

■ Sistem za avtomatsko gradnjo baze znanja iz spletnih mest

¹Ambrož Stropnik, ²Milan Zorman

¹KiviCOM, d. o. o., Kidričeva 3 a, 2380 Slovenj Gradec

²Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, 2000 Maribor
ambroz@kivi.si; milan.zorman@uni-mb.si

Izvleček

Spletni portali vsebujejo ogromno podatkov in informacij, ki so v večini primerov zapisani v nestrukturirani obliki in tako najlažje berljivi. Za računalnik je branje in obdelava takšnih podatkov in informacij vse prej kot lahko opravilo. Ker danes težimo k vse večji avtomatiki pri obdelavi informacij in ker se ne moremo sprijazniti z dejstvom, da bi informacije, objavljene na spletu, ostale neobdelane, potencial za dostop do novih znanj pa neizkoriščen, smo razvili sistem za avtomatsko gradnjo baze znanja s spletnih mest. V članku predstavljamo predlagani sistem, ki s pomočjo vizualne analize portala oziroma algoritma VIPS zgradi bazo znanja, ki jo je mogoče uporabiti v različne namene. Na koncu predlagani sistem za avtomatsko gradnjo baze znanja predstavimo na realnem primeru podjetja Elektra energije, d. o. o., pri čemer z izbrane spletne strani podjetja izdelano bazo znanja tudi predstavimo in analiziramo.

Ključne besede: spletno rudarjenje, semantični splet, besedilno rudarjenje, gradnja ontologij.

Abstract

Automatic Construction of a Knowledge Base from Websites

Web portals contain a lot of information, mostly represented in unstructured form and as such intended for reading and comprehension by humans, while the processing of such information by computer is a very complicated task. Because of an increased need for automated information processing, raw and unexploited information as well as untapped knowledge is seen wasteful. For that reason we developed a system for the automatic construction of a knowledge base from websites. The paper presents the aforementioned system for knowledge base construction using the VIPS algorithm that can be used for many different purposes. The system is demonstrated through an implementation on the website of the Elektro energija enterprise. The paper closes with an analysis and evaluation of the resulting knowledge base constructed in the course of the experiment.

Key words: Web mining, Semantic web, Text mining, Ontology building.

1 UVOD

Svetovni splet je v zadnjih desetih letih doživel pravi razcvet. Vede in nevede ga uporabljamo pravzaprav povsod, na vsakem koraku, tako v prostem času kot na delovnem mestu. Najbolj pogosto ga uporabljamo za pridobivanje različnih informacij, saj lahko trdimo, da je postal eden izmed največjih in najpogostejše izkoriščanih tovrstnih virov. Informacije, ki jih iščemo prek svetovnega spleta, pridobivamo z različnih portalov (npr. novice, tv spored, cene električne energije). Največkrat iskano informacijo preberemo ali preprosto prenesemo datoteke, ki vsebujejo iskano informacijo (npr. predstavitevno gradivo, slike). Ne glede na način, kako pridemo do iskane informacije, je pomembno, da je večina podatkov na svetovnem spletu predstavljenih s pomočjo jezika HTML (angl. Hyper Text Markup Language), ki omogoča zelo dobro predstavitev podatkov in informacij in posledično atraktiven prikaz uporabniku prek

ustrezne programske opreme (npr. Internet Explorer). Z vidika računalnika pa so informacije neurejene in nestrukturirane in posledično težje obvladljive za računalniško obdelavo. Poleg velike količine informacij, dostopnih na spletu, ki so pravilne, se vmes najdejo navedbe, ki so zavajajoče ali nepravilne. Upoštevanje opcije morebitnega novega znanja, ki se zna ravno zaradi novosti znajti med nepravilnim, še dodatno oteži nalogo. A za uspešno ločevanje med dobrimi, koristnimi podatki in informacijami s spleta, med smetmi in novim znanjem, je treba najprej nestrukturirane podatke in informacije spraviti v obliko, ki bo z računalniško ali ročno obdelavo lažje obvladljiva.

V prispevku predstavljamo naš pogled na pridobivanje podatkov in informacij s svetovnega spleta in njihov zapis v obliko, ki temelji na sodobnih tehnolo-

gijah Web 3.0. V razdelku 2 predstavimo različne pristope obdelave podatkov in informacij s svetovnega spleta, ki so pomembno vplivali na razvoj sistema. Razdelek 3 je namenjen predstavitvi tehnologij, ki so bile uporabljene v sistemu za avtomatsko gradnjo baze znanja s spletnih mest. Jedro prispevka je razdelek 4, v katerem predstavimo razvito rešitev in eksperiment na konkretnem primeru.

2 SVETOVNI SPLET IN PRIDOBIVANJE PODATKOV IN INFORMACIJ S SVETOVNEGA SPLETA

Pridobivanje podatkov in informacij s svetovnega spleta tako postaja vedno bolj pomembno tudi za računalnike, saj lahko različne povezave (angl. Links) razkrivajo mnogo več informacij, kot jih uporabnik zazna pri uporabi svetovnega spleta. Na to je v preteklosti že opozarjal eden od gurujev svetovnega spleta Tim Barnes-Lee, ki je predstavil idejo o semantičnem spletu, ki ga nekateri imenujejo tudi Web 3.0. Vendar moramo obstoječe portale in pripadajoče spletne strani ustrezno obdelati, če želimo ustvariti povezave med podatki in informacijami na spletu, ki bodo preprosto berljive za stroj oz. računalnik. Obdelavo podatkov in informacij s svetovnega spleta v grobem delimo v dve skupini, in sicer rudarjenje struktur spletnih strani (angl. Web Structure Mining) ter rudarjenje vsebine spletnih strani (angl. Web Content Mining) (Liu, 2011), ki se med seboj z različnimi metodami tudi prekrivata. Med splošno znane metode spletnega rudarjenja spadata tudi zelo znana algoritma PageRank in HITS, ki ju uporabljajo predvsem spletni iskalniki za indeksiranje vsebine spletnih strani (Liu, 2011; Han & Kamber, 2006).

Različni raziskovalci so se z razvojem tehnologij s področja spletnega rudarjenja različno lotevali pridobivanja in obdelave podatkov in informacij s spleta. Nekateri so se osredinjali le na pridobivanje, medtem ko so drugi naredili korak naprej in so na podlagi pridobljenih podatkov in informacij delali tudi semantične povezave med njimi. Tako velja omeniti primere pristopov segmentiranja spletnih strani in gradnje baze znanja, po katerih smo se zgledovali.

1) Toledo-Alvarado idr. so predstavili metodo za avtomatsko gradnjo ontologij oz. baze znanja iz zbirke besedilnih dokumentov. Njihova procedura temelji izključno na tehnologijah podatkovnega rudarjenja. Njena posebnost je v tem, da se pri gradnji ne upira na zunanje vire oz. slovarje (Tole-

do-Alvarado, Guzman-Arenas, & Martinez-Luna, 2012).

- 2) Gunasundari in Karthikeyan sta predstavila svojo metodo za pridobivanje podatkov in informacij s svetovnega spleta na podlagi povezav. Metoda lahko odkrije vsebino strani (segment), ki je v skladu s številom ločil in razmerjem števila znakov, ki niso del povezave, in s številom znakov, ki so del povezave (Gunasundari & Karthikeyan, 2012).
- 3) Gawrysiak idr. so predstavili sistem Text-Onto-Miner (TOM), ki je v bistvu sistem za analizo naravnega jezika (angl. Natural Language Analysis System). V okviru tega so razvili in izvedli nekaj algoritmov ter pristopov za polavtomatsko gradnjo ontologij. Pri tem so se upirali na tehnologije besedilnega rudarjenja in metod procesiranja naravnega jezika (Gawrysiak, Protaziuk, & Rybinski, 2012).
- 4) Mehta idr. so opisali implementiran algoritem za pridobivanje semantične strukture spletnih dokumentov, ki temelji na algoritmu VIPS in klasifikatorju Naive Bayes. Ta ima samo nalogo združevanja segmentov, ki spadajo v isti razred (Mehta, Mitra, & Karnick, 2012).

3 METODE IN TEHNOLOGIJE ZA AVTOMATSKO GRADNJO BAZE ZNANJA

Gradnja baze znanje je opravilo, ki je večinoma rezervirano za eksperte oz. skrbnike baz znanja (angl. knowledge worker), ki na podlagi popisanega in lastnega znanja ustvarijo bazo znanja za področje, ki ga obvladujejo (Vasić, 2004). Če želimo zgraditi bazo znanja, moramo imeti predznanje ali podatke in informacije, ki bodo jedro baze znanja. Pri tem moramo opozoriti, da sami podatki in informacije niso dovolj in še ne pomenijo znanja. Treba jih je ustrezno povezati med seboj (Stropnik, 2006; Vasić, 2004). Šele povezani podatki so znanje, ki ga predstavimo na želeni način. Najpogostejše znanje predstavimo z zemljevidom znanja. Če povzamemo, potrebujemo za gradnjo baze znanja te korake (Vasić, 2004):

- pridobitev podatkov in informacij, na katerih bo temeljila baza znanja;
- definiranje ustreznih povezav med predhodno pridobljenimi podatki in informacijami, ki služijo za povezavo in so surovina za bazo znanja;
- zapis in predstavitev povezanih podatkov z ustreznim jezikom oz. notacijo.

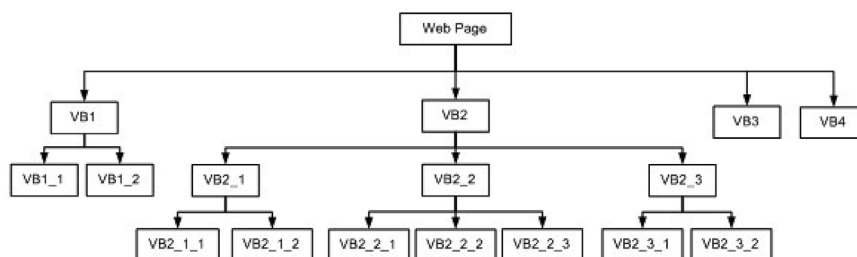
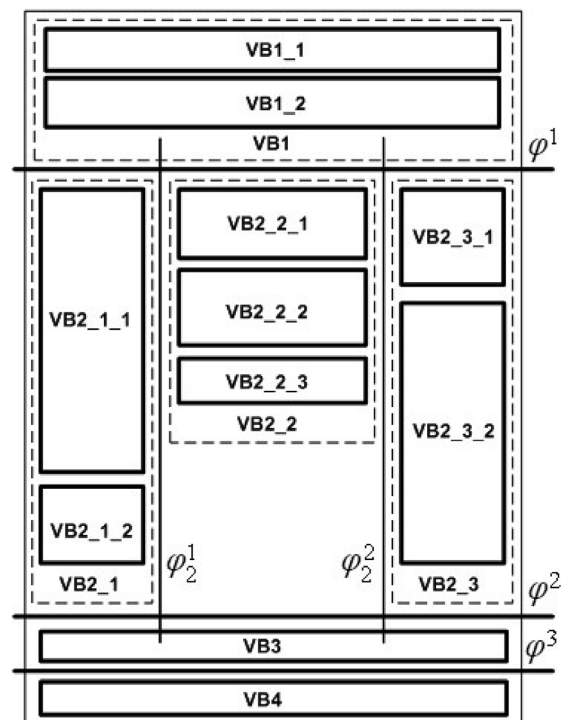
V nadaljevanju predstavimo postopek za vizualno analizo spletne strani, ki je eden od možnih načinov za pridobivanje podatkov in informacij s posamezne spletne strani, in sicer po principu deljenja strani na segmente.

3.1 Vizualna analiza spletnih strani

Spletne strani za svojo predstavitev vsebine uporabljajo jezik HTML, ki do določene mere strukturira podatke posamezne spletne strani. Osnovna struktura spletne strani je struktura DOM (angl. Document Object Model). To je drevesna struktura, pri kateri je vsaka značka HTML vozlišče v drevesni strukturi. Tako je spletna stran segmentirana z nekaterimi predefiniranimi strukturnimi značkami, ki vsebujejo pomembne informacije. Te značke so odstavek, tabela, list, glava itd. (Han & Kamber, 2006).

Na žalost je v praksi pridobivanje podatkov in informacij iz strukture spletne strani DOM zelo zapleteno opravilo, saj je jezik HTML po eni strani zelo fleksibilen, po drugi strani pa se lahko zgodi, da spletna stran ni zgrajena po standardu W3C HTML, zaradi česar ima nepravilno strukturo (Han & Kamber, 2006).

Ena izmed najbolj znanih in uspešnih metod na tem področju je algoritem VIPS (angl. Vision-Based Page Segmentation), ki so ga razvili in predstavili Microsoftovi raziskovalci (Deng, Shipeng, Ji-Rong, & Wei-Ying, 2003). Naloga algoritma je izluščiti semantično strukturo spletne strani glede na njen dizajn, in sicer tako kot bi storil uporabnik človek. Na sliki 1 je prikazan primer segmentacije spletne strani z uporabo metode VIPS (Deng, Shipeng, Ji-Rong, & Wei-Ying, 2003; Han & Kamber, 2006).



Slika 1: Prikaz delovanja algoritma VIPS (Vir: Deng, Shipeng, Ji-Rong, Wei-Ying, 2003)

Algoritem temelji na predpostavki, da je mogoče vsako spletno stran predstaviti kot t. i. trojko, ki je sestavljena iz množice območij oz. segmentov, množice ločevalnikov (vodoravne ali navpične linije, ki ločujejo segmente med seboj) in množice relacij med segmenti (Deng, Shipeng, Ji-Rong, & Wei-Ying, 2003; Han & Kamber, 2006). Omenjena predpostavka velja tudi za posamezni segment, ki ga obravnavamo kot podstran spletne strani.

Algoritem sestoji iz treh korakov: izluščevanje segmentov in ločevalnikov ter izdelava drevesne strukture. Postopek se izvaja rekurzivno za vsak segment, in sicer po principu od zgoraj navzdol. Tako algoritem spletno stran v prvem koraku razdeli na nekaj večjih segmentov, pri čemer se izluščijo ločevalniki in se rezultat doda v drevesno strukturo. V drugem koraku se postopek rekurzivno ponovi iz vsakega segmenta. Postopek se ponavlja tako dolgo, dokler posameznega segmenta ni več mogoče razdeliti na več podsegmentov (Deng, Shipeng, Ji-Rong, & Wei-Ying, 2003).

Končni rezultat algoritma je drevesna struktura segmentov, v kateri vsako vozlišče vsebuje t. i. vrednost DoC (angl. Degree of Coherence), ki določa skladnost vsebine segmenta in temelji na vizualni percepciji (Deng, Shipeng, Ji-Rong, & Wei-Ying, 2003).

3.2 Besedilno rudarjenje

Besedilno rudarjenje lahko označimo kot procesiranje nestrukturiranih besedilnih podatkov in informacij (npr. besedil), ki so zapisani v naravnem jeziku z namenom iskanja novih znanj (Han & Kamber, 2006). Pri iskanju novih znanj uporabljamo zelo podobne prijeme, tehnike in metode kot pri podatkovnem rudarjenju. Tudi sam proces rudarjenja je zelo podoben kot pri podatkovnem rudarjenju in sestoji iz enakih korakov: izbira/vzorčenje, čiščenje/predobdelava, podatkovno rudarjenje, cenitveni kriterij in predstavitev podatkov in informacij (Han & Kamber, 2006; Zorman idr., 2003). Omeniti velja, da je pri besedilnem rudarjenju zelo velik poudarek na pripravi vhodov za rudarjenje iz besedil. Od priprave je namreč zelo odvisen sam rezultat. Besedilno rudarjenje uporabljamo v več različnih opravilih, kot so (Brezovnik, 2009; Han & Kamber, 2006):

a) **iskanje informacij** (angl. information retrieval) – je eno od bolj pogostih opravil besedilnega rudarjenja in se uporablja pri različnih iskalnikih, pri čemer iskani niz predstavlja iskani vzorec besedila;

b) **klasifikacija besedil** (angl. text classification) – je opravilo, pri katerem besedila umeščamo glede na vsebino v vnaprej določene kategorije; primer uporabe klasifikacije besedil so filtri za nezaželeno pošto;

c) **razvrščanje v gruč** (angl. text clustering) – je oblika klasifikacije besedil, pri kateri kategorije niso znane vnaprej; primer uporabe razvrščanja v gruč je izpis iskalnih zadetkov;

č) **luščenje entitet in konceptov** (angl. entity extraction, concept extraction) – je opravilo, ki odkriva imenovane entitete iz besedil, medtem ko se luščenje konceptov ukvarja z odkrivanjem konceptov v besedilih;

d) **izdelava povzetkov** (angl. text summarization) – je opravilo, ki izdela krajše besedilo, ki opisuje dokument in vsebuje ključno vsebino iz dokumenta.

Poudariti moramo, da se besedilno rudarjenje ne ukvarja z razumevanjem besedil, temveč se s tem področjem ukvarja posebna veja znanosti za procesiranje naravnega jezika (angl. NLP – Natural Language Processing) (Brezovnik, 2009).

