

Najkorisniji razvojni alati

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Staša Đorđević Lazar Savić Đurđa Milošević
Bogdan Tomić

18. decembar 2025.

Sažetak

Istraživanjem je analizirano korišćenje razvojnih alata u programiranju sa ciljem identifikovanja faktora koji utiču na njihov izbor i promenu. Podaci su prikupljeni anonimnom anketom. Rezultati pružaju uvid u aktuelne trendove i razloge preferiranja pojedinih razvojnih okruženja i sistema za verzionisanje.

Sadržaj

1	Uvod	2
2	O razvojnim alatima	2
2.1	Integrisana razvojna okruženja (IDE)	2
2.2	Tekstualni editori koda	3
2.3	Sistemi za verzionisanje	3
3	Metodologija	3
3.1	Uzorkovanje	3
3.2	Struktura uzorka	3
3.3	Struktura ankete	4
4	Rezultati	4
4.1	Analiza faktora važnosti pri izboru alata	4
4.2	Analiza opštih preferencija Git platformi	7
4.3	Analiza uticaja radnog iskustva na promenu razvojnog alata	7
4.4	Analiza uticaja navike na promenu razvojnog alata	8
4.5	Uporedna analiza prioriteta: Studenti nasuprot zaposlenih	8
4.6	Analiza otvorenih odgovora ispitanika	9
5	Diskusija	10
5.1	Najkorišćeniji razvojni alati	10
5.2	Faktori koji utiču na izbor alata	11
5.3	Ograničenja istraživanja	12
6	Zaključak	12
	Literatura	13

1 Uvod

Razvojni alati predstavljaju jedan od najznačajnijih faktora koji utiču na produktivnost i efikasnost programera, i kao takvi, čine neizostavni deo njihovog svakodnevnog rada. Imajući u vidu veliki broj dostupnih razvojnih alata, koji se međusobno razlikuju po funkcionalnostima, performansama, mogućnostima prilagođavanja i korisničkom iskustvu, izbor najkorisnijeg alata nije jednoznačan. On često zavisi od ličnih preferencija korisnika, prethodnog iskustva, kao i specifičnih zahteva projekta ili okruženja u kojem se alat koristi.

Predmet ovog rada jeste analiza upotrebe razvojnih alata, sa fokusom na integrisana razvojna okruženja (IDE), tekstualne editore koda i sisteme za verzionisanje, kao tri osnovne kategorije alata koje se najčešće koriste u procesu razvoja softvera. Cilj rada jeste da se ispita koje konkretne alate iz navedenih kategorija studenti i iskusniji programeri najčešće koriste, koje faktore smatraju najvažnijim prilikom njihovog izbora, kao i šta ih motiviše da pređu na novi alat ili ostanu pri postojećem. Pored toga, poseban akcenat stavljen je na ulogu navike u korišćenju određenog alata, kao i na zadovoljstvo korisnika alatima koje trenutno koriste, u cilju identifikovanja potencijalnih unapređenja samih alata.

Značaj ovog istraživanja ogleda se u pružanju sistematičnog uvida u način na koji korisnici percipiraju i biraju razvojne alate, kao i u boljem razumevanju faktora koji utiču na donošenje odluka u procesu izbora alata u realnom akademskom i profesionalnom kontekstu.

2 O razvojnim alatima

Razvoj softvera podrazumeva upotrebu različitih alata koji programerima omogućavaju efikasnije pisanje, organizaciju, testiranje i održavanje programskog koda. Dosadašnja istraživanja pokazuju da izbor razvojnog okruženja i pratećih alata utiče na efikasnost rada, produktivnost programera i donošenje odluka tokom razvoja softvera [1]. U ovom radu razmatraju se tri osnovne kategorije razvojnih alata: integrisana razvojna okruženja (IDE), tekstualni editori koda i sistemi za verzionisanje.

2.1 Integrisana razvojna okruženja (IDE)

Integrisana razvojna okruženja (IDE – *Integrated Development Environment*) predstavljaju softverske alate koji objedinjuju više funkcionalnosti potrebnih za razvoj softvera u jedinstveno radno okruženje [2]. Tipične komponente IDE-a uključuju editor koda, kompajler ili interpreter, alate za dinamičku analizu koda, kao i dodatne funkcionalnosti za upravljanje projektima i testiranje. Primeri integrisanih razvojnih okruženja koji se često koriste u praksi uključuju Visual Studio, IntelliJ IDEA, PyCharm i Android Studio.

Glavna prednost integrisanih razvojnih okruženja ogleda se u visokom stepenu automatizacije i podrške programeru. Funkcionalnosti kao što su automatsko dopunjavanje koda, statička analiza, upozorenja na potencijalne greške i integrisana dokumentacija mogu značajno ubrzati proces razvoja i smanjiti broj grešaka u kodu. Sa druge strane, integrisana razvojna okruženja mogu zahtevati veće hardverske resurse i biti složenija za prilagođavanje individualnim potrebama korisnika.

