

Upravljanje znanjem u malim softverskim preduzećima: Koncepti i praktična iskustva

Knowledge management in small software companies: Concepts and practical experiences

Željko Stojanov, Dalibor Dobrilović, Univerzitet u Novom Sadu, Tehnički fakultet "Mihajlo Pupin" Zrenjanin, Srbija
Tijana Dabić, Univerzitet Sinergija Bijeljina, Bosna i Hercegovina

Sažetak – Softverska industrija je bazirana na znanju zaposlenih u organizacijama koje proizvode i održavaju softverske sisteme. Imajući to u vidu, ključni problemi u upravljanju znanjem u softverskim organizacijama su identifikacija, sistematizacija i očuvanje znanja s obzirom da softverski inženjeri veoma lako mogu da promene posao. Upravljanje znanjem koje postaje dostupno zaposlenima u softverskim organizacijama postaje ozbiljan izazov. Mala softverska preduzeća, koja imaju ograničene resurse, posebno su izložena problemu efikasnog upravljanja znanjem koje obezbeđuje očuvanje znanja u preduzećima s obzirom na česte promene u kadru koji radi na poslovima razvoja i održavanja softvera. U radu su prikazani osnovni koncepti i problemi u upravljanja znanjem u malim softverskim preduzećima, kao i praktična iskustva autora u malom lokalnom softverskom preduzeću. Takođe su navedene implikacije za softverska preduzeća, istraživače iz akademske zajednice, kao i smernice za dalja istraživanja.

Ključne riječi – Upravljanje znanjem; Softverska industrija; Mala softverska preduzeća; Naučene lekcije

Abstract – Software industry is based on knowledge of employees that work in organizations focused on developing and maintaining software systems. By having this in mind, the key problems in knowledge management in software organizations are identification, systematization and retention of knowledge within organizations since software engineers can easily change jobs. Management of knowledge that becomes available to all employees in software organizations becomes a serious challenge. Small software companies, with limited resources, are particularly exposed to the problem of managing and retaining knowledge due to the frequent changes in staff that develop and maintain software. This paper presents basic concepts and problems in knowledge management in small software companies, as well as practical experiences of the authors in a local small software company. Implications for software companies and researchers from academia are outlined, together with further research directions.

Keywords – Knowledge management; Software industry; Small software companies; Lessons learned

I. UVOD

Upravljanje znanjem je disciplina koja se bavi kreiranjem, primenom i širenjem znanja. Cilj upravljanja znanjem je da se implicitno znanje transformiše u eksplicitno, kao i da se eksplicitno znanje pojedinaca prenese na grupe u okviru organizacija. Preduslov za unapređenje upravljanja znanjem u okviru organizacije je da se razumeju tokovi znanja u organizaciji [1]. Ovi tokovi ukazuju na izvore novih ideja ili inicijativa, kao i na način njihovog širenja u ostatak organizacije. Takođe se na taj način mogu uočiti i delovi organizacije gde se tokovi znanja ne pojavljuju na očekivani način, što ukazuje na pravce unapređenja prakse. Efikasno upravljanje znanjem u okviru organizacije rezultuje organizacijom koja uči kroz svoju praksu.

Upravljanje znanjem ima korene u filozofiji, epistemologiji i drugim naučnim disciplinama, a sa značajnim razvojem softverskog inženjerstva kao discipline u poslednjih nekoliko decenija biva prepoznato kao jedan od ključnih elemenata unapređenja prakse [2]. Istovremeno, upravljanje znanjem u različitim oblastima softverskog inženjerstva postaje i predmet značajnog broja naučnih istraživanja širom sveta [3, 4, 5]. Znanje koje poseduju softverski inženjeri je danas ključni faktor i kapital koji poseduju softverske organizacije, što najbolje odslikava konstatacija da znanje kao intelektualni kapital ima noge i svaki dan ide kući (*"The major problem with intellectual capital is that it has legs and walks home every day"*)[6].

