

PRIMENA DEVOPS METODOLOGIJE U AGILNOM RAZVOJU SOFTVERA

Halida Karišik¹, Aldina Avdić²

Rezime: U ovom radu istražuje se integracija DevOps metodologije u agilnom razvoju softvera, sa ciljem unapređenja efikasnosti, kvaliteta i brzine isporuke softverskih rešenja. Rad obuhvata analizu osnovnih principa i praksi Agile i DevOps pristupa, pregled relevantnih istraživačkih radova, kao i predlog metodologije za njihovu kombinaciju. Poseban akcenat stavljen je na praktične aspekte implementacije, uključujući alate, procese i kulturne promene koje su ključne za uspešnu primenu ovih metodologija u savremenom softverskom razvoju. Istraživanje opisano u ovom radu primenjuje DevOps principe u okviru agilne metodologije Crystal, analizirajući efekte na efikasnost timskog rada i kvalitet isporučenog softvera.

Ključne reči: Agilne metodologije, DevOps, Kontinuirana integracija (CI), Kontinuirana isporuka (CD).

APPLICATION OF DEVOPS METHODOLOGY IN AGILE SOFTWARE DEVELOPMENT

Abstract: This paper examines the integration of DevOps methodology in agile software development, with the aim of improving the efficiency, quality and speed of delivery of software solutions. The work includes an analysis of the basic principles and practices of Agile and DevOps approaches, an overview of relevant research works, as well as a methodology proposal for their combination. Special emphasis is placed on the practical aspects of implementation, including the tools, processes and cultural changes that are critical to the successful application of these methodologies in modern software development. The research described in this paper applies DevOps principles within the Crystal agile methodology, analyzing their impact on team efficiency and the quality of delivered software.

Key words: Agile methodologies, DevOps, Continuous Integration (CI), Continuous Delivery (CD).

1. UVOD

Razvoj softvera danas prolazi kroz dinamične promene, gde su brzina, efikasnost i kvalitet ključni faktori uspeha. U tom kontekstu, agilna metodologija i DevOps pristup postaju dominantni okviri za organizaciju i realizaciju softverskih projekata. Agilna metodologija omogućava fleksibilnost i prilagođavanje promenljivim zahtevima korisnika, dok DevOps integriše razvojne i operativne timove, unapređujući automatizaciju procesa i ubrzavajući isporuku softvera.

Motivacija za istraživanje proizlazi iz potrebe za unapređenjem efikasnosti softverskog razvoja kroz sinergiju ova dva pristupa. Iako se Agile i DevOps inicijalno razvijaju kao zasebni koncepti, njihova integracija nudi značajne prednosti, uključujući smanjenje jaza između timova, povećanje kvaliteta proizvoda i skraćivanje vremena potrebnog za plasiranje proizvoda na tržište.

Cilj rada je istražiti mogućnosti uspešne implementacije DevOps metodologije u okviru agilnog razvoja softvera. Poseban fokus je na analizi sličnosti i razlika između ovih metodologija, njihovih prednosti i izazova, kao i na predlogu metodologije za njihovu integraciju.

Pregled rada: Nakon uvoda, u drugom poglavlju predstavljeni su relevantni istraživački radovi koji osvetljavaju ključne aspekte Agila i DevOps metodologija. Treće poglavlje se bavi detaljnom analizom ovih metodologija, uključujući njihove sličnosti, razlike i prednosti. U četvrtom poglavlju predstavljeno je istraživanje koje uključuje primenu DevOps principa u agilnoj metodologiji Crystal, sa detaljima o procesima, alatima i rezultatima. Rad se završava zaključkom u šestom poglavlju, gde su sumirani ključni nalazi i preporuke za buduće istraživanje.

¹Državni univerzitet u Novom Pazaru, Vuka Karadžića bb, halida.karisik6@gmail.com

²Doc. dr, Državni univerzitet u Novom Pazaru, Vuka Karadžića bb, apljaskovic@np.ac.rs

2. SRODNI RADOVI

Istraživanje uključuje sistematski pregled literature o različitim srodnim radovima. Analizirano je deset postojećih radova koji čine osnovu ovog istraživanja, sa fokusom na problematiku, istraživačke metode i ograničenja, kako je prikazano u nastavku.