2.2 Tekstualni editori koda

Tekstualni editori koda predstavljaju alate namenjene pisanju i uređivanju izvornog koda, sa fokusom na jednostavnost i fleksibilnost [4]. Za razliku od integrisanih razvojnih okruženja, tekstualni editori obično nude osnovne funkcionalnosti kao što su isticanje sintakse, jednostavna navigacija kroz fajlove i minimalna podrška za analizu koda.

Jedna od ključnih prednosti tekstualnih editora jeste njihova brzina rada i mala potrošnja sistemskih resursa. Mnogi editori omogućavaju proširivanje funkcionalnosti putem dodataka ili ekstenzija, čime se mogu prilagoditi različitim programskim jezicima i stilovima rada. Primeri često korišćenih tekstualnih editora koda su Sublime Text, Kate i Vim.

2.3 Sistemi za verzionisanje

Sistem za verzionisanje je sistem koji je zadužen za čuvanje i kontrolisanje izmena programskog koda, odnosno za upravljanje različitim verzijama softvera koji se razvija [3]. Njihova osnovna uloga jeste čuvanje istorije izmena, omogućavanje povratka na prethodne verzije koda i olakšavanje saradnje više korisnika na istom projektu.

Dele na centralizovane, koji se oslanjaju na centralni repozitorijum sa kompletnom istorijom izmena, i distribuirane, kod kojih svaki korisnik poseduje lokalnu kopiju repozitorijuma. Najpoznatiji primer distribuiranog sistema za verzionisanje je Git.

3 Metodologija

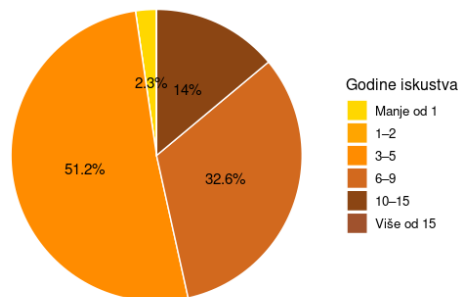
Istraživanje je sprovedeno korišćenjem anonimne onlajn ankete. Metodološki pristup je dominantno kvantitativan, budući da su podaci prikupljeni putem strukturiranih pitanja i analizirani primenom deskriptivne statistike, uz ograničene kvalitativne uvide dobijene analizom odgovora otvorenog tipa.

3.1 Uzorkovanje

U istraživanju je učestvovalo ukupno 43 ispitanika koji se bave programiranjem. Uzorak je obuhvatio studente tehničkih i srodnih fakulteta, kao i zaposlene profesionalce u IT sektoru. Uzorkovanje je sprovedeno metodom pogodnog (neprobabilističkog) uzorka, s obzirom na to da su ispitanici dobrovoljno učestvovali u anketi i bili dostupni autorima istraživanja putem onlajn kanala.

3.2 Struktura uzorka

Struktura uzorka analizirana je kroz demografske i profesionalne karakteristike ispitanika, pri čemu su prikupljeni podaci o dužini iskustva u programiranju, pohađanim fakultetima, kao i polu. Među ispitanicima, 27 je bilo muškog, a 16 ženskog pola. Ubedljivo najviše ispitanika je sa Matematičkog fakulteta 79,1%, dok je zastupljenost ostalih fakulteta mala. Na slici 1 je prikazana raspodela ispitanika po dužini iskustva u programiranju.



Slika 1: Raspodela ispitanika po dužini iskustva u programiranju.

3.3 Struktura ankete

Anketa je sadržala 16 pitanja, od kojih je 12 bilo zatvorenog tipa, dok su četiri pitanja omogućavala slobodne odgovore. Pitanja su organizovana po celinama:

- **Demografska i profesionalna pitanja** - dužina iskustva u programiranju, pol, pohađani fakultet, trenutna pozicija u IT sektoru. Pitanja su bila zatvorenog tipa sa ponuđenim odgovorima.
- **Pitanja o korišćenju razvojnih alata** - najčešće korišćeni IDE i tekstualni editori, sistemi za verzionisanje, preferencije između GitHub i GitLab, glavni razvojni alat, razlozi izbora i problemi sa alatima. Većina pitanja bila je zatvorenog tipa, dok su argumentacija izbora i opis problema bili otvorenog tipa radi dodatnih komentara.
- **Faktori izbora razvojnog alata** - pitanja o važnosti karakteristika alata, uticaju dugogodišnjih navika, ranijim promenama glavnog alata, razlozima za prelazak na novi alat i faktorima koji bi motivisali tu promenu. Za većinu ovih pitanja korišćena je *Likert skala* od 1 do 5.

