

ds

**dnevi
slovenske
informatike**

9. in 10. maj 2023

Kongresni center Bernardin, Portorož

30. konferenca Dnevi slovenske informatike
Soustvarjamo digitrajno Slovenijo

INTELIGENTNI POMOČNIKI IN NJIHOV VPLIV NA PISANJE IZVORNE KODE

Tilen Hliš, Luka Četina, Luka Pavlič

10. 5. 2023

Umetna inteligenca pri razvoju programske opreme

Umetna inteligenca

- Umetna inteligenca se vse bolj **uveljavlja v različnih področjih**.
- Uporaba metod umetne inteligence pri razvoju programske opreme **omogoča skrajšanje časa razvoja in izboljšanje kakovosti končnega izdelka**.

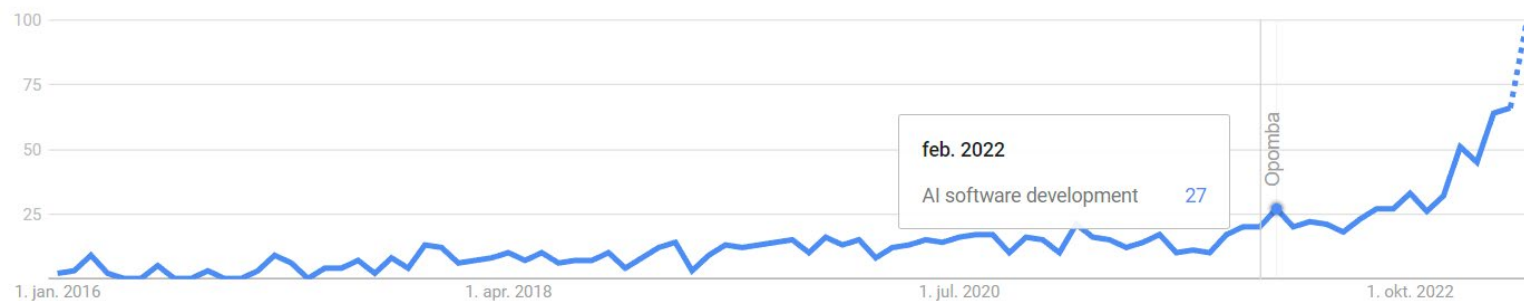


- Dve glavni možnosti:
- ***kot razčlenjevalec naravnega jezika,***
ali
- ***kot orodje za izboljšanje produktivnosti razvijalcev s samodejnim napovedovanjem in dokončevanjem izvorne kode.***

Pomočniki za dokončevanje izvorne kode

Tradicionalni / Inteligentni

- Tradicionalni pomočniki za dopolnjevanje kode **ne** ponujajo „pametnih“ predlogov na podlagi uporabe metod umetne inteligence.
- Inteligentni pomočniki za dokončevanje izvorne kode so **obetavna alternativa**, ki lahko **predvidijo** in **prilagodijo** se **namenom razvijalcev** ter tako **pospešijo razvoj** in **zmanjšajo število napak** v izvorni kodi.
- Obetavnost inteligentnih pomočnikov.
- **Omejen obseg in nabor funkcionalnosti.**
- Ali so pomočniki že v fazi, ko razvijalcem koristijo?



Tradicionalni pomočniki

Poglavitne funkcionalnosti tradicionalnih pomočnikov

Funkcionalnost
Dopolnjevanje trenutne besede (imena metod, spremenljivk, atributov)
Predvidevanje najbolj verjetne naslednje enote kode in prikaz seznama predlogov (na podlagi lokalnega konteksta)
Prikaz vseh možnih kandidatov (spremenljivk, metod, atributov) in podatkov o njih
Dopolnjevanje kode na osnovi predlog (for/while zanka, iterator)

Tradicionalni pomočniki

Poglavitne funkcionalnosti tradicionalnih pomočnikov - PRAKTIČNO

- Dopolnjevanje trenutne besede:

```
        }  
    });  
}  
public static void main(String[] args){  
    final BookStoreServerUnary |
```

