

Dizajn strukturne matrice za identifikaciju i evaluaciju kritičnih faktora uspeha softverskih projekata

ZORICA M. MITROVIĆ, Univerzitet u Beogradu,

Fakultet organizacionih nauka, Beograd

EMA D. GLIGORIJEVIĆ, Univerzitet u Beogradu,

Fakultet organizacionih nauka, Beograd

JOVANA Đ. RAKIĆEVIĆ, Univerzitet u Beogradu,

Fakultet organizacionih nauka, Beograd

Originalni naučni rad

UDC: 004.4:005.51

DOI: 10.5937/tehnika2201099M

Kritični faktori uspeha predstavljaju fokalnu tačku diskursa u oblasti uspeha softverskih projekata, kako u akademskim, tako i u krugovima profesionalaca u softverskoj industriji. Međutim, redukcionistički pristup u definisanju postojećih modela kritičnih faktora uspeha softverskih projekata, rezultirao je fokusom postojeće literature na identifikaciju specifičnih kategorija kritičnih faktora uspeha, zanemarujući interakcije koje postoje u okviru ovih modela između različitih grupa kritičnih faktora uspeha, pa i u okviru samih grupa. Svrha ovog rada je da ukaže na gep koji postoji u literaturi, a koji se odnosi na kauzalnost između elemenata modela kritičnih faktora uspeha i da predloži novi koncept za identifikaciju i evaluaciju kritičnih faktora uspeha softverskih projekata zasnovan na modelu koji prepoznaje interakcije između kritičnih faktora uspeha i posmatra uspeh kao jedan od ishoda nelinearne kauzacije u modelima. Pristup je zasnovan na teoriji grafova i dizajn strukturnim matricama, čija je osnovna svrha da podrže modelovanje interakcija projektnog sistema. Ključan rezultata ovog rada je alat za identifikaciju i evaluaciju interakcija između elemenata u mreži kritičnih faktora uspeha, čija krajnja svrha je prepoznati kritičnosti faktora na osnovu snage uticaja u celom projektnom sistemu.

Ključne reči: *kritični faktori uspeha, softverski projekti, pristup sistemskog razmišljanja, teorija grafova, dizajn strukturne matrice*

1. UVOD

Ulazak u četvrtu industrijsku revoluciju je uslovio da napredak i razvoj društva i privrede zavise od sposobnosti da se promeni ne samo način na koji se radi i doprinosi, već i način na koji se misli i poima sadašnjost i budućnost. Eksplozivni rast softverske industrije je u prethodnom periodu omogućio transformaciju i razvoj u svim sektorima, kao i na svim tržištima kroz unapređenje produktivnosti, efikasnosti i inovativnosti. Međutim, iako je softverska industrija prepoznata kao pokretačka snaga svih aspekata modernog društva, razvoj softvera je daleko od „savršenog“ procesa [1]. Naprotiv, postojeća literatura i praksa, ukazuju na nikada prisutniji izazov prevalencije neus-

peha u softverskoj industriji [2] [3] [4]. U pitanju je problem koji je još 60-ih godina prošlog veka prepoznat kao „softverska kriza“ [5], a za čije rešavanje nije pronađen odgovarajući pristup u okviru postojećih redukcionističkih paradigmi.

U softverskoj industriji, potreba da se produbi razumevanje pojma uspeha projekata i nađu načini da se upravlja performansama projekata, rezultovalo je naporima naučne i stručne zajednice da definišu i implementiraju koncept kritičnih faktora uspeha (KFU).

U opštem smislu, u literaturi projektnog menadžmenta, KFU su prepoznati kao faktori koji su direktno vezi sa uspehom projekta, odnosno faktori koji mogu uticati na povećanje verovatnoće nastupanja uspeha projekta [6] [7] [8] [9] [10] [11], pod uslovom da se uspeh projekta posmatra kao ishod delovanja elemenata projektnog sistema, a da su elementi pod kontrolom pojedinaca, odgovornih za isporuku performansi u oblastima ključnim za uspeh projekta. Sa ciljem da se što sveobuhvatnije istraže faktori uspeha

Adresa autora: Zorica Mitrović, Univerzitet u Beogradu, Fakultet organizacionih nauka, Beograd, Jove Ilića 154

e-mail: zorana.mitrovic@fon.bg.ac.rs

Rad primljen: 09.02.2022.

