

# Teorijske osnove

---



# Organizacija rada

Problem

Teorijske osnove

Rešenje

Evaluacija rešenja

Rezultati/Diskusija

Koncepti  
neophodni za  
razumevanje  
problema

Koncepti  
neophodni za  
razumevanje  
rešenja

- Detaljniji opis domena problema iz koga sledi skup zahteva koje rešenje treba da ispunи
  - Aplikativna rešenja – koji su ciljevi krajnjih korisnika?
  - Tehnička rešenja – koje tehničke probleme rešenje treba da reši?
  - Mašinsko učenje – kako prevesti problem u problem mašinskog učenja?
- Razmatranje mogućih (alternativnih) rešenja
- Argumentacija zbog čega ste se opredelili za rešenje koje predstavljate u radu
- Opis koncepata korišćenih u rešenju (tehnologije, modeli)

# Radovi orijentisani na softversko inženjerstvo

- Iz detaljnijeg opisa domena izdvojite skup zahteva
  - Šta su ciljevi interesnih grupa
  - Pokušajte da izrazite cilj u jednoj rečenici
  - Za svaku frazu u cilju definišite jedan ili više zahteva koji ispunjavaju taj deo cilja

# Primer - ITS za razvoj veština pisanja održivog koda

## Domen problema → skup zahteva koje rešenje treba da ispuni

Poglavlje II.A opisuje domen pisanja održivog koda. Kroz ovo poglavlje ilustrujemo kako ne postoji precizna definicija održivog koda, te se veština pisanja održivog koda mora učiti kroz primere. Poglavlje II.B analizira ciljeve interesnih grupa i na osnovu njih definiše zahteve koje rešenje treba da ispuni kako bi doprinelo ispunjenju tih ciljeva.

### A. Održivost koda

**NAPOMENA:** poglavlje nije napisano, ali je iznad istaknut njegov cilj. Okvirno, organizacija poglavlja bi bila: (1) definicija pojma održivosti koda, (2) konkretan primer problema održivosti koda (npr., *Long Method*). Primer bi pokazivao kako ne postoje precizne (merljive) heuristike koje jednoznačno definišu funkciju kao „dugačku“.

### B. Zahtevi koje mora da ispuni edukativni alat za razvoj veštine pisanja održivog koda

Zahtevi koji treba da ispuni edukativni alat za razvoj veštine pisanje održivog koda su izvedeni iz našeg iskustva kao instruktora i empirijskih studija koje su istraživale potrebe instruktora za analitikama učenja u učionici [2][5]. Tabela II prikazuje zahteve koje rešenje treba da ispuni i objašnjava kako oni doprinose ispunjenju ciljeva interesnih grupa koje će koristiti rešenje.

TABELA II ZAHTEVI ALATA ZA UČENJE VEŠTINE PISANJA ODRŽIVOG KODA

Cilj interesne grupe	Zahtev	Argument kako zahtev doprinosi cilju
Studenti softverskog inženjerstva žele efektivno da savladaju veštine pisanja održivog koda kako bi unapredili svoje karijere	<ol style="list-style-type: none"> <li>1. Alat treba da simulira situacije iz realnog sveta koje zahtevaju upotrebu veštine pisanja održivog koda</li> <li>2. Alat treba da automatski identifikuje nedostatke u znanju svakog studenta</li> <li>3. Alat treba da se automatski prilagodi nedostacima u znanju studenta odabirom adekvatnih zadataka i povratnih informacija</li> </ol>	<p>Realistično okruženje povećava angažovanost u učenju i rezultate učenja, pomažući učenicima da razviju veštine potrebne za dato realistično okruženje [1]. Na primer, pitanje sa višestrukim izborom je manje realistično od programskog zadatka koji se rešava u integriranom razvojnom okruženju</p> <p>Rešenje treba da automatski identifikuje nedostatke u znanju svakog studenta kako bi se omogućilo automatsko prilagođavanje instrukcije</p> <p>Rešenje treba da se adaptira nedostacima u znanju svakog studenta kako bi se optimizovali dobici u učenju i time efikasno razvile veštine učenika.</p>
Instruktori teže donošenju odluka zasnovanih na podacima kako bi poboljšali iskustvo učenja za sebe i studente	<ol style="list-style-type: none"> <li>4. Alat treba da nadgleda i prikazuje progres svakog studenta</li> <li>5. Alat treba da prati interakcije studenata sa edukativnim sadržajem da bi podržao analizu kvaliteta sadržaja</li> </ol>	<p>Pomoći pregleda aktivnosti svakog učenika, instruktor može da usmeri svoju pažnju na studente kojima treba pomoći, dok alat obučava ostale studente.</p> <p>Pomoći ove funkcionalnosti, instruktori mogu da analiziraju studentske odgovore na zadatke, vreme provedeno u svakom segmentu ili šablone u učenju. Na taj način instruktori razumeju kojim delovima procesa učenja i edukativnih materijala je potrebno unapređenje. Specifično za održivost koda, instruktor može da analizira predat kod da bi utvrdio kršenje principa, preporuka i heuristika i adekvatno razvio model veština pisanja održivog koda.</p>