Softversko inženjerstvo je oblast koja se brzo menja i razvija, a bazirana je na poslovnim aktivnostima koje uključuju znanje velikog broja ljudi u različitim fazama životnog ciklusa softvera. Pored toga, softverski inženjeri moraju posedovati i znanja iz drugih oblasti, kao što su teorija upravljanja u organizaciji ili psihologija, pošto je prepoznato da je uticaj ljudskog i organizacionih faktora veoma važan za svakodnevnu praksu. Softverske organizacije zbog toga imaju problem da identifikuju potrebna znanja i da ih na adekvatan način primene u praksi. Upravljanje znanjem omogućuje organizacijama da na adekvatan način iskoriste svoj ljudski kapital [7]. Pored znanja koja su usko vezana za tehničke i tehnološke probleme u softverskom inženjerstvu, softverske

organizacije moraju efikasno upravljati sledećim aspektima znanja [6]:

- **Domenska znanja.** Odnose se na znanja u oblasti poslovanja ili života za koju se projektuje softver (npr. medicina ili poljoprivreda). Ako organizacija ne poseduje eksperte za te oblasti potrebno ih je angažovati kao spoljne saradnike.
- **Znanja o internoj organizaciji, kulturi i pravilima u softverskoj organizaciji.** Svaka organizacija ima pravila i procedure koje uređuju poslovanje i interne odnose u organizaciji. Ovakva znanja obično postoje kao "folklorni motivi" koji se samo spominju a retko sistematično koriste u praksi.
- **Identifikovanje znanja pojedinaca.** Softverske organizacije veoma zavise od znanja pojedinaca, pa je takvo znanje potrebno identifikovati i prevesti u oblik koji je trajno dostupan organizaciji. Ovo je posebno važno zbog toga što softverski eksperti mogu veoma lako promeniti posao.
- **Saradnja i deljenje znanja.** Aktivnosti u okviru razvoja i održavanja softvera su grupne i obično uključuju više ljudi koji mogu biti i geografski udaljeni. Efikasno deljenje znanja i saradnja se moraju ostvariti nezavisno od vremenske i prostorne razlike.

Upravljanje znanjem u softverskom inženjerstvu podrazumeva [2]: prikupljanje znanja, transformaciju znanja iz skrivenog (*tacit*) u eksplicitno znanje koje se lako sistematizuje, čuvanje, razmenu i evaluaciju, i primenu znanja u novim situacijama. Mathiasen & Pourkomeylian [8] ukazuju da se upravljanje znanjem najčešće pojavljuje kao važna komponenta inicijativa za unapređenje softverskih procesa (*Software Process Improvement, SPI*). Šta više, SPI projekti se oslanjaju na znanje koje postoji u organizaciji (iskustvo) i koje predstavlja osnovu za unapređenje prakse [9]. Ove inicijative su obično ciklične, što znači da podrazumevaju više ciklusa sa aktivnostima inicijacije unapređenja procesa, dijagnostifikovanja stanja, sprovođenja aktivnosti na unapređenju i potom učenja iz iskustva. Na taj način se kroz aktivnost učenja iz sopstvenog iskustva podstiče kultura upravljanja znanjem u organizaciji. Osnovna namena SPI projekata je da se pored unapređenja prakse obezbedi kreiranje generičkog i kodifikovanog znanja koje postaje dostupno softverskoj organizaciji za buduće projekte. Ipak, alati, tehnike i metodologije koje se koriste u softverskom inženjerstvu i dalje nisu adekvatni za efikasno upravljanje znanjem kao preduslovom za unapređenje prakse [10]. Navedena uvodna razmatranja su motivisala autore za realizaciju istraživanja u oblasti upravljanja znanjem u softverskim preduzećima.