Rad prihvaćen: 17.02.2022.

projekata u softverskoj industriji, nastao je veliki broj različitih klasifikacija KFU softverskih projekata [1] [12] [13]. Zahvaljujući analitičko-redukcionističkom pristupu zastupljenom u konvencionalnim teorijama projektnog menadžmenta, a pod pretpostavkom asocijacije, postojeće klasifikacije služe da se kreiraju modeli KFU zasnovani na linearnoj vezi između faktora i uspeha. U pitanju su modeli, zasnovani na pretpostavci da se uspeh projekta može kontrolisati na osnovu ograničenog broja merljivih varijabli. Međutim, softverske projekte karakteriše kompleksnost u upravljanju granicama projekta, preko kojih se odvija interakcija između internih i eksternih stejkholdera. Pritisak na interne strukture se intenzivira, budući da se uspeh projekta ocenjuje na osnovu eksternih očekivanja, istovremeno vršeći pritisak da se projekat isporuči u okviru interno postavljenih ograničenja. U skladu sa tim, identifikovana su dva ključna nedostatka u postojećim modelima KFU softverskih projekata.

Kao prvo, postojeći modeli ne analiziraju na odgovarajući način interakcije između faktora softverskih projekata u internom i eksternom okruženju projekta. Jedna od osnovnih zamerki ovim modelima je da ne uzimaju u obzir da se događaji ne dešavaju u izolaciji, već su rezultat povratnih sprega. Na primer, ukoliko bi se analizirala dinamika veze između razvojnog tima i klijenta, može se desiti da razvojni tim nije dovoljno uključivao klijente u proces razvoja softvera zato što klijenti nisu bili zainteresovani da budu uključeni u proces. Međutim, isto tako može se desiti da klijenti nisu pokazivali interesovanje da budu uključeni u proces zato što razvojni tim nije bio dovoljno zainteresovan da ih uključi u proces. Identifikacija ovakve interakcije između KFU nije moguća u postojećim modelima, budući da su zasnovani na linearnoj kauzaciji, koja pretpostavlja jednosmeran tok događaja, odnosno razumevanje odnosa KFU i uspeha kroz uzročno-posledične veze.

Kao drugo, ne postoji pristup kojim bi se mogla izvršiti kvantifikacija opisanih interakcija. U modelima zastupljenim u literaturi, kao i u metodologijama na osnovu kojih su definisani modeli, predviđeno je da se kvantifikacija kritičnosti KFU vrši preko subjektivnih ocena prioriteta datih od strane evaluatora i to najčešće pojedinaca koji će i biti odgovorni za isporuku performansi u ovim oblastima. Kao i u prethodnom slučaju, predviđeno je da postoji povratna sprega između faktora, povratna kauzacija u vremenu, efekat nastajanja i ostali ishodi nelinearne kauzacije. Shodno tome, predviđeno je da je projekat u stvari sistem, koji predstavlja mrežu elementarnih procesora, a čije ponašanje se može razumeti samo primenom neparame-tarskih i nelinearnih modela.

Prema tome, u ovom radu, autori se osvrću na navedene probleme i predstavljaju predlog analitičkog

pristupa za identifikaciju i evaluaciju KFU softverskih projekata zasnovan na razumevanju da je projekat sistem i da interakcija između elemenata sistema određuje krajnji ishod, odnosno uspeh ili neuspeh projekta. Koncept je baziran na zajedničkoj upotrebi teorije grafova i Dizajn strukturalnih matrica (Design Structure Matrices - DSM) i naglašava važnost razumevanja koncepta interakcija elemenata u projektnom sistemu, kao i važnost evaluacije tih interakcija.

Rad je strukturiran u 6 poglavlja. Poglavlje 2 i 3 predstavljaju teorijsku pozadinu koncepta merenja uspeha i KFU softverskih projekata. Poglavlje 4 predstavlja konceptualnu ideju za zajedničku upotrebu teorije grafova i DSM za evaluaciju KFU u softverskim projektima, dok poglavlje 5 daje uvid u konceptualnu ideju analitičkog pristupa za identifikaciju i evaluaciju KFU softverskih projekata. Na kraju u poglavlju 6 prikazana su zaključna razmatranja, zajedno sa budućim pravcima istraživanja.