Marius Andersen Bjørni i Simen Haugen sproveli su intervju i posmatranja kako bi identifikovali glavne izazove agilne metodologije. Među izazovima su visok intenzitet rada tokom sprintova, nedostatak testiranja u sprintovima pre objavljivanja, loše opisani PBI (Product Backlog Item) zadaci, manjak poslovne agilnosti i nedostatak dokumentacije. Takođe su uočeni problemi poput neadekvatne pripreme PBI zadataka, nedostatka timskog unapređenja, neefikasnog procesa objavljivanja i izostanka sprint pregleda. Istraživanje, međutim, nije obuhvatilo procenu radnog opterećenja, što predstavlja značajan istraživački jaz [1].

Koi-Akrofi i saradnici sproveli su studiju o agilnoj metodologiji zasnovanu na pregledima literature. Identifikovali su ključne prednosti agilnog pristupa, uključujući prilagodljivost promenama u zahtevima korisnika, jednostavno ažuriranje, procenu prioriteta, integraciju povratnih informacija, kontinuirano testiranje i motivaciju za programere. S druge strane, istaknuti su izazovi poput nepredvidivosti razvoja, potrebe za visokim angažovanjem, većih zahteva korisnika, nedostatka dokumentacije i rizika od odstupanja projekta od ciljeva. Međutim, studija se oslanja isključivo na pregled literature, bez dodatnih istraživačkih metoda, što predstavlja ograničenje [2].

Farid i saradnici istražili su kombinaciju DevOps-a sa Lean Software Development metodologijom. Lean Software Development je predstavljen kao metodologija usmerena na smanjenje otpada i kontinuirano unapređenje, dok je DevOps opisan kao skup praksi usmerenih na unapređenje celokupnog životnog ciklusa razvoja integracijom razvoja i operacija u međufunkcionalne timove. Istraživačka metoda uključuje primenu DevOps praksi za prevazilaženje identifikovanih problema u Lean pristupu, kao što su kašnjenja, defekti, dodatne funkcije i prebacivanje između zadataka. Ograničenje ovog rada je što je primena DevOps principa testirana samo u kontekstu Lean metodologije, dok su slični slučajevi za Scrum, XP i druge metode ostavljeni za buduća istraživanja [3].

Banica i saradnici predstavili su DevOps kao metodologiju upravljanja projektima u razvoju softverskih projekata. Kao istraživačku metodu koristili su pilot projekat u kojem su implementirali DevOps alate u proces razvoja web sajta koristeći VersionOne alat, pogodan za sisteme upravljanja projektima koji podržavaju DevOps. Rezultati su pokazali da DevOps ubrzava i poboljšava proces razvoja. Međutim, zbog neiskustva timova, tokom razvoja su se pojavili problemi poput kašnjenja i propuštanja određenih ciljeva, što predstavlja ograničenje ovog istraživanja [3].

Hemon i saradnici predložili su model zrelosti za tranziciju sa Agile na DevOps kroz tri faze: Agile, kontinuirana integracija i kontinuirana isporuka. Model se fokusira na usklađivanje razvoja i operacija, ali ne obuhvata polje razvoja softvera, uprkos visokoj primeni Agile i DevOps pristupa u ovoj oblasti, što predstavlja ograničenje [4].

Dodatna studija Hemon i saradnika istražila je uloge pojedinaca, saradnju i veštine tokom tranzicije sa Agile na DevOps. Podaci prikupljeni iz studije slučaja u organizaciji sa višegodišnjim iskustvom u DevOps-u pokazuju da DevOps omogućava balansiranu saradnju, ali zahteva visok nivo veština među developerima i operatorima. Međutim, istraživanje ne pruža dovoljno podataka o širem spektru saradnje, što ostavlja prostor za buduće analize [5].

3. ANALIZA AGILA I DEVOPS-A

Agilna metodologija razvoja softvera koja se fokusira na četiri osnovna principa [6] stavlja naglasak na značaj pojedinca i interakcija u odnosu na procese i alate, radni softver u odnosu na opsežnu dokumentaciju, saradnju sa korisnicima u odnosu na pregovaranje ugovora, i odgovaranje na promene u odnosu na pridržavanje plana. Postoji mnogo metoda i tehnika za implementaciju Agilne metodologije, zavisno od opsega projekta, domena znanja i drugih faktora, uključujući DAD

(Disciplined Agile Delivery), SAFe (Scaled Agile Framework), XP (Extreme Programming), LeSS (Large Scale Scrum), itd [7]. Scrum je poseban agilni okvir dizajniran za male timove koji relativno jednostavan za korišćenje i visoko prilagodljiv promenama [8].