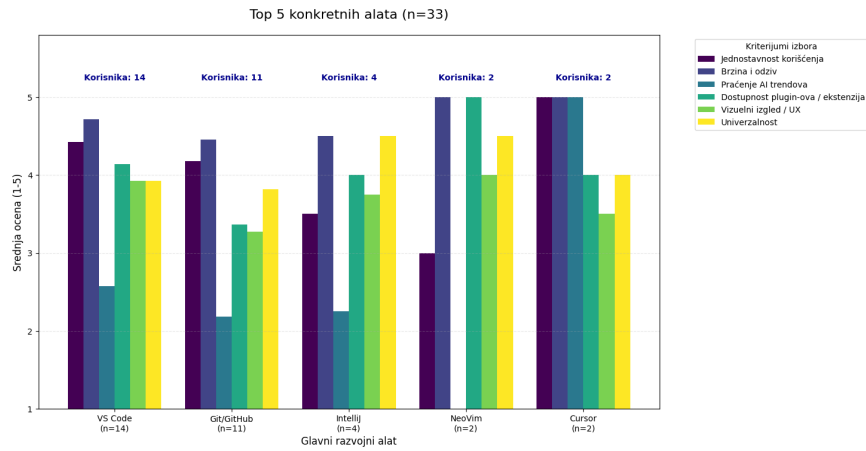
Svi odgovori su bili anonimni, a učestvovanje u anketi dobrovoljno.

4 Rezultati

U ovom poglavlju analizirani su podaci prikupljeni istraživanjem, uz poseban osvrt na korelacije između različitih parametara. Budući da uzorak obuhvata učesnike različitih nivoa stručnosti i obrazovnih profila, stanje na tržištu je sagledano iz više perspektiva. Vizuelni prikazi generisani su direktno iz prikupljenih odgovora.

4.1 Analiza faktora važnosti pri izboru alata

Fokus ovog dela rada je na kriterijumima koje programeri smatraju presudnim prilikom odabira radnog okruženja. Na slici 2 prikazani su rezultati za pet najzastupljenijih alata.



Slika 2: Važnost faktora prilikom izbora razvojnog alata (Top 5)

Za preciznije rangiranje uveden je pokazatelj reprezentativnosti alata P_A , koji ponderiše prosečnu ocenu faktora sa udelom korisnika tog alata u ukupnom uzorku. Formula za izračunavanje glasi:

$$P_A = \left(\frac{f_{jednost} + f_{brzina} + f_{AI} + f_{plugin} + f_{visual} + f_{univ}}{6} \right) \cdot \frac{n_A}{N_{top5}} \quad (1)$$

Značenje parametara korišćenih u formuli detaljno je opisano u tabeli 1.

Tabela 1: Opis parametara formule za reprezentativnost alata

Parametar	Opis faktora / Vrednost
$f_{jednost}$	Jednostavnost korišćenja
f_{brzina}	Brzina i odziv okruženja
f_{AI}	Praćenje i integracija AI trendova
f_{plugin}	Dostupnost dodataka i ekstenzija
f_{visual}	Vizuelni izgled i UX dizajn
f_{univ}	Univerzalnost alata
n_A	Broj korisnika koji koriste alat A
N_{top5}	Broj korisnika nekih od top 5 alata ($N = 33$)

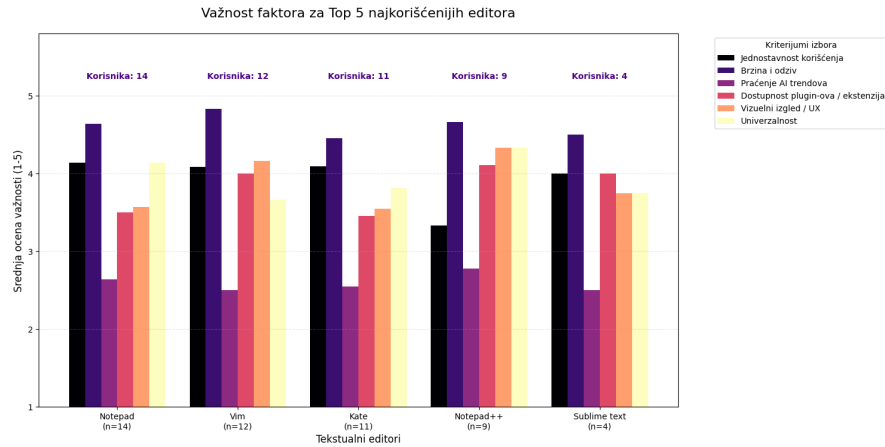
Na osnovu prikupljenih podataka, u tabeli 2 sumirani su rezultati prosečnih ocena (\bar{X}) i ključne karakteristike za svaki od analiziranih alata.

