

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

ovde pišemo abstrakt.

Sadržaj

1	Uvod	2
2	O razvojnim alatima	2
2.1	Integrirana razvojna okruženja (IDE)	2
2.2	Tekstualni editori koda	3
2.3	Sistemi za verzionisanje	3
3	Metodologija	3
3.1	Uzorkovanje	4
3.2	Struktura uzorka	4
3.3	Struktura ankete	4
3.4	Analiza dobijenih rezultata	5
4	Rezultati	5
4.1	Analiza faktora važnosti pri izboru alata	6
4.2	Analiza korišćenja razvojnih okruženja u odnosu na pol	7
4.3	Analiza preferencija Git platformi kod korisnika IntelliJ IDEA okruženja	8
4.4	Analiza uticaja radnog iskustva na promenu razvojnog alata	8
4.5	Analiza važnosti faktora kod korisnika tekstualnih editora	9
4.6	Uporedna analiza prioriteta: MATF, FON i RAF	11
4.7	Uporedna analiza prioriteta: Studenti nasuprot zaposlenih	13
5	Diskusija	14
6	Zaključak	14
	Literatura	14

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.

Istraživanje je sprovedeno putem onlajn ankete, pri čemu su prikupljeni kvantitativni i kvalitativni podaci o navikama, stavovima i iskustvima ispitanika u vezi sa pomenutim razvojnim alatima. 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 programeerima omogućavaju efikasnije pisanje, organizaciju, testiranje i održavanje programskog koda. 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 objedinuju više funkcionalnosti potrebnih za razvoj softvera u jedinstveno radno okruženje. Tipične komponente IDE-a uključuju editor koda, kompajler ili interpreter, alate za dinamičku analizu koda, kao i dodatne funkcionalnosti za upravljanje projektima i testiranje.

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.

Primeri integrisanih razvojnih okruženja koji se često koriste u praksi uključuju Visual Studio, IntelliJ IDEA, PyCharm i Android Studio.

Sa druge strane, integrisana razvojna okruženja mogu zahtevati veće hardverske resurse i biti složenija za prilagođavanje individualnim potre-

bama korisnika. Uprkos tome, njihova sveobuhvatnost čini ih posebno pogodnim kako za profesionalno okruženje, tako i za obrazovne svrhe.

2.2 Tekstualni editori koda

Tekstualni editori koda predstavljaju alate namenjene pisanju i uređivanju izvornog koda, sa fokusom na jednostavnost i fleksibilnost. 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. Ovakav modularni pristup korisnicima pruža veću kontrolu nad razvojnim okruženjem.

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. Njihova osnovna uloga jeste čuvanje istorije izmena, omogućavanje povratka na prethodne verzije koda i olakšavanje saradnje više korisnika na istom projektu.

Upotrebom sistema za verzionisanje, programeri mogu paralelno raditi na različitim delovima projekta, uz kasnije objedinjavanje izmena. Ovi sistemi značajno smanjuju rizik od gubitka podataka i doprinose boljoj timskoj organizaciji.

Sistemi za verzionisanje mogu se podeliti na centralizovane i distribuirane, u zavisnosti od načina na koji se čuva i deli istorija izmena. Centralizovani sistemi za verzionisanje podrazumevaju postojanje centralnog repozitorijuma na kojem se čuva kompletna istorija projekta, dok korisnici pristupaju tom repozitorijumu radi preuzimanja i slanja izmena. Ovakav pristup olakšava centralnu kontrolu, ali može predstavljati problem u slučaju nedostupnosti centralnog servera.

Nasuprot tome, distribuirani sistemi za verzionisanje omogućavaju svakom korisniku da poseduje potpunu lokalnu kopiju repozitorijuma. Ovakav model omogućava rad bez stalne konekcije sa centralnim serverom, veću fleksibilnost i pouzdanost, kao i efikasniji paralelni razvoj. Zbog ovih karakteristika, distribuirani sistemi za verzionisanje danas su dominantni u savremenom razvoju softvera.

Najpoznatiji primer distribuiranog sistema za verzionisanje je Git.