2. USPEH SOFTVERSKIH PROJEKATA

U teoriji konvencionalnog upravljanja projektima, uspeh projekta je posmatran kao binarna funkcija, odnosno projekat se može smatrati uspešnim ili neuspešnim [14]. Takođe, konvencionalan pristup je podrazumevao da se uspeh meri sposobnošću da se dogovoreni obuhvat isporuči u okviru definisanog roka i budžeta, kao i u skladu sa dogovorenim kvalitetom [17] [18] [19]. Prema tome, većina studija definiše uspeh kroz kriterijume/ograničenja vezana za budžet, rokove i funkcionalnosti, odnosno neuspeh kao prekoračenja ograničenja u vezi sa budžetom i rokovima ili dogovorenim funkcionalnostima [20]. Takva definicija uspeha sugerise da se projekat ne može smatrati uspešnim ukoliko bilo koji od navedenih kriterijuma nije ispunjen [21].

Pregled literature ukazuje da postoje najmanje tri različite kategorije uspeha, uspeh procesa upravljanja projektima, uspeh rezultata/proizvoda projekta i budući uspeh, koji se odnosi na poslovne i strateške aspekte uspeha projekta značajne za budućnost kompanije [7] [12] [14] [22]. Posledično, uspeh projekta se meri na različite načine [23] [24], usled velike divergencije u definicijama uspeha softverskih projekata [7] [16] [25] kao i potrebe da se koriste različiti kriterijumi uspeha za ocenu projekata sa različitim karakteristikama [13].

Sa ciljem da se produbi razumevanje stvarne pozicije projekata u kontekstu uspeha i neuspeha, istraživači u softverskoj industriji su došli na ideju uvođenja pojma uslovnog neuspeha (ili „challenged projects“), kao ocene koja se na skali nalazi između uspeha i neuspeha [4]. Na ovaj način je barem jednim delom dat odgovor na problem binarnog posmatranja uspeha projekta i omogućava posmatranja uspeha kao

složenog koncepta koji uzima u obzir različite dimenzije uspeha i omogućava balans između ovih dimenzija [14].

3. KRITIČNI FAKTORI USPEHA SOFTVERSKIH PROJEKATA

Od prvih pokušaja identifikacije i klasifikacije je KFU softverskih projekata [15] [16], lista KFU je značajno proširena. Belassi & Tuke-al [7], su bili među prvim autorima koji su dali sledeću klasifikaciju: faktori projekta, projektnog menadžera, članova projektnog tima, organizacije i eksternog okruženja. Fokusingajući se na agilne softverske projekte Chow & Cao [1] su grupisali 109 KFU u pet kategorija: organizacioni faktori, faktori vezani za ljude, zatim faktori vezani za procese, tehnički faktori i faktori vezani za projekte. Nasir & Sahibuddin [26] su identifikovali 26 faktora koji utiču na uspeh projekta i klasifikovali ih kao faktore vezane za ljude, procese i tehničke faktore. Slično, Taraweneh [27] sugeriše sledeću kategorizaciju: organizacioni, tehnički, vezani za ljude i faktore vezane za kulturu. Sledi, Ahimnisabwe et al. [13], koji su kreirali još jedan važan model i to na osnovu sistematičnog pregleda literature, u kome je identifikovano 37 KFU za projekte razvoja softvera i grupisao ih u organizacione, timske, faktore kupaca i faktore projekata. Chiyangwa & Mnkandla [28] su identifikovali KFU u oblasti agilnih projekata, kreirali su sledeće kategorije: organizacioni, ljudski, tehnološki i faktori procesa, takođe ispitali su vezu između očekivanih performansi i uspeha agilnih projekata. Sudhakar [12] je predložio jedan od najznačajnijih okvira u oblasti merenja uspeha softverskih projekata, zasnovan na konceptualnom modelu koji je ispitao 80 faktora. Autor je prepoznao sedam kategorija faktora uspeha: faktori komunikacije, faktori tima, organizacioni faktori, faktori okruženja, tehnički faktori, faktori projektnog menadžmenta i faktori proizvoda. Uz sveobuhvatnu kategorizaciju faktora uspeha, autor je uspeo da identifikuje vezu između ovih kategorija. Značaj ovog doprinosa je čak i veći, uzimajući u obzir da pored prepoznavanja identifikacije veza između grupa KFU, autor po prvi put naglašava i potrebu za kvantifikacijom istih. Iako ne postoji opšti konsenzus u vezi sa kategorizaciju KFU softverskih projekata [29], pregled literature pokazuje da najveći broj radova grupiše KFU u ključne oblasti, sugerišući da postoje alternativni okviri za kategorizaciju KFU [1] [12] [13].