Tabela 2: Analiza vodećih razvojnih alata na osnovu pojedinačnih faktora važnosti

Alat	n	f_{jed}	f_{brz}	f_{AI}	f_{plg}	f_{vis}	f_{uni}	\bar{P}_A
VS Code	14	4.4	4.7	2.6	4.1	3.9	3.9	16.35
Git/GitHub	11	4.2	4.5	2.2	3.4	3.3	3.8	7.13
IntelliJ	4	3.5	4.5	2.3	4.0	3.8	4.5	2.73
Cursor	2	5.0	5.0	5.0	4.0	5.0	5.0	1.75
NeoVim	2	3.0	5.0	2.5	5.0	4.0	4.5	1.45

Ovakav tabelarni prikaz jasno ukazuje na korelaciju između modernih zahteva (AI i brzina) i visokih ocena korisnika, dok VS Code zadržava primat u masovnosti primene.

U okviru ove analize odrađena je i analiza koja se vezuje isključivo za tekstualne editore koda. Na slici 3 prikazano je kako korisnici pet najzastupljenijih editora ocenjuju ključne faktore prilikom izbora svog primarnog alata.



Slika 3: Važnost faktora u odnosu na najčešće korišćene tekstualne editore

Na osnovu prikupljenih odgovora, analizirane su prosečne vrednosti šest ključnih kriterijuma: brzina i odziv, jednostavnost korišćenja, univerzalnost, dostupnost dodatka, vizuelni izgled i AI trendovi. Podaci sa slike 3 ukazuju na specifične prioritete unutar različitih korisničkih grupa:

- **Notepad ($n = 14$):** Najzastupljeniji editor u uzorku. Korisnici mu daju visoke ocene za brzinu i odziv, kao i za jednostavnost korišćenja, dok su AI trendovi očekivano najslabije ocenjen faktor.
- **Vim ($n = 12$):** Ovaj alat beleži izuzetno visoku ocenu za brzinu i odziv (blizu 5.0). Korisnici takođe visoko vrednuju dostupnost dodatka i vizuelni izgled/UX, što ukazuje na visok stepen personalizacije.
- **Kate ($n = 11$):** Karakteriše ga ujednačenost ocena, sa posebnim naglaskom na brzinu rada. Interesantno je da korisnici ovog editora daju stabilne ocene univerzalnosti alata.
- **Notepad++ ($n = 9$):** Kod ovog alata dominiraju ocene za brzinu, vizuelni izgled i univerzalnost, što ga pozicionira kao svestrano rešenje za brze izmene koda.
- **Sublime Text ($n = 4$):** Iako na manjem uzorku, ovaj editor zadržava visok nivo ocena za brzinu i jednostavnost, dokazujući svoju reputaciju jednog od najbržih modernih tekstualnih editora.

Ova analiza potvrđuje da korisnici tekstualnih editora apsolutni prioritet daju performansama i brzini rada. Iako su alati poput *Vim-a* i *Notepad++* ocenjeni stabilno u svim kategorijama, zajednička nit svim ispitanicima je niska važnost AI trendova u kontekstu čistih tekstualnih editora, gde se primat ostavlja stabilnosti i efikasnosti.

4.2 Analiza opštih preferencija Git platformi

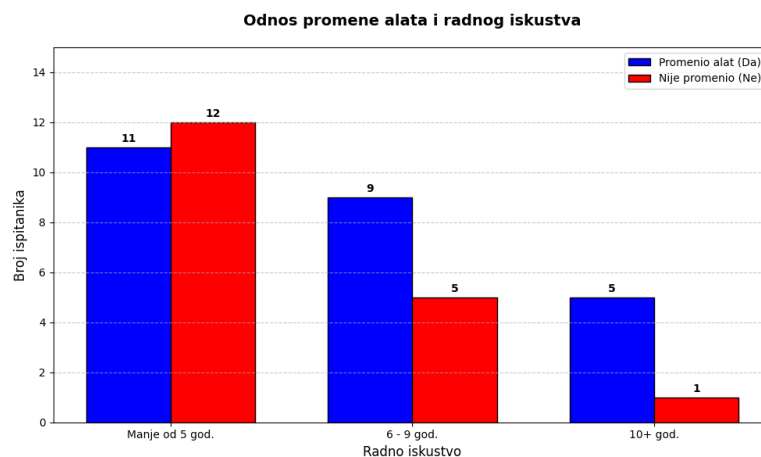
U okviru istraživanja, jedan od ciljeva bio je i utvrđivanje popularnosti vodećih platformi za upravljanje verzijama koda: *GitHub-a* i *GitLab-a*. Analiza je sprovedena na celokupnom uzorku ispitanika.

- **Apsolutna dominacija GitHub platforme:** Najveći deo uzorka, tačnije 88,4% (ukupno 38 ispitanika), opredelio se za *GitHub*.
- **Zastupljenost GitLab-a:** Znatno manji udeo ispitanika, odnosno 11,6% (ukupno 5 ispitanika), koristi *GitLab* kao primarnu platformu.