3 Metodologija

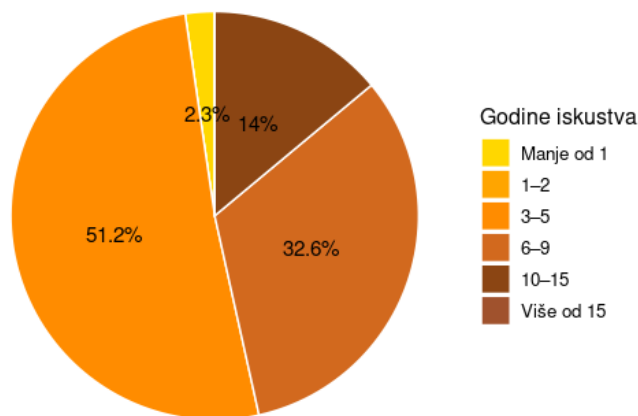
Istraživanje je sprovedeno korišćenjem anonimne onlajn ankete. Metodološki pristup je dominantno kvantitativan, budući da su podaci prikupljeni putem strukturiranih pitanja i analizirani primenom deskriptivne statistike, uz ograničene kvalitativne uvide dobijene analizom odgovora otvorenog tipa.

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. Pregled ostalih demografskih karakteristika prikazan je na slikama 1 i 2.

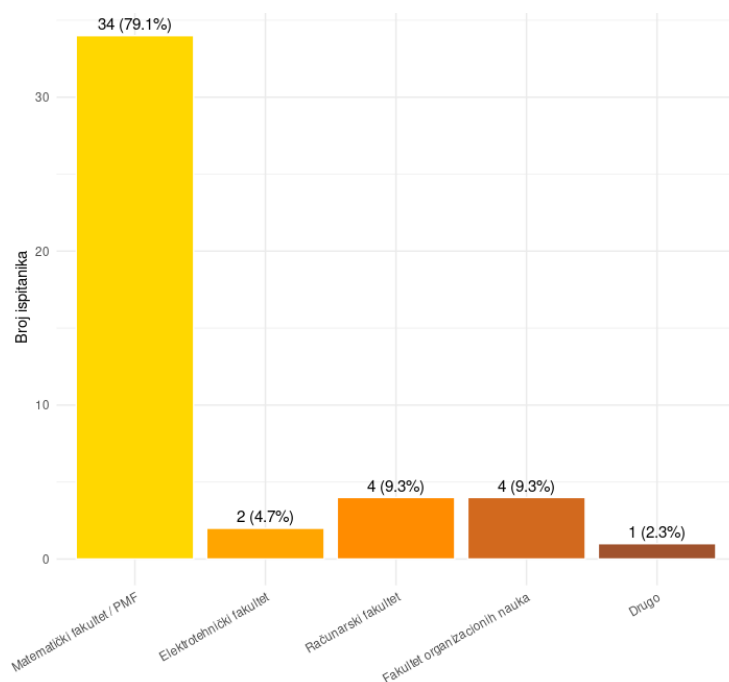


Slika 1: Raspodela ispitanika po dužini iskustva u programiranju.

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



Slika 2: Broj ispitanika po fakultetima.

alata, razlozima za prelazak na novi alat i faktorima koji bi motivirali tu promenu. Za većinu ovih pitanja korišćena je *Likert skala* od 1 do 5.

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

3.4 Analiza dobijenih rezultata

4 Rezultati

U ovom poglavlju izloženi su podaci prikupljeni sprovedenom anketom, koji daju uvid u najkorisnije razvojne alate. Uzorak obuhvata ispitanike različitih nivoa stručnosti od studenata do zaposlenih u IT industriji što omogućava analizu tržišta iz više perspektiva.

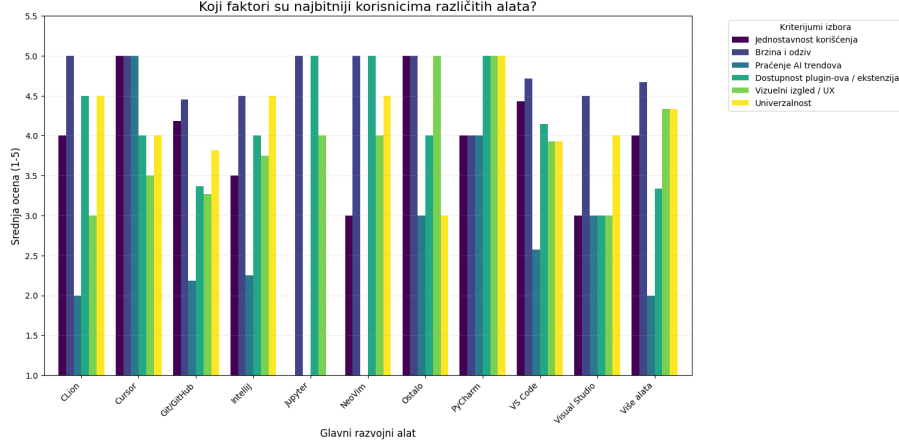
Prikazani rezultati su podeljeni u nekoliko logičkih celina:

