

▣ Kombinirano učenje – izkušnje in rešitve

Iztok Bitenc, Borut Werber, Marko Urh

Univerza v Mariboru, Fakulteta za organizacijske vede, Kidričeva cesta 55a, 4000 Kranj
 iztok.bitenc@um.si, borut.werber@um.si, marko.urh@um.si

Izvilleček

Ob uvedbi kombiniranega e-učenja smo se na fakulteti predavatelji in študenti srečali z novimi izzivi. Posebej se je to pokazalo pri učenju računalniškega programiranja in podobnih predmetih, ki od študentov zahtevajo sprotno usvajanje znanja in logično kombiniranje tega znanja pri reševanju zastavljenih problemskih nalog. Tu smo zaznali povečano odsotnost študentov, plagiatstvo in izjemno povečanje obremenitev predavateljev pri pregledovanju in ocenjevanju problemsko usmerjenih e-nalog. Rešitve smo našli na več področjih. Uvedli smo več načinov motiviranja študentov za sprotno delo. Spremenili smo vsebino e-nalog, da osnovni pregled pravilnosti oddane kode izvede računalnik. Prilagodili smo nastavitve orodja za podporo e-študiju za čim hitrejši pregled in ocenitev e-nalog. Namesto informativnih e-nalog smo izvedli e-predavanje kot videokonferenco. Z vzporednim izvajanjem predavanj in vaj smo študentom dali več časa za usvojitve novih znanj. Rezultat teh in tudi drugih ukrepov je danes za približno tretjino povečana udeležba študentov na prvem izpitnem roku.

Ključne besede: e-učenje, kombinirano učenje, Moodle, praktične izkušnje

Abstract

Blended e-learning introduced new challenges for teachers and students in our faculty. This was especially true for computer programming class and similar subjects, where students need to build up their knowledge on a regular basis and logically combine the acquired knowledge in problem-solving e-lessons. We have noticed an increased absence from classes and plagiarism, as well as an extraordinary increase of teachers' workload when grading the submitted problem-solving e-lessons. Solutions were found in different fields. We have introduced many methods to motivate students for regular study activities. Requirements for e-lessons were changed in order to use the computer as a first tool to assess the correctness of the submitted computer code. We have also changed the settings in Moodle's review module in order to streamline the grading of e-lessons. A videoconference was used instead of an e-lecture. By combining lectures and exercises in a timetable, we have given the students more time to master the new knowledge. The introduction of these and certain other measures improved the attendance of students in the first exam period by about one third.

Keywords: E-learning, blended learning, Moodle, practical experience.

1 UVOD

Po splošni definiciji je učenje proces ustvarjanja novega znanja in pridobivanja veščin skozi izkušnje. Na Fakulteti za organizacijske vede Univerze v Mariboru (FOV) nam je bolj domača definicija, ki jo ponuja Slovar slovenskega knjižnega jezika za področje šolstva: programirano učenje je individualizirano učenje po vnaprej natančno pripravljenem gradivu s pomočjo priprav za učenje ali posebnih knjig. Seve-

da bo potrebno to definicijo dopolniti, saj je silovit razvoj informacijske in komunikacijske tehnologije (IKT) z novimi učnimi metodami in orodji močno posegel tudi v domeno izobraževanja. Za FOV je že kmalu po prelomu tisočletja postalo zanimivo področje e-izobraževanja, saj smo imeli v preteklosti veliko število izrednih študentov, ki so jim naši predavatelji predavali na več kot 10 lokacijah po vsej Sloveniji. V e-izobraževanju smo videli možnost

zmanjšanja obremenitve predavateljev in znižanje stroškov, povezanih s predavanji na lokacijah. Tako smo v enem od laboratorijev pričeli poskusno uvajati e-izobraževanje s podporo učne platforme Moodle že leta 2004 (Leskovar, 2005). Ker so se tudi na nivoju univerze pričele odvijati aktivnosti v zvezi z uvedbo e-študija, smo z razvojem lastne rešitve počakali. Kot člani Univerze v Mariboru smo na osnovi strategije univerze leta 2011 vpeljali posebno obliko e-študija – tako imenovano *kombinirano učenje* (blended learning) – klasično poučevanje v kombinaciji z e-študijem, ki je bilo podprto na nivoju univerze z Moodle učno platformo. Približno polovica predavanj in vaj se je tako namesto v klasični obliki v predavalnicah začela izvajati izključno z računalnikom. Seveda smo ob uvedbi začeli zaznavati težave: od zamujanja pri oddaji e-nalog, povečane odsotnosti študentov na e-predavanjih in vajah, pojava plagiatorstva pri oddaji e-nalog do preobremenjenosti učiteljev zaradi pregledovanja velikega števila e-nalog. Več let smo tako iskali in izboljševali organizacijske in tudi vsebinske rešitve, ki jih v tem prispevku želimo predstaviti.

2 OPREDELITEV POJMOV

Preden opredelimo problem in predlagamo možne rešitve, je potrebno opredeliti osnovne pojme, ki se uporabljajo na tem področju. **E-študij** omogoča platformo, ki ob vsakem času in na vsakem mestu omogoča nadgradnjo znanj in spretnosti (Gowda & Suma, 2017). Zagotavlja platformo, v kateri posameznik dobi prilagojen paket, ki je povezan s ključnimi tematskimi področji, s pomočjo samovodenega procesa. To predstavlja sodelovanje tehnologije, pedagogike in akreditacije, in sicer s ciljem pripraviti privlačen koncept učenja, imenovanega e-učenja.

Glede na vključenost e-učenja v pedagoški proces ločimo dva koncepta: študij na daljavo in kombinirano učenje (»blended learning«). Medtem ko študij na daljavo običajno predstavlja študij, kjer študenti ves študij izvajajo od doma na osnovi gradiv, nalog in testov, sprva v papirnati obliki in niso nikoli fizično skupaj s predavatelji (Siemens idr., 2015) online learning, and blended learning. With the intent of informing future research and practice in the emerging discipline of digital learning, this tertiary study focuses on the history and state of distance education, and the understanding of the large body of empirical research as captured by secondary studies (i.e.,

