



Часопис из области економије
менаџмента и информатике
Година 2016, волумен 7, број 2, стр. 45-55



Journal of Economics, Management
and Informatics
Year 2016, Volume 7, Number 2, pp. 45-55

Стручни рад/ Professional paper

УДК/UDC: 378.016(497.11)
378:[007:004

DOI: 10.5937/bizinfo1602045J

SMERNICE ZA IZRADU KURIKULUMA ZA STUDIJSKE PROGRAME IZ OBLASTI RAČUNARSKOG INŽENJERSTVA

CURRICULUM GUIDELINES FOR UNDERGRADUATE DEGREE STUDY PROGRAMS IN COMPUTER ENGINEERING

Nenad Jovanović¹

Univerzitet u Prištini, Fakultet tehničkih nauka, Kosovska Mitrovica, Srbija

Rezime: ACM i IEEE su u decembru 2016. godine publikovali smernice za nastavne planove studijskih programa iz oblasti računarskog inženjerstva. Izveštaj CE2016 odražava tekuće zahteve u obrazovanju i praksi računarskog inženjeringa, a koji će biti aktuelni u narednoj deceniji. U izveštaju se daju određeni primeri i načini mapiranja nastavnih jedinica na studijske programe i određuju se očekivani zahtevi sa naučnog aspekta. Ovaj rad daje pregled izveštaja CE2016 i ima za cilj da pruži uputstva za implementaciju ovih smernica u studijske programe računarskog inženjerstva.

Ključne reči: studijski programi, računarsko inženjerstvo, preporuke.

Abstract: ACM and the IEEE Computer Society have released their computer engineering curriculum guidelines in December of 2016. The report, tagged CE2016, reflects current computer engineering education and practice that would be relevant for the next decade. The report gives specific examples and methods of mapping teaching units in study programs and addressing the expected requirements from a scientific point of view. This paper provides an overview of the report CE2016 and aims to provide a way to implement these guidelines in the study programs of computer engineering.

Keywords: study program, computer engineering, recommendations.

¹ nndjov@gmail.com

1. UVOD

Preporuke koje su date u dokumentu “Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering” (CE2004) [1], publikovanog od strane tri kompetentne organizacije: The Association for Computing Machinery (ACM), The Association for Information Systems (AIS) i The Computer Society (IEEE-CS), predstavljaju osnovu na kojoj se zasnivaju kurikulumi za studijske programe iz oblasti računarskog inženjerstva.

ACM i IEEE Computer Society su, nakon detaljnog razmatranja, zaključili da je dokumentu CE2004 potrebna revizija i 2011. godine su formirali upravni odbor za ažuriranje tekućeg izveštaja sa modernim i smislenim sadržajima. Posle opsežne javne rasprave, koja je obuhvatila preko dvadeset hiljada privrednih i akademskih subjekata, ACM i IEEE su objavili svoj izveštaj u dokumentu CE2016 (Computer Engineering Curricula 2016). Javna rasprava je sprovedena u industrijskoj i akademskoj zajednici kroz istraživanja, kao i kroz workshop-ove na brojnim značajnim konferencijama (Impagliazzo et al. 2012, Eric, et al. 2013, Impagliazzo and Durant, 2014).

Deo izveštaja sadrži dodavanje novih ili proširenje postojećih oblasti tehničkih veština, kao što su moderne mogućnosti umrežavanja, sistem na čipu (SoC), agilne metode i alati, projektovanje ugrađenih sistema, paralelno programiranje, hardver-softver ko-dizajn, arhitekture sa više jezgara itd.

Što se tiče laboratorijskih vežbi, izveštaj pruža niz sugestija o načinima da se takve vežbe integrišu u nastavne planove i programe. Takođe, izveštaj daje i preporuke za konfigurisanje takvih laboratorija u skladu sa studijskim programima računarskog inženjerstva.