II. UPRAVLJANJE ZNANJEM U MALIM SOFTVERSKIM PREDUZEĆIMA

Osnovna odlika malih preduzeća je da neformalno upravljaju znanjem u okviru uobičajenih poslovnih aktivnosti [11]. Takav pristup upravljanju znanjem, koje se vrlo često može posmatrati i kao zanemarivanje, je posledica nedostatka formalnih pristupa za identifikaciju, deljenje, čuvanje i upotrebu znanja koje postoji u okviru organizacije [12].

Sistematski pregled literature u oblasti upotrebe znanja u malim i srednjim preduzećima (MSP), prikazan u članku [13], ukazuje da su ovakvim preduzećima neophodni fleksibilni pristupi za razvoj resursa koji podržavaju upotrebu znanja, a koji uzimaju u obzir specifičnosti ovakvih preduzeća.

Mala softverska preduzeća predstavljaju značajan faktor u ekonomskom razvoju mnogih zemalja. Ipak, ova preduzeća imaju značajna ograničenja (pre svega finansijska i u ljudskim resursima) u svom poslovanju, zbog čega su veoma fleksibilna i brzo se prilagođavaju promenama na tržištu [14]. Rešenja za unapređenje prakse koja prihvataju mala softverska preduzeća se moraju realizovati u relativno kratkim vremenskim periodima, poželjno je da ne zahtevaju velike količine podataka i podrazumevaju kreativnost i improvizaciju [15]. Na osnovu rezultata studije čiji su ciljevi da istraži kako se upravlja znanjem u malim softverskim preduzećima, koji su kritični faktori za upravljanje znanjem i kako treba organizovati softverski tim za efikasnu upotrebu znanja [16], autori zaključuju da se upravljanje znanjem realizuje neformalno i na individualnom nivou, kao i da velika većina učesnika ima pozitivan stav prema upravljanju znanjem.

Upravljanje i upotreba znanja se u malim softverskim preduzećima koriste u različitim segmentima prakse. Najčešće se znanje prepoznaje kao ključni faktor u projektima procenjivanja i unapređenja procesa [16, 5, 17, 18]. Pored toga istraženi su aspekti upotrebe alata za deljenje iskustva i znanja u razvoju softvera [19, 9], domeni znanja potrebni za razvoj softvera i tehnologije potrebne za efikasnu upotrebu znanja u razvoju softvera [20], ili uticaj raspoloživog znanja na inovacije [21]. Uvid u dostupnu literaturu ukazuje da aspekt upravljanja znanjem u malim softverskim preduzećima nije dovoljno istražen, ali i da mala softverska preduzeća ne poklanjaju dovoljno pažnje upravljanju znanjem.

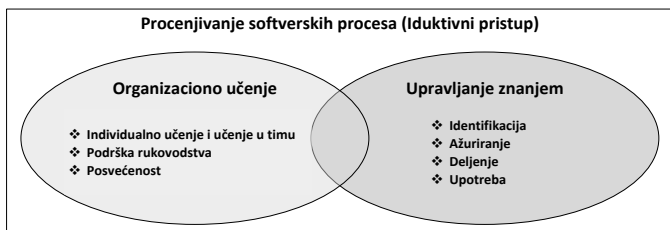
III. ISKUSTVA IZ PRAKSE

U okviru projekta "Razvoj softverskih alata za analizu i poboljšanje poslovnih procesa", koji je u periodu od 2011. do 2017. godine finansiran od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije, realizovano je istraživanje u okviru malog softverskog preduzeća sa ciljem procene i unapređenja procesa održavanja softvera. Kao segment tog projekta je realizovan podprojekat sistematizacije znanja koje se odnosi na proces održavanja softvera. Segmenti ovog istraživanja su delimično publikovani, a obuhvataju opis pristupa identifikaciji znanja u postupku procene procesa [18], diskusiju o primeni kvalitativnih metoda koje čine suštinu postupka identifikacije i sistematizacije znanja [22], prikaz induktivnih metoda u procenjivanju procesa koje su esencijalne za identifikovanje znanja o procesima [23], prikaz segmenta tematskog okvira sa znanjem o procesiranju zahteva za održavanjem softvera [24], i prikaz koncepta povratnih informacija (feedback) u procenjivanju softverskih procesa i identifikovanju znanja [25].

