obsežnejšem projektu, ki služi kot študija primera, preizkusijo obravnavane pristope, nato pa med seboj primerjajo in ovrednotijo njihove prednosti in pomanjkljivosti.

Objectives and competences:

In depth treatment and empirical evaluation of modern software development methods in comparison to traditional approach. Students work on a project that serves as a case study for evaluation of modern approaches in order to find their strengths and weaknesses.

Predvideni študijski rezultati:

Znanje in razumevanje:
Poznavanje in razumevanje:

- sodobnih pristopov k razvoju programske opreme
- ključnih faktorjev, ki vplivajo na uspešnost tovrstnih projektov
- pomena empiričnega spremljanja razvojnega procesa.

Intended learning outcomes:

Knowledge and understanding:
Knowledge and understanding of:

- modern approaches to software development
- key factors that affect the success of software development projects
- the importance of empirical evaluation of the software development process

<p>Uporaba:</p> <p>Uporaba obravnavanih metod in orodij pri delu na konkretnem projektu.</p> <p>Refleksija:</p> <p>Razumevanje okoliščin, ki vplivajo na potek razvoja programskih sistemov.</p> <p>Prenosljive spretnosti - niso vezane le na en predmet:</p> <p>Sposobnost za skupinsko delo, vodenje, načrtovanje in organizacijo, medsebojno komuniciranje, pisno in ustno poročanje.</p>	<p>Application:</p> <p>Application of modern software development methods and tools within the framework of an (almost) real software development project.</p> <p>Reflection:</p> <p>Understanding all the disparate issues affecting the implementation.</p> <p>Transferable skills:</p> <p>Increase in professional skills like team-work, management, planning and organization, written and oral communication.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Metode poučevanja in učenja:

Predavanja z aktivno udeležbo študentov (razlaga, diskusija, primeri, reševanje problemov). Laboratorijske vaje s praktičnim delom na večjem projektu, ki služi kot študija primera za ovrednotenje posameznih pristopov k razvoju programske opreme.

Learning and teaching methods:

Type (examination, oral, coursework, project):

Continuing (homework, midterm exams, project work)

Final (written and oral exam)

Grading: 6-10 pass, 1-5 fail (according to the rules of University of Ljubljana)

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
<p>Način (pisni izpit, ustno izpraševanje, naloge, projekt):</p> <p>Sprotno preverjanje (domače naloge, kolokviji in projektno delo)</p>	<p>50%</p> <p>50%</p>	<p>Type (examination, oral, coursework, project):</p> <p>Continuing (homework, midterm exams, project work)</p>

<p>Končno preverjanje (pisni in ustni izpit)</p> <p>Ocene: 6-10 pozitivno, 1-5 negativno</p> <p>(v skladu s Statutom UL)</p>		<p>Final (written and oral exam)</p> <p>Grading: 6-10 pass, 1-5 fail (according to the rules of University of Ljubljana)</p>
------------------------------------------------------------------------------------------------------------------------------	--	------------------------------------------------------------------------------------------------------------------------------

Reference nosilca / Lecturer's references:

MAHNIČ, Viljan. Teaching Scrum through team-project work : students' perceptions and teacher's observations. International journal of engineering education, ISSN 0949-149X, 2010, vol. 26, no. 1, str. 96-110, tabele. [COBISS.SI-ID 7613012]

MAHNIČ, Viljan. A case study on agile estimating and planning using Scrum. Elektronika ir elektrotehnika, ISSN 1392-1215. [Print ed.], 2011, no. 5, str. 123-128. [COBISS.SI-ID 8364116]

FÜRST, Luka, MERNIK, Marjan, MAHNIČ, Viljan. Improving the graph grammar parser of Rekers and Schürr. IET software, ISSN 1751-8806. [Print ed.], 2011, vol. 5, no. 2, str. 246-261, ilustr. [COBISS.SI-ID 8317780]

MAHNIČ, Viljan, VRANA, Ivan. Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process. Elektrotehniški vestnik, ISSN 0013-5852. [Slovenska tiskana izd.], 2007, letn. 74, št. 5, str. 241-247, ilustr. [COBISS.SI-ID 6362452]

MAHNIČ, Viljan, ŽABKAR, Nataša. Using COBIT indicators for measuring scrum-based software development. WSEAS transactions on computers, ISSN 1109-2750. [Print ed.], Oct. 2008, vol. 7, no. 10, str. 1605-1617, ilustr. [COBISS.SI-ID 6974036]