meta-analyses and systematic literature reviews, kasneje izvedeno v obliki »on-line study«, je **kombinirano učenje** mešanica klasičnega učenja v učilnici v kombinaciji z e-študijem, ki omogoča delo na daljavo z uporabo sodobnih informacijskih tehnologij (Kennedy & Newcombe, 2011; Nazarenko, 2015). Z vidika usmerjenosti ločimo v učitelja usmerjeno učenje (Teacher-Centric learning) in v učenca usmerjeno učenje (Learner Centric Learning) (Gowda & Suma, 2017). Medtem ko v **učitelja usmerjeno učenje** poudarja pomen pripravljenega predmetnika s postavljenimi temami, vsebinami in cilji, ki jih učitelj s svojim posredovanjem učencem poskuša doseči, se v **učence usmerjen študij** posveča učencem kot posameznikom, njihovim željam in potrebam, saj učenci dobijo aktivno vlogo v učnem procesu (skupinsko delo, medsebojno ocenjevanje, podajanje snovi s strani učencev, igranje vlog, učenje z igro ...), učitelj pa je predvsem v vlogi usmerjanja in organiziranja izmenjave znanj med učenci (Gowda & Suma, 2017). V več raziskavah so proučevali razlike med e-študijem in klasičnim študijem (Daymont & Blau, 2008; Gowda & Suma, 2017; Nazarenko, 2015; William T. Kaufman, 2015). Večinoma ugotavljajo, da je osip študentov večji pri e-študiju in se kot najbolj optimalna oblika kaže kombinirano učenje. V praksi se je pokazalo, da je zelo pomembno, kaj se pri posameznem predmetu uči in kaj se pri posameznem predmetu zahteva. Ugotavlja se, da je pri e-študiju zelo pomembna komunikacija predavatelja s študentom, saj slednji hitreje izgubi interes in se izgubi, če mu pri delu ne pomaga predavatelj (Moore, 2014). Še več, ni dovolj čas komuniciranja, temveč kvaliteta komuniciranja. Osebna motivacija, oblika študijskih gradiv, učno okolje in organizacijska podpora so ključni dejavniki, ki vplivajo, da več študentov prekine e-študij (Wang, Foucar-Szocki, Griffin, O'connor, & Sceiford, 2003). V raziskavi (Anderson, 2008) so proučevani dejavniki, ki vplivajo na uspešno izvedbo e-učenja. Med slednje spadajo lastnosti študentov, učiteljev, razpoložljiva tehnologija, vsebina predmetov, podpora študentom, organiziranost institucije, lastnosti družbe in stroški. V tem prispevku se osredotočamo na predavatelje, študente in organiziranost predmetov.

3 ŠTUDIJ NA FOV

Na FOV imamo **seminarski način študija** – predavanja in vaje se za vsak predmet izvajajo praktično vsak

delovni dan v tednu po 3 šolske ure, običajno po dva predmeta vzporedno, pogosto tudi trije. Prvi izpitni rok je običajno 2–3 tedne po izteku predavanj in vaj. To pomeni za študente in predavatelje precej natrpan urnik. Prav tako nimamo predpisane obvezne prisotnosti na predavanjih in vajah. Z večjim številom udeležencev (do 150 študentov) se običajno srečamo v prvem letniku, ko vsi študenti obiskujejo skupni predmetnik, predmeti pa so bolj informativni. V višjih letnikih se študenti razdelijo na 3 programe in so skupine velike od 10 do 40 študentov, zahtevnost predmetov in obveznosti študentov pa se povečajo. V študijskem letu poslušajo študenti okrog 8 predmetov in 700 ur predavanj in vaj.

Programska oprema za e-študij je **Moodle** (<http://moodle.org/>). To je odprtokodna učna platforma, ki izobraževalcem, administratorjem in učencem zagotavlja enoten, zanesljiv in integriran sistem za ustvarjanje prilagojenih učnih okolij. Z več deset tisoč namestitvami in več kot 90 milijoni uporabnikov je največja učna platforma na svetu. Je enostavna za uporabo, preko licence GNU odprtokodna, podprta z močno skupnostjo organizacij in razvijalcev. Prevedena je v slovenščino, zanjo pa obstaja tudi več sto vtičnikov, ki dodatno razširijo njeno uporabnost (Moodle, 2019). Na FOV za podporo e-študiju pretežno uporabljamo univerzitetni strežnik, dostopen na naslovu <https://estudij.um.si/>, pri enem predavatelju pa tudi lokalno namestitev (<https://swqlab.fov.um.si/moodle/>).

Eden od uporabljenih Moodlevih vtičnikov je tudi odprtokodni spletni videokonferenčni sistem **Big Blue Button** (BBB) (<http://bigbluebutton.org/>), ki Moodlevi spletni učilnici doda aktivnost »videokonferenca«. BBB podpira deljenje video in avdio vsebin, naprednih predstavitev z uporabo table (kazalec, približevanje, risanje), javne in zasebne klepetalnice, deljenje zaslona in podobno. Študent spremlja predavatelja (živa slika in glas predavatelja, deljen ekran predavatelja – »tabla«, kjer predavatelj lahko kaže prosojnice ali npr. piše programsko kodo, klepetalnica, kviz) in ostale udeležence (klepetalnica, zvok, deljen ekran) v realnem času in se aktivno odziva na način, ki mu ga omogoči moderator sestanka (Big Blue Button, 2019).

4 IZKUŠNJE Z E-ŠTUDIJEM V PRAKSI

Opazene težave pri izvajanju e-študija in njihove rešitve lahko v grobem razdelimo v dve kategoriji:

1) posamezne rešitve pri specifičnih zahtevah in 2) obvladovanje velikega števila udeležencev. Pri obeh imamo podkategoriji: a) organizacijske rešitve, kjer so spremenili način izvedbe predavanj in vaj, in b) vsebinske rešitve, kjer so dejansko spremenili vsebino predmetov.

5 REŠITVE SPECIFIČNIH ZAHTEV

Predmeti, ki so se še iz časov pred uvedbo e-študija izkazali za težavne, so predmeti, kjer študente učimo osnove računalniškega programiranja v drugem letniku informacijske smeri študija. Predmeti so imeli v zgodovini poučevanja veliko različnih imen in 5 različnih programskih jezikov (trenutno sta aktualna jezika JavaScript in Java), jedro vsebine pa večinoma ostaja isto. Študente na konkretnem višjenivojskem programskem jeziku učimo osnov programiranja: od ukazov programskega jezika preko metod, ki se uporabijo za analizo problema in za izgradnjo algoritma za programsko rešitev, do dejanskega pisanja in razhroščevanja kode. Cilj predmetov je usposobiti študente, da samostojno na podlagi zahtev izdelajo pravilno delujoč program. Predmeti so med zahtevnejšimi, saj se programiranja ne da naučiti »na pamet«, ampak mora študent dejansko **razumeti** načela programiranja in obvladati pravilno uporabo gradnikov programskega jezika. Brez tega študent na izpitu, ki se vedno izvaja na računalniku, za zadano nalogo in v omejenem času enostavno ni sposoben sestaviti algoritma delovanja in na računalniku napisati delujoče programske kode.

Zahtevano znanje se pri vsakem predmetu tokom predavanj in vaj postopno nadgrajuje, kar pomeni, da mora študent za uspešno nadaljevanje vsaj dobro poznati vsebine prejšnjih predavanj in vaj. Študent, ki izpusti dan predavanj ali vaj, mora manjkajočo snov samostojno predelati do naslednjega dneva, sicer tvega, da naslednjim predavanjem ne bo mogel uspešno slediti. Ob običajnem razporedu dveh predmetov dnevno se je tako samostojno delo izkazalo kot zahtevna naloga.