2. OSNOVNI PRINCIPI

Računarski inženjering je veoma važna oblast u ekspanziji. Upravni odbor CE2016 je definisao skup osnovnih principa na kojima se zasniva izrada nastavnih planova i programa iz ove oblasti:

1. Brza evolucija računarskog inženjeringa zahteva stalnu reviziju odgovarajućeg nastavnog plana i programa. Imajući u vidu tempo promena u ovoj disciplini, profesionalna udruženja treba da uspostave preporuke za proces revizije koje će omogućiti pravovremeno ažuriranje pojedinačnih komponenti nastavnog plana i programa.
2. Razvoj nastavnih planova i programa računarskog inženjerstva mora da prati promene u tehnologiji i pedagogiji i da uvaži značaj doživotnog učenja. U oblasti koja evoluira tako brzo kao računarsko inženjerstvo, obrazovne institucije moraju da usvoje eksplicitnu

- strategiju za reagovanje na promene. Obrazovanje u oblasti računarskog inženjerstva mora da teži da pripremi studente za doživotno učenje, koje će im omogućiti da prevaziđu današnje tehnologije i da odgovore na izazove budućnosti.
3. Važno je identifikovati osnovne veštine i znanja koje svi diplomci računarskog inženjeringa moraju posedovati.
 4. Potrebna srž oblasti znanja mora da bude što je moguće manja. Ovo je važno kako bi se obezbedila fleksibilnost i prilagodljivost u izboru ostalih delova nastavnog plana i programa i kako bi se omogućilo stvaranje individualnih studijskih programa.
 5. Računarski inženjering mora da uključi odgovarajuća i potrebna laboratorijska iskustva. Programi treba da uključe iskustva u projektovanju, izgradnji i testiranju i hardverskih i softverskih sistema.
 6. Nastavni planovi i programi iz oblasti računarskog inženjerstva su često predmet akreditacije, licenciranja ili vladinih ograničenja. Preporuke moraju da uvažavaju sva spoljna ograničenja i da daju smernice za njihovu evoluciju.
 7. Nastavni plan i program mora da obuhvati profesionalnu praksu kao svoj sastavni deo.
 8. Iako je važno da nastavni plan artikuliše široke vizije računarskog obrazovanja, uspeh bilo kog plana i programa u velikoj meri zavisi od implementacionih detalja. Da bi se to postiglo izveštaj mora da sadrži primere i modele nastavnih planova i programa.
 9. Da bi bio uspešan, proces stvaranja preporuka mora da obuhvati učešće različitih subjekata, od privrednih, vladinih i visokoškolskih institucija.
 10. Zahtevi za izradu planova i programa moraju biti zasnovani na principima koji su međunarodno prihvatljivi.
 11. Svi relevantni alati, standardi i ograničenja moraju biti uključeni u oblasti znanja.
 12. Integracija hardvera i softvera je od krucijalnog značaja, pa je potrebno obezbediti i naglasiti potrebu razvoja celog računarskog sistema u laboratorijskim uslovim koji uključuje hardver, operativne sisteme i softverske sisteme i njihovu upotrebu u kontekstu relevantnih aplikacija.

3. LABORATORIJSKE VEŽBE

Laboratorijske vežbe su veoma važan deo nastavnog plana i programa računarskog inženjerstva i one imaju višestruku ulogu u obrazovanju. U svakom inženjerstvu, pa i u računarskom inženjerstvu veoma je bitno da studenti imaju mogućnost da posmatraju, istražuju, manipulišu karakteristikama i ponašanjima aktuelnih uređaja, sistema i procesa.

Ovo uključuje osmišljavanje, primenu, ispitivanje i dokumentovanje hardvera i softvera, dizajniranje eksperimenata za prikupljanje podataka, analizu i interpretaciju tih podataka u cilju korekcije dizajna i provere specificiranih funkcija.

Tipično, laboratorijske vežbe traju dva ili tri časa i odigravaju se na mestu sa specijalizovanom opremom. Složenost ovih vežbi, razlikuje se od institucije do institucije i zavisi od broja časova unutar nastavnog plana, fizičkog prostora i raspoložive opreme. U tabeli 1. prikazane su preporuke upravnog odbora CE2016 u vezi važnosti pojedinih tipova laboratorijskih vežbu u nastavnom planu i programu računarskog inženjerstva.