- Ocene važnosti ključnih funkcionalnosti (brzina, UX, AI asistenti) u zavisnosti od kategorije alata.
- Statistički podaci o korišćenju razvojnih okruženja (IDE) s obzirom na pol i radni status ispitanika.
- Odnos između godina radnog iskustva i spremnosti na promenu primarnog alata.
- Specifičnosti u zahtevima studenata sa MATF-a, FON-a i RAF-a.

Svi grafikoni su generisani na osnovu odgovora ispitanika, a prateća analiza se fokusira na uočene trendove i korelacije u dobijenim podacima.

4.1 Analiza faktora važnosti pri izboru alata

U ovom delu rada fokusiramo se na kriterijume koji programeri smatraju presudnim prilikom odabira radnog okruženja. Na slici 3 prikazani su rezultati za 11 različitih kategorija alata.



Slika 3: Važnost faktora prilikom izbora razvojnog alata

Za svaki od 11 analiziranih razvojnih alata izračunata je prosečna ocena kvaliteta \bar{X}_A , koja predstavlja aritmetičku sredinu šest ključnih kriterijuma (brzina, jednostavnost, univerzalnost, dostupnost dodataka, vizuelni izgled i AI trendovi). Formula po kojoj je vršeno računanje za svaki alat A je:

$$\bar{X}_A = \frac{f_{brzina} + f_{jednost} + f_{univ} + f_{plugin} + f_{visual} + f_{AI}}{6} \quad (1)$$

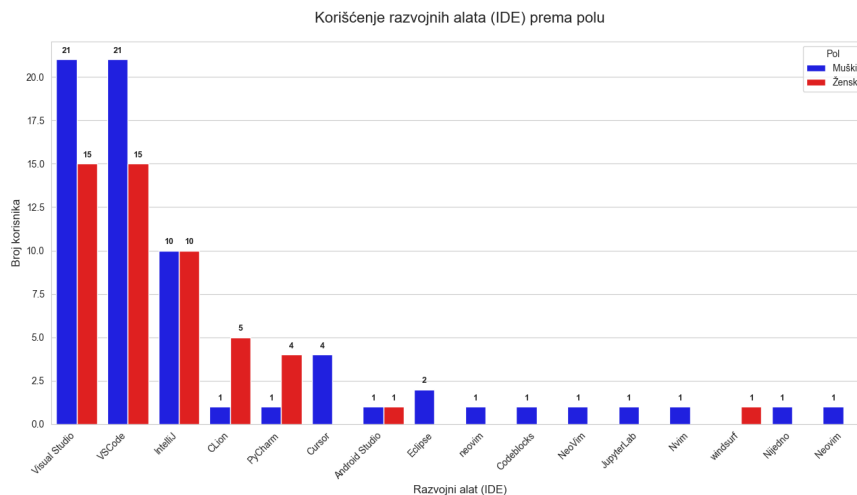
Na osnovu ove metrike, možemo preciznije rangirati alate prema ukupnom zadovoljstvu korisnika:

- **Cursor:** $\bar{X} \approx 4.41$ — Najviše rangiran alat, prvenstveno zbog maksimalnih ocena u kategorijama brzine i praćenja AI trendova.
- **CLion i IntelliJ:** $\bar{X} \approx 3.85$ — Ovi alati zadržavaju visok prosek zbog ujednačenih ocena u svim kategorijama, uprkos nižem rezultatu za AI trendove.
- **VS Code:** $\bar{X} \approx 3.78$ — Balansiran alat sa visokim ocenama za dostupnost dodataka i brzinu.
- **PyCharm:** $\bar{X} \approx 4.00$ — Visok prosek zahvaljujući ujednačenosti svih faktora, gde nijedna ocena ne pada ispod 3.
- **NeoVim i Vim:** $\bar{X} \approx 3.42$ — Specifični alati koji imaju ekstremno visoke ocene za brzinu, ali niži prosek zbog vizuelnog izgleda i AI integracija.
- **Visual Studio:** $\bar{X} \approx 3.41$ — Solidan prosek koji održava univerzalnost i brzinu, dok su AI trendovi slabije ocenjeni.

