



<b>Студијски програм:</b> Основне академске студије информатике					
<b>Назив предмета:</b> СОФТВЕРСКО ИНЖЕЊЕРСТВО					
<b>Статус предмета:</b> Обавезан на модулу Рачунарске науке и Информационо-комуникационе технологије					
<b>Број ЕСПБ:</b> 7					
<b>Услов:</b> Уписан одговарајући семестар; Положена два од три следећа предмета: Објектно-оријентисно програмирање, Базе података 1 и Клијентске веб технологије					
<b>Циљ предмета</b> Упознавање студената модула <i>Рачунарске науке</i> са основним методама и техникама за развој софтвера у свим фазама животног циклуса. Студенти ће у оквиру овог предмета имати прилику да се упознају са методама анализе и документовања захтева, и пројектовања софтвера на високом и детаљном нивоу, документовањем дизајна софтвера, принципима корисничког доживљаја и добром праксом дизајна корисничког интерфејса, техникама тестирања софтвера, као и са испоруком и одржавањем софтвера.					
<b>Исход предмета</b> Студент је способан да изврши интервјуисање клијента, прикупи, анализира и документује захтеве које софтвер треба да испуни. Студент уме да пројектује софтверски систем на високом нивоу, и да одабере одговарајуће технологије и развојно окружење. Студент уме да пројектује софтвер на детаљном нивоу и да изабере одговарајуће структуре података и алгоритме за имплементацију. Студент је оспособљен да документује дизајн софтверског решења. Студент уме да прави планове тестирања софтвера, и да дефинише и спроводи јединичне тестове. Студент је овладао знањем неопходним за квалитетну испоруку и одржавање развијеног софтвера.					
<b>Садржај предмета</b> <i>Теоријска настава</i> Увод у софтверско инжењерство. Моделовање процеса и животног циклуса. Евидентирање захтева. Важност захтева. Извођење захтева. Типови захтева. Документовање захтева (спецификација софтверских захтева). Дизајнирање система. Концептуални и технички дизајн. Модуларност. Архитектонски стилови. Карактеристике доброг дизајна. Кориснички доживљај. Случајеви коришћења. Представљање класа (дијаграми класа). Дијаграми секвенце сарадње, стања, активности,...). Документовање дизајна софтвера. Тестирање програма. Грешке и откази. Врсте грешака. Организација тестирања. Тестни тимови. Јединично тестирање. Интеграционо тестирање. Планирање теста. Алати за аутоматизовано тестирање. <i>Практична настава</i> Инсталација радног окружења. Упознавање са алатима за колаборацију. Упознавање са алатима за верзионисање кода. Тимски рад на развоју софтверског решења задатог проблема. Документовање софтверских захтева. Документовање дизајна софтвера. Презентовање решења.					
<b>Литература</b> 1. S.L. Pfleeger, J.M. Atlee, <i>Софтверско инжењерство, теорија и пракса</i> , СЕТ, Београд, 2006.					
<b>Број часова</b>	<b>активне наставе</b>	<b>Теоријска настава:</b>	<b>3</b>	<b>Практична настава:</b>	<b>1 + 2</b>
<b>Методe извођења наставе</b> Теоријска настава се изводи у виду интерактивних предавања, током којих наставник помоћу електронских презентација и традиционалних метода студентима излаже садржај предмета. Студенти активно учествују у настави кроз дискусије о различитим варијантама решавања проблема и њиховим последицама на квалитет софтверског решења у погледу испуњења функционалних, пројектних и процесних захтева, као и на процес испоруке и одржавања софтвера. Практична настава се изводи у виду лабораторијских вежби у рачунарским учионицама, на којима студенти у тимовима континуирано раде на реализацији пројекта развоја задатог софтвера. Из седмице у седмицу студенти са асистентима анализирају досадашњи развој, последице одабраних решења и дефинишу циљеве даљег развоја. Поред класичне наставе у виду предавања и вежби, студенти у посебним терминима имају могућност консултација са наставницима и асистентима у вези са проблемима у савладавању градива.					
<b>Оцена знања (максимални број поена 100)</b>					
<b>Предиспитне обавезе</b>	<b>70</b>	<b>Завршни испит</b>	<b>30</b>		
практична настава	50	усмени испит	30		
семинар-и	20				