V okviru sistema za avtomatsko gradnjo baze znanja, ki ga predlagamo v tem prispevku, je besedilno rudarjenje nadgradnja oz. dopolnitev algoritma VIPs, pri katerem s pomočjo klasifikacijskega algoritma Naive Bayes določimo pomen posameznega segmenta.

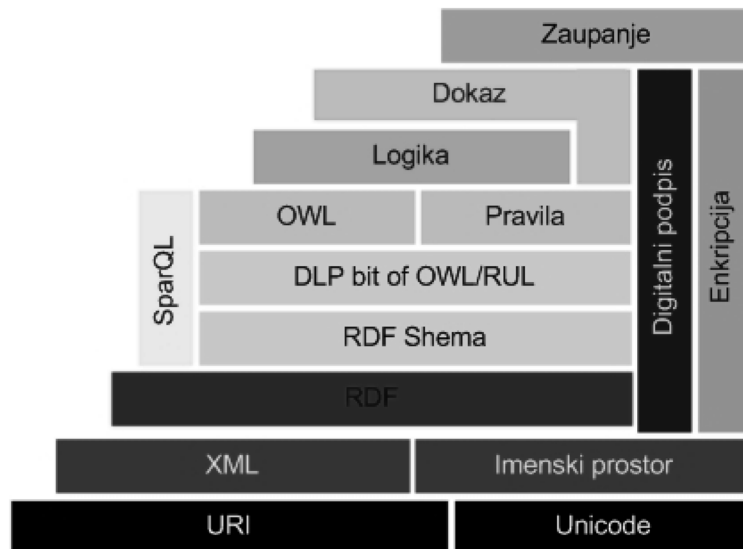
3.1 Semantični splet kot tehnologija za zapis baze znanja

Semantični splet ali Web 3.0, kot ga imenujejo nekateri, je nadgradnja obstoječega svetovnega spleta, pri kateri so spletnim informacijam pripisani nedvoumni, računalniško razumljivi metapodatki, ki bistveno olajšajo sodelovanje med ljudmi in računalniki (Leuf, 2006; Stropnik, 2006; Zarič 2004). Semantični splet definira takšno organiziranost in zapis podatkov, ki je razumljiva tako človeku kot programski opremi za potrebe razumevanja pomena podatkov in samodejnega sklepanja o pomenu in povezavah med posameznimi podatki (npr. elektronskimi dokumenti) (Stropnik, Vasič & Podgorelec, 2006).

V okviru semantičnega spleta je znanje predstavljeno v obliki semantičnih mrež, v katerih so koncepti predstavljeni z jezikom RDF (angl. Resource Description Framework), ki temelji na XML (angl. eXtensible Markup Language), in sicer s pomočjo URI-jev

(angl. Uniform Resource Identifier). Jezik RDF omogoča gradnjo podatkovnega modela za vire in relacije med njimi. Ker poenoteno poimenovanje stvari na podlagi URI-jev ni dovolj za uporabo baz znanja v inteligentnih programih, potrebujemo še dodatno orodje – ontologije (Stropnik, Vasić & Podgorelec, 2006). Ontologija definira pojem, uporabljen za opis in predstavitev področja znanja, ki ga uporabljajo lju-

dje, podatkovne baze in aplikacije, ki hočejo širiti domenske informacije (npr. medicina, popravilo avtomobilov). Zapišemo jih lahko z različnimi jeziki (npr. OIL, DAML+OIL), od katerih W3C priporoča ter najširše uporablja spletni ontološki jezik OWL (angl. Web Ontology Language) (Stropnik, 2006). Nabor tehnologij, ki tvorijo semantični splet, so predstavljene na sliki 2.



Slika 2: Nabor tehnologij semantičnega spleta (Vir: Stropnik, Vasić & Podgorelec, 2006)

Kot je razvidno iz diagrama, prvi dve ravni vzpostavljata skupno sintakso tehnologijam, pri katerih na najnižji ravni zasledimo enolični identifikator vira URI in standard Unicode za zapisovanje in izmenjavo znakov ali simbolov. Raven višje je jezik XML (angl. eXtensible Markup Language), ki definira notacijo za opisovanje metapodatkov. Besedišče dokumenta XML določa eden ali več imenskih prostorov (angl. Namespace). Tretja raven nabora tehnologij semantičnega spleta je jezik RDF, ki je namenjen opisovanju virov, pri čemer je vir opisan z identifikatorjem URI. Pomen je v dokumentih RDF zapisan v obliki trojic, ki se glede na elementarni stavek ujemajo s predstavitevijo osebek – povedek – predmet. Dokument RDF je predstavljen tudi z usmerjenim grafom, ki med seboj povezuje objekte dokumenta. Ontološki slovar je predstavljen s shemo RDF (angl. Resource Description Framework Schema Language) ali jezikom OWL (Stropnik, Vasić & Podgorelec, 2006).

Zaradi vseh omenjenih lastnosti je uporaba tehnolo-

logij semantičnega spleta zelo primerna rešitev za zapis in predstavitev baze znanja. Alternativne rešitve zapisa baze znanja predstavljajo specifični jeziki ekspertnih sistemov, kot so npr. CLIPS, Prolog idr.

4 KiviKM KOT PRAKTIČNI PRIMER REŠITVE

V okviru internega raziskovalnega projekta smo razvili platformo KiviKM (Kivi Knowledge Management) za podporo procesom upravljanja znanja (Stropnik, 2012). KiviKM smo razvili zaradi potreb naših strank, pri katerih je potreba po bolj kompleksnih iskalnikih vse večja, obstoječa orodja pa so premalo učinkovita. Primer takšne kompleksne poizvedbe je »Zakaj je moj račun za električno energijo tako visok«. Na videz preprosto vprašanje je v resnici zelo kompleksno, saj je treba pri iskanju odgovora upoštevati veliko različnih podatkov, kot so podatki, vezani na uporabnika (ponudnik električne energije, tip paketa električne energije itd.), ter drugi podatki in informacije, ki so objavljene na spletnih straneh oz.

svetovnem spletu (npr. različna priporočila, nasveti). Prepričani smo, da je za reševanje tovrstnih vprašanj ključnega pomena ustrezna organiziranost podatkov oz. baza znanja. Platforma KiviKM podpira ves cikel upravljanja z bazo znanja, ki obsega:

- a) pridobivanje znanja,
- b) kodiranje in shranjevanje znanja,
- c) prenos znanja,
- č) uporabo znanja.

Pri razvoju platforme smo dali velik pomen procesu pridobivanja in shranjevanja znanja, pri čemer smo implementirali sistem avtomatske gradnje baze znanja s spletnih mest. Pri izvedbi smo se zgledovali po algoritmu za združevanje segmentov algoritma VIPS, ki so ga v svojem delu predstavili Mehta idr. (Mehta, Mitra & Karnick, 2012), pri čemer smo uporabili podobne prijeme in tehnologije, vendar na drugačen način in za drugačne potrebe. Pri razvoju omenjenega sistema je bil ključnega pomena končni rezultat, in sicer baza znanja, zgrajena iz izbranega spletnega mesta. Ta je morala biti takšna, kot bi jo

ročno izdelal skrbnik spletne strani. Za omenjeno opravilo je bila vizualna analiza spletne strani z algoritmom VIPS najbolj primerna izbira. Ker algoritem VIPS ne omogoča identifikacije vsebine posameznih segmentov, je bilo treba izvesti ustrezno nadgradnjo z možnostjo klasifikacije segmentov v ustrezni razred (npr. novice), pri čemer smo si pomagali z metodami besedilnega rudarjenja.

4.1 Algoritem za izdelavo baze znanja s spletne strani

Algoritem, ki smo ga razvili za potrebe gradnje baze znanja s spletne strani, je pravzaprav modifikacija algoritma, ki ga je predstavil Mehta (Mehta, Mitra, & Karnick, 2012), s to razliko, da smo v algoritem vključili klasifikacijo vsebine segmentov spletne strani in predstavitev klasificiranih segmentov v obliki ontologije. Razviti algoritem lahko razdelimo na tri korake, in sicer na pridobivanje segmentov spletne strani, klasifikacijo segmentov in gradnjo ontologije. Spodaj je prikazan psevdokod razvitega algoritma.

```
AnalysePage(URL)
begin
    var V=GetVipsSegments(URL);
    if(V.Classify()==true)
        BuildOntology(V);
    Obvesti(V);
end

//pridobimo segmente spletne strani
//segmente klasificiramo
//zgradimo ontologijo
//obvestimo skrbnika o končanem postopku
```

Prvi korak algoritma, pridobivanje segmentov v drevesni strukturi s spletne strani, je postopek pri katerem je bila uporabljena obstoječa različica algoritma VIPS.

Drugi korak, imenovan klasifikacija segmentov, je eden od ključnih delov algoritma, saj v primeru, da segmenti niso klasificirani, ni mogoča izdelava ontologije. Ta poteka s pomočjo klasifikacijskega algoritma Naive Bayes, ki kot prvi pogoj za delovanje potrebuje ustrezno učno množico, ki vsebuje primere uspešno klasificiranih segmentov. Klasifikacijski algoritem posebej klasificira vsak segment iz drevesne strukture segmentov v ustrezen razred. Pri tem se upošteva tudi verjetnost klasifikacije segmenta, ki mora biti 60-odstotna ali več. V nasprotnem primeru se segment označi kot neklasificiran. Med klasifikacijo segmentov se izvede tudi pridobivanje naslovov posameznih segmentov, in sicer po pravilu stilsko

označenih besedil z značkami HTML *h1* – *h6*. Pri tem se vzame prvo besedilo iz segmenta, ki je označeno s stilskimi značkami HTML. Če segment nima ustrezno označenega besedila za naslov, se kot naslov segmenta generira besedilo, ki je sestavljeno iz naziva klasifikacijskega razreda in kombinacije številke ravni in zaporedne številke segmenta na tej ravni (primer Novica-1-2).

Zadnji, tretji korak algoritma za izdelavo baze znanja je izdelava ontologije na podlagi klasificiranih segmentov spletne strani. Gradnja ontologije poteka tako, da je vsak klasifikacijski razred iz učne množice tudi razred v ontologiji. Posledično se najprej kreirajo v ontologiji vsi razredi iz učne množice klasifikacijskega algoritma. Primerke razredov ontologije predstavimo s klasificiranimi segmenti, in sicer s predhodno izluščenimi nazivi segmentov. Za potrebe iskanja po ontologiji se vsebina segmenta zapiše

v lastnost »vsebina«. Povezave med posameznimi razredi oz. njihovimi primerki se kreirajo ob dodajanju primerkov v posamezni razred ontologije. Določajo se glede na drevesno strukturo segmentov, natančneje segment – podsegment. Postopek dodajanja primerkov v ontologijo se zaradi predstavitve segmentov v drevesni strukturi izvaja rekurzivno. Ko je končana izdelava ontologije, se ta shrani in se proži ustrezna funkcija za obveščanje skrbnika sistema.

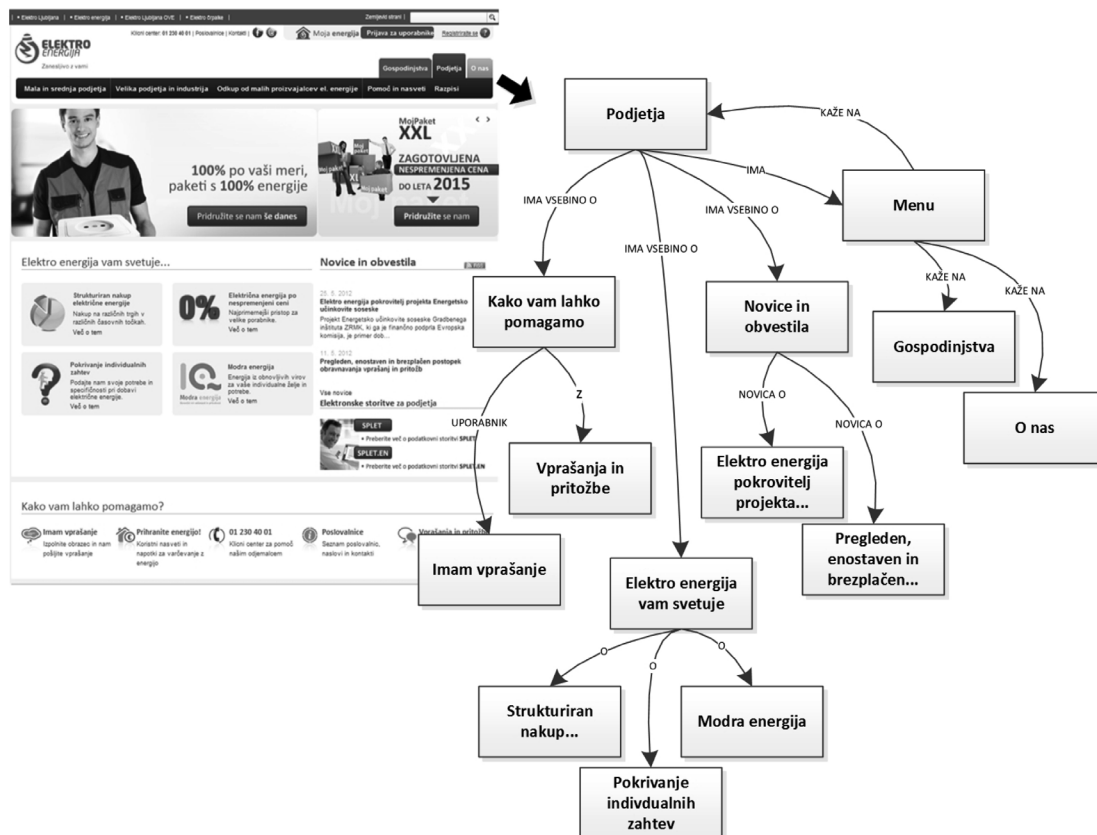
Če gradnja ontologije ni mogoča, kar je v primeru vsaj enega neuspešno klasificiranega segmenta, je skrbnik obveščen o neuspešno klasificiranih segmentih. Takrat sistem za te segmente od skrbnika zahteva, da jih klasificira v ustrezeni razred. Tako klasificirani segmenti se dodajo v učno množico segmentov za klasifikacijski algoritem.

4.2 Praktična uporaba KiviKM na primeru portala Elektro energija, d. o. o.

Algoritem za gradnjo baze znanja s spletne strani smo razvili v okviru platforme KiviKM, ki je bila razvita kot programska knjižnica (angl. Class Library),

in sicer z Microsoftovimi tehnologijami, natančneje s programskim jezikom Visual C#. Razlog za implementacijo platforme v obliki programske knjižnice je bil čisto praktične narave, saj je knjižnico mogoče uporabiti tako pri namiznih aplikacijah (angl. Desktop Application) kot pri portalnih rešitvah (angl. Web Application). Pri implementaciji platforme smo uporabili nekaj obstoječih prosto dostopnih knjižnic, s katerimi smo si olajšali delo. Uporabili smo obstoječo implementacijo algoritma VIPS, ki je bila pridobljena s portala Microsoft Research oz. Microsoftovih raziskovalcev, ki so razvili omenjeni algoritem (VIPs, 2013), medtem ko smo za potrebe besedilnega rudarjenja uporabili programsko knjižnico Weka (Weka 3, 2013) in za podporo semantičnim tehnologijam programsko knjižnico Jena.NET (Jena.NET, 2013).

Izdelani sistem za avtomatsko izdelavo baze znanja smo preskusili na portalu podjetja Elektro energija, d. o. o., oz. na njegovi podstrani Podjetja – www.elektro-energija.si/podjetja. Izbrana spletna stran je bila primerna za preverjanje delovanja izdelanega sistema, ker je dovolj preprosta, da je obvladljiva za



Slika 3: Prikaz izdelane baze znanja s spletne strani

preverjanje, in hkrati vsebuje tudi preostale gradnike spletne strani, ki pomembno vplivajo na gradnjo baze znanja (npr. pasice in vsebine).

V okviru preskusa smo se osredinjali na preverjanje rezultata implementiranega algoritma, pri čemer smo izvedli primerjavo ontologije omenjene spletne strani podjetja Elektro energija, d. o. o., izdelane z razvitim algoritmom in ontologijo, ki jo je razvil skrbnik spletnega mesta podjetja Elektro energija, d. o. o.

Pogoj za izvedbo preskusa je bila priprava učne množice klasifikatorja segmentov, pri čemer smo s pomočjo algoritma VIPS segmentirali osemdeset naključno izbranih spletnih strani portala Elektro energija, d. o. o., in jih ustrezno klasificirali v razrede: spletna stran, glava portala, noga portala, vertikalna navigacija, horizontalna navigacija, območje (segment), novice in obvestila ter besedilo.

Slika 3 prikazuje izdelano ontologijo izbrane spletne strani podjetja Elektro energija, d. o. o., ki je bila izdelana z razvitim algoritmom.