Ovakvo računanje pokazuje da moderni alati poput *Cursor-a* trenutno nude najkompletniji paket funkcionalnosti, dok tradicionalni editori (Vim, NeoVim) ostaju specijalizovani alati sa visokim performansama, ali užim fokusom na moderne vizuelne i AI standarde.

4.2 Analiza korišćenja razvojnih okruženja u odnosu na pol

Ova podsekcija analizira distribuciju najpopularnijih razvojnih alata (IDE) među muškim i ženskim ispitanicima. Podaci su prikupljeni kako bi se utvrdilo da li postoji značajna korelacija između pola i izbora softvera za programiranje.



Slika 4: Korišćenje IDE alata u zavisnosti od pola ispitanika

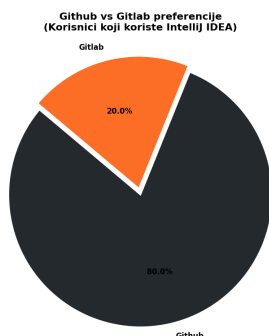
Na osnovu podataka prikazanih na slici 4, uočavamo odnos razvojnih alata u odnosu na pol ispitanika:

- **Dominantni alati:** *Visual Studio* i *VS Code* su najzastupljeniji alati u uzorku. Interesantno je da oba rešenja imaju identičnu strukturu korisnika, po 21 muški i 15 ženskih ispitanika, što ukazuje na njihovu široku prihvaćenost bez obzira na pol.
- **Uravnotežena zastupljenost:** Alat *IntelliJ IDEA* izdvaja se po potpunoj ravnopravnosti, sa tačno 10 muških i 10 ženskih korisnika, što ga čini statistički najneutralnijim alatom u pogledu preferencija polova.
- **Specifične razlike:**
 - Alati poput *CLion*-a i *PyCharm*-a pokazuju veću zastupljenost kod ženskih ispitanika u odnosu na njihov ukupan broj korisnika.
 - S druge strane, rešenja kao što su *Cursor* i *Eclipse* u ovom istraživanju su koristili isključivo muški ispitanici.

Ovi rezultati sugerišu da su industrijski standardi, poput Microsoft-ove porodice alata, univerzalno prihvaćeni u celoj zajednici. Razlike u preferencijama postaju uočljivije kod alata koji su usko specijalizovani za određene programske jezike ili kod novijih rešenja baziranih na veštačkoj inteligenciji.

4.3 Analiza preferencija Git platformi kod korisnika IntelliJ IDEA okruženja

U okviru istraživanja, specifičan fokus je stavljen na korisnike koji kao svoje primarno ili često razvojno okruženje koriste *IntelliJ IDEA*. Cilj je bio utvrditi da li izbor profesionalnog IDE alata korelira sa izborom platforme za upravljanje verzijama koda (Git). Rezultati su prikazani na slici 5.



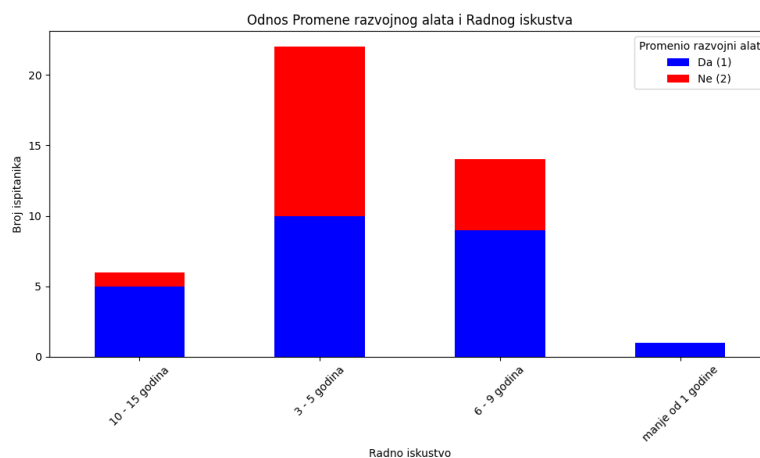
Slika 5: Odnos GitHub i GitLab korisnika među korisnicima IntelliJ IDEA okruženja