Uprkos brojnim studijama, pregled literature pokazuje da postojeći modeli KFU softverskih projekata u fokusu imaju identifikaciju i kategorizaciju faktora uspeha, ali ne i razumevanje interakcije koja postoji između različitih grupa faktora u ovim modelima. U skladu sa tim, Sudhakar [12] je jedan od retkih autora

koji je identifikovao potrebu da se analiziraju veze između grupa faktora. Model KFU softverskih projekata, koji je prikazan u radu, predstavlja acikličan graf u kome su veze između grupa faktora predstavljene kao usmerene strelice, apstrahujući moguću interakciju i povratnu spregu između grupa faktora, pa i između faktora koji pripadaju istoj grupi. Ovako predstavljen model i dalje je pogodan za tradicionalnu statističku analizu, ali ne i za ispitivanje i kauzalno zaključivanje u vezi sa odnosima između KFU i uspeha projekta.

Ograničenja ovakvog pristupa se mogu opisati na sledeći način. Kao prvo, osnovna karakteristika modela KFU se odnosi na činjenicu da su KFU definisani kao uzrok, a uspeh kao posledica. U skladu sa tim, iako se radi o jednostavnim opservacijama asocijacije između događaja, zaključak o budućnosti se dobija na osnovu ekstrapolacije veza između uzroka i posledica. Ovakva vrsta analize i zaključivanja ne daje odgovore na pitanja u vezi sa mogućim scenarijima vezanim za intervenciju, koja podrazumeva da su veze između elemenata opservacije stabilne, ali da se parametri menjaju u skladu sa okolnostima. Kao drugo, u ovim modelima veza između KFU i uspeha je bila posmatrana kroz jednosmeran tok događaja od KFU do uspeha projekta.

Međutim, za preciznije sagledavanje ponašanja projekta kao sistema nedostaju: informacije u vezi sa višestrukim petljama povratnih sprega, koje postoje između faktora i uspeha; dvosmeran tok kauzacije između makro i mikro nivoa u okviru projektnog sistema; razumevanje povratne kauzacije u vremenu, odnosno da definisanje budućih ciljeva može kroz povratne veze da utiče na trenutne događaje; pojam ekvifinalnosti; i indeterminizam umesto determinizma, kao iluzije potpune kontrole.

Svrha ovog rada je da predloži novi pristup za identifikaciju i evaluaciju KFU softverskih projekata koji uzima u obzir da postoji interakcija između elemenata modela. Pristup je zasnovana na zajedničkoj upotrebi teorije grafova i DSM sa ciljem da se naglasi važnost interakcija koje postoje između elemenata projektnog sistema.

4. METODOLOGIJA

4.1. *Definisanje interakcija između kritičnih faktora uspeha*

U prethodnim poglavljima definisani su pojmovi uspeha projekta i KFU. Takođe, projekat je bio definisan kao projektni sistem, odnosno mreža elementarnih procesora, koji u slučaju ovog rada predstavljaju KFU. Shodno tome, pre razmatranja alata koji bi se mogli primeniti u okviru novog pristupa, a koji bi obuhvatio interakcije koje postoje između

elemenata modela, potrebno je definisati koncept interakcija KFU. U skladu sa tim, sledi definicija:

- Interakcija postoji između dva faktora KFU_i i KFU_j ukoliko pojavljivanje KFU_j može zavisiti od pojave KFU_i i obrnuto.
- Interakcija je orijentisana od strane jednog faktora ka drugom, što znači da je prioritet veza između faktor važan za definisanje interakcije KFU.

Definisanje interakcija KFU omogućava da se uvedu Dizajn strukturne matrice - DSM i teorija grafova kao alati za identifikaciju i evaluaciju interakcija KFU.

4.2. Dizajn strukturna matrica i teorija grafova: alati i definicije

Dizajn strukturne matrice (Design Structure Matrix - DSM) su kvadratne matrice, a koje predstavljaju dizajn projektnog sistema. Kao celina, pružaju veoma jednostavan, vizualan način da se izvrši analiza projektnog sistema koji je kompleksan usled interakcija i međuzavisnosti između brojnih elemenata sistema [30]. Pristup DSM je široko korišćen za modelovanje kompleksnih sistema u različitim oblastima [31] [32] [33] [34] [35] [36] [37], a posebno u oblasti projektnog menadžmenta [31] [38] [39] [40].