Ontologija, ki jo je razvil skrbnik sistema, se je nekoliko razlikovala od ontologije, ki je bila razvita s platformo KiviKM, kar je bilo tudi pričakovano. Vendar je bila ta razlika vidna pri poimenovanju razredov ontologije (npr. novice – novice in obvestila). Z vidika vsebine sta bili ontologiji enaki.

Ker pa primerjava izdelane ontologije s pomočjo razvitega algoritma z ontologijo, ki jo je razvil domenski ekspert, še ne pomeni uporabne vrednosti, smo izvedli eksperiment, s katerim smo preverili uporabnost izdelane ontologije. Pri tem smo uporabili pogosta vprašanja (angl. Frequently Asked Questions – FAQ) na spletni strani podjetja Elektro energija, d. o. o.

ELEKTRO ENERGIJA
Zanesljivo moja

ELEKTRIKA
GOSPODINJSTVA

ELEKTRIKA
POSLOVNI ODJEMALCI

PLIN
do 30.000 Sm³ letne porabe

PLIN
NAD 30.000 Sm³ letne porabe

O nas

Mala in srednja podjetja **Velika podjetja in industrija** **Odkup el. energije** **Pomoč in nasveti** **Razpisi** **Video vsebine**

100% razsvetljujemo poslovni svet, s 100% energije

Razlaga računa
Pogosta vprašanja in odgovori

Energetska učinkovitost
Storitve in nasveti za energetske učinkovitost.
Preberite več

Elektro energija | Podjetja | Pomoč in nasveti | Pogosta vprašanja in odgovori

Pogosta vprašanja in odgovori

Pogoji dobave za gospodinjstva

Opis osnovnih energetskega pojmov

Vprašanja poslovnih odjemalcev

Vprašanja gospodinskih odjemalcev

Prekinitve dobave električne energije

Arhiv vprašanj

Ali izmensko delo vpliva na doseženo povprečno ceno v preteklem obračunskem obdobju? [Odgovor](#)

Ali lahko vplivamo na ceno za porabljeno električno energijo s tipom dnevnega obremenitvenega diagrama? [Odgovor](#)

Ali zastarela tehnologija vpliva na porabo električne energije? [Odgovor](#)

Kaj je omrežnina? [Odgovor](#)

Slika 4: Stran Pogosta vprašanja in odgovori portala podjetja Elektro energija, d. o. o.

Za ta vprašanja smo se odločili, ker jih postavljajo uporabniki najpogosteje in nanje že poznamo odgovore. To je pomemben podatek za preverjanje pravilnosti rezultatov iskanja po ontologiji. Ker je predstavljena rešitev gradnje baze znanja pilotska rešitev, ki je omejena na določeno spletno stran, smo se omejili le na tista vprašanja, ki se nanašajo na spletno stran, s katere je bila izdelana baza znanja.

Vprašanja smo pretvorili v ustrezno obliko povpraševalnega jezika SPARQL (angl. Simple Protocol And RDF Query Language), s katerim smo izvedli poizvedbe po podatkih iz ontologije. Pri tem smo iskali po lastnostih razredov vsebina, v katerih je zapisana vsebina posameznega segmenta. Rezultate, ki smo jih dobili, smo primerjali z odgovori, ki so bili podani poleg vprašanj v okviru spletne strani. Pozitivne rezultate smo dosegli pri vprašanjih, pri katerih je bil odgovor del vprašanja objavljene novice oziroma članka. Tako smo kot odgovor na vprašanje dobili povezavo na članek, ki vsebuje odgovor na vprašanje. Medtem ko pri vprašanjih, ki niso bila del

odgovora, oziroma pri člankih, objavljenih v okviru spletne strani, nismo našli novic. Razlog za to gre iskati v samem obsegu ontologije, ki je bil osredinjen na točno določeno spletno stran in na novice, objavljene na njej.

4.3 Analiza delovanja sistema

V okviru preskusa delovanja sistema smo izvedli dva eksperimenta, in sicer primerjavo ontologij, pri čemer je bila ena izdelana na razviti platformi, drugo pa je izdelal skrbnik sistema za izbrano spletno stran, v okviru drugega eksperimenta pa smo preverjali uporabnost ontologije, razvite na platformi. Na podlagi prvega preskusa ugotavljamo, da je sistem upravičil naše zahteve in izdelal bazo znanja z izbrane spletne strani, tako kot je bilo pričakovano in zahtevano – kot bi jo izdelal domenski ekspert.

V tabeli 1 predstavljamo analizo SWOT razvitega sistema za avtomatsko gradnjo baze znanja, ki je bila izdelana na podlagi uporabe sistema na prikazanem primeru.

Tabela 1: Analiza SWOT

Prednosti	Slabosti
- Baza znanja, izdelana na zelo podoben način, kot bi jo izdelal ekspert - Pridobitev podatkov in informacij s portalov - Za računalnik in človeka lažje berljivi podatki - Možnost zelo široke uporabe	- Na začetku velika odvisnost od skrbnika sistema - Počasna obdelava portalov zaradi časovne odvisnosti algoritma VIPS - Potreben nadzor skrbnika baz znanja - Vsa vsebina posameznega segmenta, predstavljena v lastnostih ontologije
Priložnosti	Pasti
- Možnost povezave več izdelanih baz znanja v enotno bazo znanja – povezani splet - Predlagane tehnologije (semantični splet) omogočajo preprost prenos znanja prek spleta	- Uspešnost implementirane metode je odvisna tudi od domenskega eksperta (definirani razredi). - Možnost napak pri klasifikaciji segmentov in zaradi tega napačno zgrajena baza znanja - Verzije jezika HTML (npr. verzija 5) in nadgradnja algoritma VIPS

Iz izdelanega primera in prvega preskusa smo ugotovili, da je ena od prednosti tega pristopa gradnje baze znanja vsekakor preprost zapis znanja v smislu primerjave baze znanja in same spletne strani. Kot veliko priložnost velja omeniti preprosto povezljivost tako razvite baze znanja z drugimi ontologijami, kar lahko pomeni veliko dodano vrednost – preprost prenos znanja prek spleta. Po drugi strani pa je cena za takšen pristop časovna odvisnost, predvsem algoritma VIPS (Deng, Shipeng, Ji-Rong & Wei-Ying, 2003), in odvisnost od skrbnika sistema pri ustrezni pripravi učnih vzorcev klasifikatorja segmentov. Prav tako je ena od večjih pasti vsekakor vpliv skrbnika sistema pri označevanju neklasificiranih segmentov.

V tem primeru lahko pride do napačne klasifikacije, kar pomeni napačno delovanje sistema.

Drugi del preskusa, v okviru katerega smo preverjali uporabnost razvite ontologije s pomočjo platforme, nakazuje nekatere pomanjkljivosti. Čeprav je bil preskus uspešen in smo na večino vprašanj dobili ustrezne odgovore, so bili ti povprečni. Odgovor na vprašanje »Kako ugotovimo povprečno dnevno porabo?« iz ontologije je bil »Odgovor na vaše vprašanje je zapisan v članku Pokrivanje individualnih zahtev«, kar je primerljivo z iskalnimi rezultati konvencionalnih iskalnikov (npr. Google). Razlog za povprečne odgovore gre iskati v sami gradnji ontologije,

ki se osredinja le na gradnjo iz segmentov posamezne spletne strani. Vsebina posameznih segmentov, ki je lahko eden od pomembnejših nosilcev znanja, ostaja premalo obdelana s semantičnimi tehnologijami. Nadaljnji razvoj platforme bo v prihodnje usmerjen v več smeri, in sicer izdelavo ontologije na ravni celotnega portala, semantično obdelavo posameznih segmentov in ne nazadnje vključitev pilotne rešitve v okviru e-storitev, ki so na voljo uporabnikom portala Elektro energija, d. o. o.

5 SKLEP

V članku smo predstavili v okviru platforme KiviKM razvit sistem za avtomatsko gradnjo baze znanja s spletnega mesta, ki temelji na različnih tehnologijah, kot so VIPS, besedilno rudarjenje in semantične tehnologije. Delovanje razvitega sistema smo demonstrirali in analizirali na izbranem primeru spletne strani, pri čemer so bili rezultati sistema v okviru pričakovanih. Naše nadaljnje delo v podjetju bo usmerjeno v nadgradnjo in prilagoditev razvite platforme za potrebe naših projektov, katero bomo v prihodnosti tudi vključevali tako pri nadgradnjah obstoječih projektov (npr. Elektro energija, d. o. o.) kot tudi pri novih.

6 VIRI IN LITERATURA

- [1] Berners-Lee, T. (2013). *Putting the Web back into Semantic Web*. ISWC2005 Keynote. <http://www.w3.org/2005/Talks/1110-iswc-tbl> (19. 2. 2013).
- [2] Brezovnik, J. (2009). *Programsko ogrodje za procesiranje besedil v naravnem jeziku*. Magistrsko delo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.
- [3] Deng Cai, Shipeng Yu, Ji-Rong Wen & Wei-Ying Ma (2003). *VIPS: a Vision-based Page Segmentation Algorithm*. Microsoft Research Technical Report, November 1.
- [4] Gawrysiak, P., Protaziuk, G. & Rybinski, H. (2012). *Experiments with semi automated ontology building using text ontology miner*. <http://iis.ipipan.waw.pl/2008/proceedings/iis08-31.pdf> (26. 9. 2012).
- [5] Gunasundari, R. & Karthikeyan, S. (2012). *A Study of content Extraction From Web Pages Based on links*. <http://airccse.org/journal/ijdkp/papers/2312ijdkp03.pdf> (26. 9. 2012).
- [6] Han, J. & Kamber, M. (2006). *Data Mining Concept and Techniques*. Second Edition, Elsevier.
- [7] Jena .NET (2013). *Flexible .NET port of the Jena semantic web toolkit*. http://semanticweb.org/wiki/Jena_.NET (19. 2. 2013).
- [8] Leuf, B. (2006). *The semantic web*. Crafting Infrastructure for Agency. Sloko CIGRÉ, Ljubljana.
- [9] Liu, B. (2011). *Web Data Mining Exploring Hyperlinks, Context and usage data*. Second edition, Springer.
- [10] Mehta, R. R., Mitra, P. & Karnick, H. (2012). *Extracting Semantic Structure of Web Documents Using Content and Visual Information*. http://cse.iitkgp.ac.in/~pabitra/paper/www05_semantic.pdf (4. 12. 2012).
- [11] Stropnik, A., Vasič, V. & Podgorelec, V. (2006). *Potencial tehnologij semantičnega spleta v industrijskem podjetju*. GiB – Bilten strokovnih informacij Gorenja, Velenje.
- [12] Stropnik, A. (2006). *Potencial tehnologij semantičnega spleta v razvojnem podjetju*. Diplomsko delo. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.
- [13] Stropnik, A. (2012). *Spletna mesta tretje generacije*. Poročilo v okviru podiplomskega študija. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.
- [14] Toledo-Alvarado, J. I., Guzman-Arenas, A. & Martinez-Luna, G. L. (2012). *Automatic building of an ontology from a corpus of text documents using data mining tools*. www.jart.ccadet.unam.mx/vol10_3/automatic_9.pdf (4. 12. 2012).
- [15] Vasič, V. (2004). *Prenos znanja v Gorenju – Zemljevid znanja*. GiB – Bilten strokovnih informacij Gorenja, Velenje.
- [16] VIPS. (2013). *a Vision based Page Segmentation Algorithm*. <http://www.cad.zju.edu.cn/home/dengcai/VIPS/VIPS.html> (18. 2. 2013).
- [17] WEKA 3. (2013). *Data Mining with Open Source Machine Learning Software in Java*. <http://www.cs.waikato.ac.nz/ml/weka> (19. 2. 2013).
- [18] Zarič, D. (2004). *Semantični splet*. Diplomsko delo. Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor.
- [19] Zorman, M., Podgorelec, V., Lenič, M., Povalej, P., Kokol, P. & Tapajner, A. (2003). *Inteligentni sistemi in profesionalni vsakdan*. Univerza v Mariboru, Center za interdisciplinarne in multidisciplinarne raziskave in študije UM, Maribor.

Ambrož Stropnik je leta 2006 diplomiral na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru in se istega leta zaposlil v podjetju Gorenje, d. d., kot razvijalec spletnih aplikacij za podporo poslovanja. Od leta 2010 je zaposlen v podjetju KiviCOM, d. o. o., kot vodja razvoja. Je tudi študent doktorskega študija na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, v okviru katerega se ukvarja z raziskovanjem, pridobivanjem in predstavitvijo znanja s spletnih mest.

Milan Zorman je redni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Njegovo področje raziskav so klasični in hibridni inteligentni sistemi ter njihova aplikacija na različnih področjih s poudarkom na medicini in zdravstvu. Je predstojnik Inštituta za kompleksne in inteligentne sisteme na Centru za interdisciplinarne in multidisciplinarne raziskave in študije Univerze v Mariboru ter vodja projekta in pisarne Evropske podjetniške mreže na tem centru.

Primerjalna uporaba metod DEX in AHP v procesu odločanja

¹Tadej Košljar, ²Vladislav Rajkovič

²Univerza v Mariboru, Fakulteta za organizacijske vede, Kidričeva cesta 55a, 4000 Kranj
tadej.kosljar@gmail.com; vladislav.rajkovic@gmail.com

Izvleček

V odločitvenem procesu izbire ponudnika storitev v oblaku smo uporabili metodi DEX in AHP. Uporabili smo programski orodji DEXi in Super Decision. S tem želimo prispevati h kritičnemu razmisleku o uporabi ene in druge metode. Ocenitev alternativ, to je ponudnikov, je podana po korakih. Pristopi obeh metod se ujemajo v nekaterih korakih, v nekaterih pa se pomembno razlikujejo. Razlike so predvsem v določanju uteži, podajanju funkcij koristnosti in možnostih interpretacije rezultatov.

Ključne besede: večparametrsko odločanje, ponudniki storitev v oblaku, ocenjevanje, metoda DEX, metoda AHP.

Abstract

Comparative Use of the AHP and DEX Methods in the Decision-Making Process

We used the DEX and AHP methods in the decision-making process for choosing the most appropriate service provider in the cloud. The DEXi and Super Decision programs were applied and compared. By this we want to contribute to critical thinking about the use of each of the two methods. The evaluation of alternatives - in our case cloud providers - is presented step-wise. The approaches used by these methods are similar in some steps, but significantly different in others. The differences lie mainly in weight identification, utility function presentation and the possibilities of interpreting the evaluation results.

Key words: multi-attribute decision-making, service providers in the cloud, evaluation, DEX method, AHP method.

1 UVOD

Odločanje je ključnega pomena pri vsakem upravljanju in vodenju (od posameznika prek poslovnih sistemov in države do globalne družbe). Tako se posamezniki in skupine, npr. menedžerji in politiki srečujejo z reševanjem problemov, ki zahtevajo sprejemanje odločitev (Adair, 2013; Hayes, 2013). Obstajajo različne metode in pristopi, ki nam lahko pomagajo pri reševanju odločitvenega problema (Alessio in Nemery, 2013; Bohanec, 2006). Zastavlja se vprašanje, katero metodo izbrati v določenih okoliščinah. Dobrodošle so primerjalne analize metod. V prispevku bomo pri izbiri ponudnika računalniških storitev v oblaku primerjalno uporabili metodi DEX in AHP. S tem skušamo prispevati k razmisleku, kdaj je v določenem primeru uporaba posamezne metode primernejša in zakaj.

Model, ki smo ga razvili za oceno ponudnika računalniških storitev v oblaku, upošteva potrebe in možnosti malih in srednjih podjetij pri tej odločitvi. Storitve v oblaku pomenijo za podjetje pomemben

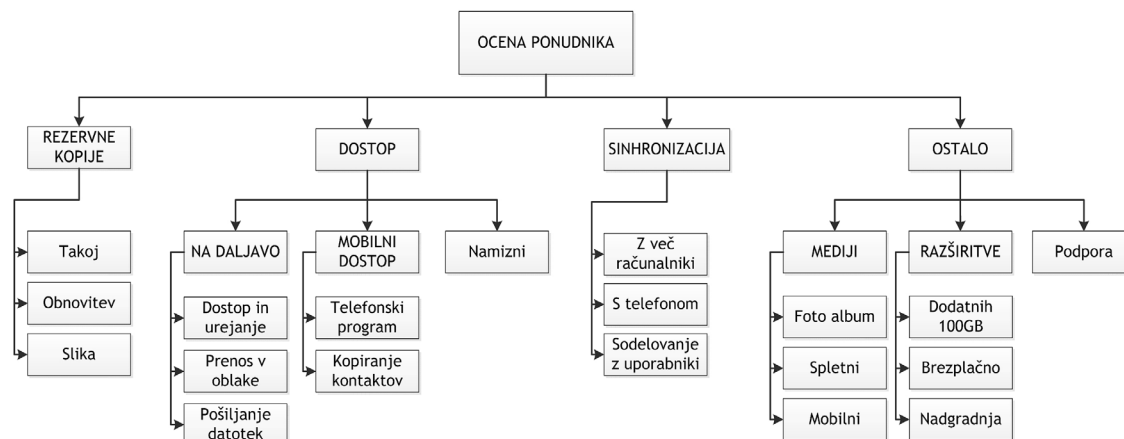
tehnični in ekonomski izziv (Kragelj in Rajkovič, 2012). Zato je tehten premislek, kako v oblak in s kom, več kot umesten.