Tabela 1. Tipovi laboratorija za računarsko inženjerstvo

Tip laboratorije	Mora	Trebalo bi	Dopunske
Elektronika i električna kola	x		
Arhitektura računara i projektovanje			x
Digitalno procesiranje signala			x
Digitalna logika i projektovanje sistema	x		
Ugrađeni sistemi	x		
Uvod u inženjerstvo			x
Umrežavanje		x	
Projektovanje softvera		x	
Napredno projektovanje	x		

Izvor: Computer Engineering Curricula 2016, Final Curriculum Report, str. 34.

Laboratorije treba da obuhvate i primenu softvera za simulaciju projektovanja računarskih sistema, uključujući i digitalne sisteme. Alati za simulaciju od suštinske su važnosti u profesionalnoj inženjerskoj praksi.

Simulacije su poželjne i često neophodne kada se proučavaju složeni procesi i sistemi čija je analiza sa realnom opremom napratična ili ograničena vremenom i raspoloživim resursima. Ovi sistemi treba da se koriste i za hardverske i za softverske sisteme.

Laboratorijske vežbe treba da pomognu studentima da izuče praktične veštine kao što su:

- Bezbedno ponašanje u laboratorijama, naročito kada električna oprema predstavlja opasnost.
- Pravilna upotreba računara i opreme za testiranje.
- Izgradnja električnih i elektronskih kola i sistema.
- Izgradnja i testiranje softvera.
- Razumevanje procesa i pitanja u vezi sa razvojem proizvoda.

- Prepoznavanje mogućnosti za kompromisima i biti u stanju da se donesu odluke u ovoj oblasti.

3. RAZMATRANJE SOFTVERA

Softverski alati i paketi koji se koriste u nastavnom planu i programu računarskog inženjerstva razlikuju se u zavisnosti od filozofije i potreba svakog studijskog programa. U tabeli 2. prikazane su preporuke u vezi softverskih paketa koji bi trebalo da se koriste u specifičnim laboratorijskim postavkama. Ove preporuke su ilustrativnog karaktera, nije predviđeno da svaki program mora da uključi sve ove aplikacije, već svaki program treba, na osnovu svojih potreba da odredi svoju postavku, imajući u obzir i najnovije verzije ovih aplikacija.

Tabela 2. Preporučeni softverski paketi u računarskom inženjeringu

Projekovanje modelovanje i simulacije	Razvoj softvera	Opšte računarstvo
<ul style="list-style-type: none"> • Nivo kola (npr. SPICE) • Gate nivo (šematski ulaz) • Digitalni sistemi (VHDL, Verilog) • Analogna kola (VHDL-AMS, Verilog-AMS) • Sistemsko projektovanje (System Verilog, System C) 	<ul style="list-style-type: none"> • Integrirano razvojno okruženje (IDE) • Kompajleri (C, C++, Java, Pyton, C#) • Podrška operativnim sistemima • Otklanjanje grešaka na nivou izvornog koda • Podrška za platforme (mobilni telefoni, tableti) 	<ul style="list-style-type: none"> • Web browser • Email • Office • PDF reader/editor • Photo viewer/editor • Multimedije players/editors • Kompresija fajlova • Transfer fajlova (FTP, SCP) • Terminal emulatori
Kreiranje prototipa hardvera		Alati sistemskog inženjerstva
<ul style="list-style-type: none"> • FPGA/CPLD razvojni paket • Component/IP biblioteka za podršku • Programiranje uređaja • Interaktivno otklanjanje grešaka 	Integrirano projektovanje circuit/ASIC	<ul style="list-style-type: none"> • Project management (GANTT,PERT charts) • Requirements and specifications management (UML tools)
Sistemsko projektovanje mikrokontrolera	<ul style="list-style-type: none"> • Projektovanje i simulacija • Sinteza • Grafički editor • Automatsko raspoređivanje komponenti • Verifikacija projektovanja • Projektovanje za testiranje, automatsko generisanje test obrasca 	Ostali alati
<ul style="list-style-type: none"> • Integrirano razvojno okruženje (IDE) • Kompajler, assembler, linker • Simulator procesora • Programiranje uređaja • Testiranje/debugovanje 	Computer-aided design and modeling (CAD tools)	<ul style="list-style-type: none"> • Razvoj softvera za robotiku • Poluprovodnički uređaji i modelovanje procesa (TCAD) • Mikrotalasna tehnika • Simulacije elektromagnetnog polja • MEMS
	Laboratorije za	