DSM je alat koji se koristi da izlista sve elemente sistema, kao i interakcije koje postoje između ovih elemenata. Elementi sistema se definišu u prvom redu i prvoj koloni matrice. Dijagonalne ćelije predstavljaju elemente sistema, a ćelije van dijagonale ukazuju na interakcije ili zavisnosti između njih.

Za potrebe ovog rada, sastavni elementi DSM matrice su KFU (slika 1) i ćelije van dijagonale u i -tom redu i j -toj koloni, koje ukazuju na vezu između dva KFU. Ovo znači da KFU u j -toj koloni utiče na KFU u i -tom redu, što znači da i -ti KFU prima input od j -tog KFU i j -ti KFU šalje output i -tom KFU. Dakle, DSM se koristi da opiše informacije neophodne za početak određene aktivnosti i smer u kome vode ove informacije [36]. U kontekstu ovog rada, cilj je prepoznati međuzavisnosti između KFU.

	KFU 1	KFU 2	KFU 3	KFU 4	KFU 5
KFU 1					
KFU 2					
KFU 3					
KFU 4					
KFU 5					

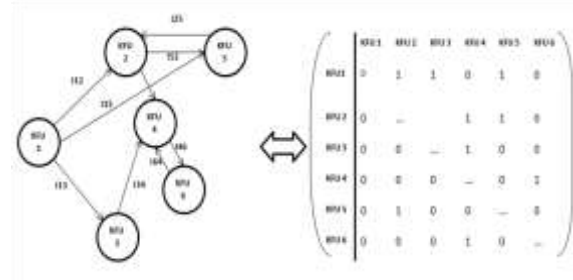
Slika 1 – Dizajn strukturna matrica kritičnih faktora uspeha

Implementacija pristupa DSM za identifikaciju i analizu interakcija KFU može biti argumentovana na

sledeći način. Kao prvo, pristup DSM omogućava reprezentaciju velikog broja KFU i njihovih veza na veoma koncizan način, koji je bio skoro nemoguć do sada. Kao drugo, DSM može biti korišćen da ukaže na prisustvo petlje povratnih sprega, što znači da može da otkrije ukoliko postoje samo-ojačavajuće veze između KFU. Konačno, pristup DSM omogućava primenu različitih tehnika analize uključujući kvantifikaciju interakcija između elemenata sistema neophodnih za proučavanje kompleksnih sistema. DSM se fokusiraju na međuzavisnosti i mogu omogućiti analizu promene uticaja KFU u različitim fazama životnog ciklusa projekta razvoja softvera.

Za predlog analitičkog pristupa za identifikaciju i evaluaciju KFU, predlaže se primena matrice reprezentacije modela kompleksne projektne mreže sastavljene iz KFU i to zajedno sa teorijom grafova. Kako bi se to uradilo, u ovom radu biće korišćen pristup sistemskog razmišljanja u skladu sa radovima Le Moigne [40] i Vidal [36].

Uzimajući u obzir osnovne koncepte i elementarne alate teorije grafova [41], kao i definicije interakcije faktora koja je predložena u ovom radu, mreža KFU projektnog sistema može biti definisana kao konačan usmeren težinski graf (slika 2). Može se primetiti da je DSM matrica slična matrici susedstva u teoriji grafova. Razlika je u dijagonalnim elementima, koji su nula u matrici susedstva. U svrhe ovog rada, cilj je da se koristi teorijski koncept DSM-a za identifikaciju KFU kao objekata i da se iz toga razvija matrica susedstva KFU kao usmeren težinski graf, u kome su interakcije između KFU predstavljene pomoću ivica i vrhova.



Slika 2 – Mreža KFU i matrica susedstva

Na osnovu prikazanih alata i definisanja njihove funkcionalne uloge u razvoju modela KFU, u nastavku sledi prikaz predloga analitičkog pristupa za identifikaciju i evaluaciju interakcija u ovim modelima.