2 ODLOČITVENI PROBLEM

Odločitveni problem, ki ga bomo obravnavali, se nanaša na izbiro ponudnika storitev v oblaku. Pomagati želimo pri odločitvi, kateri ponudnik storitev v oblaku je najprimernejši. Podjetje lahko s primerno storitvijo v oblaku izboljša svoj konkurenčni položaj v svoji panogi (Columbus, 2012).

Najprej smo identificirali alternative, to je ponudbe storitev v oblaku. Potrebne podatke smo pridobili prek spleta.

Zatem smo definirali drevo kriterijev (slika 1), s štirimi glavnimi podsklopi kriterijev, to so rezervne kopije, dostop, sinhronizacija idr. Iz slike 1 je razvidno, kako so posamezni sklopi kriterijev nadalje razgrajeni na merljive oz. neposredno ocenljive kriterije.



Slika 1: Drevo kriterijev

2.1 ODLOČITVENA METODA DEX

Metoda DEX obravnava kriterije, katerih zaloge vrednosti so diskretne. S stališča razumljivosti je običajno primerneje, da je posamezna vrednost izražena z besedo, kot pa s številko. Funkcije koristnosti, ki agregirajo podredne kriterije v nadredne, so podane tabelarično s pravili. Tabela 1 prikazuje funkcijo koristnosti »ocene ponudnika«, predstavljeno z devetimi točkami. Vsako točko lahko razumemo kot pravilo, ki določa vrednost/vpliv kriterija »rezervne kopije« in kriterija »ostalo« na »oceno ponudnika«. Pri tem so zaloge vrednosti parametrov: nesprejemljiv, sprejemljiv, dober, odličen. Pravilo 8 preberemo takole: Če je ponudnik po kriteriju »rezervne kopije« ocenjen z »zelo sprejemljivo« in v pogledu kriterija »ostalo« z oceno »sprejemljiv«, potem je »ocena ponudnika« »dober«.

Kaj dosežemo s tako izraženo funkcijo koristnosti? V večini drugih večkriterijskih metod so zaloge vrednosti kriterijev zvezne, funkcije koristnosti pa podane analitično, npr. kot utežene vsote, ko je povezava med kriteriji linearna. S tabelarično podano funkcijo v metodi DEX lahko izrazimo tudi nelinear-

Tabela 1: Funkcija koristnosti parametra ocena ponudnika

	REZERVNE KOPIJE	OSTALO	OCENA PONUDNIKA
1	Nesprejemljivo	Nesprejemljivo	Nesprejemljivo
2	Nesprejemljivo	Sprejemljivo	Sprejemljivo
3	Nesprejemljivo	Zelo sprejemljivo	Sprejemljivo
4	Sprejemljivo	Nesprejemljivo	Nesprejemljivo
5	Sprejemljivo	Sprejemljivo	Sprejemljivo
6	Sprejemljivo	Zelo sprejemljivo	Dober
7	Zelo sprejemljivo	Nesprejemljivo	Nesprejemljivo
8	Zelo sprejemljivo	Sprejemljivo	Dober
9	Zelo sprejemljivo	Zelo sprejemljivo	Odličen

ne povezave med kriteriji. Tabelarično funkcijo lahko primerjamo tudi s spremenljivimi utežmi, pri čemer lahko izrazimo spremembo uteži v odvisnosti od spremembe vrednosti kriterija (Bohanec in Rajkovič, 1990).

2.2 ODLOČITVENA METODA AHP

Analitično hierarhični proces (AHP) je metoda za določanje pomembnosti kriterijev pri večkriterijskem odločanju, s katero določimo uteži (Saaty, 1994; Špendl idr., 2003). Po tem postopku kriterije uredimo v hierarhijo, tako da na vsakem nivoju izberemo le majhno število kriterijev (3–5).

Pri metodi AHP opišemo pomembnost kriterijev z matriko primerjav pomembnosti kriterijev. Relativno pomembnost kriterijev i in j ocenjujemo z vrednostmi od 1 do 9. Pomen teh vrednosti podaja tabela 2 (Špendl idr., 2003; Nolberto, 2011; Alessio in Nemery, 2013; Creative Decisions Foundation, 2014a).

Tabela 2: Pomen elementov matrike pri parni primerjavi

1	Kriterija sta enako pomembna.
3	Kriterij i je malo bolj pomemben kot kriterij j .
5	Kriterij i je opazno bolj pomemben kot kriterij j .
7	Kriterij i je bistveno bolj pomemben kot kriterij j .
9	Kriterij i je absolutno bolj pomemben kot kriterij j .

Lestvica 1–9 je sorazmerno preprosta za uporabo. Soočamo pa se lahko tudi z negotovostjo lastnega dojemanja posamezne številke.

Pri metodi AHP parno primerjamo kriterije, pa tudi alternative. Z matriko primerjav kriterijev (tabe-

la 3) smo ocenili, da je kriterij razširitve pomembnejši od kriterija mediji in bistveno pomembnejši od kriterija podpora.

Tabela 3: Matrika primerjav za parameter rezervne kopije

	Mediji	Podpora	Razširitve
Mediji	1	4	1/2
Podpora	1/4	1	1/6
Razširitve	2	6	1

Iz pridobljenih matrik primerjav kriterijev izračunamo uteži tako, da izračunamo lastni vektor matrike in normiramo elemente, tako da je vsota elementov enaka 1. Elementi normiranega lastnega vektorja so uteži kriterijev. Za naš primer v tabeli 3 so tako dobljene uteži kriterijev mediji, podpora in razširitve po vrsti enake 0,323, 0,089 in 0,588.

3 REZULTATI VREDNOTENJA ALTERNATIV PO METODAH DEX IN AHP

V prispevku smo odločitveni proces z metodama DEX in AHP izvedli po korakih (Recchia idr., 2011). Sosledje korakov omogoča sistematično zbiranje in urejanje podatkov in znanja, ki je potrebno za ocenitev alternativ in za sprejem odločitve.

Pri metodi DEX smo uporabili program DEXi, s pomočjo katerega smo vrednotili alternative. Po identifikaciji odločitvenega problema in identifikaciji alternativ smo določili drevo kriterijev. Za tem smo določili tudi zaloge vrednosti kriterijev in definirali funkcije koristnosti, ki določajo vpliv nižjenivojskih kriterijev na višje ležeče kriterije. Zatem smo opisali vsako alternativo z vrednostmi posameznih kriterijev na listih drevesa (slika1). Temu je sledilo vrednotenje alternativ (ponudnikov) s programom DEXi (Bohanec, 2013a; Jereb idr., 2003; Bohanec, 2013b). Ta-

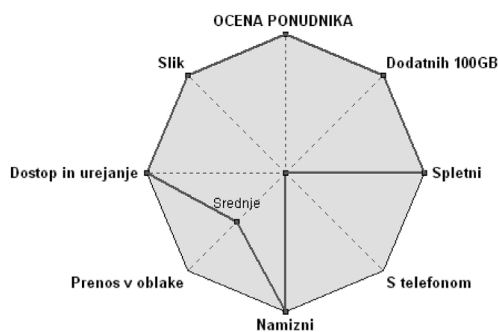
Varianta	Ponudnik 1	Ponudnik 2	Ponudnik 3	Ponudnik 4
OCENA PONUDNIKA	Odličen	Nesprejemljivo	Dober	Odličen
REZERVNE KOPIJE	Zelo sprejemljivo	Sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo
... Takoj	Zelo sprejemljivo	Sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo
... Obnovitev	Zelo sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo
... Slik	Dobro	Srednje	Srednje	Dobro
DOSTOP	Zelo sprejemljivo	Sprejemljivo	Sprejemljivo	Zelo sprejemljivo
... NA DALJAVO	Zelo sprejemljivo	Sprejemljivo	Sprejemljivo	Zelo sprejemljivo
... Dostop in urejanje	Zelo sprejemljivo	Sprejemljivo	Sprejemljivo	Zelo sprejemljivo
... Prenos v oblake	Srednje	Srednje	Dobro	Dobro
... Pošiljanje datotek	Da	Da	Da	Da
MOBILNI DOSTOP	Zelo sprejemljivo	Sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo
... Telefonski program	Dobro	Srednje	Dobro	Dobro
... Kopiranje kontaktov	NE	NE	NE	NE
... Namizni	Zelo sprejemljivo	Nesprejemljivo	Sprejemljivo	Zelo sprejemljivo
SINHRONIZACIJA	Sprejemljivo	Nesprejemljivo	Sprejemljivo	Zelo sprejemljivo
... Z več računalniki	Zelo sprejemljivo	Sprejemljivo	Zelo sprejemljivo	Zelo sprejemljivo
... S telefonom	Slabo	Slabo	Srednje	Dobro
... Sodelovanje z uporabniki	Sprejemljivo	Sprejemljivo	Sprejemljivo	Zelo sprejemljivo
OSTALO	Sprejemljivo	Nesprejemljivo	Sprejemljivo	Sprejemljivo
MEDIJI	Dobro	Srednje	Dobro	Dobro
... Foto album	Srednje	Srednje	Dobro	Dobro
... Spletni	Dobro	Dobro	Srednje	Dobro
... Mobilni	Dobro	Srednje	Dobro	Dobro
RAZŠIRITVE	Sprejemljivo	Nesprejemljivo	Sprejemljivo	Zelo sprejemljivo
... Dodatnih 100GB	pod 120 EUR	Ni možno	od 181 EUR do 240 EUR	pod 120 EUR
... Brezplačno	do 2 GB	od 4 GB dalje	od 4 GB dalje	od 4 GB dalje
... Nadgradnja	Ne	Ne	Ne	Ne
... Podpora	Srednje	Slabo	Srednje	Srednje

Tabela 4: Rezultati vrednotenja alternativnih ponudnikov s programu DEXi

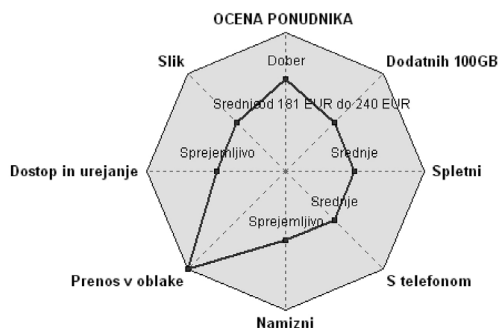
bela 4 prikazuje rezultate vrednotenja od listov drevesa kriterijev prek nadrednih kriterijev do končne ocene posamezne alternative.

Kakor je razvidno iz tabele 4, sta ponudnik 1 in ponudnik 4 dosegla oceno »odličen«. Na sliki 2 so prikazane ocene ponudnikov z osmimi izbranimi kriteriji: skupna ocena ponudnika, slik, dostop in urejanje, prenos v oblake, namizni, s telefonom, spletni in dodatnih 100 GB. Te kriterije smo izbrali zato, ker menimo, da so zanimivi za primerjanje ponudnikov. Izbrani kriteriji nam pomagajo razmišljati o razlikah med alternativami. Vidimo, da so razlike med ponudniki tudi v primeru, ko so vsi ocenjeni z oceno odlično. Iz primerjave lahko sklepamo, da je glede na zastavljene kriterije najprimernejši ponudnik 4.

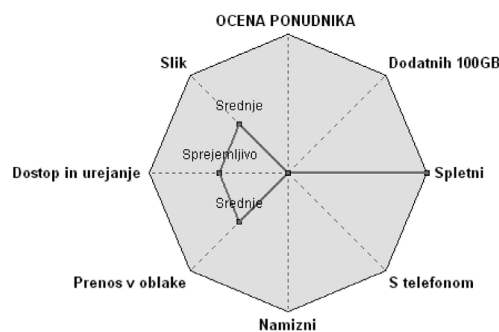
Ponudnik 1



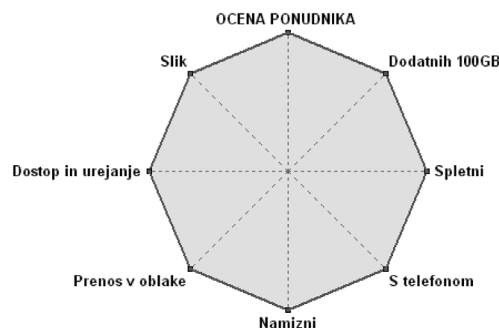
Ponudnik 3



Ponudnik 2



Ponudnik 4



Slika 2: Prikaz rezultatov vrednotenja s programom DEXi

DOSTOP		0,18
OSTALO		0,36
REZERVNE KOPIJE		0,28
SINHRONIZACIJA		0,28

Slika 3: Hierarhija osnovnih kriterijev, podana z utežmi

Pri metodi AHP so prve tri faze procesa odločanja enake kot v procesu odločanja z metodo DEX. Tako so procesi identifikacija odločitvenega problema, identifikacija alternativ in določitev drevesa kriterijev enaki. Bistvena razlika nastopi s parno primerjavo kriterijev glede na zastavljeni cilj in s parno primerjavo alternativ glede na vsak zastavljeni kriterij.

Za izvedbo procesa odločanja po metodi AHP smo uporabili program Super Decision (Creative Decisions Foundation, 2014a; Creative Decisions Foundation, 2014b). Ta izračuna preferenčnost posameznih kriterijev glede na ostale kriterije na istem nivoju. Na sliki 3 je prikazana hierarhija (pomembnost) osnovnih kriterijev, podana z utežmi posameznega kriterija.

Ko končamo vse ocenitve parnih primerjav kriterijev in parne primerjave alternativ, lahko ugotovimo, katera je »zmagovalna« alternativa. V našem primeru je to ponudnik 4, sledijo mu ponudnik 3, ponudnik 1 in ponudnik 2, kot je prikazano v tabeli 5.

Stolpec normiranih ocen predstavlja rezultate v obliki prioritet. Stolpec razmerja izračunamo iz stolpca normiranih ocen tako, da posamezni rezultat deli-

mo z največjo vrednostjo v stolpcu, v našem primeru je to 0,29. Najboljša alternativa tako dobi vrednost 1.

Tabela 5: Prikaz rezultatov vrednotenja s programom Super Decisions

Varianta	Razmerja	Normirane ocene
Ponudnik 1	0,86	0,25
Ponudnik 2	0,63	0,18
Ponudnik 3	0,93	0,27
Ponudnik 4	1,00	0,29

Ostale vrednosti v tem stolpcu interpretiramo tako: ponudnik 3 ima vrednost 0,93, kar pomeni, da dosega 93 odstotkov rezultata, ki ga ima zmagovalni ponudnik 4.

4 SKLEP

Lahko rečemo, da uporaba obeh metod, DEX in AHP, pomembno prispeva k sistematični organizaciji odločitvenega procesa in posledično k boljšim odločitvam. Razvejana in strokovno dodelana podpora obeh metod – tako v programskem smislu kot tudi v pogledu priročnikov in objav primerov uporabe – omogoča njuno uporabo v neposredni praksi.

Primerjava metod pri reševanju našega problema izbire ponudnika storitev v oblaku je pokazala, da je DEX preprostejša in razumljivejša metoda, AHP pa omogoča ločljivost rezultatov tudi pri zelo podobnih ocenah variant, vendar so bili rezultati manj razumljivi. Rezultati metode DEX razgrnejo slabosti in prednosti posameznega ponudnika.

Metodi DEX in AHP bi veljalo preizkusiti tudi v kombinaciji, ko bi AHP uporabili za ločevanje variant znotraj istega razreda po metodi DEX. Kot že rečeno, je bistvena prednost metode AHP ločljivost podobnih variant. Taki dvostopenjski modeli oziroma kombinacije lahko delujejo bolje od uporabe posameznih metod.