A. Identifikacije i sistematizacije znanja u malom softverskom preduzeću

Istraživanje je realizovano u malom softverskom preduzeću kao segment projekta unapređenja procesa održavanja softvera. Preduzeće proizvodi poslovne softverske aplikacije i

fokusirano je na klijente u Srbiji. Oblast održavanja softvera je odabrana za unapređenje pošto je najveći deo poslova u preduzeću posvećen održavanju softvera [18]. Pristup procenjivanju i unapređenju procesa je dizajniran zajedno sa zaposlenima preduzeća, i podrazumeva aktivno učešće zaposlenih koji poseduju potrebno znanje za identifikovanje aspekata procesa koje treba unaprediti. Na taj način je kroz projekat procenjivanja i unapređenja procesa podržano i organizaciono učenje koje je omogućilo identifikovanje znanja koje poseduju zaposleni [2, 26], a odnosi se na odabrani segment prakse (sl. 1).



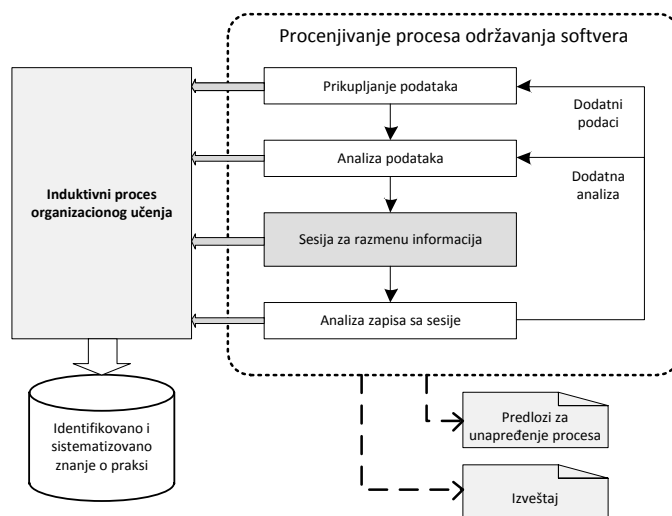
Sl. 1. Organizaciono učenje i upravljanje znanjem u okviru procenjivanja softverskih procesa

Organizaciono učenje i podrška za identifikaciju i sistematizaciju znanja (aktivnosti u okviru upravljanja znanjem) su omogućene primenom induktivnog pristupa u procenjivanju softverskih procesa [23], što je zapravo pristup odozdo-na-gore (*bottom-up*) i samim tim omogućuje identifikaciju znanja koje postoji u organizaciji (u glavama zaposlenih). Osnovne karakteristike ovog pristupa su:

- **Induktivan.** Istraživanje i identifikacija znanja polaze od stvarne prakse u organizaciji, čime se obezbeđuje da je identifikovano znanje relevantno za posmatrani kontekst.
- **Participativan.** Priprema i sprovođenje projekta podrazumevaju aktivno učestvovanje zaposlenih u organizaciji. Zaposleni učestvuju u svim aktivnostima od planiranja, prikupljanja i analize podataka, pa do validacije rezultata.
- **Učestala razmena povratnih informacija (frequent feedback).** Redovno dostavljanje povratnih informacija o toku i rezultatima projekta omogućuje proveru svih koraka i međurezultata dobijenih analizom podataka, što obezbeđuje utemeljenost identifikovanih znanja u stvarnoj praksi.
- **Triangulacija izvora podataka i metoda analize podataka.** Projekat je radi obezbeđivanja validnosti rezultata baziran na različitim izvorima informacija u okviru organizacije (triangulacija izvora podataka), a na osnovu toga su odabrane odgovarajuće metode analize podataka (triangulacija metoda za analizu podataka). Izvori podataka su: posmatranje prakse, polustrukturirani intervjui, dokumenti u preduzeću, repozitorijum sa evidentiranim poslovima (SQL baza podataka). U analizi podataka su korišćene i kvantitativne i kvalitativne metode.

