

1.	Наслов на наставниот предмет	<b>Надежност на софтвер</b> Software Reliability		
2.	Код	СОЦД-И-04		
3.	Студиска програма	Систем во чип (ТЕМПУС)		
4.	Организатор на студиската програма (единица, односно институт, катедра, оддел)	Факултет за информатички науки и компјутерско инженерство – ФИНКИ		
5.	Степен (прв, втор, трет циклус)	втор циклус		
6.	Академска година / семестар <b>2 / зимски / изборен</b>	7. Број на ЕКТС кредити	<b>6</b>	
8.	Наставник	Доц д-р Соња Филиповска, Доц. Д-р Иван Чорбев		
9.	Предуслови за запишување на предметот	нема		
10.	Цели на предметната програма (компетенции): Курсот има за цел да им понуди на студентите познавање на основните концепти за отпорност на грешка, основните техники за постигнување на отпорност на грешка во софтверските, комуникациските и електронските системи. Здобивање на вештини за моделирање и евалуирање на архитектури отпорни на грешка, нивна овозможност и сигурност. Да се здобијат со знаење за можните извори на грешка и начини за нивно предвидување и спречување. Да знае да дизајнира и развива високо-надежен софтвер.			
11.	Содржина на предметната програма: Дефиниција на надежност на софтвер, Дефиниција на расположливост на софтвер, Потреба од надежен софтвер, Примена на софтвер толерантен на откази. Развоен циклус за софтвер, Модели за грешки во софтверот, Извори на грешки во софтверот, Стратегија на развој на софтвер со многу мал број на грешки. Користење на стандарди при кодирањето. Примена на различни структури на софтверот, Декомпозиција по модули, Партиционирање, Затварање на модулите, Аномични акции. Детекција на грешки, Временски проверки, Проверка наназад, Кодови за детекција и корекција на грешка, Проверка на логиката, Структурни проверки. Обработка на исклучоци, Дизајн на системот за обработка на исклучоци. Дебагирање, Дебагирање со индукција, Дебагирање со дедукција. Логирање на информации, Примена на логирањето за дебагирање, Примена на логирањето за детекција на грешки. тестирање, Автоматизирање на процесот на тестирање, Unit тестови, Интеграциски тестови, Acceptance тестови. Техники кај софтвер кој се прави во една верзија, Контролни точки и Повторно стартување, Пар од процеси, Разновидност на влезните податоци. ехники кај софтвер во повеќе верзии, Опоравувачки блокови, Програмирање со N-верзии, N Само-Проверувачки програми, Опоравувачки блокови со консензус, t/(n-1)- Варијантно програмирање.			
12.	Методи на учење: Предавања поддржани со презентации преку слајдови, интерактивни предавања, вежби (користење на опрема и софтверски пакети), тимска работа, пример случаи, поканети гости предавачи, самостојна изработка и одбрана на проектна задача и семинарска работа, учење во електронско опкружување (форуми, консултации).			
13.	Вкупен расположив фонд на време	6 ECTS x 30 часа = 180 часа		
14.	Распределба на расположивото време	30 + 15 + 135 = 180 часа		
15.	Форми на наставните активности	15.1.	Предавања- теоретска настава	30 часови
		15.2.	Вежби (лабораториски, аудиториски), семинари, тимска работа	15 часови
16.	Други форми на активности	16.1.	Проектни задачи	60 часови
		16.2.	Самостојни задачи	25 часови

		16.3.	Домашно учење	50 часови
17.	Начин на оценување			
	17.1.	Тестови		45 бодови
	17.2.	Семинарска работа/ проект ( презентација: писмена и усна)		45 бодови

19.	Услов за потпис и полагање на завршен испит		реализирани активности 15.1 и 15.2			
20.	Јазик на кој се изведува наставата		македонски и англиски			
21.	Метод на следење на квалитетот на наставата		механизам на интерна евалуација и анкети			
	Литература					
	22.1.	Задолжителна литература				
		Ред. број	Автор	Наслов	Издавач	Година
		1.	Israel Koren, C. Mani Krishna	Fault-Tolerant Systems	Morgan Kaufmann	2007
2.	Laura L. Pullum	Software Fault Tolerance Techniques and Implementation	Artech House Publishers	2001		

22.	3.	Kopetz, Hermann	Real-Time Systems, Design Principles for Distributed Embedded Applications	Springer	2011
	Дополнителна литература				
	Ред. број	Автор	Наслов	Издавач	Година
	1.	Mitch Lacey	The Scrum Field Guide: Practical Advice for Your First Year (Agile Software Development Series)	Addison-Wesley Professional	2012
	2.	Dan Skwire	First Fault Software Problem Solving: A Guide for Engineers, Managers and Users	Opentask	2009
22.2.	3.	Sam Guckenheimer, Neno Loje	Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback (2nd Edition) (Microsoft .NET Development Series)	Addison-Wesley Professional; 2nd edition	2011