Matematički paketi	automatizaciju i instrumentaciju	projektovanje
	<ul style="list-style-type: none"> • (IntuiLink, LabVIEW) 	
<ul style="list-style-type: none"> • Rešavanje problema • Analiza podataka • Modelovanje i simulacija 		

Izvor: Computer Engineering Curricula 2016, Final Curriculum Report, str. 35.

4. SADRŽAJ NASTAVNOG PROGRAMA

Nastavni program računarskog inženjerstva zahteva izbor različitih oblika znanja, praktičnih veština i stavova koje sve treba razmotriti unutar jednog okvira. Jedno od ključnih pitanja jeste kako izvršiti distribuciju, po godinama studiranja, relativno nepromenljivog gradiva (matematika, logička kola...) u odnosu na materijale modernih tehnologija. Tradicionalni pristup ima tendenciju da se ovi bazični materijali pojave na početku programa. Međutim, studenti koji se odluče da studiraju računarski inženjering, često su motivisani praktičnim inženjerskim znanjima i ako na početku studija imaju samo bazične predmete oni mogu biti razočarani i frustrirani zbog toga.

Izveštaj CE2016 promovise fleksibilnost u kreiranju nastavnog plana i programa i ne propisuje kako treba uključiti bazične kurseve u plan i program i ostavlja prostor za izborne i inovativne predmete.

Iako kreiranje nastavnih planova i programa zahteva značajnu lokalnu adaptaciju, ipak moguće je definisati nekoliko ključnih principa koji mogu da pomognu u procesu donošenja odluka:

- Nastavni plan i program mora da odražava integritet i karakter računarskog inženjerstva kao nezavisne discipline.
- Nastavni plan i program mora da odgovori na brze tehničke promene.
- Nastavni plan i program treba da bude zasnovan na očekivanim rezultatima i ishodima učenja.
- Nastavni plan i program treba da odražava konzistentan etički princip, koji promovise inovacije, kreativnost i profesionalizam.

Upravni odbor CE2016 u svom modelu nastavnog plana i programa, prvo definiše primarne discipline, tzv. oblasti znanja (eng. knowledge areas), koje zajedno čine telo znanja (eng. Body of Knowledge). Oblasti znanja se dele na manje entitete pod nazivom jedinice znanja (eng. knowledge units) ili nastavne jedinice koje predstavljaju pojedinačne teme u okviru oblasti. Jedinice znanja se opisuju na osnovu očekivanih ishoda učenja. Oblasti znanja u računarskom inženjerstvu predstavljeni su u tabeli 3.

Tabela 3. Oblasti znanja u računarskom inženjerstvu

Tag ID	Knowledge Area
CE-CAE	Elektronika i električna kola
CE-CAL	Računarski algoritmi
CE-CAO	Arhitektura i organizacija računara
CE-DIG	Digitalno projektovanje
CE-ESY	Ugrađeni sistemi
CE-NWK	Računarske mreže
CE-PPP	Priprema za profesionalnu praksu
CE-SEC	Bezbednost informacija
CE-SGP	Procesiranje signala
CE-SPE	Sistemska i projektno inženjerstvo
CE-SRM	Upravljanje sistemskim resursima
CE-SWD	Projektovanje softvera

Izvor: Computer Engineering Curricula 2016, Final Curriculum Report, str. 24.

Set nastavnih jedinica se povezuje sa svakom od ovih oblasti znanja i svakoj nastavnoj jedinici se dodaju očekivani ishodi učenja.

Kao i kod svih nastavnih planova i programa, oblasti znanja nisu kursevi. Na svakom nastavnom programu je da usaglasi ove oblasti znanja i nastavnih jedinica sa kursom. Izveštaj CE2016 daje neke primere koji omogućavaju ovo usaglašavanje. Jedan primer tela znanja za računarsko inženjerstvo prikazan je u tabeli 4.