LITERATURA

- [1] Adair, J. (2013). Decision Making and Problem Solving (Creating Success). Croydon: CGI Group Ltd.
- [2] Alessio, I. & Nemery, P. (2013). Multi-criteria Decision Analysis: Methods and Software. Hoboken, NJ: John Wiley & Sons Ltd.
- [3] Bohanec, M. & Rajkovič, V. (1990). DEX: An Expert System Shell for Decision Support. *Sistemica*, 1(1), str. 145–157.
- [4] Bohanec, M. (2006). Odločanje in modeli. Ljubljana: DMFA – Založništvo.
- [5] Bohanec, M. (2013a). Program DEXi. Available at <http://kt.ijs.si/MarkoBohanec/dexi.html>; zadnji ogled 1. 6. 2014.
- [6] Bohanec, M. (2013b). DEXi: Program for Multi-Attribute Decision Making, User's Manual, Version 4.00. IJS Report DP-11340. Ljubljana: Jožef Stefan Institute.
- [7] Columbus, L. (2012). Cloud Computing and Enterprise Software Update. *Forbes*, 2012 [online]. Available at <http://www.forbes.com/sites/louiscolumbus/2012/11/08/cloud-computing-and-enterprise-software-forecast-update-2012/>; objavljeno 8. 11. 2012.
- [8] Creative Decisions Foundation. (2014a). Program Super Decisions. Available at <http://www.superdecisions.com/download.php3>; zadnji ogled 1. 6. 2014.
- [9] Creative Decisions Foundation. (2014b). Tutorial on Hierarchical Decision Models (AHP). Available at <http://beta.superdecisions.com/tutorial-1-building-ahp-models/manual-for-building-ahp-decision-models>; zadnji ogled 1. 7. 2014.
- [10] Hayes, J. (2013). Operational Decision-Making in High-Hazard Organizations. Burlington: Ashgate.
- [11] Jereb, E., Bohanec, M. & Rajkovič, V. (2003). DEXi – Računalniški program za večparametrsko odločanje. Kranj: Moderna organizacija.
- [12] Kragelj, P. & Rajkovič, V. (2012). Kako oceniti ponudnika storitev v oblaku. *Uporabna informatika*, 20(4), str. 244–249.
- [13] Nolberto, M. (2011). A Strategy for Using Multicriteria Analysis in Decision-Making. Dordrecht: Springer Science+Business Media B.V., str. 77–78.
- [14] Recchia, L., Boncinelli, P., Cini, E., Vieri, M., Garbati, F. & Sarri, D. (2011). Multicriteria Analysis and LCA Techniques. London: Springer-Verlag, str. 6–7.
- [15] Saaty, T. (1994). Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process. Pittsburgh: RWS Publications.
- [16] Špendl, R., Bohanec, M. & Rajkovič, V. (2003). Comparative Analysis of AHP and DEX Decision Making Methods. V A. Mrvar & A. Ferligoj (ur.) Program and abstracts of the International Conference on Methodology and Statistics, str. 65–66. Ljubljana: Center of Methodology and Informatics, Institute of Social Sciences at Faculty of Social Science.

Tadej Košljar je diplomiral leta 2007 na Fakulteti za organizacijske vede Univerze v Mariboru, smer organizacija in management kadrovskih in izobraževalnih procesov, leta 2012 pa je pridobil še drugo diplomo na smeri informacijski sistemi. Njegovo področje dela je kadrovske menedžment, v okviru katerega vodi ključne kupce.

Vladislav Rajkovič je zaslužni profesor in predstojnik Laboratorija za odločitvene procese in ekspertne sisteme na Fakulteti za organizacijske vede Univerze v Mariboru ter raziskovalni sodelavec Odseka za inteligentne sisteme na Institutu Jožef Stefan. Njegovo področje so računalniški informacijski sistemi s posebnim poudarkom na uporabi metod umetne inteligence v procesih odločanja ter vzgoje in izobraževanja.

Dokumentiranje zahtev pri razvoju mobilnih aplikacij

Tina Schweighofer, Marjan Heričko

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, 2000 Maribor

tina.schweighofer@um.si; marjan.hericko@um.si

Izvelek

Pravilno zastavljene in dobro oblikovane zahteve so eden izmed pomembnejših gradnikov uspeha projekta. Pomembno je, da se zbiranja in kasnejšega zapisa lotimo sistematično ter v skladu s priporočili in dobrimi praksami. Izjema ni niti razvoj aplikacij za mobilne telefone. Kljub temu da gre za razmeroma sodoben programski produkt, velja enako — dobre zahteve so ključ do uspeha. V okviru zbiranja zahtev je treba dovolj časa posvetiti oblikovanju specifikacije zahtev programske opreme (SZPO). Pri izdelavi dokumenta SZPO za mobilno aplikacijo v razvoju hitro ugotovimo, da bi bilo treba prilagoditi posamezna poglavja ali pa celo dodati nova glede na obstoječe dobre prakse. V prispevku podrobneje predstavljamo proces oblikovanja dokumenta SZPO za mobilne aplikacije in njegovo priporočeno strukturo. Pregledali in analizirali bomo splošna priporočila za izdelavo dokumentov SZPO, jih dopolnili s pregledom literature in analizo praktičnih primerov dokumentov SZPO za mobilne aplikacije. Vse skupaj bomo nadgradili z izkušnjami, pridobljenimi v okviru aktualnega projekta razvoja mobilnih aplikacij. Na podlagi zbranih podatkov bomo predstavili posamezna dodatna poglavja in dopolnitve, specifične za mobilne aplikacije, ter izpostavili lastnosti mobilnih naprav in tehnologij, na katere moramo biti pozorni. Vse ugotovitve bomo povzeli ter jih v sklepu predstavili v obliki priporočil in dobrih praks za izdelavo dokumenta SZPO za mobilne aplikacije.

Ključne besede: specifikacije zahtev programske opreme (SZPO), mobilne aplikacije, mobilne tehnologije, IEEE 830, ISO/IEC/IEEE 29148.

Abstract

Documenting the Requirements for Mobile Applications Development

Project success greatly depends on well-formed requirements. Therefore it is important to collect and record software requirements systematically, according to given recommendations and best practices. Requirements engineering is also an important part of mobile applications development. It is very important to devote enough time to the creation of the Software Requirements Specification (SRS) document. When constructing the SRS document for a mobile application several challenges have to be dealt with. We soon realize that existing recommendations are not enough and that we need some changes and updates. In this paper we present the process of creating an SRS document for a mobile application in development and its proposed structure. We focus on additional sections and the supplementation of existing sections for the document. Next, we review general recommendations for writing SRS documents which we extend with a review of literature and practical examples of SRS documents for mobile applications. Finally, we present experiences gained in the process of a mobile application development project. Based on collected data we present some additional chapters and amendments to the SRS document for mobile applications. All findings are summarized in the conclusion in the form of recommendations and best practices for writing SRS documents for mobile applications.

Key words: software requirements specification (SRS), mobile application, mobile environment, IEEE 830, ISO/IEC/IEEE 29148.

1 UVOD

Že začetni koraki pri razvoju mobilnih aplikacij imajo pomemben vpliv tako na nadaljnji razvoj kot tudi na kasnejšo uspešnost aplikacije na trgu. Poleg vseh izzivov, s katerimi se spopadamo ob razvoju, je pomembno, da mobilno aplikacijo izoblikujemo tako, da izpolni pričakovanja naročnika. Do tega nas v veliki meri pripelje primerno in ustrezno zbiranje zahtev. Korak zbiranja zahtev je v življenjskem ciklu produkta zelo pomemben. Je namreč eden izmed najbolj zahtevnih delov razvoja pro-

gramskega produkta (Hofmann & Lehner, 2001; Kamaruddin, Yusop & Ali, 2011). Neustrezne zahteve lahko pripeljejo do številnih posledic: podpora napačnim funkcionalnostim, nedoseganje tržnih potreb, zamujanje ter številni hrošči, napake, pomanjkljivosti in nepravilnosti v produktu (Maccari, 1999).

Pomanjkljive zahteve so tako lahko eden izmed pomembnih vzrokov neuspeha projekta, ki lahko sega od nenadnega zvišanja stroškov pa vse do propa-

da projekta (Belfo, 2012; Giakoumakis & Xylomenos, 1996). Povedano drugače, pravilno zbiranje, oblikovanje in zapis zahtev so med najpomembnejšimi in zahtevnejšimi nalogami v okviru projekta v razvoju (Hofmann & Lehner, 2001).

Končni produkt zbiranja zahtev je običajno zahtevnik oziroma dokument specifikacija zahtev programske opreme (SZPO). Ta je eden izmed temeljev nadaljnjega razvoja programskega produkta tudi v okviru razvoja mobilnih aplikacij. Dobro oblikovan SZPO projektu prinese številne prednosti, kot npr. vzpostavitev temeljev dogovora med naročnikom in izvajalcem z vidika zahtev sistema, služi lahko kot podlaga za oceno truda in stroškov, ponuja pa tudi podlago za preverjanje ter za nadaljnje izboljšave razvojnega produkta (IEEE Recommended Practice for Software Requirements Specifications, 1998). Priporočila za izdelavo dokumenta SZPO so izdale številne organizacije, oblikovani pa so bili tudi posamezni standardi, ki podpirajo izdana priporočila posameznih organizacij (Giakoumakis & Xylomenos, 1996). Med množico organizacij zagotovo najbolj izstopa IEEE, kjer so svoja priporočila zbrali v standardu IEEE 830,¹ ki je danes del širšega standarda ISO/IEC/IEEE 29148.² Standard opisuje vsebino in lastnosti dokumenta SZPO, predstavljeni pa so tudi posamezni praktični primeri dokumentov specifikacij zahtev. Poudariti je treba, da se omenjena priporočila osredinjajo predvsem na splošni vidik izdelave dokumenta. Pri tem avtorjem dajejo napotke glede oblike in strukture, podajajo pa tudi pričakovane lastnosti dobro oblikovanega dokumenta SZPO.

Če priporočila uporabimo kot primer dobre prakse pri izdelavi dokumenta SZPO za mobilne aplikacije, kar kmalu zaznamo potrebo po dodatnih podatkih in definicijah v okviru dokumenta. Mobilne aplikacije se namreč v marsičem razlikujejo od tradicionalne programske opreme in pomembno je, da to upoštevamo tudi pri njihovem razvoju. Razlike so tako vidne že pri določanju zahtev, to pa se odraža tudi pri izdelavi SZPO. Izkazuje se, da SZPO za mobilne aplikacije potrebuje nekatera dodatna poglavja in zahteve, vezane na specifično delovanje in funkcionalnosti mobilnih telefonov in aplikacij. V prispevku bomo tako raziskali, katere postavke, poglavja in zahteve se razlikujejo v okviru dokumentiranja

zahtev za mobilne aplikacije. S pomočjo literature in praktičnih izkušenj bomo identificirali posamezna poglavja dokumenta SZPO za mobilne aplikacije, ki se razlikujejo od dokumenta SZPO za tradicionalno programsko opremo. Ugotoviti želimo, katera poglavja je dokumentu smiselno dodati ter katera dopolniti, prav tako pa izpostaviti informacije in postavke, ki so bistvene za uspešen nadaljnji razvoj mobilne aplikacije.

V drugem razdelku prispevka bomo predstavili sorodna dela, obravnavana v okviru pregleda literature z obravnavanega področja. Sledil bo povzetek trenutnih trendov s področja mobilnih tehnologij, mobilnih telefonov ter mobilnih aplikacij. V četrtem razdelku bomo povzeli dobre prakse oblikovanja strukture dokumenta ter splošne lastnosti, povezane z dokumentom. V istem razdelku bomo predstavili tudi priporočila za prilagoditve dokumenta SZPO za mobilne aplikacije. Najprej bomo predstavili lastnosti mobilnih tehnologij in telefonov, ki imajo vpliv na dokument SZPO. Sledil bo povzetek rezultatov pregleda literature in praktičnih primerov dokumentov SZPO za mobilne aplikacije. Vse skupaj bomo dopolnili s prikazom praktičnih izkušenj pri izdelavi dokumenta SZPO za mobilne aplikacije v razvoju. Prispevek bomo sklenili s povzetkom ugotovitev in predlogom pomembnih dodatnih informacij, ki jih je po naših izkušnjah smiselno vključiti v zapis specifikacij zahtev za mobilne aplikacije.

2 SORODNA DELA

S področjem dokumentiranja zahtev in njihovim zapisom v dokument specifikacija zahtev za programsko opremo so se ukvarjali že številni avtorji. Na tem področju obstajajo številna uveljavljena priporočila in dobre prakse, med katerimi so nekatere prerasle tudi v standarde. Najbolj uveljavljen standard na tem področju je zagotovo IEEE 830 (IEEE Recommended Practice for Software Requirements Specifications, 1998), ki je danes del obsežnejšega standarda ISO/IEC/IEEE 29148 (Systems and software engineering – Life cycle processes – Requirements engineering, 2011), ki so ga pripravile organizacije ISO/IEC s sodelovanjem z IEEE. Pomembno podlago na področju razvoja programske opreme in dokumentiranja procesa so postavili tudi na oddelku za obrambo Združenih držav Amerike. Trenutno aktualni standard MIL-STD-498 (Software development and documentation, DoD Military Standard MIL-STD-498,

¹ IEEE Recommended Practice for Software Requirements Specifications

² Systems and software engineering – Life cycle processes – Requirements engineering

1994) priporočila in navodila predstavlja v okviru dokumenta DI-IPSC-81433 (Software requirements specification DID, DoD Military Standard DI-IPSC-81433, 1994). Poleg omenjenih standardov so priporočila in navodila izdale tudi nekatere druge organizacije, npr. NASA in ESA (ESA Board for Software Standardisation and Control, 1991; NASA, 2009), ki s svojimi navodili pripomoreta tako k zbiranju zahtev, kot tudi k zapisu le-teh v dokument SZPO. Vse omenjene dokumente so pri svojih raziskavah uporabili avtorji, ki so se ukvarjali z vplivi uspešnega procesa zbiranja zahtev na nadaljnje korake (Hammer, Huffman, & Rosenberg, 1998; Hofmann & Lehner, 2001) in kakovost dokumentov SZPO (Belfo, 2012; Boehm, 1984; Davis idr., 1993). Omenjeni standardi so bili tudi pregledani z namenom ovrednotenja glede na merila za namen uporabe. Avtorji dela so izbrane standarde ovrednotili glede na osem izbranih meril: neodvisnost, integracija, natančnost, splošnost, organiziranost, vsebinska popolnost, popolnost vidikov in prilagodljivost. Na podlagi podeljenih vrednosti za omenjena merila sodelujočim pri izdelavi dokumenta SZPO ponujajo hiter vpogled v lastnosti posameznega standarda. S tem pa jim na preprost način pomagajo pri odločitvi, kateri izmed standardov je najbolj primeren za izdelavo dokumenta SZPO za njihov produkt (Giakoumakis & Xylomenos, 1996).

S pojavom mobilnih aplikacij so postale aktualne študije povezane z izdelavo dokumenta SZPO za mobilne aplikacije. V svojem delu avtorji Economou, Gavalas, Kenteris in Tsekouras (2008) ugotavljajo, katere so zahteve za mobilne aplikacije, ki jih je treba upoštevati pri razvoju in posledično tudi v dokumentu SZPO. Kot navajajo, moramo v dokumentu SZPO biti pozorni na mobilni kontekst, produkt načrtovati za namen prenosljivosti ter upoštevati širok spekter znanj in izkušenj uporabnikov. Izpostavljajo še omejene vire mobilnih telefonov, vhodne teh izhodne poti in zbiranje informacij s pomočjo senzorjev, ki se nahajajo na mobilnih telefonih. Podobno se z zahtevami za konkretno definirano domeno ukvarjajo v še enem delu (Doherty, McKnight & Luz, 2010), v katerem so se avtorji osredinili na zahteve za mobilne zdravstvene aplikacije. Iz obstoječih ogradij s področja zdravstva so sestavili skupno ogrodje, katerega namen je identifikacija in analiza zahtev za mobilni produkt. Podobno so se z ogradjem za zbiranje zahtev za aplikacije, a za drugačno domeno, ukvarjali tudi v sorodnem delu (Finkelstein & Savi-

gni, 2001). Omejili so se na aplikacije, ki zaznavajo kontekst, ter na zajemanje zahtev na tem področju. Kot navajajo, je zajemanje zahtev zelo kompleksno in zahtevno delo, tudi zaradi značilnosti mobilnih telefonov.

Posamezna dela se ukvarjajo tudi s pristopi identifikacije zahtev v pripadajočem procesu in izdelavo dokumenta SZPO za mobilne aplikacije. Avtorji Gil, Andersson, Milrad in Sollervall (2012) predlagajo pristop identifikacije s pomočjo evolucijskega procesa. Čeprav delujejo v domeni mobilnega učenja, navajajo, da bi lahko tehniko uporabili tudi v drugih domenah. Avtorji Kamaruddin idr. (2011) pa za namen analize zahtev za mobilno domeno predlagajo uporabo teorije aktivnosti.

Na področju mobilnih aplikacij so zelo razširjene mobilne igre. Tako so se raziskave s področja dokumentiranja in zapisa zahtev pojavile tudi na tem področju. Kot npr. predstavljajo avtorji Salazar, Mitre, Olalde in Sanchez (2012), se tudi v domeni mobilnih iger pojavljajo številni izzivi. V okviru domene mobilnih iger igra ključno vlogo dokument GDD,³ ki je po značilnostih zelo podoben dokumentu SZPO. V bistvu gre za dokument SZPO, ki podpira razvoj mobilnih iger. V delu so poiskali in ovrednotili posamezne omenjene dokumente ter jih izboljšali glede na izbrana priporočila za dokumente SZPO.