Agilna metodologija pruža mnoge prednosti u odnosu na tradicionalne metodologije razvoja softvera [9]. Glavne prednosti uključuju niže zahteve za dokumentacijom, jaku saradnju sa korisnicima, bržu isporuku i kontinuirano usklađivanje sa poslovnim potrebama i ciljevima. Agilna metodologija ima nekoliko značajnih nedostataka, kao što su separacija sektora razvoja, testiranja i isporuke, kao i ograničena međufunkcionalnost timova [1]. Nedostatak kontinuirane integracije povratnih informacija tokom razvoja može usporiti proces i smanjiti efikasnost razvoja softvera. Pored toga, ograničena međufunkcionalnost unutar razvojnih timova može dovesti do nepovezanog i neusklađenog procesa razvoja, testiranja i isporuke, što često rezultira neskladom između završetka projekta i planiranih procena.

Međutim, DevOps (Development-Operations) predstavlja proširenje agilne metodologije koje može harmonizovati procese/sektore razvoja i operacija i obezbediti međufunkcionalnost unutar timova. Na taj način, DevOps može ublažiti neke od nedostataka agilne metodologije. DevOps je metodologija razvoja softvera koja se fokusira na saradnju između developera i operatera, ali nema konkretno definisanje i često se smatra skupom veština, setom praksi ili čak opisom posla [10]. DevOps se fokusira na međufunkcionalnost između različitih sektora razvoja, operacija i obezbeđenja kvaliteta, što bi rezultiralo visoko koordinisanim razvojem, testiranjem i isporukom, što bi dovelo do lakšeg napredovanja projekta [11].

Međutim, čist DevOps sigurno ne može da se fokusira na ispunjavanje stalno promenljivih zahteva korisnika putem prikupljanja i uključivanja povratnih informacija u inkrementalne cikluse. Takođe, čist DevOps ima visoke zahteve za veštine koje ne mogu ispuniti softverski inženjeri koji nemaju obimnu obuku i iskustvo u DevOps kulturi i praksama. Dakle, potpuno uključivanje DevOps-a u javne sektore, kao što su srednji i niski nivoi razvoja softvera, dovelo bi do poteškoća i nedostataka koji bi bili neprihvatljivi za efikasan i efektivan proces razvoja softvera [4] [5] [12].

Kako bi se prevazišli postojeći izazovi, predlaže se integracija Agile i DevOps metodologija u hibridni pristup nazvan DevOps Enabled Agile za razvoj softvera. Analiza i istraživanja ukazuju na to da kombinacija ovih metodologija može pružiti optimalne rezultate i smanjiti nedostatke prisutne u obe metodologije. Ovaj pristup se oslanja na primere iz sličnih istraživanja koja preporučuju kombinovanje Agile-a i DevOps-a kako bi se iskoristile njihove komplementarne prednosti [2] [3][13].

Naša predložena metodologija, DevOps Enabled Agile, dizajnirana je da kombinuje kolaborativne i međufunkcionalne aspekte DevOps-a sa jednostavnošću i niskim zahtevima za veštine Agila kako bi se zatvorili nedostaci obe metodologije, čime se obezbeđuje brz napredak i maksimalna efikasnost sektora razvoja, testiranja i isporuke u projektima razvoja softvera. Stoga, ova analiza opravdava implementaciju DevOps Enabled Agile metodologije za razvoj softvera.

3.1. Razlike između Agile i DevOps metodologija

U poređenju sa Agilnom metodologijom, DevOps predstavlja širi okvir koji objedinjuje razvoj, operacije i kvalitet. Agile se primarno fokusira na fleksibilnost zahteva, brzi razvoj funkcionalnosti i saradnju sa korisnicima, dok DevOps pruža infrastrukturnu podršku za kontinuirane procese integracije i isporuke. Ključne razlike su:

- **Pristup:** Agilne metodologije upravljaju projektima, dok DevOps integriše procese razvoja i operacija kroz tehničke prakse i automatizaciju.
- **Fokus:** Agile teži razvoju softverskih funkcionalnosti kroz iteracije, dok DevOps osigurava kontinuitet kroz optimizaciju procesa poput automatizovane isporuke i monitoringa.
- **Implementacija:** Agile često koristi okvire kao što su Scrum, SAFe i LeSS, dok DevOps nije ograničen specifičnim okvirom, već se zasniva na prilagodljivim praksama.