Pri klasičnem podajanju predmeta pred uvedbo e-študija smo sprotno delo poskusili »vsiliti« z vmesnimi kolokviji, ki so se na koncu seštelih v oceno predmeta, vendar je na ta način uspelo zaključiti predmet le boljšim študentom oz. v povprečju manj kot tretjini. Začetne kolokvije je pisala večina študentov, izrazit osip se je pokazal, ko je snov postala zahtevnejša, npr. delo z datotekami, sploh pa pri upora-

bi eno- in dvodimenzionalnih tabel (array). Ko smo iskali razloge za osip študentov, smo pogosto dobili odgovor, da je snov postala prezahtevna in da bodo ta predmet pustili »za konec«, ko bodo imeli več časa. Iz rezultatov drugih študij izhaja, da nimamo s tem težav le pri nas (Daymont & Blau, 2008). Tak način študentskega »dela« ima pri e-študiju dodatno neprijetno posledico: povzroči dodatno obremenitev pedagoških delavcev, ki morajo v dneh pred izpitnimi roki znova nadzorovati oddaje e-nalog predmetov, ki so se dejansko že iztekli in zanje nimajo več predvidenih pedagoških ur.

E-študij smo kot člani Univerze v Mariboru vpejljali v obliki kombiniranega učenja – klasično poučevanje v kombinaciji z e-študijem. Ker nismo imeli veliko predhodnih izkušenj, je bila prva delitev časa med njima približno 50 % predavanj in vaj v e-obliki. Novonastale e-naloge so dobile dve vsebini: samostojno učenje manjših omejenih vsebin (npr. predstavitve in uporaba tekstovnih, računskih, datumskih ... funkcij in metod) in utrjevanje že obdelane snovi. Primer dela naloge za utrjevanje predstavljene snovi je prikazan spodaj.

Vse e-naloge so bile zastavljene tako, da bi jih boljši študenti rešili v približno eni uri, slabši pa v treh urah. Le redki študenti so reševali e-naloge v tistih urah, ki se bile predvidene po urniku in ko smo predavatelji sedeli pred svojimi računalniki in čakali na njihova vprašanja (sinhrona e-predavanja ali vaje). Študenti so e-naloge razumeli kot asinhrona predavanja ali vaje in so e-naloge oddali enkrat v tistem dnevu, običajno v večernih urah. Zato smo predavatelji hitro prilagodili svoj način dela in čas za oddajo e-nalog temu »urniku«. Tako je zadnji čas za oddajo e-nalog (običajno 20:00) postal dejansko čas, ko smo začeli pregledovati oddane e-naloge. Delali smo torej

izven rednega delovnega časa, vendar smo to kmalu uredili s pogodbami o delu na domu. Na žalost se je prav tako hitro pokazalo, da je veliko (tudi do polovice) študentov razumelo e-naloge kot dodaten prosti čas in jih niso rešili na sprejemljiv način, jih sploh niso rešili ali pa so v skrajnih primerih oddali rešitve svojih sošolcev, to je plagiate. Ker niso predelali predvidene snovi, vsi ti študenti niso imeli dobre osnove za naslednja predavanja, zato so jim sledili z opaznimi težavami.

Rešitev, kako pripraviti študente k redni oddaji e-nalog, smo najprej poiskali v prisili: oddane sprejemljive e-naloge so postale pogoj za pristop k izpitu. Še vedno pa nismo rešili težave glede prepoznega oddajanja e-nalog. Prisile, to je »zapiranja« nalog po pretečenem roku za oddajo, nismo mogli uporabiti iz več razlogov:

- že v prvem poskusu časovne omejitve oddaje se je pokazalo, da je kasneje potrebno za vsakega zamudnika posebej ročno nastaviti možnost oddaje,
- oddane e-naloge so pogoj za pristop k izpitu in zato oddaja ne sme biti onemogočena,
- za zaključek pa smo dobili še pravno mnenje, da časovno ne smemo omejiti oddaje, ker to ni v skladu z neobveznim obiskom vaj in predavanj, kar pomeni, da študent lahko zamudi oddajo.

Zato smo se namesto »prisile« odločili uporabiti »spodbudo«. Uvedli smo sistem točkovanja: za oddano e-nalogo, ki je sprejemljiva, je dobil študent točko. Vsaj eno točko za vsako e-nalogo je moral dobiti, da je lahko pristopil k izpitu. Študent je lahko dobil dodatno točko, če je v roku oddal e-nalogo. Če je zbral več kot 85 % možnih točk, je to pomenilo, da je pravočasno oddal štiri od petih e-nalog, zato se mu je končna ocena predmeta zvišala za 1 oceno. Namesto

```
/*----- Tekstovne funkcije in metode
Kako bi iz e-poštnega naslova ime.priimek@nekPonudnik.domena
dobili ime in priimek te osebe v dveh tekstovnih spremenljivkah?
Na primer iz začetnega e_mail="janez.novak@fov.uni-mb.si" naredite
ime = "Janez"
priimek = "Novak"
V pomoč vam je dejstvo, da je med imenom in priimkom vedno znak ".", ter da
se sklop "ime.priimek" vedno zaključí z znakom "@". V rešitvi lahko uporabite
več korakov in vmesne spremenljivke. Ne pozabite tudi na velike začetnice.
Rešitev, ki deluje samo za ime in priimek, dolga 5 znakov, ni pravilna.
*/
```

Poštevanka do 10

	!	1	2	3	4	5	6	7	8	9	10
1	!	1	2	3	4	5	6	7	8	9	10
2	!	2	4	6	8	10	12	14	16	18	20
3	!	3	6	9	12	15	18	21	24	27	30
4	!	4	8	12	16	20	24	28	32	36	40
5	!	5	10	15	20	25	30	35	40	45	50
6	!	6	12	18	24	30	36	42	48	54	60
7	!	7	14	21	28	35	42	49	56	63	70
8	!	8	16	24	32	40	48	56	64	72	80
9	!	9	18	27	36	45	54	63	72	81	90
10	!	10	20	30	40	50	60	70	80	90	100

Spletna stran naj z uporabo FOR zanke izpiše poštevanko do 10

Poštevanka do 10

← Header

	!	1	2	3	4	5	6	7	8	9	10
1	!	1									
2	!	2	4								
3	!	3	6	9							
4	!	4	8	12	16						
5	!	5	10	15	20	25					
6	!	6	12	18	24	30	36				
7	!	7	14	21	28	35	42	49			
8	!	8	16	24	32	40	48	56	64		
9	!	9	18	27	36	45	54	63	72	81	
10	!	10	20	30	40	50	60	70	80	90	100

Jedro naloge
(poštevanka)



Stranski stolpec

Za 3 točke poravnajte izpisana števila v stolpce in izpišite poštevanko brez ponavljanj števil

Slika 1: Primer e-naloge.

8, ki jo je »prislužil« na izpitu, je dobil 9. Predstavljeni model spodbude smo uporabili pri drugi generaciji študentov, ki so imeli e-študij. Izkazal se je za sicer uporabnega, vendar je bilo preveč enostavno priti do dodatne točke pri končni oceni predmeta. Zato smo ga delno spremenili.