Posameznih del, ki bi izključno podajala priporočila in dobre prakse pri izdelavi dokumenta SZPO za mobilne aplikacije, pri pregledu digitalnih knjižnic ter na spletu nismo zasledili. Tako smo se kot logično nadaljevanje v naslednjem koraku lotili preučevanja dostopnih praktičnih primerov dokumentov SZPO za mobilne aplikacije. Na svetovnem spletu je teh na voljo veliko, podrobnejšo analizo primerov pa predstavljamo v nadaljevanju v razdelku o prilagoditvah dokumentov SZPO za mobilne aplikacije.

3 MOBILNE TEHNOLOGIJE

Število mobilnih telefonov z vsakim letom strmo narašča. Po podatkih analitične hiše Gartner je bilo samo v tretji četrtini leta 2012 prodanih skoraj 428 milijonov mobilnih telefonov. Znotraj te številke pametni mobilni telefoni predstavljajo skoraj 40 odstotkov (Gartner, 2012). Podobno se dogaja tudi na področju mobilnih naročniških razmerij. Konec leta 2012 je bilo na svetu aktivnih okoli 6,8 milijarde mo-

³ Game Design Document

bilnih naročniških razmerij. To pomeni, da je bilo v aktivnem naročniškem razmerju kar 96 odstotkov svetovne populacije. Trenutno je stopnja prodora mobilnih naročniških razmerij v svetovnem merilu 96 odstotkov, če se osredinimo samo na Evropo, pa znaša kar 126 odstotkov (ITU, 2013). Številke bodo še naraščale. Do leta 2016 naj bi bilo na svetu kar osem milijard mobilnih naročniških razmerij (MobiThinking, 2012; Portio Research, 2012).

S številom mobilnih telefonov narašča tudi število mobilnih aplikacij. Eden pomembnejših mejnikov na tem področju je bil dosežen decembra 2011, ko je število mobilnih aplikacij, ki so dostopne uporabnikom, preseglo milijon (The New York Times, 2011). Konec leta 2012 je mobilne aplikacije uporabljalo okoli 1,1 milijona mobilnih uporabnikov. Po napovedih bo številka hitro naraščala, za skoraj 30 odstotkov letno, in bo tako konec leta 2017 znašala 4,4 milijona uporabnikov (Whitfield, 2013). Mobilne aplikacije so leta 2012 prinesle okoli 12 milijard dolarjev prihodkov, prenesenih pa je bilo okoli 46 milijard mobilnih aplikacij (Portio Research, 2012). Tudi na tem področju lahko pričakujemo hitro rast. Leta 2013 naj bi uporabniki prenesli še dodatnih 82 milijard mobilnih aplikacij (Whitfield, 2013).

4 SPECIFIKACIJE ZAHTEV PROGRAMSKE OPREME

Z vse večjo prisotnostjo mobilnih aplikacij postaja vse pomembnejši proces njihovega razvoja v okviru življenjskega cikla mobilne aplikacije. Eden od prvih korakov pri razvoju le-teh je – podobno kot pri razvoju tradicionalne programske opreme – zbiranje in dokumentiranje zahtev za produkt. Končni produkt omenjenega koraka je dokument specifikacija zahtev programske opreme (SZPO). Dokument SZPO je zaradi svojih značilnosti in namena izjemno pomemben gradnik pri razvoju programske opreme, bodisi tradicionalne ali mobilne. V okviru razvojnega cikla izstopajo specifikacije zahtev kot zelo pomembna in ključna postavka (Giakoumakis & Xylomenos, 1996). Kot pravi definicija slovarja IEEE, dokumentacija SZPO združuje bistvene zahteve, funkcionalnosti, zmogljivosti, oblikovne omejitve in attribute programskega produkta ter njegovih zunanjih vmesnikov (IEEE Standard Glossary of Software Engineering Terminology, 1990). Dokument SZPO združuje specifikacije posameznega programskega produkta, programa ali skupine programov, ki izvajajo funkcije v okviru specifičnega okolja. Pri oblikovanju do-

kumenta lahko sodeluje eden ali več predstavnikov proizvajalca, eden ali več predstavnikov naročnika, ali pa dokument nastane s sodelovanjem obeh strani (IEEE Recommended Practice for Software Requirements Specifications, 1998).

Izdelava dokumenta SZPO je del procesa zbiranja zahtev oziroma inženiringa zahtev programske opreme. Omenjeni proces so v izolaciji ali kot del celotnega življenjskega cikla opisovale številne organizacije (Giakoumakis & Xylomenos, 1996). Izdale so tudi različna priporočila, navodila ali celo ogrođja, s pomočjo katerih razvijalcem programske opreme in udeležencem procesa zbiranja zahtev ponujajo številne predloge in dobre prakse. Omenimo nekaj primerov od najbolj splošnega do bolj podrobnega. Navodila in priporočila za oblikovanje dokumenta SZPO lahko najdemo v okviru opisa procesa zbiranja zahtev, opisanega znotraj celotnega življenjskega cikla programske opreme. To prikazujeta npr. IBM Rational Unified Process (RUP) (Rational, 1998) in standard, ki ga je objavila organizacija ESA (ESA Board for Software Standardisation and Control, 1991). Nekoliko nižje ravni – le opisa procesa zbiranja zahtev – so se lotili pri organizaciji NASA (NASA, 2009). Najbolj podrobno so se s samim dokumentom SZPO ukvarjali pri IEEE in US DoD⁴ (IEEE Computer Society, 1998; Software requirements specification DID, DoD Military Standard DI-IPSC-81433, 1994).

S tem, kako izdelati dokument SZPO, da bo ta primerna in trdna podlaga za nadaljnji razvoj, se je ukvarjalo že veliko avtorjev. Prav vsi so poudarili pomen inženiringa zahtev. Kot navajata avtorja Giakoumakis in Xylomenos, viri, ki opisujejo dokument SZPO, spadajo nekam med strogo definicijo dokumenta in priporočila ter navodila za izdelavo. Priporočila in standardi so navadno oblikovani na način, ki podaja okvirno obliko dokumenta skupaj s pojasnili, kako zapisati posamezne vrste zahtev. Prilagodljivost vsebine se spreminja od standarda do standarda in od priporočila do priporočila. Avtorji navadno dodajajo, da gre le za priporočila, ki si jih vsak lahko prilagodi glede na svoje potrebe. Z uporabo dokumentacijskega standarda pri izdelavi dokumenta SZPO lahko avtorji povečajo svojo učinkovitost, tako da sledijo znanim orisom. Pomagajo si lahko tudi z uporabo seznama zahtev ter seznama elementov, ki naj bi bila del dokumenta v izdelavi (Giakoumakis & Xylomenos, 1996).

⁴ United States Department of Defense

Tako sta korak zbiranja zahtev kot tudi njihov zapis v okviru dokumenta SZPO zahtevna naloga. Pomembno je, da se ne glede na tip produkta, ki ga razvijamo, držimo nekaterih ustaljenih standardov in smernic. Nekateri od njih, ki so po našem mnenju pomembni tudi za dokumente SZPO mobilnih aplikacij, bomo predstavili v nadaljevanju.

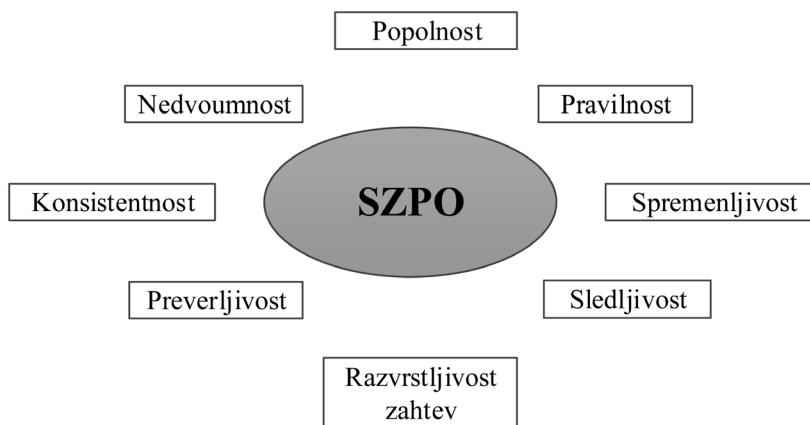
Dokument SZPO je prvi dokument, ki podrobno opisuje programski produkt v razvoju. Zato je uporabljen v vseh nadaljnjih aktivnostih in predstavlja uporabniške potrebe, zahteve za implementacijo ter kasnejšo točko za preverjanje (Giakoumakis & Xylomenos, 1996). Za ustrezen oblikovan dokument specifikacije zahtev programske opreme je priporočljivo, da avtorji naslovijo nekaj ključnih vprašanj, vezanih na posamezna področja razvoja (IEEE Recommended Practice for Software Requirements Specifications, 1998):

- *funkcionalnosti* (kaj naj bi izdelek delal),
- *zunanji vmesniki* (kako bo produkt sodeloval z uporabniki, s sistemsko in drugo strojno opremo ter z drugo programsko opremo),

- *zmogljivost* (kakšna je zahtevana hitrost, zaželena razpoložljivost ter pričakovani odzivni čas in čas obnovitve posameznih funkcij produkta),
- *atributi* (kako so upoštevani vidiki prenosljivosti, pravilnosti, vzdrževanja in varnosti),
- *vpliv načrtovalnih omejitev na implementacijo* (ali obstajajo kakšne omejitve, povezane s potrebnimi standardi, programskim jezikom, pravili, viri ali operacijskim okoljem, ki vplivajo na implementacijo).

4.1 Značilnosti dobrega SZPO

Priporočljivo je, da ustrezen dokument specifikacij zahtev programske opreme ustreza posameznim lastnostim, ki so predstavljene na sliki 1 (Belfo, 2012; Davis idr., 1993; ESA Board for Software Standardisation and Control, 1991; Hammer idr., 1998; IEEE Recommended Practice for Software Requirements Specifications, 1998; Software requirements specification DID, DoD Military Standard DI-IPSC-81433, 1994).



Slika 1: Lastnosti dokumenta SZPO

Dokument specifikacij mora ustrezati lastnosti *pravilnosti*. Produkt v razvoju naj bi upošteval vsako zahtevo, navedeno v dokumentu SZPO. Napisan mora biti *nedvoumno*, tako da ima lahko ena podana zahteva samo eno možno interpretacijo. Ker je dokument SZPO pomemben skozi ves življenjski cikel programskega produkta – od analize in načrtovanja pa vse do preverjanja, testiranja in usposabljanja –, je pomembno, da je napisan nedvoumno in razumljivo za vse potencialne uporabnike, ne le za avtorje dokumenta. Če se srečamo s primerom, pri katerem

je mogoča razlaga na več načinov, je treba zahtevo dodatno definirati in izraze, ki imajo dvoumen pomen, posebej razložiti v slovarju dokumenta. Pogosto se zgodi, da uporabniki SZPO nimajo enakega predznanja, zato je pomembno, da so specifikacije zapisane v naravnem jeziku. Naslednja lastnost, ki jo mora izpolnjevati dober SZPO, je *popolnost* dokumenta. Dokument je popoln, kadar vsebuje vse pomembne zahteve, ki se nanašajo na funkcionalnosti, zmogljivosti, oblikovne omejitve ali zunanje vmesnike, definicije odgovorov produkta na vse identi-

ficirane vhodne podatke vseh situacij, označbe vseh tabel, slik ter diagramov ter definicije vseh izrazov in merskih enot. Če je kje v okviru dokumenta uporabljena oznaka »potrebno definirati«, ⁵ potem v večini primerov SZPO ni popoln. Pojavi pa se lahko situacija, pri kateri je mogoče tudi v popolnem dokumentu SZPO uporabiti oznako »potrebno definirati«. V tem primeru mora biti oznaka opremljena z opisom stanja, ki je pripeljal do nedefiniranosti, in dopolnilom, ki pojasnjuje postopek definiranja zahteve, kdo je odgovoren ter pripadajoči časovni rok za definiranje (Belfo, 2012; IEEE Recommended Practice for Software Requirements Specifications, 1998).

Priporočljivo je, da je SZPO notranje *konsistenten*, kar pomeni, da zahteve, zapisane v dokumentu, med seboj niso nasprotujoče. Pogosto prihaja do konfliktov pri opredelitvi karakteristik objektov iz realnega okolja. Do logičnih ali začasnih konfliktov prihaja tudi pri definiranju akcij. Tako lahko pride do primera, da dve ali več zahtev opisujejo enak objekt iz realnega okolja, a zanj uporabljajo različne izraze. Te lahko razrešimo z uporabo standardne terminologije in definicij. Dokument SZPO zadošča lastnosti *razvrstitve* glede na pomembnost in/ali stabilnost, če ima vsaka zahteva identifikator, ki ustrezno označuje pomembnost oziroma stabilnost zahteve. Tipično vse zahteve v dokumentu niso enako pomembne. Zahteve lahko na podlagi pomembnosti v grobem razdelimo na bistvene, pogojne in opsijske. Dokument SZPO ustreza lastnosti *preverljivosti*, če je preverljiva vsaka zahteva, zapisana v dokumentu. Zahteve veljajo za preverljive, če obstaja proces, s katerim lahko oseba oziroma naprava preveri, ali produkt izpolnjuje posamezno zahtevo. Če lastnost povežemo z lastnostjo nedvoumnosti, v splošnem velja, da vsaka zahteva, ki je dvoumna, ni preverljiva (Belfo, 2012; IEEE Recommended Practice for Software Requirements Specifications, 1998).

SZPO zadošča lastnosti *spremenljivosti*, če sta struktura in stil dokumenta takšni, da lahko vsako spremembo zahtev izvedemo enostavno in konsistentno. Pri tem se ohranita struktura in stil dokumenta. Dokument je lažje spremenljiv, če ima SZPO usklajeno in enostavno strukturo s kazalom, se izogiba redundanci in ima zahteve predstavljene ločeno od drugih. Zadnja izmed lastnosti dobrega SZPO

je *sledljivost*. Dokument je sledljiv, če je izvor vsake zahteve jasen in omogoča sklicevanje vsake zahteve v nadaljnjem razvoju. Priporoča se sledljivost nazaj, kar pomeni sledljivost do prejšnjih korakov v razvoju, in sledljivost naprej, kar pomeni sledljivost do vseh dokumentov, ustvarjenih na podlagi SZPO (Belfo, 2012; IEEE Recommended Practice for Software Requirements Specifications, 1998).

4.2 Struktura dokumenta SZPO

Struktura dokumenta specifikacije zahtev programske opreme je bila že natančno raziskana in preučena ter opisana v številnih delih. Okvirna poglavja v svojih priporočilih in standardih predlagajo posamezne organizacije, ki se ukvarjajo s vsebino dokumenta SZPO (ESA Board for Software Standardisation and Control, 1991; IEEE Recommended Practice for Software Requirements Specifications, 1998; Software requirements specification DID, DoD Military Standard DI-IPSC-81433, 1994; NASA, 2009). Kljub vsem priporočilom in navodilom pa struktura dokumenta ni natančno predpisana in s tem strogo začrtana. Dokument si tako lahko vsak oblikuje glede na svoje želje in potrebe, pri čemer je pomembno, da je dokument zasnovan tako, da je bralcem v pomoč in ne v oviro.

Kljub omenjeni svobodi je pri izdelavi dokumenta SZPO priporočljivo upoštevati posamezne standarde, priporočila in dobre prakse. Te avtorjem ponujajo številna praktična priporočila, nasvete in predloge strukture dokumenta SZPO. Dodajajo tudi sezname konkretnih poglavij, za katera je priporočljivo, da jih vsebuje dobro oblikovan dokument SZPO. V rokah avtorjev tako ostane predvsem oblika in struktura posameznih podpoglavij, prav tako pa tudi predstavitev in oblikovanje posameznih zahtev. Te lahko avtorji predstavijo s pomočjo besedila, grafične podobe ali kako drugače, primerno vsebini (NASA, 2009).