Na ovaj način je uz potpunu posvećenost rukovodstva preduzeća i aktivno učešće svih zaposlenih obezbeđen pristup relevantnim podacima koji su poslužili za identifikaciju znanja o praksi. Pristup identifikovanju i sistematizaciji znanja kroz organizaciono učenje u okviru procenjivanja softverskih

procesa je prikazan na sl. 2. Sesije za razmenu informacija, realizovane kao radni sastanci u preduzeću, su ključni element ovog pristupa. Na sesijama su analizirani tekući rezultati i problemi, na osnovu čega se utvrđivalo da li je potrebno prikupiti nove podatke ili uraditi neku dodatnu analizu podataka. Sesijama su prisustvovali direktor preduzeća, vodeći istraživač (prvi autor rada), istraživači koji su bili uključeni u tekuću fazu istraživanja i zaposleni čije je prisustvo bilo potrebno da bi se ispravno protumačili podaci i identifikovalo korektno znanje vezano za praksu.



Sl. 2. Pristup za identifikaciju i sistematizaciju znanja u okviru procenjivanja softverskih procesa

Rezultat dvogodišnjeg rada na identifikaciji i sistematizaciji znanja je tematski okvir sa znanjima identifikovanim i sistematizovanim u odgovarajuće teme i pod-teme. Za razvoj tematskog okvira sa znanjem je upotrebljena metoda induktivne tematske analize (*inductive thematic analysis*) [27]. Tematska analiza spada u kvalitativne metode analize podataka i podrazumeva kreativno razmišljanje i zaključivanje prilikom izvođenja nalaza istraživanja iz podataka prikupljenih na terenu (u preduzeću) [28]. Za analizu je korišćen softverski alat MAXQDA, dok je razvoj tema i njihovo razvrstavanje u tematske oblasti radeno crtanjem na papiru. Nacrane šeme sa temama su korišćene kao materijal za analizu na sesijama. Zbog složenosti procesa održavanja softvera znanje je sistematizovano u tri tematske oblasti koje odlikavaju praksu održavanja softvera:

- **Ljudski faktor.** Ova tematska oblast uključuje teme koje se odnose na karakteristike programera u preduzeću i korisnika koji su i glavni akteri u procesu održavanja softvera. Korisnici prijavljuju zahtev za održavanjem, potom programeri kroz komunikaciju u preduzeću ili sa korisnikom rešavaju prijavljeni zahtev. Zbog kompleksnosti obrade zahteva za održavanjem softvera od presudnog značaja su iskustvo, kognitivne karakteristike i organizacione sposobnosti programera.
- **Poslovna politika i organizacija preduzeća.** Procesiranje zahteva podrazumeva upravljanje radom programera, upravljanje odnosima sa klijentima (ugovori o održavanju softvera, klasifikacija klijenata) i upravljanje softverskim

proizvodima (različiti klijenti mogu koristiti različite verzije softverskih proizvoda).

- **Procesiranje zahteva za održavanjem softvera.** Ovo je centralna tematska oblast koja se odnosi na tehničke i tehnološke aspekte održavanja softvera. Detaljna specifikacija procesa održavanja, kao i atributa samog zahteva za održavanjem su ključni za efikasnost održavanja softvera.

Ove tematske oblasti su međusobno povezane s obzirom da procesiranje zahteva za održavanjem u praksi realizuju programeri i njihove organizacione sposobnosti i iskustvo utiču na način rešavanja problema u procesiranju zahteva. Organizaciona politika preduzeća kroz odgovarajuću raspodelu odgovornosti i opterećenja programera takođe bitno utiče na procesiranje zahteva.