Sedaj že več let uporabljamo 3-točkovno lestvico. Prvo in obvezno točko dobi študent, ko odda sprejemljivo nalogo. Drugo dobi za pravočasno oddajo, tretjo pa, če izpolni dodatne, natančno opredeljene zahteve. Primer take naloge je prikazan na sliki 1. Osnovna naloga je, da študent napiše kodo, ki bo izpisala tabelo množenja do 10.

Precej časa smo prihranili, ko smo način pregledovanja in ocenjevanja nalog prilagodili postopku, ki ga podpira Moodle. Na začetku smo pregledovali e-naloge in dajali ocene v modulu za ocenjevanje, toda v eni od Moodlovih posodobitev prikaza oddanih e-nalog ni bilo več mogoče urediti po datumu oddanih e-nalog. To je povzročilo preveč zamude pri iskanju naknadno oddanih nalog, zato smo ta način pregledovanja in ocenjevanja hitro nadomestili z modulom *Ogled vseh oddaj*. V tem modulu vidimo in ocenimo več e-nalog naenkrat, lahko pa ga tudi do neke mere prilagodimo našim potrebam, kot je prikazano na sliki 2.

Slika 2 je bila narejena na širini ekrana 1920 točk, a smo morali prilagoditi postavitev stolpcev, da imamo na enem ekranu vidne vse podatke, ki jih potrebujemo za ocenjevanje. V stolpcih, ki so skriti, bi bili sicer prikazani: slika uporabnika (študenta), ime in priimek, številka ID, naslov e-pošte in stanje oddaje, vendar bi bil zato stolpec *Komentarji odziva*, ki ga

uporabljamo za sporočila študentom, izven vidnega področja okna. Viden bi postal šele z dodatnim premikanjem levo-desno, za kar pa potrebujemo miško z dodatno funkcionalnostjo srednjega gumba.

Postopek pregledovanja in ocenjevanja e-naloga se začne, ko študent prilepi (»paste«) kodo naloge kot navadno besedilo v okvir za oddajo naloge. Prednost take oddaje pred oddajo datoteke je, da je za ocenjevalce vsebina naloge »oddaljena« samo en klik. Ocenjevalci besedilo izberemo in »prilepimo« v pripravljeno ogrodje programa, pregledamo kodo in jo poskusimo izvesti. Na podlagi rezultata nato damo ali oceno ali pa pripombe (komentarje) na nalogo. Pripombam na vrhu dodamo datum in čas nastanka pripombe, ki ga uporabimo za preverjanje, ali je študent nalogo dopolnil. Resda bi za to lahko uporabili tudi stolpec *Komentarji oddaje*, kamor se samodejno zapiše čas oddaje komentarja, toda vsebina stolpca je privzeto skrita in zahteva dodaten klik za razkritje, komentarji pa so prikazani po naraščajočem zaporedju (zadnji je vedno na dnu), kar zahteva še dodatno pomikanje vsebine okna. Držimo se tudi pravila, da študentu ne damo ocene, dokler njegova naloga ni sprejemljiva. Tako se izognemo nekaj klikom, ki so potrebni za podaljšanje roka za oddajo naloge, ki se privzeto zapre po ocenitvi.

Seveda smo uvedli tudi druge spremembe v izvedbi predavanj in vaj. Kot vsebinsko spremembo smo na vseh kontaktnih predavanjih in vajah začeli **močno poudarjati pomen rednega in sprotnega dela**. Takoj na začetku študentom povemo, da bodo morali začetniki pri programiranju za pozitivno oceno vložiti nekaj deset ur v tipkanje in preskušanje

The screenshot shows a Moodle submission interface. At the top, there are columns for 'Izberi', 'Ocena', 'Uredi', 'Besedilo preko spleta', 'Komentarji oddaje', 'Zadnja sprememba (ocene)', and 'Komentarji odziva'. The 'Izberi' column has a '+' sign. The 'Ocena' column shows '2,0 / 3,0'. The 'Uredi' column has a dropdown menu. The 'Besedilo preko spleta' column shows the submission content, including HTML tags like '<!DOCTYPE html>' and '<head>'. The 'Komentarji oddaje' column shows 'Komentarji (0)'. The 'Zadnja sprememba (ocene)' column shows a '-' sign. The 'Komentarji odziva' column shows a date '22.1. 22:25' and the text 'Preveč delov naloge ste iznutili. Prosim dokončajte'. Annotations with arrows point to various elements: 'Prikaz urejen po datumu' points to the sort order; 'Skriti stolpci' points to the 'Izberi' column; 'Dostop do besedila naloge' points to the submission content; 'Ni ocene, dokler naloga ni soriemliiva' points to the 'Ocena' column; 'Študent naloge še ni dopolnil' points to the submission date; and 'Zadnja sprememba (oddaje)' points to the submission time.

Slika 2: Prilagojeno okno za ocenjevanje vseh oddaj e-nalog

delovanja programske kode, ker je to edini znani način, kako se naučiti samostojnega programiranja. V prvi polovici predavanj ali vaj jim to tudi stalno omenjamo. Prvi dan običajno predstavimo tudi najslabši scenarij, ki se pogosto zgodi študentom, ki »odložijo« predmet. Čaka jih intenziven študij v avgustu, s plačevanjem inštruktorja, udeležbo na zadnjem izpitnem roku v študijskem letu in, za okrog polovico takih študentov, ponavljanje letnika. Uspešnost takega pristopa je zaradi relativno nizkega števila udeleženi študentov (8–15 študentov) težko točno izmeriti. V zadnjih nekaj letih sta na preskusnem izpitnem roku (neuradni izpit, ki ga pišejo takoj ob zaključku predmeta, rezultat pa se upošteva pri uradnem izpitnem roku samo v primeru pozitivne ocene) običajno prisotni okrog dve tretjini študentov, kar je opazen napredek glede na eno tretjino iz obdobja pred e-študijem.

Kot organizacijsko prilagoditev smo uvedli **vzporredno izvajanje predavanj in vaj**: Po dveh ali treh dneh predavanj (spoznavanja z novo snovjo) sledita dan ali dva vaj (utrjevanje nove snovi). Tako lahko študenti, ki so npr. manjkali na predavanju ali snovi na predavanju niso dobro razumeli, na vajah še vedno usvojijo uporabo nove snovi. S tem odpadejo težave zaradi neznane snovi pri nadaljnjih predavanjih.

Predavateljem veliko časa prihrani uporaba **preprečevanja oddaje nalog** (zaklepanje), ko se predmet izteče in je bil izveden prvi izpitni rok. Ker študenti – zamudniki ne morejo več normalno oddati zahtevanih e-nalog, stopijo v stik s predavateljem ali asistentom. S tem pa ga opozorijo nase in zato asistentu ali predavatelju ni potrebno dva tedna pred vsakim izpitnim rokom več dni zapored pregledovati oddaje e-nalog za predmete, ki so se zaključili že pred več meseci. Moodle sicer lahko nastavimo, da ob vsaki oddaji e-naloga dobi predavatelj ali asistent e-poštno sporočilo, toda v praksi se je izkazalo, da to močno obremeni e-poštni predal in lastnika predala, zato smo to možnost izključili. Večino močno zamujenih e-nalog tako študenti – zamudniki opravijo s pošiljanjem nalog preko e-pošte, kar je za izvajalca bolj priročno kot globoko v Moodlevih menijih odklepati za vsakega zamudnika vsako e-nalogo posebej.