Na osnovu vizuelnog prikaza sa slike 5, uočavaju se jasni trendovi u izboru platformi za kontrolu verzija među korisnicima *IntelliJ IDEA* okruženja:

- **Dominacija GitHub platforme:** Velika većina ispitanika, tačnije 80%, opredelila se za *GitHub*. Ovako visok procenat potvrđuje da je GitHub primarni standard u zajednici, verovatno zbog njegove široke zastupljenosti u open-source projektima i industriji.
- **Zastupljenost GitLab-a:** Preostalih 20% korisnika koristi *GitLab*. Iako je ovaj udeo značajno manji, on ukazuje na prisustvo dela zajednice koji se fokusira na specifične *DevOps* funkcionalnosti ili rad u okviru privatnih repozitorijuma koje GitLab tradicionalno podržava.
- **Zaključak o homogenosti uzorka:** Dobijeni rezultati pokazuju da je grupa korisnika *IntelliJ* alata prilično ujednačena u pogledu korišćenja pratećih alata, sa izrazitom preferencijom ka globalnom standardu koji predstavlja *GitHub*.

4.4 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 6 pružaju uvid u to koliko su programeri u različitim fazama karijere skloni usvajanju novih tehnologija.



Slika 6: Uticaj radnog iskustva na promenu razvojnog alata

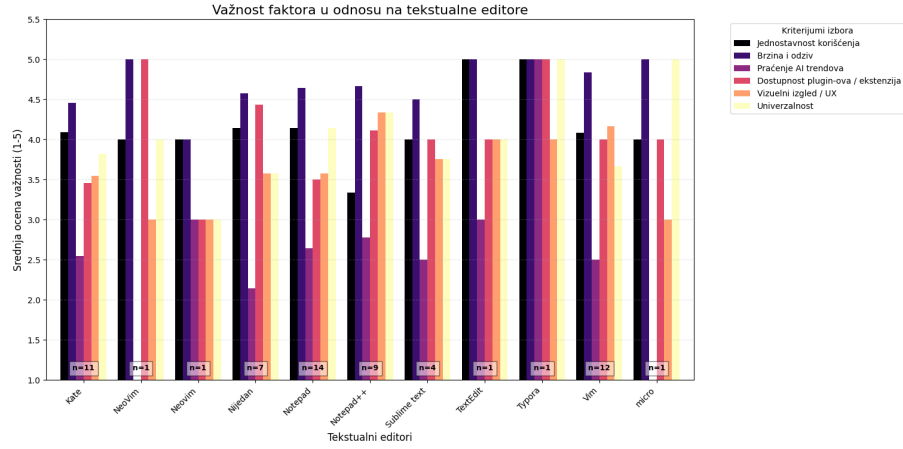
Na osnovu podataka prikazanih na slici 6, analizirana je spremnost ispitanika da promene svoj primarni razvojni alat u zavisnosti od godina radnog iskustva. Rezultati ukazuju na sledeće trendove:

- **Iskusni programeri (10–15 godina):** Ova grupa pokazuje izuzetno visoku stopu promene alata. Od ukupno 6 ispitanika, čak 5 je prešlo na novo okruženje, što sugerise da iskusniji kadar ne okleva da usvoji modernija rešenja ukoliko ona nude veću efikasnost u radu.
- **Grupa sa 6–9 godina iskustva:** Kod ovih ispitanika je takođe uočen visok stepen prilagođavanja, gde je 9 od 14 programera promenilo svoj primarni alat.
- **Srednji nivo iskustva (3–5 godina):** Ovo je najbrojnija grupa u uzorku, u kojoj je uočljiva najveća podeljenost u navikama. Od 22 ispitanika, 10 je promenilo alat, dok je 12 ostalo pri svom prvobitnom izboru.
- **Početnici (manje od 1 godine):** Iako je uzorak u ovoj kategoriji minimalan, zabeleženo je da se promena alata dešava čak i u najranijoj fazi karijere.