Ove razlike osvetljavaju komplementarnu prirodu ovih metodologija, pri čemu DevOps nadopunjuje ograničenja Agile-a pružajući kontinuitet nakon razvoja softvera.

3.2. Sličnosti između Agile i DevOps metodologija

Iako imaju različite pristupe, Agile i DevOps dele nekoliko ključnih zajedničkih principa, što omogućava njihovu integraciju u različitim kontekstima razvoja softvera. Sličnosti se ogledaju kroz sledeće principe:

- Brzina i prilagodljivost: Oba pristupa podržavaju brzu isporuku softvera kroz iterativne procese.
- Saradnja i povratne informacije: U oba slučaja, saradnja između timova i povratne informacije korisnika igraju centralnu ulogu u poboljšanju krajnjeg proizvoda.
- Fleksibilnost: Oba pristupa su dovoljno prilagodljiva da se integrišu u različite industrijske i organizacione modele.

Ova sinergija ukazuje na potencijal za kombinovanje Agile i DevOps metodologija kako bi se povećala efikasnost i efektivnost razvoja softverskih rešenja.

3.3. Prednosti DevOps pristupa

DevOps donosi značajne prednosti u unapređenju ključnih aspekata softverskog razvoja. Ubrzava proces razvoja i isporuke softvera, smanjuje troškove integracijom kvaliteta u procese razvoja i omogućava timovima da se fokusiraju na unapređenja, a ne na rešavanje problema. Takođe, DevOps podstiče timsku motivaciju i efikasnost kroz zajedničke ciljeve i dinamičan radni model. Kontinuirana integracija (CI) omogućava automatsku proveru i integraciju promena u softveru, dok kontinuirana isporuka (CD) osigurava pravovremenu isporuku funkcionalnog softvera kroz kratke iterativne cikluse. Ovi procesi značajno smanjuju vreme potrebno za identifikaciju i rešavanje problema, povećavajući pouzdanost i stabilnost softverskih rešenja.

3.4. Prakse kontinuiranih procesa u DevOps-u

DevOps integracija naglašava važnost kontinuiranih procesa za postizanje stabilnog i skalabilnog razvoja softvera:

Kontinuirana integracija obezbeđuje česte i automatske integracije promena u softveru. Ovaj proces smanjuje vreme potrebno za identifikaciju i rešavanje problema, čime se omogućava brža isporuka funkcionalnog softvera.

Kontinuirana isporuka omogućava pravovremenu isporuku softvera kroz kratke iterativne cikluse. Ovo uključuje procese automatizacije i standardizacije koji osiguravaju pouzdanost isporuke. CD dodatno rešava ograničenja Agile-a kroz bolje upravljanje komunikacijom i procesima razvoja.

3.5. Uloga alata u DevOps-u i izazovi DevOps metodologije

Alati predstavljaju ključni element DevOps prakse, omogućavajući visoku efikasnost u radu timova. Alati poput Puppet-a i RANCID-a omogućavaju jednostavno upravljanje konfiguracijama u složenim softverskim projektima. Scrum i Kanban su među najčešće korišćenim okvirima za upravljanje radnim procesima. Scrum naglašava ljudski faktor, dok Kanban unapređuje vidljivost i otklanjanje uskih grla u procesima.

Iako DevOps donosi brojne prednosti, njegovo usvajanje nosi i specifične izazove:

- Visoki zahtevi za veštinama: Uspešno implementiranje DevOps-a zahteva značajne tehničke veštine i iskustvo, što može predstavljati prepreku za mnoge organizacije.
- Usklađivanje razvojnih i operativnih timova: Različiti ciljevi i prioriteti timova mogu otežati uspostavljanje efektivne saradnje.

4. ISTRAŽIVANJE: PRIMENA DEVOPS-A U AGILNOJ METODOLOGIJI CRYSTAL

U ovom istraživanju, fokusirali smo se na primenu DevOps principa u agilnoj metodologiji Crystal, s ciljem unapređenja razvoja softvera kroz automatizaciju, kontinuiranu integraciju i bolju saradnju u timu. Istraživanje je sprovedeno na malom projektu uz učešće tima od samo dve osobe, što je dodatno naglasilo važnost efikasne organizacije i komunikacije.