Omeniti velja tudi, da smo v študijskem letu 2017–18 za nekatere predmete **zmanjšali število ur** e-vaj in e-predavanj in povečali število kontaktnih ur. Razlog za to najbolje ilustrira primer odlične študentke, ki je dosegala nadpovprečne rezultate pri vseh predmetih, bila uvrščena na dekanovo listo najuspešnejših študentov, bila vedno prisotna na predavanjih in vajah, a je kljub temu za nalogo, planirano za eno šolsko uro, zaradi neobičajne tipkarske napake porabila

celih 5 ur samostojnega dela. Če bi isto nalogo delala v klasični predavalnici, bi ji predavatelj pomagal v nekaj sekundah. Tukaj se kaže, kako so predmeti, ki zahtevajo logično razmišljanje, povezavo na predhodna znanja in sklepanje (matematika, statistika, računalniško programiranje ...), manj primerni za e-način učenja kot predmeti, ki zahtevajo pomnjenje dejstev, primerjanje, analizo, sintezo in razvrščane. Tukaj pričakujemo kritiko bralcev s pripombo, da to ne drži, saj je večina tečajev programiranja narejena v spletnih verzijah s primeri. Res je, tudi mi študentom kot dopolnilno gradivo predlagamo uporabo spletnih učilnic, kot je [www3school](http://www3school.com), kjer najdejo razlage in primere osnovnih ukazov. Opazili pa smo, da študenti pristopajo k problemom na parcialen način, to pomeni, da ne preučijo snovi od začetka do konca, ampak poiščejo samo rešitev za trenuten problem, ostalo pa preskočijo. Tako pogosto v e-nalogah dobimo sicer delujoče rešitve, ki pa uporabljajo metode in kodo, ki je študentom sploh nismo predstavili. Študenti je običajno niti ne zanjo razložiti, ker so jo samo prekopirali s spleta. Seveda to postavlja pod vprašaj, ali so se s takim načinom kopiranja kode sploh kaj naučili o uporabi te kode. Težava je tudi v tem, da večina spletnih učilnic brezplačno ponuja samo osnove, ki pogosto niso dovolj za naše potrebe. Za rešitve višjega nivoja pa je pri mnogih spletnih učilnicah potrebno plačilo.

Značilnost učenja programiranja je nadgradnja in povezovanje predhodnih ukazov z novimi in tudi iskanje in odpravljanje napak. Zato smo v predavanju uvedli vprašanja v obliki kratkih praktičnih nalog, v katerih študenti takoj preverijo uporabo in delovanje predstavljene snovi. Študent, ki je prisoten na predavanjih, te naloge reši tekom predavanj, odsoten študent pa mora zamujeno opraviti sam, za kar potrebuje bistveno več časa, kot če bi se udeležil predavanj in vaj. Rešitve teh kratkih nalog študent odda v spletni učilnici predmeta. Oddaja hkrati služi kot »seznam prisotnosti« za tisti dan predavanj ali vaj, saj Moodle takoj označi naloge, ki niso bile oddane pravočasno, v tem primeru do konca predavanj.

Druga prednost sprotnih nalog je, da se študent spozna z napakami v kodi in njihovim odpravljanjem. Da bi študenti čim prej spoznali večino možnih napak začetnikov, smo uvedli način predavanj s sodelovanjem študentov za katedrom. Prostovoljec (zanimivo, da se študenti sami radi javijo) rešuje nalogo za katedrom s projekcijo svojega dela na platno,

predavatelj pa sledi reševanju in hkrati pomaga ostalim študentom v predavalnici. Na ta način se tudi pokažejo napake, ki jih predavatelj zaradi praktičnih izkušenj ne dela, a včasih vzamejo več deset minut predavanj, preden so skupaj odpravljene. Dodatno tudi spodbujamo boljše študente, ki so svojo nalogo že zaključili, da kot demonstratorji pomagajo sošolcem pri pisanju kode in odpravljanju napak in na tak način neposredno izvajajo učenje med seboj.

Kot dejavnik (ne)uspešne uporabe e-učenja se je pokazala tudi kultura študentov. Precej študentov pristopa k opravljanju študija po pragmatičnem načelu: do cilja z najmanj napora. Ena od »metod« tega načina študija je kopiranje rešitev v stilu »samo da nekaj oddam«. Ta dejanja ne le da povzročajo luknje v znanju študentov, ampak tudi nepotrebno dodatno delo za predavatelje, saj morajo ob vsebini preverjati tudi **plagiatorstvo**. Še posebej je to vidno pri e-nalogah, kjer vsi študenti doma rešujejo isto nalogo in imajo čas, da poiščejo rešitve prejšnjih generacij ali dobijo rešitev od boljšega študenta. Pretekle rešitve enostavno onemogočimo z delnim spreminjanjem besedila naloge (npr. namesto »izpišite najboljše tri ...« zahtevamo »izpišite najboljše štiri ...«), za tekoče pa imamo v okviru univerze na strežniku za e-študij nameščen vtičnik *Detektor podobnih vsebin 2.0*, ki v nekaterih primerih pomaga, pri oddaji računalniške kode pa pogosto ne. Glavna težava je, ker študenti v besedilu e-naloge že dobijo dele kode ali pripravljene podatke, detektor pa to označi kot plagiat. Tako se dejanski plagiat skrijejo v množici lažnih zadetkov.

Da bi olajšali proces učenja, študentom omogočamo uporabo povzetkov ukazov, ki vsebujejo vse ukaze, potrebne za izdelavo izpitne naloge. Povzetki niso zapisani kot rešitev, temveč kot sestavni deli kode, ki jo mora študent ustrezno združiti. Kljub opozorilom, naj takoj začnejo uporabljati povzetek ukazov, študenti z nizko motivacijo to naredijo šele tik pred izpitom. Tako jim povzetek ukazov ne pomaga veliko, ker zaradi pomanjkanja praktičnih izkušenj, ki bi jih pridobili z rednim delom, enostavno ne znajo teh delov kode smiselno uporabiti.

6 OBVLADOVANJE VELIKEGA ŠTEVILA UDELEŽENCEV

Veliko število pomeni več deset študentov, ki v istem dnevu oddajo svoje e-naloge, ki jih je treba pregledati in oceniti do naslednje izvedbe vaj ali predavanj. Tu moramo ločiti naloge glede na zvrst:

- informativne, v katerih študenti samostojno pre-delajo določene vsebine,
- formativne, v katerih študenti usvojijo izbrane veščine, in
- problemske, kjer morajo združiti zanje in veščine pri reševanju zastavljenega problema.