B. Naučene lekcije

Uobičajeni način učenja iz prakse predstavlja sumiranje iskustava i sagledavanje naučenih lekcija nakon realizacije projekata [29]. Lekcije koje su sagledane nakon realizacije projekta identifikacije i sistematizacije znanja o procesu održavanja softvera su:

- **Lekcija 1: Posvećenost rukovodstva organizacije.** Za uspeh projekta u oblasti unapređenja procesa baziranog na identifikaciji i sistematizaciji znanja je neophodna posvećenost rukovodstva koje jasno postavlja ciljeve projekta i obezbeđuje dostupnost resursa neophodnih za sprovođenje projekta.
- **Lekcija 2: Aktivno učešće zaposlenih.** Zaposleni su "nosioci" organizacionog znanja koje je najčešće skriveno i nestruktuirano, pa se njihovim uključenjem obezbeđuje pristup znanju koje oni primenjuju u svakodnevnoj praksi, a potom i korektno strukturiranje identifikovanog znanja.
- **Lekcija 3: Zajednički rad istraživača i zaposlenih.** Zaposleni su osnovni izvor znanja u organizaciji, dok istraživači poznaju različite metode istraživanja i mogu odabrati najpogodnije metode za posmatranu organizaciju, pa je zbog toga njihova saradnja neophodna u svim fazama projekta. Takođe, analiza podataka može da se odradi kao zajednički posao istraživača i zaposlenih, čime se povećava validnost rezultata istraživanja.
- **Lekcija 4: Višestruki izvori podataka.** Razumevanje prakse i identifikovanje znanja je moguće ako se prikupi što više podataka iz različitih izvora u odabranom okruženju. Na taj način se podaci međusobno dopunjuju, a analiza jedne grupe podataka može zahtevati prikupljanje novih podataka iz drugih izvora.
- **Lekcija 5: Multidisciplinarni pristup.** Znanje o praksi se nalazi u različitim izvorima - najčešće u glavama zaposlenih, ali se može naći i u dokumentima i elektronskim bazama podataka. Različiti izvori podataka podrazumevaju i različite metode za prikupljanje i analizu koje nisu samo iz domena inženjerskih disciplina već iz društvenih i humanističkih (npr. etnografsko posmatranje prakse ili induktivna tematska analiza podataka).

IV. IMPLIKACIJE

Empirijsko istraživanje realizovano u okviru projekta unapređenja procesa, sa podprojektom identifikacije i sistematizacije znanja, ima značajne implikacije kako za industrijsku praksu tako i za istraživače.

Softverski praktičari, menadžeri i softverske organizacije mogu pronaći uputstva kako da pristupe projektu unapređenja procesa i sistematizaciji znanja koje je obično skriveno u samoj organizaciji kroz saradnju sa akademskim institucijama.

Istraživači mogu pronaći uputstva kako da strukturiraju projekat u saradnji sa organizacijama iz privrede, i kako da kroz zajednički rad povećaju validnost rezultata istraživanja. Takođe, rezultati istraživanja se mogu iskoristiti kao polazna tačka za naredna istraživanja koja bi se detaljnije fokusirala na specifične segmente prakse (npr. istraživanje komunikacije sa korisnicima ili istraživanje automatizacije pojedinih faza u procesiranju zahteva za održavanjem softvera).

V. ZAKLJUČAK

Upravljanje znanjem dobija na sve većoj važnosti u softverskim preduzećima s obzirom na trendove razvoja tržišta i sve veće zahteve za radnicima koji poseduju specifična znanja. Problem sa pronalaženjem i očuvanjem kvalitetnih radnika u softverskim preduzećima još više naglašava potrebu da se znanjem efikasno upravlja u okviru samih organizacija. Prikazana studija i navedene naučene lekcije ukazuju da je moguće pristupiti ovom problemu kroz zajednički rad istraživača iz akademije i zaposlenih u preduzeću. Prikazani pristup je i ključni doprinos ovog rada, dok se naučene lekcije mogu iskoristiti kao putokazi kako pristupiti istraživanjima u ovoj oblasti ili generalno u istraživanju industrijske prakse.