Odnos od skoro 9:1 u korist GitHub-a ukazuje na to da korisnici u velikoj većini biraju platforme koje omogućavaju najlakšu kolaboraciju i najveću bazu zajednice.

4.3 Analiza uticaja radnog iskustva na promenu razvojnog alata

U ovom delu istraživanja analizira se korelacija između dužine radnog staža ispitanika i njihove tendencije ka promeni primarnog razvojnog alata. Rezultati prikazani na slici 5 pružaju uvid u to koliko su programeri u različitim fazama karijere skloni usvajanju novih tehnologija.



Slika 4: Uticaj radnog iskustva na promenu razvojnog alata

Slika 5: Odnos promene razvojnog alata i radnog iskustva (grupisano)

Kako bi se jasnije uočili trendovi, podaci su grupisani u tri ključne kategorije, a rezultati ukazuju na sledeće:

- **Manje od 5 godina iskustva:** Ova objedinjena grupa (koja obuhvata početnike i programere sa 3–5 godina staža) predstavlja najbrojniji deo uzorka. Primećuje se umerena podeljenost u navikama: od ukupno 23 ispitanika, 11 je promenilo alat, dok je 12 ostalo pri prvobitnom izboru.

- **Srednji nivo (6–9 godina iskustva):** Kod ovih ispitanika uočen je visok stepen prilagođavanja trendovima. Većina programera u ovoj grupi (9 od ukupno 14) odlučila se za promenu primarnog alata.
- **Iskusni kadar (10+ godina iskustva):** Grupa sa najdužim stažom pokazuje najveću proaktivnost. Od 6 ispitanika, čak 5 je prešlo na novo okruženje.

Ukupni podaci sugerišu da akumulacija radnog staža (posebno preko 6 godina) korelira sa većom verovatnoćom promene alata. Razlog za ovakav trend verovatno leži u kompleksnosti projekata, kao i u integraciji novih tehnologija poput veštačke inteligencije u modernim razvojnim okruženjima.

4.4 Analiza uticaja navike na promenu razvojnog alata

Jedan od ciljeva istraživanja bio je ispitivanje uloge navike u procesu donošenja odluke o promeni glavnog razvojnog alata. U tom kontekstu analizirana je povezanost između ocene značaja navike odgovora ispitanika na pitanje da li su u prethodnom periodu menjali svoj primarni razvojni alat.

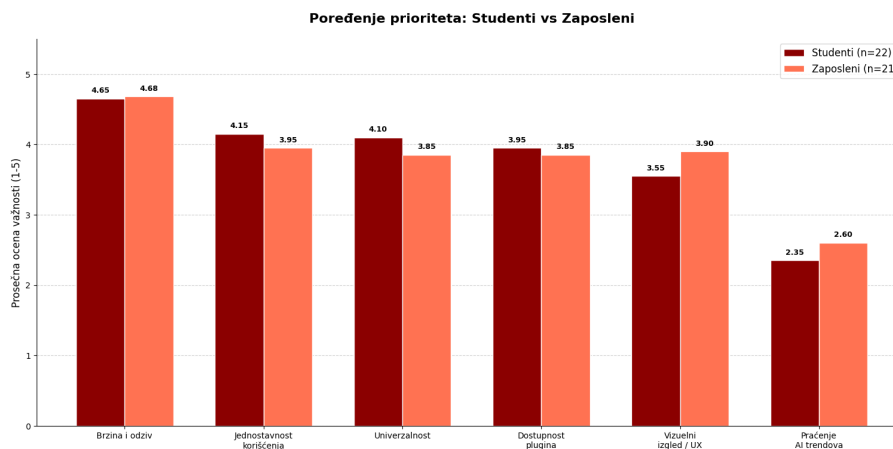
Analiza prosečnih ocena pokazuje da ispitanici koji su menjali glavni razvojni alat navici pridaju visoku važnost, sa prosečnom ocenom od približno 4.1, dok ispitanici koji nisu menjali alat naviku ocenjuju prosečnom ocenom od oko 4.0 na skali od 1 do 5.

Rezultati pokazuju da obe grupe ispitanika naviku ocenjuju kao važan faktor u izboru razvojnog alata. Iako je početna pretpostavka bila da će ispitanici kojima navika ima manji značaj biti skloniji promeni alata, uočava se da i ispitanici koji su menjali alat daju relativno visoke ocene uticaju navike. Razlika između prosečnih vrednosti ove dve grupe je prisutna, ali nije izražena, što ukazuje na to da navika ima snažan, ali ne i presudan uticaj na odluku o promeni razvojnog alata.

Dobijeni rezultati sugerišu da promena razvojnog alata ne podrazumeva odsustvo navike, već pre spremnost korisnika da postojeće navike prevaziđu u situacijama kada novi alat nudi jasne funkcionalne prednosti ili bolje odgovara zahtevima konkretnog projekta.