Tabela 4. Oblasti znanja i nastavne jedinice

CE-CAE Elektronika i električna kola [50]	
CE-CAE-1	Istorija i pregled [1]
CE-CAE-2	Relavantni alati, standardi i inženjerska ograničenja[3]
CE-CAE-3	Električne veličine i osnovni elementi [4]
CE-CAE-4	Električna kola [11]
CE-CAE-5	Elektronski materijali, diode, tranzistori [7]
CE-CAE-6	MOS kola, tajming i snaga [12]
CE-CAE-7	Arhitektura ćelija za skladištenje [3]
CE-CAE-8	Logički interfejsi [3]
CE-CAE-9	Operacioni pojačavači [3]
CE-CAE-10	Projektovanje mixed-signal kola [3]
CE-CAE-11	Parametri projektovanja
CE-CAE-12	Modelovanje kola i simulacioni modeli
CE-CAO Arhitektura i organizacija računara [60]	
CE-CAO-1	Istorija i pregled [1]
CE-CAO-2	Relavantni alati, standardi i inženjerska ograničenja [1]
CE-CAO-3	Set instrukcija [10]

CE-CAO-4	Merenje performansi [3]
CE-CAO-5	Računarska aritmetika [3]
CE-CAO-6	Organizacija procesora [10]
CE-CAO-7	Arhitektura i organizacija memorijskog sistema [9]
CE-CAO-8	Ulaz/Izlaz interfejsi i komunikacija [7]
CE-CAO-9	Periferni sistemi [7]
CE-CAO-10	Arhitekture sa više jezgara [5]
CE-CAO-11	Arhitektura distribuiranih sistema [4]
CE-ESY Ugrađeni sistemi [40]	
CE-ESY-1	Istorija i pregled [1]
CE-ESY-2	Relavantni alati, standardi i inženjerska ograničenja [2]
CE-ESY-3	Karakteristike ugrađenih sistema [2]
CE-ESY-4	Osnovne softverske tehnike [3]
CE-ESY-5	Paralelni ulaz i izlaz [3]
CE-ESY-6	Asinhrona i sinhrona serijska komunikacija [6]
CE-ESY-7	Periodični prekidi, generisanje talasa, vremenska merenja [3]
CE-ESY-8	Akvizicija podataka, kontrola, senzori [4]
CE-ESY-9	Implementacione strategije za kompleksne ugrađene sisteme [7]
CE-ESY-10	Tehnike za operacije niske snage [3]
CE-ESY-11	Mobilni i mrežni ugrađeni sistemi [3]
CE-ESY-12	Napredni ulazno/izlazni uređaji [3]
CE-ESY-13	Računarske platforme za ugrađene sisteme
CE-PPP Profesionalna praksa [20]	
CE-PPP-1	Istorija i pregled [1]
CE-PPP-2	Relavantni alati, standardi i inženjerska ograničenja [1]
CE-PPP-3	Strategije efektivne komunikacije [1]
CE-PPP-4	Pristup interdisciplinarnom timu [1]
CE-PPP-5	Filozofski i kulturni okviri [2]
CE-PPP-6	Inženjerska rešenja i socijalni efekti [2]
CE-PPP-7	Profesionalna i etička odgovornost [4]
CE-PPP-8	Intelektualna svojina i pravna pitanja [3]
CE-PPP-9	Savremeni problemi [2]
CE-PPP-10	Poslovna i upravljačka pitanja [3]
CE-PPP-11	Kompromisi u profesionalnoj praksi
CE-SGP Procesiranje signala [30]	
CE-SGP-1	Istorija i pregled [1]
CE-SGP-2	Relavantni alati, standardi i inženjerska ograničenja [3]
CE-SGP-3	Konvolucija [3]
CE-SGP-4	Transformaciona analiza [5]
CE-SGP-5	Frekvencija odziva [5]
CE-SGP-6	Odabiranje [3]
CE-SGP-7	Digitalni spektar i diskretne transformacije [6]
CE-SGP-8	Projektovanje filtera sa impulsnim odzivom [4]
CE-SGP-9	Prozorske funkcije
CE-SGP-10	Procesiranje mutimedija
CE-SGP-11	Kontrolni sistemi
CE-SRM Menadžment sistemskih resursa [20]	
CE-SRM-1	Istorija i pregled [1]