Moodle ponuja pester nabor gradnikov in orodij za razvoj **informativnih** nalog pa tudi za samodejno preverjanje kakovosti informacij, ki naj bi jih študenti z nalogo usvojili. Ta zvrst nalog tako potrebuje veliko začetnega vloženega dela: od oblikovanja vsebine za vsako nalogo, prenosa te vsebine z Moodlovimi gradniki v e-nalogo do zasnove, oblikovanja in izdelave testnih nalog in samodejnega točkovanja. Ko pa je taka e-naloga na Moodle pripravljena, potek njenega reševanja v celoti izvaja računalnik in tako postane število udeleženi študentov nepomembno. Predavatelj ne dela več na nalogi posameznega študenta, ampak le nadzoruje uspešnost študentov in v primeru odstopanj od pričakovanih rezultatov ali velikega števila vprašani študentov spremeni ali dopolni e-nalogo. Praktične izkušnje so pokazale, da se na tak način lahko nekatere predmete izvede skoraj v celoti, vključno z zaključnim izpitom.

Preverjanje pravih rešitev **formativnih** nalog je bolj zapleteno, saj so običajno odgovori opisni in je preverjanje pravilnosti odgovorov težko realizirati z razpoložljivimi Moodlovimi orodji. Najbolj zahtevne za preverjanje pa so **problemske** naloge, ki so pri učenju programiranja najbolj zastopane. V Moodle nimamo orodij, ki bi preverjala pravilnost oddane programske kode, zato je vse delo na strani predavateljev. Ko smo prvič oblikovali e-naloge, v katerih so morali študenti samostojno reševati zastavljene probleme in oddati delujočo kodo rešitve, se je izkazalo, da že pri približno 15 oddanih e-nalogah le-teh ni bilo mogoče kakovostno pregledati v predvidenem času treh šolskih ur.

E-naloge so bile na začetku kopije nalog, ki so jih študenti delali v računalniški učilnici. Tam je predavatelj ali asistent sproti na računalniku vsakega študenta lahko spremljal njegovo delo in preverjal, ali je koda pravilna. Pri e-nalogah pa je pregledovanje potekalo na enem samem računalniku in te »paralelnosti« ni bilo. Prav tako se je pojavilo dodatno delo, ker je bilo vsako oddano e-nalogo v obliki stisnjene (»zip«) datoteke Visual Basic projekta potrebno prenesti s strežnika, prekopirati vsebino na disk predavateljevega računalnika in nato projekt odpreti z Visual Studijem.

Postopek je trajal manj kot minuto, toda ko to pomnožimo z nekaj deset nalogami, se je že tako predolg čas pregledovanja nalog opazno podaljšal.

Ko smo analizirali stanje, smo ugotovili, da moramo prilagoditi e-naloge. Naloga študentov je oddati delujočo kodo, mi pa smo e-nalogam poleg zahtev, kako naj koda deluje, dodali še zahtevo, da koda naredi podrobno opisan izpis. Primer take naloge je predstavljen na sliki 1 zgoraj. Logika je enostavna – če je koda napisana pravilno, bo tudi izpis rezultatov njenega delovanja pravilen. Tako smo za osnovno »pregledovanje« uporabili računalnik – če je izpis delovanja bil tak, kot je bilo zahtevano, nam ni bilo več treba podrobno pregledovati kode vrstico za vrstico, ampak smo samo še pregledali ključne sklope kode. Če pa izpis ni bil zadosti podoben zahtevanemu, smo nalogo enostavno zavrnil, ne da bi pregledovali kodo. Na tak način smo uspeli skrajšati čas s približno 7 ur za 37 e-nalog na 3–4 ure, kar je bilo bistveno bližje predvidenim trem šolskim uram.

Zbrati več deset ljudi ob istem času na istem mestu je težava, s katero se vsak dan srečujemo v predavalnicah. Tudi pri tem smo si pomagali z uporabo sodobne IKT. Pri predmetu *Informatika in računalništvo* v prvem letniku že od leta 2004 uporabljamo lokalno verzijo Moodle (Leskovar, 2005) in od leta 2014 že prej omenjeni Moodle vtičnik **Big Blue Button** (BBB). Namestitev vtičnika je enostavna, za delovanje pa se potrebuje spletni brskalnik z nameščenim javanskim izvedbenim okoljem JRE 1.8. Moodle je nameščen lokalno na virtualnem računalniku s štirimi procesorji, štirimi GB pomnilnika, 100 GB prostora na disku in 64-bitnim Linux Ubuntu operacijskim sistemom. Strežnik je nekaj let star HP-jev strežnik Proliant DL380 G7 z 12 jedri, 64 GB pomnilnika, 2,5 TB velikim diskovnim poljem, VMware programsko opremo in 1 GB hitro omrežno povezavo. Na tej konfiguraciji je brez vidnih težav potekalo e-predavanje z neposrednim prenosom slike in zvoka ter deljenjem ekrana s 43 hkrati prijavljenimi študenti. Strojne zahteve so sicer nižje – večina začetnih testnih sej z do 16 uporabniki je potekala na običajnem osebem računalniku s 4-jedrnim procesorjem, 16 GB pomnilnika in 100 MB omrežne povezave. Na tej konfiguraciji smo izvedli tudi dejansko videokonferenco s 7 udeleženci.

Izvedba »videokonferenčnega« predavanja (seje) z BBB je precej podobna klasičnemu predavanju, le da ga študenti spremljajo preko računalnika. Zato je pred začetkom predavanja treba poskrbeti za dobro

osvetlitev delovnega mesta predavatelja in primerno ozadje. Administrator Moodla zažene sejo in predavatelju dodeli pravico moderatorja. Predavatelj mora pred začetkom javnega dela seje v BBB naložiti potrebne dokumente (podprta je večina oblik, kot so pdf, doc, ppt ...) in v primeru, da pri izvedbi predavanja potrebuje tudi tablo, le-to nadomestiti z deljenjem dela ekrana svojega računalnika s študenti. Dobra rešitev je, da je deljeni del ekrana na istem mestu in iste velikosti, ko je okno BBB za prikazovanje naloženih dokumentov. Priporočljivo je tudi, da ima na svojem računalniku že vnaprej pripravljeno besedilo vprašanj, ki jih namerava zastaviti študentom tekom predavanja, in da že zažene aplikacije ali programe, ki jih namerava uporabiti med predavanjem. Na začetku prve seje naj bi bil poleg predavatelja prisoten tudi administrator Moodla, da pomaga študentom, ki imajo težave pri prijavi na sejo. Ker so se te težave vedno pojavile, se je prvo predavanje običajno začelo z 10- do 30-minutno zamudo.