4.5 Uporedna analiza prioriteta: Studenti nasuprot zaposlenih

Finalni segment analize fokusira se na razlike u prioritetima između studenata ($n = 22$) i zaposlenih u IT sektoru ($n = 21$). Poređenje rezultata na objedinjenom prikazu (slika 6) omogućava precizan uvid u evoluciju potreba programera tokom profesionalnog razvoja.



Slika 6: Uporedni prikaz važnosti faktora za grupe: STUDENTI i ZAPOSLENI

Na osnovu uporednog pregleda srednjih ocena važnosti (\bar{X}) sa slike 6, uočavaju se sledeće ključne razlike:

- **Neprikosnovenost performansi:** Obe grupe su *Brzinu i odziv* ocenile kao najvažniji faktor. Zaposleni daju nijansu višu ocenu (4.68) u odnosu na studente (4.65), što potvrđuje da je efikasnost alata apsolutni prioritet bez obzira na profesionalni status.
- **Jednostavnost nasuprot profesionalnom UX-u:**
 - **Studenti** daju značajno veći prioritet *Jednostavnosti korišćenja* (4.15), što reflektuje potrebu za brzim savladavanjem alata tokom procesa učenja.
 - **Zaposleni** pokazuju blagi pad važnosti puke jednostavnosti (3.95), ali zato daju приметно veću važnost faktoru *Vizuelni izgled / UX* (3.90 naspram studentskih 3.55). Ovo ukazuje na to da profesionalci, usled dugotrajnog svakodnevnog rada u alatima, više vrednuju ergonomiju i kvalitet dizajna interfejsa.
- **Ekstenzibilnost i univerzalnost:** Kod zaposlenih je uočena potpuna ujednačenost u vrednovanju *Univerzalnosti* i *Dostupnosti plugina* (obe ocene su 3.85), dok studenti daju blagu prednost univerzalnosti alata (4.10) u odnosu na ekstenzije (3.95).
- **Odnos prema AI inovacijama:** Profesionalni sektor pokazuje veću otvorenost ka novim tehnologijama, pa **zaposleni** daju višu ocenu *Praćenju AI trendova* (2.60) u odnosu na **studente** (2.35). Ovo sugeriše da iskusniji programeri brže prepoznaju upotrebnu vrednost AI asistencija u realnim produkcionim uslovima.

Može se zaključiti da, dok su bazični zahtevi za brzinom rada univerzalni, profesionalizacija donosi povećan fokus na korisničko iskustvo (UX) i AI integracije, dok važnost same jednostavnosti primene blago opada u korist naprednijih i bolje dizajniranih funkcionalnosti.

4.6 Analiza otvorenih odgovora ispitanika

Pored kvantitativnih podataka, ispitanici su pružili detaljnije obrazloženje svojih preferencija kroz otvorena pitanja o razlozima izbora glav-

nog alata i problemima sa kojima se susreću. Analiza ovih odgovora otkriva nekoliko ključnih obrazaca:

- **Razlozi za izbor alata:** Dominantan faktor kod korisnika *Visual Studio Code*-a je balans između jednostavnosti i funkcionalnosti, pri čemu se često ističu intuitivnost i dobar korisnički doživljaj (UX). S druge strane, korisnici *JetBrains* porodice (IntelliJ, PyCharm, CLion) primarno biraju ove alate zbog robusne podrške za specifične programske jezike poput Jave i Spring framework-a. Zanimljivo je da značajan broj ispitanika kao razlog navodi eksterne faktore poput zahteva fakulteta ili radnog mesta („nametnuo posao”, „moram zbog faksa”), dok napredni korisnici (npr. *NeoVim*) fokus stavljaju na brzinu i efikasnost kroz eliminaciju upotrebe miša.
- **Problemi i ograničenja:** Kritike su jasno podeljene prema tipu alata. Kod kompleksnijih okruženja poput *Visual Studio*-a i *IntelliJ*-a, korisnici se najviše žale na veliku potrošnju memorije (RAM) i prisustvo „nepotrebnih opcija” koje opterećuju rad. Nasuprot tome, korisnici lakših editora (VSC) uočavaju nedostatke u naprednim funkcijama, kao što su slabija signalizacija grešaka u poređenju sa punim IDE rešenjima, problemi sa C++ podrškom, ili povremeni otkaz opcije „Go to Definition”.
- **Uloga pomoćnih alata i AI:** Iako su pitanja bila usmerena na razvojna okruženja, veliki broj ispitanika je spontano izdvojio *Git* kao svoj najkorišćeniji alat, definišući ga kao „apsolutno neophodan”. Takođe, primećuje se prodor novih alata poput *Cursor*-a, gde korisnici ističu visoku produktivnost, ali i uočavaju rane probleme poput AI halucinacija.

