

Наставна дисциплина	Надежност на софтверот				
Семестар	Вид	Фонд на часови	Кредити	Јазик	Институт
X	Изборен	2+0+1+3	5	МК	КТИ
Предуслови					
Компетенции*	Запознавање на концептите за градба на високонадежен софтвер. Дизајн и развој на надежен софтвер.				
Содржина	<p>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. естирање, Автоматизирање на процесот на тестирање, Unit тестови, Интеграциски тестови, Acceptance тестови.</p> <p>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, Оповарувачки блокови, Програмирање со N-верзии, N Само-Проверувачки програми, Оповарувачки блокови со консензус, t/(n-1)- Варијантно програмирање.</p> <p>Литература:</p> <ol style="list-style-type: none"> 1. Laura L. Pullum, <i>Software Fault Tolerance Techniques and Implementation</i>, Artech House, 2001 2. Wilfredo Torres-Pomales, <i>Software Fault Tolerance:A Tutorial</i>, NASATM-2000-210616, 2000 3. Glenford J. Myers, <i>The Art of Software Testing, Second Edition</i>, John Wiley & Sons, 2004 4. Paul Hamill, <i>Unit Test Frameworks</i>, O'Reilly, 2004 				