Ukupni podaci ukazuju na to da programeri sa više radnog staža (preko 6 godina) zapravo češće menjaju alate nego što se to možda intuitivno očekuje. Razlog za ovo verovatno leži u evoluciji projekata na kojima rade, kao i u pojavi naprednijih alata baziranih na novim tehnologijama poput veštačke inteligencije.

4.5 Analiza važnosti faktora kod korisnika tekstualnih editora

Pored kompleksnih integrisanih razvojnih okruženja (IDE), značajan deo ispitanika oslanja se na tekstualne editore. Na slici 7 prikazano je kako korisnici različitih editora ocenjuju ključne faktore prilikom izbora svog primarnog alata.



Slika 7: Važnost faktora u odnosu na najčešće korišćene tekstualne editore

Na osnovu prikupljenih odgovora, za svaki od analiziranih tekstualnih editora izračunata je prosečna ocena zadovoljstva korisnika. Ova vrednost predstavlja aritmetičku sredinu šest ključnih kriterijuma (brzina, jednostavnost, univerzalnost, dodaci, vizuelni izgled i AI trendovi), a računa se po formuli:

$$\bar{X}_E = \frac{f_{brzina} + f_{jednost} + f_{univ} + f_{plugin} + f_{visual} + f_{AI}}{6} \quad (2)$$

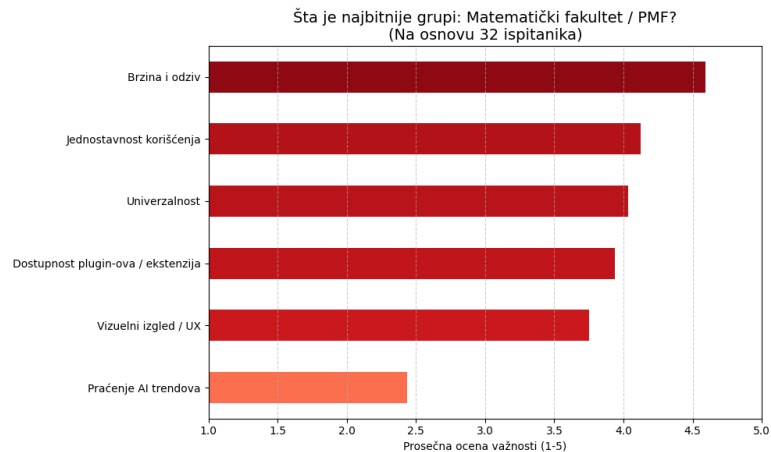
Analiza podataka sa slike 7 ukazuje na sledeće prosečne ocene po alatima:

- **Typora:** $\bar{X} \approx 4.83$ — Alat sa najvišim prosekom, gde su skoro svi faktori, uključujući i AI trendove, ocenjeni maksimalnim ocenama.
- **NeoVim i Vim:** $\bar{X} \approx 3.65$ — Ovi editori beleže ekstremno visoke ocene za brzinu i dostupnost dodataka, ali ukupni prosek vukom naniže nešto slabije ocene za vizuelni izgled i AI integracije.
- **Sublime Text:** $\bar{X} \approx 3.75$ — Karakteriše ga visok stepen ujednačenosti, sa posebnim naglaskom na brzinu i jednostavnost korišćenja.
- **Notepad++:** $\bar{X} \approx 3.73$ — Korisnici ovog alata najviše vrednuju univerzalnost i jednostavnost, dok su AI trendovi očekivano slabije ocenjeni.
- **Kate:** $\bar{X} \approx 3.63$ — Solidan prosek koji održavaju stabilne ocene za brzinu i univerzalnost formata.
- **Micro i TextEdit:** $\bar{X} \approx 3.50$ — Ovi editori se percipiraju kao jednostavna i brza rešenja, ali bez značajnog fokusa na proširivost ili moderan UX.

Ova analiza potvrđuje da korisnici tekstualnih editora prioritet daju performansama i brzini rada. Iako su modernija rešenja poput *Typora*-e visoko ocenjena u svim kategorijama, tradicionalna zajednica (Vim, Notepad++) ostaje verna alatima koji nude stabilnost i efikasnost, čak i uz svesno žrtvovanje vizuelnog ugođaja ili najnovijih tehnoloških dodataka.