5. ANALITIČKI PRISTUP ZA IDENTIFIKACIJU I EVALUACIJU INTERAKCIJA U MODELIMA KRITIČNIH FAKTORA USPEHA

Svrha predloga analitičkog pristupa koji će biti predstavljen u nastavku je da se odgovori na dva ključna problema u vezi sa modelima KFU softverskih projekata: 1) postojeći modeli KFU ne analiziraju na

odgovarajući način interakcije između faktora softverskih projekata; 2) ne postoji pristup kojim bi se mogla izvršiti kvantifikacija interakcija između elemenata modela. Da bi se dao odgovor na ove probleme, bilo je potrebno sagledati posledice primene redukcionističko-analitičkog pristupa u razvoju modela KFU, ali i na mogućnosti primene kauzalnog zaključivanja i pristupa sistemskog razmišljanja kroz uvođenje informacija u vezi sa višestrukim petljama povratnih sprega, koje postoje između faktora i uspeha, zatim razumevanja da postoji dvosmeran tok kauzacije između makro i mikro nivoa u okviru projektnog sistema, kao i razumevanje povratne kauzacije u vremenu. Rešenje je pronađeno u definisanju interakcija i predlogu primene DSM i teorije grafova, čime je kreirana metodološka osnova za definisanje predloga analitičkog pristupa koji će omogućiti razvoj integrisanog modela KFU i uspeha projekta.

Pristup je organizovan u četiri koraka (slika 3). Svaki korak se sastoji od opisa procesa i očekivanih rezultata, kao i od opisa elementarnih alata i tehnika u svakom koraku.

Ovaj analitički pristup je konceptualni model u razvoju i njegova primena se može dobiti tek nakon sprovođenja niza empirijskih studija. Takođe, usled ograničenja obuhvata ovog rada, pojedini koncepti predloženi u pristupu nisu detaljno objašnjeni.

Proces bi trebalo da bude sproveden na sledeći način:



Slika 3 – Predlog analitičkog pristupa za identifikaciju i evaluaciju modela KFU softverskih projekata

6. DISKUSIJA I ZAKLJUČAK

Svrha ovog rada je bila da se da kritički osvrt na postojeće modele KFU softverskih projekata i da se ukaže na mogućnost unapređenja istih. Ključan rezultat ovog rada je analitički pristup zasnovan na zajedničkom korišćenju teorije grafova i DSM, dok se najznačajniji benefiti analitičkog pristupa mogu sumirati na sledeći način.

Kao prvo, identifikacija KFU i njihovih kauzalnih veza će omogućiti bolje razumevanje ponašanja softverskog projekta kao sistema koji evoluiraju. Kao

- Prvi korak analize je opis projektnog sistema i identifikacija KFU primenom Systemics for Complex Organisational Systems Design (SCOS'D), kao metodologije za organizacioni dizajn zasnovana na pristupu sistemskog razmišljanja. Rezultat ove faze bi trebalo da bude opis projektnog sistema i balansirani set KFU zasnovan na kategorizaciji koju je predložio Sudhakar [12].
- Drugi korak je identifikacija i evaluacija interakcija između faktora uspeha korišćenjem teorije grafova i DSM pristupa [31] [42]. Dva ključna rezultata koja bi trebalo da budu dobijena u ovoj fazi su struktura KFU i numerička matrica.
- Treći korak je identifikacija potencijalnih petlji između faktora i ispitivanja potencijala propagacije specifičnih faktora uspeha u okviru mreže faktora uspeha projekta. U skladu sa sugestijama Gebala and Eppinger [43], proces parcionisanja može biti korišćen za prikaz zavisnih faktora (faktora koji su uključeni u prioritetne veze), međuzavisnih faktora (faktori uključeni u međusobno zavisne veze) i nezavisnih faktora (faktori koji nisu povezani ni sa jednim drugim faktorom).
- Na kraju, četvrti korak uključuje klasterovanje KFU kako bi se analizirala kritičnost KFU na osnovu njihovog uticaja na različite dimenzije uspeha [14].

drugo, evaluacija interakcija između KFU kroz kvantifikaciju i analizu moguće propagacije specifičnih faktora u okviru mreže faktora uspeha projekta će omogućiti identifikaciju KFU sa najvećim uticajem u projektnom sistemu. Kao treće, klasterovanje KFU i mera performansi će omogućiti merenje uticaja KFU na različite dimenzije uspeha. Na kraju, rad predstavlja inovativan način za identifikaciju i analizu KFU fokusiranjem na interakcije faktora korišćenjem pristupa sistemskog razmišljanja i povezivanje faktora uspeha sa uspehom projekta. Dalja istraživanja u ovoj oblasti bi trebalo da budu bazirana na empirijskim

podacima i da daju osvrt na predložene korake analitičkog pristupa, kako bi se ispitala njihov izvodljivost i predlog unapređenja koji će dati prave benefite u nauci i praksi.