Analiza otvorenih odgovora pokazuje da se izbor razvojnog alata zasniva na balansu između funkcionalnosti, performansi i navike korisnika. Ispitanici su spremni da prihvate veću složenost ili hardversku zahtevnost alata ukoliko on donosi značajnu uštedu vremena i olakšava rad na kompleksnim projektima, dok se potencijalni pravci unapređenja uglavnom odnose na optimizaciju performansi, unapređenje signalizacije grešaka i bolje prilagođavanje specifičnim oblastima primene.

5 Diskusija

U ovom radu su analizirani obrasci korišćenja razvojnih alata među studentima tehničkih fakulteta i profesionalcima u IT sektoru, sa posebnim fokusom na kriterijume izbora i spremnost na promenu alata. Dobijeni rezultati se u velikoj meri poklapaju sa literaturom, ali ukazuju i na određene specifičnosti konteksta.

5.1 Najkorišćeniji razvojni alati

Analizirani su najkorišćeniji razvojni alati među ispitanicima, a rezultati su upoređeni sa relevantnom literaturom, uključujući izvore [5] za IDE i tekstualne editore, [6] za sisteme za verzionisanje, kao i [7] za platforme poput GitHub-a i GitLab-a.

- **IDE alati:** Među našim ispitanicima je najkorišćeniji alat *Visual Studio/Visual Studio Code*, što je u skladu sa literaturom u kojoj se ovaj alat nalazi na prvom mestu liste najkorišćenijih razvojnih

okruženja. Pored toga, razvojna okruženja iz *JetBrains* porodice imaju izraženu zastupljenost u našem uzorku, gde *IntelliJ IDEA* koristi 46,5% ispitanika, *PyCharm* 11,6%, a *CLion* 14%. Ovakva zastupljenost je takođe u saglasnosti sa literaturom, prema kojoj se *JetBrains IDEs* nalaze na četvrtoj poziciji najkorišćenijih razvojnih alata.

- **Tekstualni editori:** Najzastupljeniji tekstualni editori kod naših ispitanika su *Notepad++* (32,6%), *Vim* (27,9%) i *Kate* (25,6%), dok *Sublime Text* koristi svega 9,3% ispitanika. Ovo predstavlja značajno odstupanje u odnosu na literaturu, u kojoj se *Sublime Text* nalazi čak na drugom mestu liste najkorišćenijih razvojnih alata. Uočena razlika može se objasniti uticajem akademskog okruženja i ličnih navika korisnika, kao i relativno malim obimom uzorka.
- **Sistemi za verzionisanje koda:** Kada je u pitanju verzionisanje koda, *Git* se izdvaja kao ubedljivo najpopularniji sistem, kako prema literaturi, tako i u našem istraživanju, gde ga koristi čak 100% ispitanika. Ovakav rezultat potvrđuje status *Git*-a kao industrijskog standarda u savremenom razvoju softvera, nezavisno od korišćenog razvojnog okruženja ili nivoa iskustva programera. U našoj anketi se jasno vidi znatna dominacija *GitHub*-a nad *GitLab*-om kao platformama za upravljanje verzijama koda — *GitHub* koristi 88,4% ispitanika, dok *GitLab* koristi 11,6%. Ovaj rezultat ukazuje na snažnu preferencu prema *GitHub*-u i u našem uzorku, što se poklapa sa opštim trendom u IT industriji, gde *GitHub* generalno ima znatno veći udeo u korišćenju u odnosu na druge *Git* platforme.

5.2 Faktori koji utiču na izbor alata

Pored same zastupljenosti pojedinih alata, rezultati istraživanja pružaju uvid u obrasce ponašanja i navike programera prilikom izbora i promene razvojnog okruženja. Analiza faktora izbora pokazuje da su *brzina i odziv* alata dosledno ocenjeni kao najvažniji kriterijum, nezavisno od pola, radnog statusa ili nivoa iskustva ispitanika. Ovakav rezultat potvrđuje da programeri primarno vrednuju efikasnost i stabilnost, budući da razvojni alati predstavljaju centralni deo svakodnevnog rada.

Takođe, analiza spremnosti na promenu razvojnog alata ukazuje da programeri sa većim radnim iskustvom češće menjaju svoje primarno okruženje. Ovo sugeriše da se sa razvojem karijere smanjuje vezanost za konkretan alat, dok raste fokus na prilagodljivost i efikasnost u različitim tehnološkim kontekstima. Nasuprot tome, manje iskusni programeri češće ostaju pri jednom alatu, što može biti posledica formiranja početnih navika i potrebe za stabilnim okruženjem tokom učenja.

Razlike su uočljive i u odnosu prema savremenim tehnologijama. Zaposleni programeri pokazuju veću otvorenost ka praćenju *AI trendova* i korišćenju *AI* alata u razvoju softvera, što može biti posledica rada na realnim produkcionim projektima i potrebe za većom produktivnošću. Studenti, iako upoznati sa ovim tehnologijama, u manjoj meri ih doživljavaju kao važan faktor pri izboru razvojnog alata.