4.6 Uporedna analiza prioriteta: MATF, FON i RAF

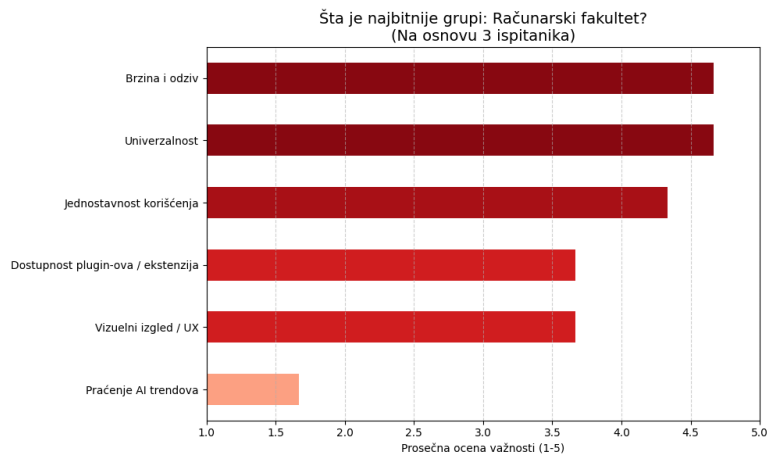
Poseban segment istraživanja posvećen je analizi prioriteta ispitanika u zavisnosti od fakulteta na kojem studiraju ili su diplomirali. Na slikama 8, 9 i 10 prikazani su rezultati za Matematički fakultet (MATF), Fakultet organizacionih nauka (FON) i Računarski fakultet (RAF).



Slika 8: Prioriteti ispitanika sa Matematičkog fakulteta



Slika 9: Prioriteti ispitanika sa Fakulteta organizacionih nauka



Slika 10: Prioriteti ispitanika sa Računarskog fakulteta

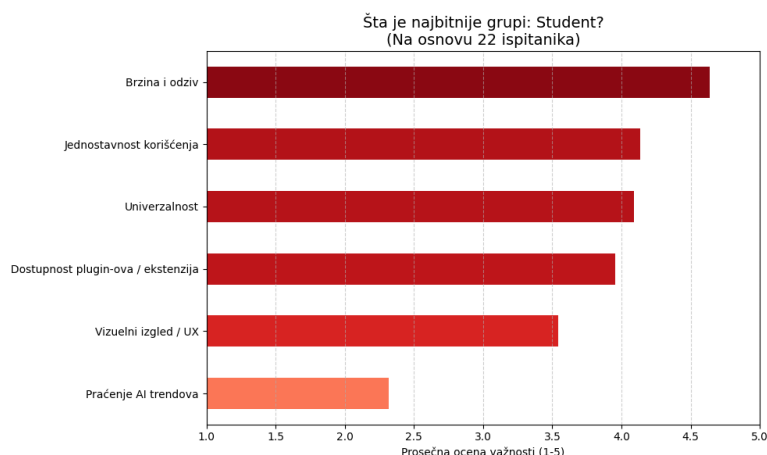
Komparativnom analizom prosečnih ocena važnosti \bar{X} za ove tri grupe, mogu se uočiti sledeći trendovi:

- **Zajednički konsenzus o performansama:** Kod svih grupa, *Brzina i odziv* je najviše rangiran faktor sa ocenama iznad 4.5. Ovo potvrđuje univerzalnu potrebu programera za alatima koji ne usporavaju proces pisanja koda.
- **Razlike u pragmatičnosti (Jednostavnost vs. Univerzalnost):**
 - Ispitanici sa **MATF-a** (na bazi 32 ispitanika) visoko vrednuju *Jednostavnost korišćenja* (≈ 4.12), stavljajući je ispred univerzalnosti alata.
 - Ispitanici sa **RAF-a** daju podjednako visok značaj *Univerzalnosti* i *Brzini* (obe ≈ 4.67), što može ukazivati na rad u različitim tehnologijama.
 - Ispitanici sa **FON-a** najviše vrednuju *Dostupnost plugin-ova* (≈ 4.0) odmah nakon brzine, što sugeriše sklonost ka prilagođavanju okruženja specifičnim potrebama.
- **Odnos prema AI trendovima:** Uočen je zanimljiv trend najnižih ocena za faktor *Praćenje AI trendova*. Dok je na MATF-u taj prosek oko 2.44, na RAF-u pada na samo 1.67, a na FON-u je nešto viši (≈ 2.67). Ovi podaci sugerišu da je akademska zajednica i dalje primarno fokusirana na stabilnost i jezgro funkcionalnosti alata u odnosu na trenutne AI inovacije.
- **Vizuelni izgled:** Najmanju važnost estetskom izgledu pridaju ispitanici sa FON-a (≈ 3.33), dok je na MATF-u i RAF-u taj prosek nešto viši ($\approx 3.67 - 3.75$).