Ko se predavanje (javni del seje) začne, predavatelj takoj prosi študente, da v BBB onemogočijo svoje mikrofone in da naj za komunikacijo uporabljajo le vgrajeno okno za klepet. S tem preprečimo zvočne motnje, ki običajno nastanejo na strani študentov. Med predavanjem predvaja prosojnice in ostale pripravljene dokumente, ki jih lahko tudi označuje ali po njih riše z vgrajenimi orodji. Ko potrebuje prikaz računalniškega ekrana (npr. za prikaz delovanja aplikacije), potegne okno te aplikacije v področje, ki ga s študenti deli na svojem ekranu. Za sprotno preverjanje razumevanja predavanja (in ugotavljanje dejanske prisotnosti študentov pred njihovimi ekran) prilepiti (»paste«) prej pripravljena besedila vprašanj in odgovorov v modul kviz, ki ga z dvema klikoma postavi v klepetalnico. Ker Moodle beleži vse dejavnosti študentov v vgrajeno bazo podatkov, lahko te podatke uporabimo za različne statistike (Leskovar in Baggia, 2015) ali celo za zbiranje točk za končno oceno.

Tak način izvedbe predavanj zahteva od predavatelja nekaj prilagajanja. Poleg nenavadnega občutka ob predavanju ekranu, mora stalno spremljati klepetalnico, da odgovori na vprašanja študentov ali dodatno pojasni snov. Paziti mora tudi na pravilno preklapanje (prestavljanje v deljeno območje ekrana) med okni aplikacij, katerih vsebino predstavlja študentom.

Prednosti take izvedbe e-predavanja so predvsem na strani predavatelja. Na tak način lahko hitro predstavi nove vsebine, za katere še nima pripravljenih

modulov na Moodlu. Ker je snov študentom predstavljena v obliki predavanja in ne e-naloge, mu ni treba preverjati odgovorov študentov na e-naloge, kar pri nekaj 10 študentih zahteva nesorazmerno veliko časa. Na študentski strani cenijo tako obliko predavanj predvsem študenti iz oddaljenih krajev in tisti, ki zjutraj težko vstanejo. Najbolje opiše njihovo mnenje anekdota, ki smo jo doživeli na enem prvih predavanj, ko je študent pri vklopljenem mikrofону nekemu navdušeno razlagal: »Pet do osmih vstanem, si skuham kavo, vklopim računalnik in poslušam predavanje.« Navsezadnje imajo e-predavanja tudi ekološko prednost, saj prihranijo nekaj deset litrov goriva, ko študentom ni treba osebno priti na predavanja.

7 KLASIČNA V PRIMERJAVI Z E-PREDAVANJI

Sedaj po večletnih izkušnjah lahko z gotovostjo potrdimo, da je s strani izvajalcev za e-predavanja potrebno več dela in več časa kot za klasična predavanja, kar pa klub večjemu vloženemu delu ne pomeni boljšega uspeha med študenti (Daymont & Blau, 2008; William T. Kaufman, 2015). Za e-predavanja moramo najprej proučiti teorijo, izbrati neko poglavje in ga v strnjeni obliki podati v elektronski obliki kot besedilo, kodo, primere, rešitve, povezave, filme ali slike. Dodatno moramo pripraviti e-učilnico, naložiti ustrezne dokumente, predstavitev, primere in filme na ustrezna mesta in jih časovno ustrezno odpreti študentom. Sledi zbiranje rezultatov študentov in njihova ocena. Na koncu predmeta je potrebno za vsakega študenta zbrati rezultate za vsa predavanja in vaje in na osnovi tega ovrednotiti njihovo delo. Ker se skoraj nikoli ne zgodi, da bi vsi študenti v roku oddali svoje e-naloge, lahko vsak dan posebej preverjamo, če je kak od zamudnikov oddal naloge. Dodatno moramo odgovarjati na vprašanja, ki jih študenti postavljajo med izdelavo e-predavanj ali vaj. Za iskanje ene napake lahko porabimo tudi več ur kljub izkušnjam v programiranju. Dodaten problem povzroča razlika med posameznimi generacijami študentov. Redko se namreč zgodi, da druga generacija predela isto količino snovi kot predhodna, zato je potrebno že pripravljena gradiva za predavanja vsako leto prilagajati glede na hitrost sprejemanja znanj posamezne generacije. Kakor koli gledamo, je porabljen čas nesorazmeren z rezultati (Daymont & Blau, 2008) in glede na odzive študentov v anketah smo pri predmetu osnov programiranja iz 50 % prešli na 20 % e-predavanj in e-vaj.

Asinhron in sinhron način izvedbe e-predavanj in e-vaj je naslednje vprašanje, ki se nam je porajalo ob prehodu na e-študij. Sinhron način pomeni, da se v času e-predavanj izvaja neposredna komunikacija s študenti v obliki predavanj, informacij in navodil. Kljub temu da Moore ugotavlja pomen sprotnih komunikacij s študenti (Moore, 2014), samo en sodelavec prakticira tak način poučevanja in odziv med študenti ni najboljši, saj s tem izničimo bistveno prednost e-študija – časovno in krajevno neodvisnost. Če mora študent točno ob določenem času biti prijavljen in poslušati predavanja preko elektronske povezave, ne vidi prednosti pred klasičnimi predavanji s tem, da tak način predavanj spremlja še slabša slišnost, slaba vidljivost, možnost prekinitev in možnost motenja v domačem okolju. Drugačen je odziv med študenti, če imajo na razpolago komunikacijo s predavateljem, ko potrebujejo nasvet ali pomoč, tudi če to ni takrat, ko sami izvajajo e-naloge. Vprašanja študentov med in po predavanjih so največji pokazatelj o tem, ali študenti spremljajo in v kolikšni meri obvladajo podano snov. Zanimivo je, da študenti postavijo več vprašanj na klasičnih predavanjih kot na e-predavanjih, kljub temu da imajo možnost zastaviti vprašanje znotraj Moodlea le klik stran. Tudi če potrebujejo pomoč pri iskanju napake, se redki odločijo in pošljejo prošnjo za pomoč. Prednost klasičnih predavanj in vaj se v tem primeru pokaže tudi zaradi tega, ker ostali slišijo odgovor na postavljeno vprašanje, pri e-predavanjih pa je odgovor le za enega študenta.

Uporaba e-učenja za potrebe predštudija (Anderson, 2008) oziroma spoznavanja osnov se je pokazala kot odlična, a je žal odvisna od kulture študentov. Medtem ko je v nekaterih zahodnih državah običajno, da študenti predelajo uvodno snov, preden pridejo na predavanja, je ta kultura pri nas bolj izjema kot pravilo.

