

1.	Наслов на наставниот предмет	Вреднување и тестирање на софтверот			
2.	Код	СЕ-И1-02			
3.	Студиска програма	Заеднички студии по софтверско инженерство			
4.	Организатор на студиската програма (единица, односно институт, катедра, оддел)	Факултет за информатички науки и компјутерско инженерство			
5.	Степен (прв, втор, трет циклус)	Студии од вториот циклус			
6.	Академска година / семестар	прва година, прв семестар	7.	Број на ЕКТС кредити	6
8.	Наставници	доц. д-р. Анастас Мишев проф. д-р. Катерина Здравкова проф. д-р. Сузана Лошковска			
9.	Предуслови за запишување на предметот	Реализирани најмалку 240 кредити на универзитетски студии			
10.	Цели на предметната програма	<ul style="list-style-type: none"> • Со зголемувањето на важноста на компјутерските системи во индустријата и општеството, техниките кои го поддржуваат производството на доверлив софтвер стануваат позначајни. Комплексноста на многу компјутерски системи побарува примена на низи вакви техники. • Двата најветувачки приода се формалните методи и тестирањето на софтверот. Вообичаено, формалните методи и тестирањето на софтверот се сметаат за спротивставени. Затоа, вземната размена на информациите и интеракцијата меѓу нивните застапници недостасува. Последниве години дојде до компромис, според кој овие два приода се комплементарни. Тоа доведе до истражување на начините на кои овие два приода се надополнуваат. • Користењето на формалните спецификации или модели ја елиминира двосмисленоста и на таков начин ја намалува опасноста грешките да бидат вклучени при развивањето на софтверот. Во случаите кога постои формален метод, и изворниот код и спецификациите можат да се третираат како формални објекти кои можат да се анализираат и со кои може да се ракува. Користењето на формалните спецификации ја овозможува формалната, па евентуално и автоматската анализа на односите меѓу спецификациите и изворниот код. Се очекува ова да има форма на доказ, но ваквиот доказ не ја гарантира оперативната точност. Затоа, дури и тогаш кога доказ постои, неопходно е да се примени и динамичкото тестирање. • Тестирањето на софтверот е значајно и традиционално е исклучително скап дел во процесот на развивањето софтвер. Студиите покажуваат дека во голем број случаи тестирањето чини дури 50% од целокупната цена за развој. Таму каде што постојат формалните спецификации и модели, тие можат да бидат користени како основа за автоматзација на процесот на тестирањето. Ова може да води кон поефикасно и поефективно тестирање. Оттука произлегува дека автоматзацијата на делови од тестирањето на софтверот е една од најзначајните придобивки од користењето на формалните спецификации. Врската меѓу тестирањето и формалните методи води кон генерирањето тестови од формалните спецификации. • Предметов има цел да ги воведи и критички да ги анализира моменталните техники за тестирање на софтверот, со посебно внимание на формалните методи. <p>Компетенции По успешното завршување на овој предмет, студентот ќе биде во состојба:</p> <ul style="list-style-type: none"> • критички да ја процени важноста од тестирањето на софтверот; 			

	<ul style="list-style-type: none"> да ги почитува потребите и полезноста од формалните методи во процесот на тестирањето; да развива интегриран пристап за тестирање на софтверот и формалните теории 			
11.	<p>Содржина на предметната програма:</p> <ul style="list-style-type: none"> Воведување во основите на тестирањето. Структурирано тестирање. Функционално тестирање. Основи за комбинирање на формалните методи и тестирањето. Модел на база на формалните методи. Тестирање базирано на конечните автомати. Алгебарски специфицирано тестирање. Тестирање со UML динамичките дијаграми. Временска логика. Проверка на моделите и нивната улога во тестирањето. Управување со процесот на тестирањето софтвер 			
12.	Методи на учење: Предавања, вежби, самостојна работа, проектни задачи, семинарски работи			
13.	Вкупен расположив фонд на време	6 ЕКТС x 25 часа = 150 часа		
14.	Распределба на расположивото време	30+30+30+20+40 = 150 часа		
15.	Форми на наставните активности	15.1.	Предавања- теоретска настава	30 часови
		15.2.	Вежби (лабораториски, аудиториски), семинари, тимска работа	30 часови
16.	Други форми на активности	16.1.	Проектни задачи	30 часови
		16.2.	Самостојни задачи	20 часови
		16.3.	Домашно учење	40 часови
17.	Начин на оценување			
	17.1.	Тестови		20 бодови
	17.2.	Семинарска работа/ проект (презентација: писмена и усна)		70 бодови
	17.3.	Активност и учество		10 бодови
18.	Критериуми за оценување (бодови/ оценка)		до 50 бода	5 (пет) (F)
			од 51 до 60 бода	6 (шест) (E)
			од 61 до 70 бода	7 (седум) (D)
			од 71 до 80 бода	8 (осум) (C)
			од 81 до 90 бода	9 (девет) (B)
			од 91 до 100 бода	10 (десет) (A)
19.	Услов за потпис и полагање на завршен испит	реализирани 15.1 и 15.2		
20.	Јазик на кој се изведува наставата	македонски или англиски		
21.	Метод на следење на квалитетот на	интерна евалуација и анкети, според модел		

наставата	даден во глава Error! Reference source not found.
-----------	--

22.	Литература				
22.1.	Задолжителна литература				
	Ред. број	Автор	Наслов	Издавач	Година
	1.	Paul Ammann and Jeff Offutt	Introduction to Software Testing	Cambridge University Press	2008
	2.	Ilene Burnstein	Practical Software Testing	Springer professional computing	2003
3.	C. Kaner, J. Falk, H. Q. Nguyen	Testing Computer Software	Wiley	1999	
22.2.	Дополнителна литература				
	Ред. број	Автор	Наслов	Издавач	Година
	1.	P. C. Jorgensen	Software Testing: A Craftsman's Approach, second edition	CRC Press	2004
	2.	W. E. Lewis	Software Testing and Continuous Quality Improvement	Auerbach	2004
	3.	J. Rubin, D. Chisnell	Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests	Wiley Publishing, Inc.	2008
	4.	Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled	Model Checking	MIT Press	1999
5.	B. Beize	Software Testing Techniques	International Thomson Press	1990	