Tradicionalni pomočniki

Poglavitne funkcionalnosti tradicionalnih pomočnikov - PRAKTIČNO

- Prikaz vseh možnih kandidatov (metod):

```

    },
}
public static void main(String[] args) {
    final BookStoreServerUnary greetServer = new BookStoreServerUnary();
}
! usage
static class BookStoreImpl extends BookStoreGrpc.BookStoreImplBase {
    @Override
    public void first(BookSearch searchQuery, StreamObserver<Book> responseObserver) {
        logger.info(msg: "Searching for book with title: " + searchQuery.getName());
        List<String> matchingBookTitles = bookMap.keySet().stream().filter(title ->
            title.startsWith(searchQuery.getName().trim())).collect(Collector

        Book foundBook = null;
        if(matchingBookTitles.size() > 0) {
            foundBook = bookMap.get(matchingBookTitles.get(0));
        }
        responseObserver.onNext(foundBook);
    }
}

```

Tradicionalni pomočniki

Poglavitne funkcionalnosti tradicionalnih pomočnikov - PRAKTIČNO

- Dopolnjevanje kode na osnovi predlog (for):

```
@Override  
public com.google.protobuf.ByteString  
    getGenreBytes() {  
    Object ref = genre_;  
    if (ref instanceof String) {  
        com.google.protobuf.ByteString b =  
            com.google.protobuf.ByteString.copyFromUtf8(  
                (String) ref);  
        genre_ = b;  
        return b;  
    } else {  
        return (com.google.protobuf.ByteString) ref;  
    }  
}
```

2 usages

```
private byte memoizedIsInitialized = -1;  
@Override
```

Inteligentni pomočniki

Poglavitne funkcionalnosti inteligentnih pomočnikov

Funkcionalnost
Dopolnjevanje trenutne besede (imena metod, spremenljivk, atributov)
Generiranje predlogov nadaljevanja programa glede na kontekst
Prikaz informacij o metodah in atributih trenutnega objekta (raziskovanje API-ja)
Generiranje predlogov imen novih spremenljivk ali metod
Generiranje predlogov za nadaljevanje kode na osnovi naravnega jezika (upošteva imena metod in spremenljivk ter komentarjev)

Poglavitne funkcionalnosti inteligentnih pomočnikov - PRAKTIČNO



20
21
22
23
24

Inteligentni pomočniki

Poglavitne funkcionalnosti preizkušenih inteligentnih pomočnikov

Pomočnik→ Funkcionalnost↓	GitHub Copilot	Tabnine	Kite	IntelliCode
Dopolnjevanje trenutne besede	✓	✓	✓	✓
Generiranje nadaljevanje programa	Besede	✓	✓	✓
	Vrstice	✓	✓	×
	Odsek	✓	×	×
Raziskovanje knjižnic	×	✓	✓	✓
Predlaganje knjižnic	✓	✓	✓	✓
Poimenovanje spremenljivk	✓	×	×	×
Generiranje predlogov za nadaljevanje kode na osnovi naravnega jezika	✓	✓	×	×
Učenje modela na lastni kodi (za podjetja)	×	✓	✓	✓