Pravci budućih istraživanja se odnose na istraživanje ostalih segmenata prakse u odabranom softverskom preduzeću (npr. istraživanje oblasti testiranja softvera), kao i na primenu prikazanog pristupa u drugim softverskim preduzećima, čime bi se dobila dodatna potvrda o upotrebljivosti pristupa. Razvoj softverskih alata za specifične aspekte analize podataka, kao što je induktivna tematska analiza, ili za upravljanje sistematizovanim znanjem sa fokusom na evolutivnu prirodu znanja je takođe interesantan i izazovan pravac budućeg istraživanja.

ZAHVALNICE

Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije podržava ovo istraživanje u okviru projekta TR32044 "Razvoj softverskih alata za analizu i poboljšanje poslovnih procesa", 2011-2017.

LITERATURA

- [1] B. Hansen, K. Kautz (2004) Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organisations. In the Proceedings of the 11th European Conference on Software Process Improvement (EuroSPI 2004), pp 126-137. Trondheim, Norway, DOI: 10.1007/978-3-540-30181-3_12.
- [2] K. Schneider (2009) Experience and Knowledge Management in Software Engineering. Springer-Verlag Berlin Heidelberg, Germany DOI: 10.1007/978-3-540-95880-2.
- [3] F. O. Bjørnson, T. Dingsøyr (2008) Knowledge management in software engineering: A systematic review of studied concepts, findings and