LITERATURA

- [1] Chow T. & Cao, D. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, No. 81, pp. 961-971, 2008.
- [2] Flyvbjerg B, Budzier A. Why your IT project may be riskier than you think. *Harvard Business Review*, Vol. 89, No. 9, pp. 83-85, 2011.
- [3] Bloch, M., Blumberg, S., Laartz, J. Delivering large-scale IT projects on time, on budget and value. *McKinsey Finance*, Vol. 45, pp. 28-35, 2013.
- [4] Standish Group. The CHAOS Report, *The Standish Group International, Inc, Dennis, MA*, 2014.
- [5] Naur, P. & Randell, B. Software engineering - Report on a conference sponsored by the NATO Science Committee. *NATO Scientific Affairs Division*, Brussels, Belgium, 1969. Preuzeto. sa <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>
- [6] Boynton A. C. & Zmud R. W, An assessment of critical success factors. *Sloan Management Review*, Vol. 25, No. 4, pp. 17-27, 1986.
- [7] Prabhakar G, What is project success: A literature review. *International Journal of Business and Management*, Vol. 9, No. 3, pp. 3-10, 2008.
- [8] Kerzner, H. *Project Management Best Practices: Achieving Global Excellence*. New York: John Wiley & Sons, Inc, 2010.
- [9] Aamir Khan, R., Spang, K. Critical success factors for international projects. *IDAACS*, Vol. 2, pp. 879-883, 2011.
- [10] Hairul M, Md Nasir M, Sahibuddin S, Critical success factors for software projects: A comparative study. *Scientific Research and Essays*, Vol 6, pp. 2174-2186, 2011.
- [11] Lech P, Causes and remedies for the dominant risk factors in enterprise system implementation projects: The consultant's perspective. *SpringerPlus*, Vol. 5, pp. 238, 2016.
- [12] Sudhakar, G. P, A model of critical success factors for software projects. *Journal of Enterprise Information Management*, Vol. 25, No. 6, pp. 537-558, 2012.
- [13] Ahimbisibwe A, Cavana R. & Daellenbach U, A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, Vol. 28, No. 1, pp. 7-33, 2015.
- [14] Dalcher, D. Rethinking success in software projects: looking beyond the failure factors. *In Software Project Management in a Changing World. Berlin Heidelberg: Springer*, pp. 27- 49, 2014.
- [15] Pinto J, Prescott J. Variations in critical success over the stages of in the project life cycle. *Journal of Management*. Vol. 14, pp. 5-18, 1988.
- [16] Pinto J. K. and Slevin D. P, Critical success factors across the project life cycle, *Project Management Journal*, Vol. 19 No. 3, pp. 67-75, 1988.
- [17] Atkinson, R. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management*, Vol. 17, No. 6, pp. 337-342, 1999.
- [18] Chappell, D. The three aspects of software quality: functional, structural, and process. http://www.davidchappell.com/writing/white_papers/TheThree_Aspects_of_Software_Quality_v1.0-Chappell.pdf, 2013..
- [19] Barnes M (2013) Private communication, September 2013 BBC Radio 4 news, 15.5.2003. <http://www.silicon.com/news/500022/1/4169.html>
- [20] Mitrović Z, Petrović D. & Mihić M, Rethinking success in software projects. *Tehnika*, Vol. 75., pp. 639-645, 2020. 10.5937/tehnika2005639M.
- [21] McManus J, & Wood-Harper T Understand,ing the Sources of Information System Project Failure. *Management Services*, pp. 38-43, 2007.
- [22] Baccarini D, The logical framework method for defining project success. *Project Management Journal*, Vol. 30, No. 4, pp. 25-32, 1999.
- [23] Ika, L. Project success as a topic in project management journals, *Project Management Journal*, Vol. 40 No. 4, pp. 6-19. 2009.
- [24] Jugdev, K. and Muller, R. A retrospective look at our evolving understanding of project success, *Project Management Journal*, Vol. 36 No. 4, pp. 19-31, 2005.
- [25] Shenhar A. J, Levy O, Dvir D, Mapping the dimensions of project success. *Project Management Journal*, Vol. 28, No. 2, pp. 5-13, 1997.
- [26] Nasir M. H. & Sahibuddin, S. Critical success factors for software projects: A comparative study. *Scientific*