Med pregledom literature smo identificirali tri vire, ki natančneje prikazujejo predlagano vsebino dokumenta SZPO v obliki poglavij in podpoglavij. Prvi je nastal v okviru organizacije NASA (NASA, 2009), drugi v okviru organizacije US DoD (Software requirements specification DID, DoD Military Standard DI-IPSC-81433, 1994), tretji pa v okviru organizacije IEEE (IEEE Recommended Practice for Software Requirements Specifications, 1998). Če primerjamo pregledana priporočila, ugotovimo, da vključu-

⁵ TBD – »to be determined«

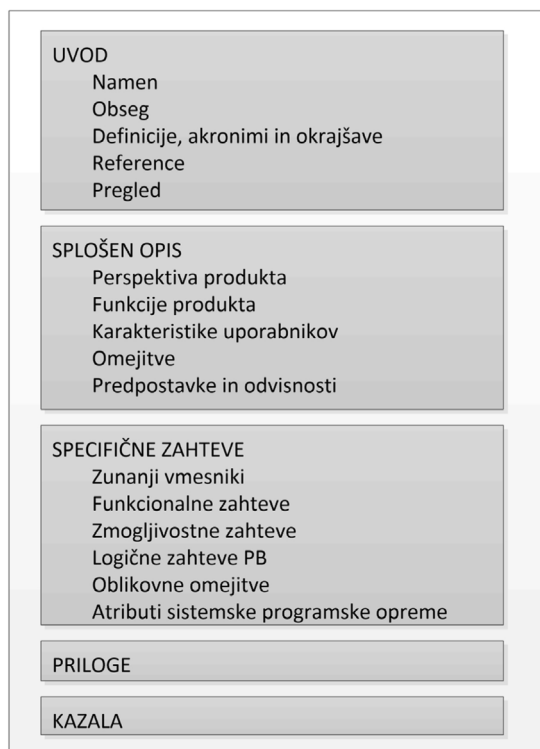
jejo podobna poglavja z razliko pri razporeditvi glavnih poglavij. Vsi predlagajo, da naj bo v dokumentu osrednje poglavje namenjeno zahtevam ter naj vsebuje podpoglavja, oblikovana glede na vrsto zahtev. Vsi predlagajo še vključitev splošnih informacij ter splošen opis dokumenta. Glede na pregledano literaturo smo ugotovili, da večina avtorjev, med drugimi tudi standard za programsko inženirstvo ESA, kot referenco za izdelavo dokumenta SZPO uporablja standard IEEE (Belfo, 2012; Davis idr., 1993; ESA Board for Software Standardisation and Control, 1991).

Tako smo se odločili, da predstavimo strukturo dokumenta, ki ga predstavlja organizacija IEEE. V dokumentu natančneje opisujejo priporočene prakse, navodila in predloge za izdelavo dokumenta specifikacije zahtev programske opreme. Prva verzija dokumenta je bila izdana leta 1984, danes pa IEEE 830 vključuje standard ISO/IEC/IEEE 29148 (Systems and software engineering – Life cycle processes – Requirements engineering, 2011). Omenjeni standard opisuje inženiring zahtev v okviru življenjskega cikla in tako vključuje tudi poglavje o oblikovanju in sestavi dokumenta SZPO. Standard IEEE je zelo vpliven

na omenjenem področju, saj gre za zapise neodvisne organizacije, ki želi pokriti čim več možnih področij in domen, prav tako pa gre za zelo berljiv, jedrnat in razumljiv dokument (Giakoumakis & Xylomenos, 1996).

Slika 2 prikazuje okvirno strukturo dokumenta SZPO glede na priporočila organizacije IEEE. Sama struktura ni prilagojena mobilnim aplikacijam, temveč gre za splošno oblikovana priporočila ter navodila. Glede na pridobljene izkušnje lahko trdimo, da se dokument SZPO za mobilne aplikacije razlikuje od dokumenta SZPO za tradicionalne programske produkte. Razlike se pojavijo predvsem pri poglavju *Specifične zahteve*. Zaradi tega se bomo osredinili predvsem na strukturo omenjenega poglavja. Poglavje o specifičnih zahtevah prikazuje zahteve za produkt, razgrajene do stopnje, ki omogoča razvijalcem načrtovanje produkta z določenimi lastnostmi, prav tako pa ponuja temelje testerjem, da produkt lahko preizkusijo glede na zapisane lastnosti. Navadno je v okviru dokumenta definiranih več skupin zahtev, osredinjenih na zunanje vmesnike, funkcionalnosti, zmogljivosti, logične zahteve podatkovne baze, oblikovne omejitve in sistemske attribute. Pri tem so v okviru specifikacij zunanjih vmesnikov predstavljeni podrobni opisi vseh vhodov v produkt in izhodov iz produkta. Funkcionalne zahteve definirajo akcije, ki jih izvaja produkt, zahteve zmogljivosti pa specifikirajo dinamične in statične numerične zahteve v povezavi s produktom ter pri sodelovanju uporabnika s produktom. Logične zahteve podatkovne baze specifikirajo logične zahteve za vse informacije, ki se hranijo v podatkovni bazi. V okviru specifičnih zahtev so navedene tudi oblikovne omejitve, ki so lahko posledica splošno sprejetih standardov oziroma dobrih praks ali pa posledica notranje oblikovanih stilnih vodičev. Med omejitve spadajo tudi omejitve strojne in programske opreme ter drugih virov, povezanih z mobilnimi telefoni. Zadnji so med specifičnimi zahtevami navedeni tudi sistemski atributi, med katere spadajo zahteve, povezane z zanesljivostjo, razpoložljivostjo, varnostjo, vzdrževanjem in prenosljivostjo (IEEE Recommended Practice for Software Requirements Specifications, 1998; Systems and software engineering – Life cycle processes – Requirements engineering, 2011).

Za vsak nekoliko zahtevnejši in obsežnejši produkt bo seznam zahtev zelo dolg in obsežen. V ta namen je priporočeno, da se zahteve po temeljitem



Slika 2: Primer strukture poglavij dokumenta SZPO po priporočilih IEEE (IEEE Recommended Practice for Software Requirements Specifications, 1998)

razmisleku organizirajo in razporedijo v kategorije, ki so najbolj smiselne in preproste za razumevanje. Ne obstaja namreč razdelitev, ki bi globalno ustrezala vsem sistemom, zato je pomembno, da se zahteve razporedijo specifično glede na produkt razvoja (IEEE Recommended Practice for Software Requirements Specifications, 1998; Systems and software engineering – Life cycle processes – Requirements engineering, 2011).

5 PRILAGODITEV DOKUMENTA SZPO ZA MOBILNE APLIKACIJE

Splošne lastnosti dokumenta SZPO, ki veljajo tudi pri izdelavi dokumenta za mobilne aplikacije, smo že predstavili. Zdaj pa se bomo osredinili na prilagoditve dokumenta za mobilne aplikacije. Predstavili bomo dele dokumenta SZPO za mobilne aplikacije, ki se nekoliko razlikujejo od dokumentov SZPO za tradicionalno programsko opremo, predvsem glede na tehnološke razlike ter lastnosti mobilnih telefonov. Pregled prilagoditev smo pripravili na podlagi pregleda literature ter praktičnih primerov, kar smo kasneje dopolnili s svojimi izkušnjami, pridobljenimi pri pripravi dokumenta SZPO za mobilne aplikacije, razvite v okviru raziskovalno-razvojnega projekta. Najprej bomo predstavili, kako se razlikujejo mobilne tehnologije in mobilne naprave in na kaj moramo paziti pri razvoju za mobilno okolje. Sledil bo povzetek praktičnih primerov dokumentov SZPO za mobilne aplikacije. Pri tem bomo identificirali posamezne razlike in prilagoditve dokumenta SZPO, nastale na podlagi specifičnih lastnosti mobilnih aplikacij. V zadnjem razdelku sledi predstavitev praktičnega primera izdelave dokumenta SZPO z poudarkom na pridobljenih praktičnih izkušnjah.

5.1 Specifične lastnostni mobilnih telefonov

Tradicionalna programska oprema nas obkroža že kar nekaj časa, medtem ko so programski produkti in aplikacije, namenjeni mobilnim telefonom, med nami zadnjih nekaj let. Zavedati se moramo, da se mobilni telefoni po nekaterih lastnostnih bistveno razlikujejo od osebnih računalnikov. To moramo upoštevati pri razvoju programskih produktov za mobilno okolje. Posebno je pomembno, da pri razvoju upoštevamo specifične lastnosti mobilnih telefonov, saj bomo le tako lahko izoblikovali uspešen in prilagojen mobilni programski produkt. Če povzamemo eno izmed definicij, ki pravi, da obstaja več tipov mobilnih tele-

fonov in da mednje lahko uvrstimo običajne mobilne telefone, mobilne telefone, ki omogočajo prenos podatkov, mobilne telefone s podporo protokolu WAP⁶ in pametne mobilne telefone (Hribar, 2001), ki so danes najbolj razširjeni. Pametni mobilni telefon je mobilna naprava, ki ima tipkovnico QUERTY in/ali tipkovnico na dotik ter za neodvisne aplikacije odprt operacijski sistem, kot npr. Android, iOS, Windows Phone in BlackBerry (Garner, 2011). Prav pametni mobilni telefoni so torej tisti, ki omogočajo namesnitev in uporabo različnih mobilnih aplikacij, zato je pomembno, da se pri razvoju aplikacij osredinimo na njihove specifične lastnosti, kot so npr. mobilna podatkovna povezava, GPS in Bluetooth, ter lastnosti, povezane z zasloni, občutljivimi na dotik, in jih upoštevamo.

Literatura navaja številne lastnosti mobilnih tehnologij in mobilnih naprav, ki vplivajo na razvoj programskih produktov v mobilni domeni. Te lastnosti imajo pomemben vpliv tudi pri izdelavi dokumenta SZPO in predhodnih korakih zbiranja zahtev. Če povzamemo le nekaj najbolj pogostih, ki jih navajajo avtorji, so to izzivi, povezani z lastnostmi povezovanja mobilnih aplikacij z drugimi aplikacijami, omejitve virov, veliko število dostopnih različnih mobilnih telefonov in zanje prilagojeni uporabniški vmesnik. Ta je tesno povezan s fizičnimi dimenzijami mobilnih telefonov in ločljivostjo zaslona. Specifične lastnosti so povezane tudi z načinom vnosa podatkov prek zaslona na dotik, varnostnimi omejitvami ter tudi z novimi družinami tako strojne kot tudi programske opreme, uporabljene pri razvoju mobilnih aplikacij (Kirubakaran & Karthikeyani, 2013; Satyanarayanan, 1996; Wasserman, 2010). Podobne lastnosti navajajo tudi drugi avtorji, npr. Economou idr. (2008). Kot še dodajajo, moramo biti pri izdelavi dokumenta SZPO pozorni na mobilni kontekst, načrtovanje aplikacij za prenosljivost, načrtovanje za uporabnike s širokim spektrom znanj in izkušenj, omejitve vhodnih in izhodnih poti ter še posebno na zbiranje informacij s pomočjo senzorjev na mobilnih telefonih. Vse to so lastnosti, na katere moramo paziti tudi pri koraku zbiranja in zapisa zahtev za mobilne aplikacije. Zahteve, kot so različne resolucije zaslonov, zmogljivostne omejitve in zahteve, pasovna širina idr., so v dokumentih SZPO za mobilne aplikacije obvezni ele-

⁶ Wireless Application Protocol

menti. Namreč, če zahtev ne prilagodimo omenjenim lastnostim in tega ustrezno ne zabeležimo, se lahko kar hitro zgodi, da končni produkt ne bo predstavljal zaključene celote, namenjene mobilnemu okolju.

5.2 Pregled praktičnih primerov SZPO za mobilne aplikacije

Po pregledu literature v obliki člankov in drugih dostopnih del smo na svetovnem spletu poiskali prosto dostopne dokumente SZPO, oblikovane v okviru razvoja mobilnih aplikacij. Ker je dokument SZPO pri izdelavi produkta praviloma poslovna skrivnost, je bilo na spletu prisotnih le nekaj omejenih primerov dokumentov. Dokumente smo sistematično pregledali in se pri tem osredinili predvsem na:

- *dopolnitve obstoječih standardnih poglavij* in v njih poiskali zahteve ter lastnosti, povezane z mobilnimi aplikacijami,
- *dodatna poglavja* v okviru dokumenta, ki obravnavajo tematiko, specifično za mobilne aplikacije.

Vsi pregledani dokumenti SZPO so bili oblikovani v skladu s priporočili IEEE 830, pri čemer upoštevajo predlagano strukturo in lastnosti, ki jih predlaga dokumentacija. Najprej smo se osredinili na *dopolnitve v okviru obstoječih standardnih poglavij*. Poglavje *Uvod* v dokumentih nima posebnih dopolnitev, vezanih na mobilne aplikacije oziroma mobilne tehnologije. Razlike pa so kar hitro vidne že v naslednjem poglavju, *Splošni opis*, in pripadajočih podpoglavjih. V podpoglavju *Perspektiva produkta* so v vseh dokumentih poudarili, da gre za razvoj mobilne aplikacije, ki je bila običajno povezana s storitvami drugih sistemov in rešitev. V istem podpoglavju so v nekaterih dokumentih dodali dodatno podpoglavje *Operacijsko okolje*, v katerem so navedli tip operacijskega sistema, za katerega bo razvita aplikacija (npr. Android, iOS ali Windows Phone), ter dodali tudi najnižjo verzijo operacijskega sistema, podprto v okviru projekta. V nekaterih dokumentih je bila navedena tudi referenčna naprava, za katero je mobilna aplikacija optimizirana.

Največ razlik in dopolnitev v primerjavi z dokumenti SZPO tradicionalne programske opreme je opaziti v poglavju *Specifične zahteve*. Če se znova najprej osredinimo na *dopolnitve v okviru obstoječih standardnih poglavij*, opazimo tudi specializacijo posameznih podpoglavij za domeno mobilnih aplikacij. Razlika je opazna že v podpoglavju *Zunanji vmesniki*. V okviru omenjenega poglavja so med drugim navedene tudi zahteve za uporabniški vmesnik, kar je

dopolnitev v primerjavi z dokumenti SZPO za tradicionalno programsko opremo. Pri tem v enem izmed pregledanih dokumentov dodajajo še opis posameznih lastnosti oblikovanja, specifičnih za posamezno platformo. Podobne podatke včasih zasledimo v okviru poglavja *Splošni opis* v podpoglavju *Omejitve*, v katerem lahko zasledimo tudi seznam odstopanj pri realizaciji rešitve na različnih platformah. Predvsem uporabniški vmesniki so običajno realizirani in prikazani nekoliko različno, v skladu z dobrimi praksami in stilnimi vodiči razvoja v okviru posameznega operacijskega sistema. V okviru specifičnih zahtev so v vseh pregledanih dokumentih SZPO *dodali dodatne zahteve*, specifične za mobilne telefone oziroma mobilne aplikacije. Dodali so zahteve, povezane s tehnologijami, ki jih uporabljajo mobilni telefoni, kot npr. GPS in Bluetooth. V zahteve so vpletli še hitrost mobilnega prenosa podatkov, prav tako pa so se dotaknili tudi velikosti zaslonov in s tem povezanih področij, predvsem z oblikovnega vidika in preglednosti.

5.3 Študija primera izdelave dokumenta SZPO za mobilno aplikacijo

V okviru raziskovalno-razvojnega projekta razvijamo produkt, pri katerem sodelujejo strokovnjaki z različnih področij: kineziologije, psihologije, medicine in informatike. Gre za rešitev, ki uporabnike spodbuja k bolj zdravemu, manj stresnemu in čim bolj aktivnemu življenju. Projekt predstavlja celovito skupino produktov, ki je sestavljena iz portalne rešitve in mobilnih aplikacij. Prav pri razvoju le-teh aktivno sodelujemo in tako pripomoremo k ustvarjanju mobilnih aplikacij za različne mobilne platforme, Android, iOS in BlackBerry.

Na začetku razvoja smo se kaj kmalu srečali z izzivom zbiranja in zapisa zahtev ter oblikovanja dokumenta SZPO za mobilne aplikacije. Zavedali smo se, da bo dokument SZPO za mobilne aplikacije nekoliko drugačen, dopolnjen in prilagojen glede na že znane dokumente SZPO, oblikovane za tradicionalno programsko opremo. Izziv nas je spodbudil, da smo se lotili *pregleda literature*, ki obravnava izdelavo dokumenta zahtev za mobilne aplikacije. Pri pregledu literature in praktičnih primerov smo bili pozorni na priporočila in dobre prakse, prav tako pa tudi na dopolnitve obstoječih poglavij ter dodatna poglavja v dokumentu, potrebna za popolno definiranje zahtev za uspešen razvoj mobilnih aplikacij.

Ko smo se seznanili z dobrimi praksami oblikovanja dokumentov SZPO za mobilne aplikacije, smo začeli z zbiranjem zahtev in oblikovanjem pripadajočega dokumenta. Pri oblikovanju zahtev so sodelovali tako strokovnjaki kineziologije, medicine in psihologije, ki so skrbeli za vsebinsko področje, kot tudi strokovnjaki s področja informatike, ki so skrbeli za tehnične zahteve. Oblikovanje ter izdelava dokumenta SZPO je potekala po več korakih.

- Postavili smo okvirno strukturo dokumenta, in sicer na podlagi dobrih praks, predstavljenih v standardih IEEE 830 in ISO/IEC/IEEE 29148.
- Vpeljali smo dodatna poglavja oziroma dopolnili obstoječa s specifičnimi podatki ali zahtevami za mobilne aplikacije. Poglavja smo glede na pregledane praktične primere in lastne praktične izkušnje ustrezno dodali oziroma razširili.
- Začeli smo z dodajanjem vsebine v dokument SZPO, pri čemer smo zajeli vse potrebne zahteve z različnih področij. Pri razvoju so sodelovali tudi razvijalci, ki so na podlagi preteklih izkušenj z razvojem mobilnih aplikacij predlagali nekatere dopolnitve, ki poskrbijo, da ne pride do podvajanja truda in spreminjanja že razvitih funkcionalnosti. Dokument smo sproti pregledovali in po potrebi dopolnjevali.
- Dokument SZPO je pregledal in potrdil projektni svet.