- research methods used. *Information and Software Technology*, 50(11): 1055-1068. DOI: 10.1016/j.infsof.2008.03.006.
- [4] A. Aurum, F. Daneshgar, J. Ward (2008) Investigating Knowledge Management practices in software development organisations – An Australian experience. *Information and Software Technology*, 50(6): 511-533. DOI: 10.1016/j.infsof.2007.05.005.
- [5] E. Galvis-Lista, J. M. Sánchez-Torres (2013) A Critical Review of Knowledge Management in Software Process Reference Models. *Journal of Information Systems and Technology Management*, 10(2): 323-338. DOI: 10.4301/S1807-17752013000200008.
- [6] I. Rus, M. Lindvall (2002). Knowledge management in software engineering. *IEEE Software*, 19(3): 26-38. doi: 10.1109/MS.2002.1003450.
- [7] J. Kaljević, J. Đorđević Boljanović (2015) Primena informacionih tehnologija u fazi deljenja znanja u organizacijama. *Sinergija University International Scientific Conference (Sinergija 2015)*, pp. 26-30. Bijeljina, Bosna i Hercegovina. DOI: 10.7251/ZRSNG1501026K.
- [8] L. Mathiassen, P. Pourkomeylian (2003) Managing knowledge in a software organization. *Journal of Knowledge Management*, 7(2): 63-80. DOI: 10.1108/13673270310477298.
- [9] M. Ivarsson, T. Gorschek (2012) Tool support for disseminating and improving development practices. *Software Quality Journal*, 20(1): 173–199. DOI 10.1007/s11219-011-9139-6.
- [10] J. Ward, A. Aurum (2004) Knowledge management in software engineering - describing the process. In *Proceedings of the 15th Australian Software Engineering Conference (ASWEC 2004)*, pp. 137-146. Melbourne, Australia. DOI: 10.1109/ASWEC.2004.1290466.
- [11] V. Hutchinson, P. Quintas (2008) Do SMEs do Knowledge Management?: Or Simply Manage what they Know? *International Small Business Journal*, 26(2): 131-154, doi: 10.1177/0266242607086571.
- [12] M. B. Nunes, F. Annansingh, B. Eaglestone, R. Wakefield, (2006) Knowledge management issues in knowledge-intensive SMEs. *Journal of Documentation*, 62(1): 101-119. DOI: 10.1108/00220410610642075.
- [13] R. Thorpe, R. Holt, A. Macpherson, L. Pittaway (2005) Using knowledge within small and medium-sized firms: A systematic review of the evidence. *International Journal of Management Reviews*, 7(4): 257–281. DOI: 10.1111/j.1468-2370.2005.00116.x.
- [14] I. Richardson, C. G. von Wangenheim (2007) Why are Small Software Organizations Different? *IEEE Software*, 24(1): 18-22. DOI: 10.1109/MS.2007.12.
- [15] T. Dyba (2000) Improvisation in Small Software Organizations. *IEEE Software*, 17(5): 82-87. DOI=10.1109/52.877872.
- [16] S. Basri, R. V. O'Connor (2011) A Study of Knowledge Management Process Practices in Very Small Software Companies. *American Journal of Economics and Business Administration*, 3(4): 636-644.
- [17] R. O'Connor, S. Basri (2014) Understanding the role of knowledge management in software development: a case study in very small companies. *International Journal of Systems and Service-Oriented Engineering*, 4 (1): 39-52. DOI: 10.4018/ijssoe.2014010103.
- [18] Z. Stojanov, J. Stojanov. D. Dobrilovic (2015) Knowledge Discovery and Systematization through Thematic Analysis in Software Process Assessment Project. In the proceedings of IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY 2015), pp. 25-30. Subotica, Serbia. DOI: 10.1109/SISY.2015.7325405.
- [19] J. García, A. Amescua, M-I. Sánchez, L. Bermón (2011) Design guidelines for software processes knowledge repository development. *Information and Software Technology*, 53(8): 834-850. DOI:10.1016/j.infsof.2011.03.002.
- [20] M. Kumar, S. Paul, S. Tadisina (2005) Knowledge management practices in Indian software development companies: Findings from an exploratory study. *Asian Academy of Management Journal*, 10(1): 59–78.
- [21] J. L. Carlo, K. Lyytinen, G. Rose (2012) A Knowledge-Based Model of Radical Innovation in Small Software Firms. *MIS Quarterly*, 36(3): 865-895.
- [22] Z. Stojanov (2015) Qualitative research on practice in small software companies. In *Encyclopedia of Information Science and Technology*, Third Edition, Mehdi Khosrow-Pour (editor), pp. 650-658. IGI Global. Hershey, PA, USA. DOI: 10.4018/978-1-4666-5888-2.ch062.
- [23] Z. Stojanov (2016) Inductive Approaches in Software Process Assessment. In *Proceedings of the 6th International Conference on Applied Internet and Information Technologies (AIIT 2016)*, pp. I- XV. Bitola, Macedonia. DOI: 10.20544/AIIT2016.I01.
- [24] Z. Stojanov, J. Stojanov (2016) Exploring software maintenance process characteristics by using inductive thematic analysis. In *Proceedings of the 6th International Conference on Applied Internet and Information Technologies (AIIT 2016)*, pp. 9-17. Bitola, Macedonia. DOI: 10.20544/AIIT2016.02.
- [25] Z. Stojanov, D. Dobrilovic (2017) The Role of Feedback in Software Process Assessment. In *Encyclopedia of Information Science and Technology*, Fourth Edition, Mehdi Khosrow-Pour (editor), pp. 7514-7524. IGI Global. Hershey, PA, USA. DOI: 10.4018/978-1-5225-2255-3.ch654.
- [26] M. Ivarsson (2010) Experience driven software process assessment and improvement. PhD Dissertation. Department of Computer Science & Engineering, Chalmers University of Technology. Göteborg, Sweden.
- [27] V. Braun, V. Clarke (2006) Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77-101. DOI: 10.1191/1478088706qp063oa.
- [28] M. Q. Patton (2001) *Qualitative research and evaluation methods*, Third edition. SAGE Publications. Thousand Oaks, CA, USA.
- [29] A. Hartmann, A. Dorée (2015) Learning between projects: More than sending messages in bottles. *International Journal of Project Management*, 33(2): 341-351. DOI: 10.1016/j.ijproman.2014.07.006.