4.1. Proces istraživanja

Projekat je započet definisanjem zahteva korisnika, što je odrađeno kroz seriju razgovora sa korisnikom kako bi se precizno identifikovali ključni ciljevi i prioriteti. Crystal metodologija, koja naglašava važnost fleksibilnosti i komunikacije, bila je prilagođena malom timu, omogućavajući transparentan tok informacija i brze iteracije.

Zahtevi su podeljeni na manje jedinice rada, organizovane u iteracije (sprintove). Svaka iteracija trajala je po nekoliko dana i obuhvatala je sledeće korake:

- Planiranje i razvoj: Prva članica tima bila je odgovorna za razvoj funkcionalnosti, oslanjajući se na povratne informacije dobijene tokom planiranja.
- Testiranje i implementacija: Drugi član tima fokusirao se na testiranje implementiranih funkcionalnosti, koristeći DevOps alate za automatizaciju testova i praćenje grešaka.

Za upravljanje projektom i automatizaciju, korišćeni su alati poput Git-a za verzionisanje, Jenkinsa za automatizaciju CI/CD procesa i Dockera za kreiranje izolovanih okruženja za testiranje. Automatizovani procesi omogućili su timu da brzo identifikuje i ispravi greške, dok je svaki sprint završavan isporukom funkcionalne verzije softvera.

4.2. Uloga DevOps-a u istraživanju

Primena DevOps-a omogućila je timu da prevaziđe izazove koji dolaze sa radom u malom timu. Jedan od ključnih aspekata bila je kontinuirana integracija, koja je omogućila brzu proveru svake promene u kodu. Na primer, nakon svake implementacije nove funkcionalnosti, automatizovani testovi su proveravali njen uticaj na postojeći sistem. Ovo je smanjilo rizik od regresija i omogućilo timu da održi stabilnost aplikacije.

DevOps je takođe unapredio saradnju između članova tima. Iako je tim bio mali, jasno definisane uloge i upotreba alata za praćenje zadataka i rezultata, poput ClickUp-a, omogućile su da se procesi odvijaju bez zastoja. Kroz redovne sastanke i razmenu povratnih informacija, tim je mogao da brzo reaguje na promene i prilagodi prioritete.

U prvoj iteraciji, Crystal metodologija je korišćena sa tradicionalnim pristupom, gde su članovi tima radili odvojeno na razvoju, testiranju i isporuci. Tokom petodnevnog perioda, performanse su praćene kroz Burndown grafike, koji su otkrili sledeće: sektor razvoja bio je efikasan i završio zadatke pre roka, dok sektori testiranja i isporuke nisu napredovali sve dok razvoj nije završen. Ovo je ukazalo na problem nedostatka paralelne saradnje i međufunkcionalnog rada.

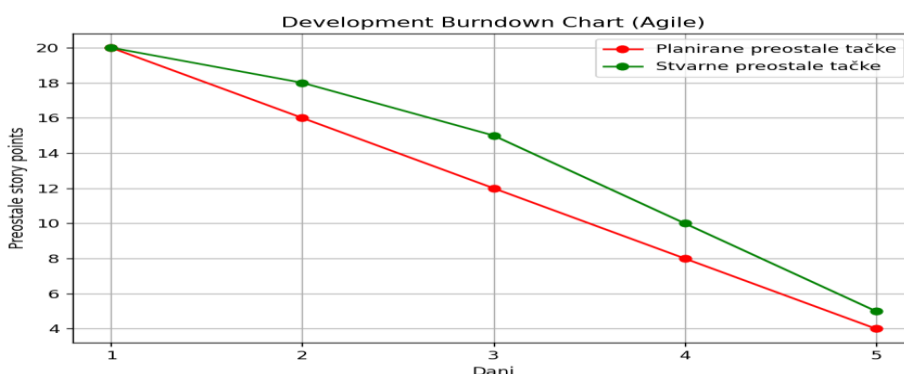
U drugoj iteraciji, tim je prešao na DevOps Enabled Agile metodologiju, gde su članovi radili zajedno na razvoju, testiranju i isporuci. Ovaj pristup omogućio je kontinuiranu saradnju, čime su svi sektori napredovali paralelno. Automatizovani procesi, kao što su CI/CD kroz Jenkins i izolovana testna okruženja putem Dockera, dodatno su povećali efikasnost. Burndown grafici iz ove iteracije pokazali su stalni napredak u svim sektorima i završetak zadataka pre planiranog roka.