- Research and Essay*, Vol. 6, No. 10, pp. 2174-2186, 2011.
- [27] Tarawneh, H. A Suggested Theoretical Framework for Software Project Success. *Journal of Software Engineering and Applications*, Vol. 4, pp. 646-651, 2011.
- [28] Chiyangwa B, Ernest T. M, Modelling the critical success factors of agile software development projects in South Africa. *SA Journal of Information Management*, 2017.
- [29] Fortune J. & White, D. Framing of project critical success factors by a systems model. *International Journal of Project Management*, Vol. 24, No. 1, pp. 53-65, 2006.
- [30] Steward, D. The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, EM-28, pp. 71-74, 1981.
- [31] Eppinger S. D, Whitney D. E. and Gebala D. A, Organizing Tasks in Complex Design Projects: Development of Tools to Represent Design Procedures – NSF Design and Manufacturing Systems Conference, Atlanta, GA, 1992.
- [32] Browning, Tyson. Applying The Design Structure Matrix To System Decomposition And Integration Problems: A Review And New Directions. *Engineering Management, IEEE Transactions*, No. 48. Pp. 292 – 306, 2001. 10.1109/17.946528.
- [33] Danilovic, M., Browning, T.R. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management*, Vol. 25, No. 3, pp. 300–314, 2007.
- [34] Sangal N, Jordan E, Sinha V, Jackson D, Using dependency models to manage complex software architecture. *Proceedings of the 20th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*. San Diego, CA, USA. ACM, pp. 167–176, 2005.
- [35] Yassine, A.A., An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method. *Project Development Research Laboratory*, University of Illinois at Urbana-Champaign, 2004.
- [36] Vidal, L.-A. Thinking Project Management in the Age of Complexity. *Particular Implications on Project Risk Management*, 2009. Paris: Retrived from: <https://tel.archivesouvertes.fr/tel-00546695/document> (datum pristpa: 10.10.2018).
- [37] Fu, Y. Li, M. Chen, M. Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management*, Vol. 23, No. 3, pp. 223–230, 2011.
- [38] Carrascosa M, Eppinger S. D, and Whitney D. E, Using the design structure matrix to estimate product development time - *ASME Design Engineering Technical Conferences*. Atlanta, Georgia, USA, 1998.
- [39] Sosa M, Eppinger S. and Rowles C, The misalignment of product architecture and organizational structure in complex product development. *Management Science*, Vol. 50, No.12, pp. 1674-1689, 2004.
- [40] Maheswari J. U, Varghese K, Project scheduling using dependency structure matrix. *International Journal of Project Management*, Vol. 23, No. 3, pp. 223–230, 2005.
- [41] Le Moigne, J.-L. *La théorie du système général. Théorie de la modélisation*. Paris: Presses Universitaires de France, 1994.
- [42] Biggs N, Lloyd E. and Wilson R, *Graph Theory*. Oxford University Press, pp. 1736-1936, 1986.
- [43] Gebala D. A. & Eppinger, S. D, Methods for Analyzing Design Procedures, *ASME Conference on Design Theory and Methodology*, Miami, September, pp. 227–233, 1991.

SUMMARY

DESIGN STRUCTURE MATRIX FOR EVALUATION OF CRITICAL SUCCESS FACTORS FOR SOFTWARE PROJECTS

The topic of critical success factors draws high attention of both academia and practitioners working in the software industry. Literature review shows that existing models of critical success factors for software projects are more concentrated on specific categories of success factors, neglecting the interaction that exists within these models between different groups of factors. The purpose of this paper is to present a new concept of interaction-based model for identification and evaluation of critical success factors for software projects. The approach is based on conjoint use of graph theory and Design Structure Matrices to highlight important interactions of a project system. The key result of the paper is the tool for evaluation of the interaction between critical success factors through quantification and the analysis of the possible propagation of specific factor within project success factor network, which will make possible the identification of the critical factors with the highest influence on the entire project system.

Key Words: *Critical success factors, software projects, systems thinking, graph theory, design structure matrix*