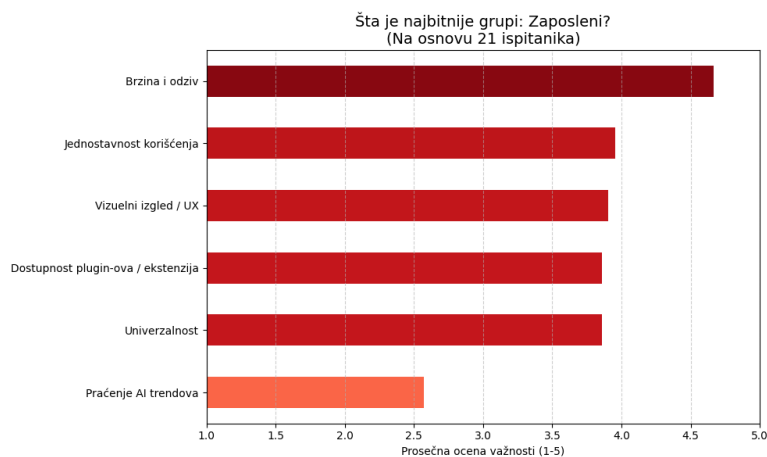
Iako su uzorci za FON i RAF znatno manji u odnosu na MATF, poređenje ukazuje na specifične nijanse u profilisanju studenata: dok MATF teži balansu jednostavnosti i performansi, RAF naglašava univerzalnost, a FON fleksibilnost kroz ekstenzije.

4.7 Uporedna analiza prioriteta: Studenti nasu-prot zaposlenih

Finalni segment analize fokusira se na razlike u prioritetima između studenata (22 ispitanika) i zaposlenih u IT sektoru (21 ispitanik). Poređenje rezultata prikazanih na slikama 11 i 12 omogućava uvid u evoluciju potreba programera tokom karijere.



Slika 11: Faktori koji su najbitniji studentima



Slika 12: Faktori koji su najbitniji zaposlenima u IT sektoru

Na osnovu uporednog pregleda srednjih ocena važnosti \bar{X} , uočavaju se sledeće ključne razlike:

- **Neprikosnovenost performansi:** Obe grupe su *Brzinu i odziv* ocenile gotovo identično visokom ocenom (≈ 4.67), što potvrđuje da je efikasnost alata apsolutni prioritet bez obzira na profesionalni status.

- **Jednostavnost vs. Profesionalni UX:**

- **Studenti** daju znatno viši značaj *Jednostavnosti korišćenja* (≈ 4.14), što je logično s obzirom na proces učenja i potrebu za brzim savladavanjem alata.
- **Zaposleni** pokazuju blagi pad važnosti jednostavnosti (≈ 3.95), ali zato daju veću važnost faktoru *Vizuelni izgled / UX* (≈ 3.90 naspram studentskih 3.55), što ukazuje na to da profesionalci koji provode više vremena u alatima više vrednuju ergonomiju i dizajn interfejsa.

- **Ekstenzibilnost i univerzalnost:** Zaposleni pokazuju veću ujednačenost u vrednovanju univerzalnosti i dostupnosti plugin-ova, dok studenti daju blagu prednost univerzalnosti alata.

- **Odnos prema AI inovacijama:** Interesantno je da **zaposleni** daju veću ocenu *Praćenju AI trendova* (≈ 2.57) u odnosu na **studente** (≈ 2.32). Ovo sugerise da profesionalni sektor brže prepoznaje praktičnu vrednost AI asistenata u produkcionom kodu.

Zaključno, dok su osnovni zahtevi za brzinom isti, prelazak u profesionalnu sferu donosi veći fokus na korisničko iskustvo (UX) i AI integracije, dok važnost puke jednostavnosti korišćenja blago opada u korist naprednijih funkcionalnosti.

5 Diskusija

6 Zaključak

ovde pišemo zaključak

Literatura

- [1] Naziv izvora at:
[link](#)