SMERNICE ZA IZRADU KURIKULUMA ZA STUDIJSKE PROGRAME IZ OBLASTI RAČUNARSKOG INŽENJERSTVA

CE-SRM-2	Relavantni alati, standardi i inženjerska ograničenja [1]
CE-SRM-3	Upravljanje sistemskim resursima [8]
CE-SRM-4	Projektovanje Real-time operativnih sistema [4]
CE-SRM-5	Operativni sistemi za mobilne [3]
CE-SRM-6	Podrška za konkurentno procesiranje [3]
CE-SRM-7	Evaluacija sistemskih performansi
CE-SRM-8	Podrška za virtualizaciju
CE-CAL Računarski algoritmi [30]	
CE-CAL-1	Istorija i pregled [1]
CE-CAL-2	Relavantni alati, standardi i inženjerska ograničenja [1]
CE-CAL-3	Analiza osnovnih algoritama [4]
CE-CAL-4	Algoritamske strategije [6]
CE-CAL-5	Klasični algoritmi [3]
CE-CAL-6	Analiza i projektovanje specifičnih algoritama [6]
CE-CAL-7	Paralelni algoritmi i višenitni rad [6]
CE-CAL-8	Kompleksnost algoritama [3]
CE-CAL-9	Algoritmi raspoređivanja
CE-CAL-10	Osnove teorije izračunljivosti
CE-DIG Digitalno projektovanje [50]	
CE-DIG-1	Istorija i pregled [1]
CE-DIG-2	Relavantni alati, standardi i inženjerska ograničenja [2]
CE-DIG-3	Brojni sistemi i kodovanje podataka [3]
CE-DIG-4	Bulova algebra [3]
CE-DIG-5	Osnovna logička kola [6]
CE-DIG-6	Modularni dizajn kombinacionih kola [8]
CE-DIG-7	Modularni dizajn i sekvencijalna kola [9]
CE-DIG-8	Control and datapath design [9]
CE-DIG-9	Projektovanje programabilnom logikom [4]
CE-DIG-10	Ograničenja sistemskog projektovanja [5]
CE-DIG-11	Model greške, testiranje
CE-NWK Računarske mreže [20]	
CE-NWK-1	Istorija i pregled [1]
CE-NWK-2	Relavantni alati, standardi i inženjerska ograničenja [1]
CE-NWK-3	Mrežna arhitektura [4]
CE-NWK-4	LAN i WAN mreže [4]
CE-NWK-5	Bežične i mobilne mreže [2]
CE-NWK-6	Mrežni protokoli [3]
CE-NWK-7	Mrežne aplikacije [2]
CE-NWK-8	Network management [3]
CE-NWK-9	Komunikacija podataka
CE-NWK-10	Procena performansi
CE-NWK-11	Bežične senzorske mreže
CE-SEC Bezbednost informavija [20]	
CE-SEC-1	Istorija i pregled [2]
CE-SEC-2	Relavantni alati, standardi i inženjerska ograničenja [2]
CE-SEC-3	Bezbednost i integritet podataka [1]
CE-SEC-4	Ranjivost: tehnički i ljudski faktori [4]
CE-SEC-5	Model zaštite resursa [1]
CE-SEC-6	Kriptografija [3]