8 UGOTOVITVE

Kljub prednostim, ki jih e-študij omogoča študentom, se kaže, da v primerjavi s klasičnimi predavanji nima večjega uspeha, poleg tega pa še od študentov zahteva večjo samodisciplino in organiziranost (Daymont & Blau, 2008). Kot bolj optimalna rešitev se ponuja kombinirano učenje, ki pa samo po sebi še ne zagotavlja stimulatívne učnega okolja, zato se predlaga, da predavatelji v večji meri vključijo sodelovanje študentov in sodelovanje med njimi samimi

(Kenney & Newcombe, 2011). Večinoma te ugotovitve lahko potrdimo tudi iz naših izkušenj. Predmeti, ki zahtevajo intenzivno kombiniranje predhodnih znanj in logičnega razmišljanja – tipičen predstavnik je učenje računalniškega programiranja – so manj primerni za izvedbo v obliki e-študija. Vendar tudi tu obstaja veliko rešitev, ki olajšajo delo predavateljem in povečajo uspeh študentom: predavatelji neprestano vzpodbujamo študente za samostojno, redno in sprotno delo, izboljšali smo razporeditev tega dela s primernim kombiniranjem predavanj (nova snov) in vaj (utrjevanje snovi). Predavateljem smo olajšali pregledovanje e-nalog, ko smo za osnovno oceno pravilnosti delovanja oddane kode uporabili računalnik in ko smo prilagodili orodja za ocenjevanje (Moodlea) pri pregledovanju e-nalog, ali celo uporabili videokonferenco pri izvedbi e-predavanj.

Po več letih takega načina izvajanja predmetov osnove računalniškega programiranja smo z rezultati zadovoljni. Težko je sicer narediti zanesljivo primerjavo s prejšnjimi leti, ker je način dela in s tem tudi zbrani podatki precej drugačni. Če pa primerjamo stari način dela, ko je približno tretjina študentov zaključila predmet s kolokviji, s številom študentov, ki so v zadnjih nekaj letih pristopili k »preskusnemu izpitu«, je sedaj številka bistveno višja – vedno je nad 60 odstotki, v zadnjem študijskem letu smo dosegli celo 82 odstotkov. Najbolj razveseljivo pa je, da večina študentov ta preskusni izpit tudi uspešno opravi.

LITERATURA

- [1] Anderson, T. (2008). *The Theory and Practice of Online Learning (Fifth Prin)*. AU Press, Athabasca University. Pridobljeno od aupress@athabascau.ca
- [2] Big Blue Button. *Overview (pridobljeno jul. 2019) dosegljivo na spletnem naslovu <https://bigbluebutton.org/>*
- [3] Daymont, T., & Blau, G. (2008). *Student performance in online and traditional sections of an undergraduate management course*. Journal of Behavioral and Applied Management, 275–295. Pridobljeno od http://www.ibam.com/pubs/jbam/articles/vol9/no3/jbam_9_3_3.pdf?origin=publication_detail&sa=U&ei=8OdxU5W9l4eeyAS1-IDoCA&ved=0CCEQFjAB&usq=AFQjCNFZdwkwCvxuftcomZNP1wWZfNINCg
- [4] Gowda, R. S., & Suma, V. (2017). *A comparative analysis of traditional education system vs. e-Learning*. IEEE International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2017 - Proceedings, (Icimia), 567–571. <https://doi.org/10.1109/ICIMIA.2017.7975524>
- [5] Kenney, J., & Newcombe, E. (2011). *Adopting a blended learning approach: Challenges encountered and lessons learned in an action research study*. Journal of Asynchronous Learning Network, 15(1), 45–57. <https://doi.org/10.1016/j.sbspro.2014.01.992>

- [6] Leskovar, Robert. *Podpora izvajanja študija s paketom „Moodle“ - izkušnje in perspektive = Studies implementation support using the „moodle“ package - experience and perspectives*. V: KALUŽA, Jindřich (ur.). Sinergija metodologij : zbornik 24. mednarodne konference o razvoju organizacijskih znanosti, Slovenija, Portorož, 16.-18. marec 2005 = Synergy of methodologies : proceedings of the 24th International Conference on Organizational Science Development, Slovenia, Portorož, March 16-18, 2005. Kranj: Moderna organizacija. 2005, str. 550-557.
- [7] Leskovar, R. & Baggia, A. (2015). *Zločin in kazen: udeležba na avditornih in videokonferenčnih predavanjih ter uspešnost reševanja nalog*. V Bernik M. in Rajkovič U. (Ur.), Vzgoja in izobraževanje v informacijski družbi - VIVID 2015, Kranj: Fakulteta za organizacijske vede, str. 287-296.
- [8] Moodle, pregled (pridobljeno jul. 2019), https://docs.moodle.org/37/en/About_Moodle
- [9] Moore, J. (2014). *Effects of online interaction and instructor presence on students' satisfaction and success with online undergraduate public relations courses*. Journalism and Mass Communication Educator, 69(3), 271–288. <https://doi.org/10.1177/1077695814536398>
- [10] Nazarenko, A. L. (2015). *Blended Learning vs Traditional Learning: What Works? (A Case Study Research)*. Proceedings - Social and Behavioral Sciences, 200(October), 77–82. <https://doi.org/10.1016/j.sbspro.2015.08.018>
- [11] Siemens, G., Skrypnik, O., Joksimovic, S., Kovanovic, V., Dawson, S., & Gasevic, D. (2015). *The history and state of blended learning*. Preparing for the Digital University: A review of the history and current state of distance, blended, and online learning, 234. <https://doi.org/10.13140/RG.2.1.3515.8483>
- [12] Wang, G., Foucar-Szocki, D., Griffin, O., O'connor, C., & Scelford, E. (2003). *Departure, abandonment, and dropout of e-learning: Dilemma and solutions*. James Madison University, (October).
- [13] William T. Kaufman. (2015). *Traditional vs. Electronic Learning Environment. Education and Human Development Master's Theses*. Paper 537. Pridobljeno od http://digitalcommons.brockport.edu/cgi/viewcontent.cgi?article=1554&context=ehd_theses.

■

Iztok Bitenc je zaposlen kot predavatelj na Fakulteti za organizacijske vede Univerze v Mariboru na področju Informacijski sistemi, kjer sodeluje pri predmetih s področja osnov informatike, različnih programskih jezikov, razvoja spletnih rešitev in baz podatkov. Je avtor več strokovnih in znanstvenih člankov, raziskovalno pa se udejstvuje pri iskanju praktičnih rešitev z uporabo sodobne informacijske tehnologije.

■

Borut Werber je docent na Fakulteti za organizacijske vede Univerze v Mariboru na študijskem programu Organizacija in management informacijskih sistemov. Glavna področja njegovega raziskovanja so informacijski sistemi, IKT, podkožni mikročipi in informatika v mikro-podjetjih. Aktivno sodeluje kot evalvator NAKVIS-a pri akreditacijah in reakreditacijah visokošolskih študijskih programov in institucij v Sloveniji in tujini. Kot evalvator sodeluje tudi pri notranjih evalvacijah Univerze v Mariboru. Kot recenzent aktivno sodeluje na domačih in mednarodnih konferencah ter domačih in tujih publikacijah.

■

Marko Urh je višji predavatelj na Fakulteti za organizacijske vede Univerze v Mariboru, svojo strokovno in znanstveno pot je začel v Policiji kot razvijalec geografskih informacijskih sistemov. Glavna področja njegovega raziskovanja so informacijski sistemi, e-izobraževanje, management in razvoj kadrov. Objavil je več domačih in mednarodnih člankov in prispevkov z omenjenih področij. Aktivno sodeluje na domačih in mednarodnih konferencah ter je recenzent domačih in tujih publikacij.