Rezultati jasno pokazuju da je DevOps Enabled Agile metodologija superiornija u odnosu na tradicionalnu Crystal metodologiju, posebno u malim timovima. Paralelni rad, podržan automatizacijom i efikasnim alatima, smanjio je kašnjenja i omogućio brže postizanje projektnih ciljeva.

5. DISKUSIJA

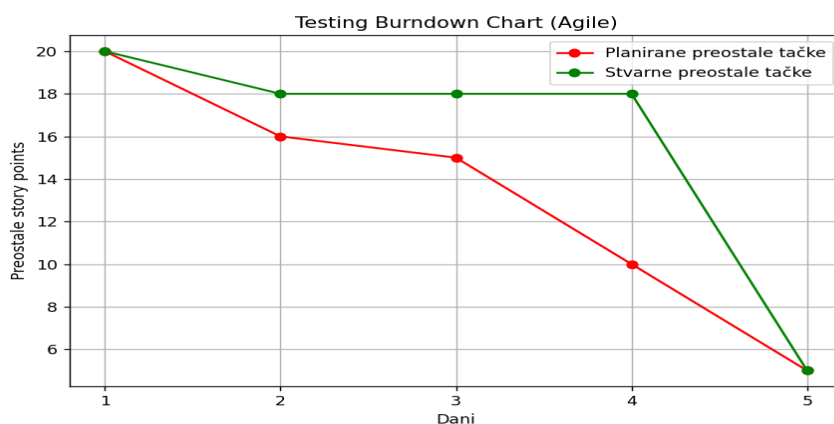
Primena DevOps-a u okviru Crystal metodologije donela je značajne rezultate. Prva iteracija, koja je koristila standardni Crystal pristup, otkrila je nedostatke u koordinaciji između sektora. Testiranje i isporuka nisu napredovali dok razvoj nije bio završen, što je rezultiralo kašnjenjem u ukupnom napretku projekta. Burndown grafici su jasno pokazali ovu neefikasnost, iako je sektor razvoja završio zadatke pre roka. Na Slici 1 prikazan je Burndown grafikon koji ilustruje napredak tokom razvoja, a Slika 2 prikazuje dinamiku testiranja kroz tok iteracije.

Druga iteracija, koja je koristila DevOps Enabled Agile metodologiju, pokazala je značajno poboljšanje. Paralelni rad članova tima na razvoju, testiranju i isporuci omogućio je stalni napredak u svim sektorima. Automatizacija kroz CI/CD alate smanjila je rizik od grešaka i ubrzala isporuku. Burndown grafici su prikazali stalni pad preostalih zadataka, što je rezultiralo završetkom projekta pre planiranog roka.



Slika 1 – Burndown grafikon razvoja

Ova analiza potvrđuje da je integracija DevOps principa u Crystal metodologiju efikasna strategija za unapređenje koordinacije i smanjenje kašnjenja u malim timovima. DevOps Enabled Agile omogućava paralelni rad i povećava ukupnu efikasnost projekta.



Slika 2 – Burndown grafikon testiranja

6. ZAKLJUČAK

Agil i DevOps su različite metodologije, iako njihova povezanost često dovodi do pretpostavke da su iste. Ovo šteti i agilnim i DevOps metodologijama. Oba pristupa pomažu timovima da rade brže i efikasnije, isporučujući kvalitetan rad i zadovoljavajuće rezultate za korisnike. DevOps je okvir za izgradnju saradnje između tehnoloških i operativnih timova. To omogućava korisnicima da brzo i automatski implementiraju kod u produkciju. DevOps pomaže ubrzanju isporuke proizvoda i usluga kroz organizaciju. Sa ovim uvidima, jasno je da i agil i DevOps teže efikasnijem načinu pružanja vrednosti krajnjim korisnicima – ali sa različitim aspektima.

Može se definisati kao usklađivanje razvoja i IT operacija. Agil nastoji da poboljša efikasnost za programere i rasporede izdanja, dok DevOps okuplja operativne timove kako bi omogućio kontinuiranu implementaciju i kontinuiranu isporuku. Takođe, pomogao je u promociji inovativnih razmišljanja koja prevazilaze samo agilni razvoj i višefunkcionalne (DevOps) timove, kao i end-to-end procese koji utiču na softverske kompanije i njihove korisnike. Agile i DevOps metodologije nisu međusobno isključive, već se dopunjuju u cilju pružanja što efikasnijeg razvoja softvera. Agile omogućava fleksibilnost i brzu isporuku funkcionalnosti, dok DevOps osigurava stabilnost, automatizaciju i kontinuitet procesa. Kombinovanjem ovih pristupa, organizacije mogu ostvariti značajna poboljšanja u kvalitetu, brzini i pouzdanosti softverskih rešenja.