CE-SEC-7	Autentifikacija porukama [1]
CE-SEC-8	Mrežna i web bezbednost [3]
CE-SEC-9	Autentifikacija [1]
CE-SEC-10	Poverljivo računarstvo [1]
CE-SEC-11	Napadi sa strane kanala [1]
CE-SPE Sistemski i projektni inženjering [35]	
CE-SPE-1	Istorija i pregled [1]
CE-SPE-2	Relavantni alati, standardi i inženjerska ograničenja [3]
CE-SPE-3	Principi projektnog menadžmenta [3]
CE-SPE-4	Iskustva korisnika [6]
CE-SPE-5	Rizik, pouzdanost, bezbednost i tolerancija na greške [3]
CE-SPE-6	Hardverski i softverski procesi [3]
CE-SPE-7	Analiza zahteva [2]
CE-SPE-8	Specifikacija sistema [2]
CE-SPE-9	Projektovanje i evaluacija sistema [4]
CE-SPE-10	Projektovanje konkurentni sistema [3]
CE-SPE-11	Sistemska integracija, testiranje i validacija [3]
CE-SPE-12	Održavanje i održivi razvoj [2]
CE-SWD Projektovanje softvera [45]	
CE-SWD-1	Istorija i pregled [1]
CE-SWD-2	Relavantni alati, standardi i inženjerska ograničenja [3]
CE-SWD-3	Programske konstrukcije i paradigme [12]
CE-SWD-4	Strategije rešavanja problema [5]
CE-SWD-5	Strukture podataka [5]
CE-SWD-6	Rekurzija [3]
CE-SWD-7	Objektno orijentisano propjektovanje [4]
CE-SWD-8	Testiranje i kvalitet softvera [5]
CE-SWD-9	Modelovanje podataka [2]
CE-SWD-10	Baze podataka [3]
CE-SWD-11	Konkurentno i programiranje vođeno događajima [2]
CE-SWD-12	Programski interfejsi
CE-SWD-13	Data mining
CE-SWD-14	Virtualizacija podataka

Izvor: Computer Engineering Curricula 2016, Final Curriculum Report, str. 25-26.

5. ZAKLJUČAK

Računarski inženjering kao akademska oblast obuhvata široke oblasti elektrotehničkog inženjerstva i računarskih nauka. Računarsko inženjerstvo se definiše kao disciplina koja obuhvata projektovanje, konstrukciju, implementaciju i održavanje softverskih i hardverskih komponentata modernih računarskih sistema i računarski kontrolisane opreme.

Računarski inženjering je veoma važna oblast u ekspanziji i profesionalna udruženja treba da uspostave preporuke za izradu nastavnog plana i programa iz ove oblasti.

ACM i IEEE su u decembru 2016. godine publikovali smernice za izradu nastavnih planova i programa računarskog inženjerstva. Ovaj izveštaj je poznat pod imenom CE2016 i on odražava tekuće zahteve u obrazovanju i praksi računarskog inženjeringa, a koji će biti aktuelni u narednoj dekadi.

LITERATURA

1. IEEE Computer Society, 2004. *Curriculum guidelines for undergraduate degree programs in computer engineering - Final Report*. [pdf] IEEE Computer Society. Dostupno na: <http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf> [Datum pristupa 18.12.2016.]
2. IEEE Computer Society, 2016. *Computer Engineering Curricula 2016, Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, - Final Curriculum Report*. [pdf] IEEE Computer Society. Dostupno na: <<https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>> [Datum pristupa 18.12.2016.]
3. Impagliazzo, J., Conry, S., Durant, E., McGettrick, A., Wilson, T. and Thornton, M., 2012. *Special session: computer engineering review task force report, ACM Special Interest Group on Computer Science Education, (SIGCSE) Conference*. [pdf] February 29 – March 3, 2012 Raleigh, North Carolina, USA. Dostupno na: <https://lyle.smu.edu/~mitch/ftp_dir/pubs/sigcse12.pdf> [Datum pristupa 19.12.2016.]
4. Durant, E., et al. 2013. Computer engineering curriculum guidelines. *IEEE Frontiers in Education Conference (FIE)*. [pdf] 2013. Dostupno na: <http://lyle.smu.edu/~mitch/ftp_dir/pubs/fie13b.pdf> [Datum pristupa 19.12.2016.]
5. Impagliazzo, J. and Durant, E., 2014 Toward a modern curriculum for computer engineering. *In Teaching, Assessment and Learning (TALE)*, December 2014 International Conference on IEEE.

Рад је примљен: 22.12.2016.

Прихваћен за објављивање: 27.12.2016.

Received: 22 December 2016

Accepted: 27 December 2016