Poseban značaj ima i uticaj dugogodišnjih navika na izbor razvojnog alata. Većina ispitanika je ocenila ovaj faktor veoma visoko — čak 62,8% ispitanika dalo je ocenu 4, a dodatnih 23,3% ocenu 5, dok zanemarljiv broj smatra da navike imaju mali ili nikakav uticaj. Ovakav rezultat ukazuje

da se programeri često zadržavaju na alatima koje dobro poznaju, čak i kada postoje alternative sa sličnim ili naprednijim funkcionalnostima.

U celini, rezultati istraživanja ukazuju da izbor razvojnog alata nije zasnovan isključivo na tehničkim karakteristikama, već predstavlja kombinaciju profesionalnih potreba, stečenih navika i konteksta u kojem programer radi ili se obrazuje.

5.3 Ograničenja istraživanja

Iako dobijeni rezultati pružaju uvid u aktuelne trendove korišćenja razvojnih alata i preference korisnika, istraživanje ima nekoliko ograničenja koja treba uzeti u obzir pri tumačenju rezultata.

Prvo, uzorak od 43 ispitanika nije dovoljno obiman da bi se izvukli generalni zaključci za širu populaciju programera. Za značajnije statističke procene i pouzdano određivanje trendova, bio bi potreban znatno veći i raznovrsniji uzorak.

Drugo, sastav uzorka je specifičan — većina ispitanika su studenti Matematičkog fakulteta, što može povući pristrasnost prema određenim alatima, budući da se na fakultetu koriste uglavnom određeni razvojni alati i okruženja.

Treće, rezultati su zasnovani na samoproceni ispitanika, što može biti pod uticajem subjektivnih stavova, navika ili trenutnog iskustva. Navike korišćenja alata, prethodno iskustvo i lične preferencije mogu značajno uticati na odgovore, što ograničava mogućnost da se dobijeni podaci tretiraju kao apsolutno reprezentativni.

Konačno, anketa nije obuhvatila sve moguće faktore koji utiču na izbor alata, kao što su specifični zahtevi projekata, industrijski standardi u kompanijama ispitanika ili regionalne razlike, što znači da određene varijable koje mogu biti značajne nisu uzete u obzir.

Uzimajući sve ovo u obzir, rezultati treba interpretirati sa oprezom i kao okvirne smernice, a ne kao definitivan prikaz navika i preferencija svih programera.

6 Zaključak

Na osnovu sprovedene ankete i analize dobijenih rezultata, može se zaključiti da su *Visual Studio* i *Visual Studio Code* dominantni alati u praksi, što je u skladu sa savremenim industrijskim trendovima i relevantnom literaturom.

Istraživanje je pokazalo da su performanse alata, odnosno brzina i odziv, ključni kriterijumi pri izboru razvojnog okruženja, dok faktori poput vizuelnog izgleda i praćenja AI trendova imaju manji, ali ne zanemarljiv značaj. Uočene razlike između studenata i zaposlenih ukazuju na promenu prioriteta tokom profesionalnog razvoja.

Rezultati takođe potvrđuju dominaciju sistema *Git* kao standarda za verzionisanje koda, kao i znatno veću popularnost platforme *GitHub* u odnosu na *GitLab*. Ovakvi nalazi dodatno naglašavaju homogenost savremenih razvojnih praksi i oslanjanje na globalno prihvaćene alate i platforme.

Iako istraživanje ima određena ograničenja, pre svega u pogledu veličine i strukture uzorka, dobijeni podaci pružaju relevantan uvid u aktuelne navike i preferencije programera. Ovaj rad može poslužiti kao osnova za buduća, obimnija istraživanja koja bi uključila veći broj ispitanika, širi spektar tehnologija i dublju analizu uticaja novih trendova, poput veštačke inteligencije, na izbor razvojnih alata.

Literatura

- [1] I. Zayour, H. Hajjdiab, “How Much Integrated Development Environments (IDEs) Improve Productivity?”, 2013.
- [2] I. Sommerville, *Software Engineering*, 9th ed., Addison-Wesley, Boston, 2011.
- [3] E. Sink, *Version Control by Example*, Pyrenean Gold Press, 2011.
- [4] The Pros and Cons of Using a Code Editor vs. Integrated Development Environment (IDE) at:
<https://medium.com/@taradwyer17/the-pros-and-cons-of-using-a-code-editor-vs-integrated-dev>
- [5] Top 27 Software Development Tools and Platforms (2025 List) at:
<https://spacelift.io/blog/software-development-tools>
- [6] Version Control Systems at:
<https://www.geeksforgeeks.org/git/version-control-systems/>
- [7] GitLab vs GitHub at:
<https://evbn.org/gitlab-vs-github-explore-their-major-differences-and-similarities-1678449>