Integracija Agile i DevOps metodologija, iako različitih pristupa, omogućava značajno unapređenje razvoja softvera. Agile pruža fleksibilnost i brzu isporuku funkcionalnosti, dok DevOps osigurava stabilnost, automatizaciju i kontinuitet procesa. Istraživanje je pokazalo da hibridni pristup, poput DevOps Enabled Agile, omogućava sinergiju ovih metodologija, smanjuje kašnjenja, povećava efikasnost i omogućava paralelni rad timova. Uvođenje automatizovanih procesa, kao što su CI/CD, i korišćenje alata poput Dockera i Jenkinsa, dodatno doprinosi stabilnosti i brzini razvoja. Ova kombinacija metodologija predstavlja optimalno rešenje za savremeni softverski razvoj, posebno u malim i srednjim timovima.

U okviru budućih istraživanja planirano je dalje istraživanje integracije DevOps-a u različite Agile okvire kako bi se procenila njihova efikasnost u većim i kompleksnijim projektima. Planirano je testiranje naprednih DevOps alata i tehnika, poput orkestracije kontejnera i unapređenih CI/CD praksi, kako bi se dodatno optimizovali procese razvoja i isporuke. Ova istraživanja će doprineti razumevanju kako organizacije mogu maksimizovati koristi od ovih metodologija u različitim industrijskim kontekstima.

7. LITERATURA

- [1] M. A. Bjørni and S. Haugen, "Challenges with Agile in a System Development Department: A Case Study," *IEEE Software*, vol. 35, no. 1, pp. 78–83, 2019.
- [2] G. Y. Koi-Akrofi, J. Koi-Akrofi, and H. A. Matey, "Understanding the Characteristics, Benefits and Challenges of Agile IT Project Management: A Literature Based Perspective," *arXiv Preprint*, vol. 10, pp. 25–44, 2019.
- [3] A. B. Farid, Y. M. Helmy, and M. M. Bahloul, "Enhancing Lean Software Development by Using DevOps Practices," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, pp. 280–286, 2017.
- [4] L. Banica, M. Radulescu, D. Rosca, and A. Hagi, "Is DevOps Another Project Management Methodology?" *Informatica Economica*, vol. 21, pp. 39–51, 2017.
- [5] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "From Agile to DevOps: Smart Skills and Collaborations," *Inf. Syst. Front.*, vol. 22, pp. 927–945, 2020.
- [6] S. M. R. A. Masud, M. Masnun, M. A. S. Sultana, F. Ahmed, and N. Begum, "DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development," *Dept. of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh*.

- [7] S. M. Mohammad, "DevOps Automation and Agile Methodology," Sr. Associate, KPMG LLP, Dept. of IT, Wilmington University.
- [8] M. A. Subih, B. H. Malik, I. Mazhar, U. Sabir, T. Wakeel, W. Ali, A. Yousaf, B. bin-Ijaz, H. Nawaz, and M. Suleman, "Comparison of Agile Method and Scrum Method with Software Quality Affecting Factors," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 509–515, 2019.
- [9] R. Shankarmani, R. Pawar, S. S. Mantha, and V. Babu, "Agile Methodology Adoption: Benefits and Constraints," *Int. J. Comput. Appl.*, vol. 58, pp. 31–37, 2012.
- [10] F. Erich, C. Amrit, and M. Daneva, "A Mapping Study on Cooperation between Information System Development and Operations," in *Proc. Int. Conf. Product-Focused Software Process Improvement*, 2014, pp. 277–280.
- [11] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE Software*, vol. 33, no. 3, pp. 94–100, 2016.
- [12] R. T. Yarlalagadda, "How Public Sectors Can Adopt the DevOps Practices to Enhance the System," *Int. J. Emerg. Technol. Innov. Res.*, pp. 82–91, 2018.
- [13] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "Conceptualizing the Transition from Agile to DevOps: A Maturity Model for a Smarter IS Function," in *IFIP Int. Conf.*, 2019, pp. 209–223.