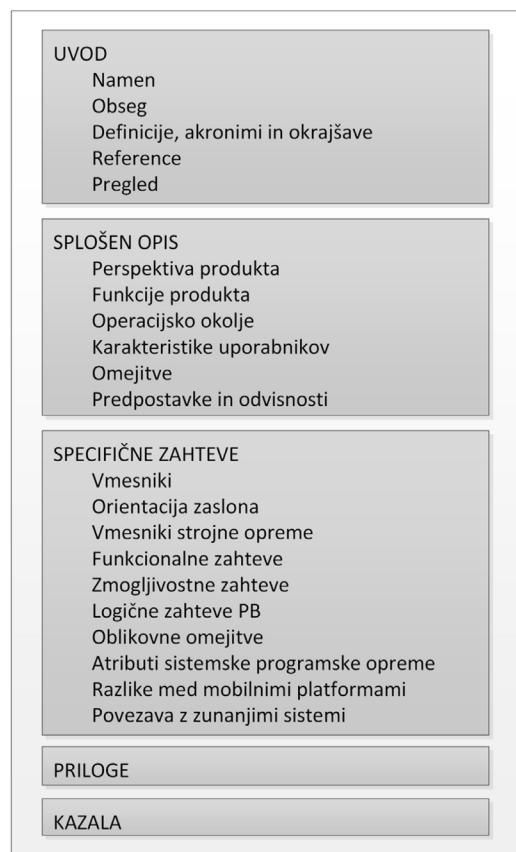
Če povzamemo *dopolnitve obstoječih* po standardu oblikovanih *poglavij* s podatki, vezanimi na lastnosti mobilnih aplikaciji, smo v okviru SZPO za aplikacijo v razvoju dodali:

- v poglavju *Splošni opis* smo v podpoglavju *Perspektiva produkta* navedli, da gre za razvoj mobilne aplikacije in ustrezno predstavili namen in način sodelovanja s storitvami v oblaku, portalno rešitvijo in podprtimi zunanji vmesniki; v podpoglavju *Omejitve* smo navedli najnižjo podprto verzijo operacijskega sistema in nekatere dodatne omejitve, vezane na lastnosti mobilnih telefonov: določili smo najmanjšo podprto ločljivost zaslona, ločljivost zaslona, optimizirano za aplikacijo v razvoju, priporočljivo pasovno širino za optimalen prenos multimedijskih vsebin in izvedbo sinhronizacije ter delovanje vmesnikov GPS in Bluetooth, ki omogočata pridobivanje podatkov;
- v poglavju *Specifične zahteve* smo v okviru podpoglavja *Vmesniki* pri uporabniških vmesnikih predstavili zaslonske maske, prilagajene za mobilne

telefone (t. i. StoryBoard), predvsem z vidika enostavnosti in preglednosti, dodali pa smo tudi zahtevo za orientacijo zaslona.

Če povzamemo še *dodana poglavja* v dokumentu SZPO, ki dopolnjujejo predlagano strukturo, lahko rečemo, da smo v dokument SZPO dodali:

- v poglavju *Splošni opis* smo dodali poglavje *Operacijsko okolje*, v katerem smo podrobneje navedli, za katere operacijske sisteme bo mobilna aplikacija razvita in na voljo;
- v poglavju *Specifične zahteve* smo dodali poglavje *Razlike med mobilnimi platformami*, ki je služilo kot temelj, da smo lahko na podlagi le enega dokumenta SZPO mobilno aplikacijo razvili za več različnih platform; na podlagi izkušenj z razvojem za različne platforme smo namreč ugotovili, da zaradi dobrih praks in načina razvoja posameznih platform lahko prihaja do manjših odstopanj pri realizaciji posameznih funkcionalnosti; razlike smo v poglavju razdelili v tri skupine, in sicer na razlike v izgledu, izvajanju in sočasnosti;



Slika 3: Struktura dokumenta SZPO za mobilne aplikacije

- v poglavju *Specifične zahteve* smo dodali nekatere zahteve, vezane na lastnosti mobilnih aplikacij in tehnologij. V podpoglavju *Zmogljivostne zahteve* smo podali zahtevo, povezano z mobilnim prenosom podatkov, ki omogoča optimalno delovanje produkta. V okviru podpoglavja *Vmesniki* smo dodali sekcijo *Orientacija zaslona*, v kateri smo navedli, v katerih primerih je ležeča orientacija zaslona omogočena in v katerih ne. V okviru podpoglavja *Vmesniki* smo dodali še opis vmesnikov strojne opreme, v katerem smo opisali delovanje strojnih gumbov v okviru aplikacije v skladu z dobrimi praksami funkcionalnosti posameznih gumbov, ter opis komunikacijskih vmesnikov, na podlagi katerih aplikacija sodeluje z zunanjimi napravami. Dodali smo še podpoglavje *Povezava z zunanjimi sistemi*, v katerem smo opisali sodelovanje mobilne aplikacije z zunanjimi sistemi.

Strukturo dokumenta SZPO za mobilne aplikacije, ki smo jo oblikovali, natančneje prikazuje slika 3. Struktura temelji na priporočilih standarda IEEE 830, dodana pa so poglavja, identificirana na podlagi pregledane literature, praktičnih primerov in praktičnih izkušenj v domeni razvoja mobilnih aplikacij.

6 SKLEP

Glede na preučeno literaturo, praktične primere ter izkušnje, pridobljene pri izdelavi dokumenta SZPO za mobilne aplikacije, lahko z gotovostjo ugotovimo, da se dokument specifikacije zahtev za mobilne aplikacije nekoliko razlikuje od dokumenta specifikacije zahtev za tradicionalne programske produkte. Razlika se pojavi že v začetnih poglavjih splošnega opisa, v katerih je treba poudariti, da gre za razvoj mobilne rešitve, dodati pa še mobilne platforme, za katere razvijamo aplikacijo. Priporočeno je zapisati tudi omejitve, povezane s tehnologijami, kot sta GPS in Bluetooth, ki sta pomembni tehnologiji mobilnih telefonov, ter omejitve, povezane z velikostjo in ločljivostjo zaslona mobilnega telefona ter z drugimi viri.

Še več razlik se pojavi pri definiranju specifičnih zahtev, npr. pri zahtevah zmogljivosti. Tam je smiselno dodati zahteve, povezane z mobilnim prenosom podatkov, in zahteve, povezane z uporabniškimi vmesniki. Pri zadnjih je treba pri oblikovanju upoštevati omejeno velikost in ločljivost zaslona, prav tako pa določiti tudi podporo različnim orientacijam zaslona pri določenih funkcionalnostih. Če aplikacijo razvijamo za več mobilnih platform, je

priporočljivo predstaviti tudi dopustne in potrebne razlike med platformami, predvsem zaradi specifik posameznih mobilnih operacijskih sistemov.

Če povzamemo, lahko na podlagi praktičnih izkušenj in pregledane literature ugotovimo, da se v dokumentih SZPO za mobilne aplikacije pojavljajo nekateri elementi in poglavja, ki jih ne zasledimo v dokumentih SZPO za tradicionalne programske produkte. Predlagamo, da je za doseganje dobrih praks v dokument SZPO za mobilne aplikacije primerno vključiti:

- pojasnilo, da gre za razvoj mobilne aplikacije, ter dodatne morebitne povezave z drugimi sistemi ali rešitvami,
- omejitve, povezane s platformo razvojnega produkta, v okviru programskih jezikov ter dobrih praks,
- navedene omejitve delovanja funkcionalnosti za množico dostopnih mobilnih telefonov in operacijskih sistemov,
- omejitve in zahteve, povezane z omejenimi viri pomnilnika, energije, spomina in drugih omejenih elementov,
- morebitne zahteve in omejitve, povezane s souporabo virov,
- zahteve, povezane z mobilnimi omrežji ter internetnimi povezavami,
- zahteve in omejitve, povezane z grafičnim uporabniškim vmesnikom, ki ga določajo fizična velikost zaslona in njegova ločljivost, prav tako pa tudi dobre prakse in izoblikovani stilni vodiči,
- zahteve, povezane z vnosom podatkov prek zaslona na dotik, in
- zahteve, povezane s senzorji mobilnih telefonov, ki zajemajo podatke.

Dobro napisan dokument SZPO je trdna podlaga za nadaljnji razvoj ter aktivnosti v življenjskem ciklu razvojnega produkta. Zaradi tega je pomembno, da si za zbiranje zahtev vzamemo dovolj časa in dokument oblikujemo po tehtnem premisleku, v skladu s priporočili in dobrimi praksami. Le tako bomo kot rezultat zbiranja zahtev dobili produkt, ki bo skozi ves razvojni cikel v pomoč, hkrati pa bo pomagal najti odgovore na vprašanja, ki se bodo morda pojavila pri nadaljnjem razvoju.

ZAHVALA

Študija je nastala v okviru programa usposabljanja mladih raziskovalcev, ki ga izvaja Javna agen-

cija za raziskovalno dejavnost RS. V članku so med drugim uporabljeni tudi nekateri rezultati projekta, izvedenega v okviru Razvojnega centra IKT Savinja Žalec. Operacija se izvaja v okviru Operativnega programa krepitev regionalnih razvojnih potencialov za obdobje 2007–2013 v okviru izvajanja prve razvojne prioritete Konkurenčnost podjetij in raziskovalna odličnost, prednostne usmeritve Izboljšanje konkurenčnih sposobnosti podjetij in raziskovalna odličnost.

VIRI IN LITERATURA

- [1] Belfo, F. (2012). People, Organizational and Technological Dimensions of Software Requirements Specification. *Procedia Technology*, 5(0), 310–318. doi:<http://dx.doi.org/10.1016/j.protcy.2012.09.034>.
- [2] Boehm, B. W. (1984). Verifying and Validating Software Requirements and Design Specifications. *Software, IEEE*. doi:10.1109/MS.1984.233702.
- [3] Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., ... Theofanos, M. (1993). Identifying and measuring quality in a software requirements specification. *Software Metrics Symposium, 1993. Proceedings., First International*. doi:10.1109/METRIC.1993.263792.
- [4] Doherty, G., McKnight, J. & Luz, S. (2010). Fieldwork for requirements: Frameworks for mobile healthcare applications. *International Journal of Human-Computer Studies*, 68(10), 760–776. doi:<http://dx.doi.org/10.1016/j.ijhcs.2010.06.005>.
- [5] Economou, D., Gavalas, D., Kenteris, M. & Tsekouras, G. E. (2008). Cultural applications for mobile devices: Issues and requirements for authoring tools and development platforms. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(3), 18–33. doi:10.1145/1462141.1462145.
- [6] ESA Board for Software Standardisation and Control. (1991). ESA Software Engineering Standards. Prentice-Hall International. Dostopno na <ftp://ftp.estec.esa.nl/pub/wm/anonymo-us/wme/bssc/PSS050.pdf> (1. 8. 2013).
- [7] Finkelstein, A. & Savigni, A. (2001). A Framework for Requirements Engineering for Context-Aware Services. In *1st International Workshop From Software Requirements to Architectures*.
- [8] Garner, R. (2011). *Mobile Payments: The importance of trust and familiarity and the need for co-operation* (str. 1–12).
- [9] Gartner. (2012). Gartner Says Worldwide Sales of Mobile Phones Declined 3 Percent in Third Quarter of 2012; Smartphone Sales Increased 47 Percent. Dostopno na <http://www.gartner.com/newsroom/id/2237315> (19. 2. 2013).
- [10] Giakoumakis, E. A. & Xylomenos, G. (1996). Evaluation and selection criteria for software requirements specification standards. *Software Engineering Journal*.
- [11] Gil, D., Andersson, J., Milrad, M. & Sollervall, H. (2012). Towards a Decentralized and Self-Adaptive System for M-Learning Applications. *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on*. doi:10.1109/WMUTE.2012.37.
- [12] Hammer, T. F., Huffman, L. L. & Rosenberg, L. H. (1998). Doing Requirements Right the First Time. Dostopno na <http://www.crosstalkonline.org/storage/issue-archives/1998/199812/199812-Hammer.pdf>.
- [13] Hofmann, H. F. & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *Software, IEEE*. doi:10.1109/MS.2001.936219.
- [14] Hribar, U. (2001). Tehnologije za mobilno poslovanje. V *Dnevi slovenske informatike*.
- [15] IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (str. 39).
- [16] IEEE Recommended Practice for Software Requirements Specifications. (1998). *IEEE Std 830-1998*. doi:10.1109/IEEE-STD.1998.88286.
- [17] IEEE Standard Glossary of Software Engineering Terminology. (1990). *IEEE Std 610.12-1990*. doi:10.1109/IEEE-STD.1990.101064.
- [18] ITU. (2013). The World in 2013 - ICT Facts and Figures. Dostopno na <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf> (22. 7. 2013).
- [19] Kamaruddin, K. A., Yusop, N. S. M. & Ali, M. A. M. (2011). Using activity theory in analyzing requirements for mobile phone application. *Software Engineering (MySEC), 2011 5th Malaysian Conference in*. doi:10.1109/MySEC.2011.6140636.
- [20] Kirubakaran, B. & Karthikeyani, V. (2013). Mobile application testing – Challenges and solution approach through automation. *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, 79–84. doi:10.1109/ICPRIME.2013.6496451.
- [21] Maccari, A. (1999). The challenges of requirements engineering in mobile telephones industry. *Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on*. doi:10.1109/DEXA.1999.795189.
- [22] MobiThinking. (2012). Global mobile statistics 2012 Part A: Mobile subscribers; handset market share; mobile operators. Dostopno na <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#subscribers> (19. 2. 2013).
- [23] NASA. (2009). NASA Software Engineering Requirements. Dostopno na August 01, 2013, from http://nodis3.gsfc.nasa.gov/npg_img/N_PR_7150_002A/N_PR_7150_002A_.pdf.
- [24] Portio Research. (2012). Your Portio Research Mobile Factbook 2012. Dostopno na <http://www.portioresearch.com/en/free-mobile-factbook.aspx> (19. 2. 2013).
- [25] Rational. (1998). Rational Unified Process, Best Practices for Software Development Teams. Dostopno na http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (1. 8. 2013).
- [26] Salazar, M. G., Mitre, H. A., Olalde, C. L. & Sanchez, J. L. G. (2012). Proposal of Game Design Document from software engineering requirements perspective. *Computer Games (CGAMES), 2012 17th International Conference on*. doi:10.1109/CGAMES.2012.6314556.
- [27] Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing – PODC '96*, 1–7. doi:10.1145/248052.24805.
- [28] Software development and documentation, DoD Military Standard MIL-STD-498. (1994). Washington, DC, USA: US Department of Defence.
- [29] Software requirements specification DID, DoD Military Standard DI-IPSC-81433. (1994). Washington, DC, USA: US Department of Defence.
- [30] Systems and software engineering – Life cycle processes – Requirements engineering. (2011). *ISO/IEC/IEEE 29148:2011(E)*. doi:10.1109/IEEESTD.2011.6146379.
- [31] The New York Times. (2011). One Million Mobile Apps, and Counting at a Fast Pace. Dostopno na <http://www.nytimes.com/2011/12/12/technology/one-million-apps-and-counting.html> (19. 2. 2013).

- [32] Wasserman, A. I. (2010). Software engineering issues for mobile application development. *Proceedings of the FSE/SDP workshop on Future of software engineering research – FOSER '10*, 397. doi:10.1145/1882362.1882443.
- [33] Whitfield, K. (2013). Fast growth of apps user base in booming Asia Pacific market. *Portio Research*. Dostopno na <http://www.portioresearch.com/en/blog/2013/fast-growth-of-apps-user-base-in-booming-asia-pacific-market.aspx> (22. 7. 2013).

■

Tina Schweighofer je mlada raziskovalka na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, kjer je leta 2012 magistrirala in si pridobila strokovni naziv magistrica inženirka informatike in tehnologij komuniciranja. Trenutno je doktorska študentka bolonjskega doktorskega študija Računalništvo in informatika. Njeno raziskovalno delo obsega mobilne tehnologije, razvoj mobilnih aplikacij ter pripadajoče postopke testiranja, dokumentiranje razvoja mobilnih aplikacij, prav tako pa tudi razvojni cikel mobilnih aplikacij s pristopom programskih produktivnih linij.

■

Marjan Heričko je redni profesor za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, kjer je nosilec več predmetov, ki so v pristojnosti Inštituta za informatiko. Je namestnik predstojnika inštituta ter vodi laboratorija za informacijske sisteme. Doktoriral je leta 1998 na Univerzi v Mariboru na področju zagotavljanja kakovosti objektno orientiranega razvoja programske opreme. Njegovo raziskovalno delo zajema vsa področja razvoja sodobnih informacijskih rešitev in storitev s poudarkom na naprednih pristopih k modeliranju in načrtovanju informacijskih sistemov, načrtovalskih vzorcih in metrikah.