Izvedba empirične raziskave

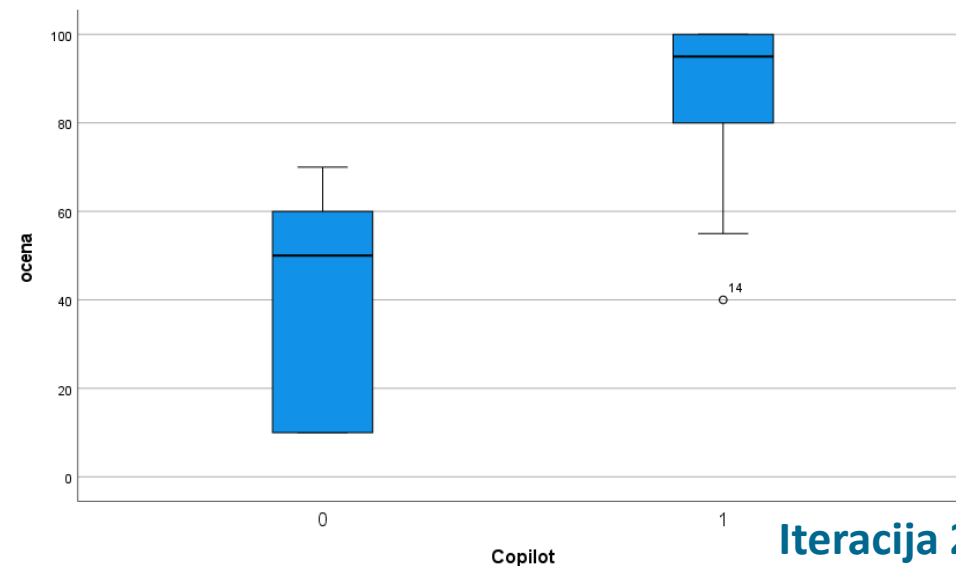
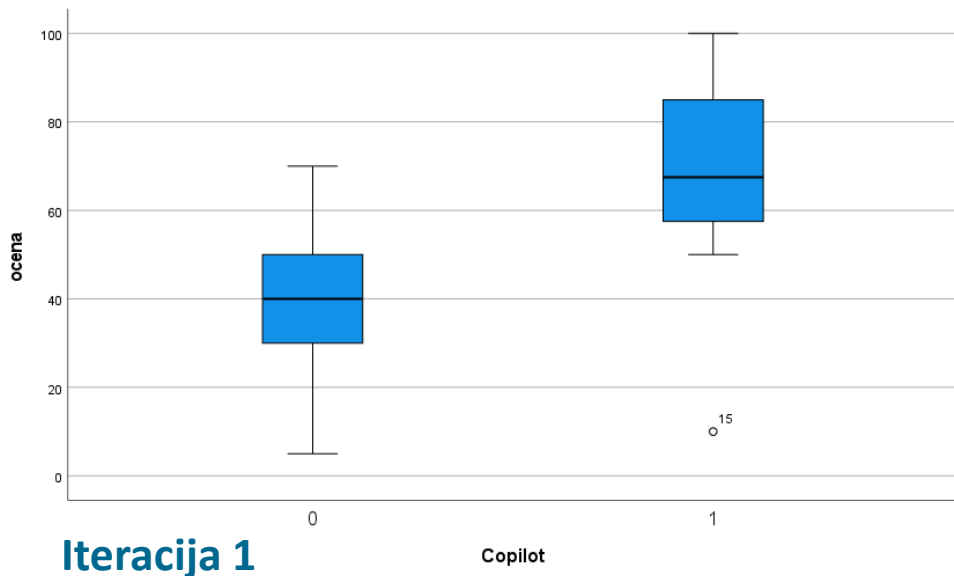
Predstavitve raziskave

- Eksperiment z uporabo orodja **GitHub Copilot**.
- **Namen:** *raziskati uporabnost in vpliv pomočnika na razvoj in dopolnjevanje obstoječe informacijske rešitve.*
- Izbran kot **najbolj funkcionalen inteligentni pomočnik** na tem področju.
- Študenti 1. letnika magistrskega študija smeri Informatika in podatkovne tehnologije.
- Naključna razdelitev v dve skupini
- Vsaka skupina **je enkrat uporabila** GitHub Copilot, drugič pa ne.
- Izvedba dveh 30-minutnih iteracij.
- **Iteracija 1:** dopolnitev spletnega javanskega projekta, ki temelji na ogrodju Spring Boot.
- **Iteracija 2:** dopolnitev javanske konzolne aplikacije, ki se povezuje na spletno storitev.

Analiza oddanih nalog

Rezultati raziskave

Iteracija	GitHub Copilot	Vrstic kode (\bar{x})	Pomanjkljivosti (\bar{x})	Ocena (\bar{x})
1	DA	560,75	12,75	67,50 %
	NE	563,46	11,54	39,23 %
2	DA	249	46,75	85,83 %
	NE	217,46	38,00	40,77 %



Zaključek

Ugotovitve raziskave

- Na umetni inteligenci temelječi inteligentni pomočniki za dopolnjevanje izvorne kode **imajo velik potencial za izboljšanje produktivnosti razvijalcev.**
- **Ključna funkcionalnost** inteligentnih pomočnikov je **kontekstno generiranje predlogov** za nadaljevanje programa.
- Primerjali smo inteligentne pomočnike in potrdili, da je **GitHub Copilot zelo učinkovit pri izboljšanju produktivnosti razvijalcev.**
- **Inteligentni pomočniki** so še v **razvoju** in niso široko dostopni, zato je še **veliko prostora za izboljšave.**
- Potrebno bo veliko dela, da se inteligentni pomočniki sprejmejo kot zelo uporabni in se razvijejo v preprosta orodja z visokim nivojem pragmatične uporabniške izkušnje, dostopna tudi za manj izkušene razvijalce.

Zaključek

Hvala za pozornost!