# Primer - ITS za razvoj veština pisanja održivog koda

- Razmatranje mogućih (alternativnih) rešenja
  - Argumentacija zbog čega ste se opredelili za rešenje koje predstavljate u radu
  - Opis koncepata korišćenih u rešenju (tehnologije, modeli)
- C. Teorijske osnove rešenja koje ispunjava izvedene zahteve*
- Definicija ITS tehnologije
  - Kako ITS odgovara na zahteve
  - Postoji više pristupa dizajnu ITS... Mi smo odabrali KLI jer...
  - Opis KLI koncepata (koncepti korišćeni u rešenju)

# Radovi orijentisani na mašinsko učenje

- Dublja analiza problema radi:
  - Određivanja kakve podatke treba prikupiti
  - Kako da biznis cilj prevedemo u metriku performansi koju optimizujemo
  - Šta je kriterijum uspeha
  - Koje rešenje je odabранo i kako ono odgovara na zahteve

# Primer – detekcija čestica polena

## Šta nam je biznis cilj?

- Opis problema – kako utiče polen na zdravlje ljudi
- Potrebno nam je da znamo količinu polena da bismo...
- Zašto vredi automatizovati proces detekcije čestica polena

# Primer – detekcija čestica polena

Kakvi nam podaci trebaju, kako dolazimo do zlatnog standarda?

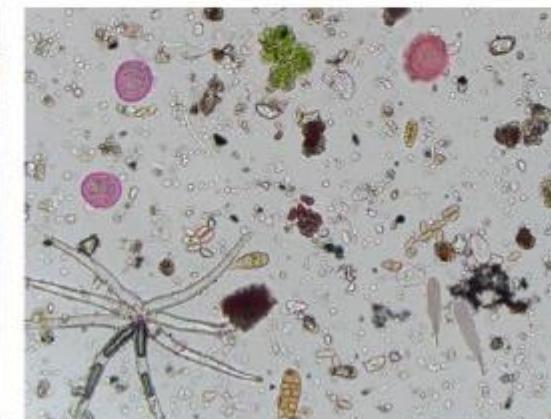
- Šta znači „detekcija čestica polena“ – koje čestice treba da detektujemo?
- Šta čestice od interesa razlikuje u odnosu na druge čestice (koja obeležja bismo merili)?
- Da li je bitna vremenska komponenta?
- Kako dolazimo do zlatnog standarda sa kojim poredimo model?
- Koliko je taj zlatni standard pouzdan?



a) Hirst device



b) identification methodology



# Primer – detekcija čestica polena

## Šta nam je cilj u smislu mašinskog učenja?

- Koje metrike performansi koristimo za poređenje? ( $F$ -mera, rad u realnom vremenu)
- Koje su dopustive vrednosti metrika (da bi rešenje bilo praktično primenljivo)?
- Da li rešenje treba da radi u uslovima šuma (kakav šum može da se javi)?
- Da li je važno da model bude interpretabilan?

## Primer – detekcija čestica polena

Koje rešenje je odabрано i kako ono odgovara na zahteve

- Da li koristiti duboko učenje ili statističke modele mašinskog učenja?
- Odabir arhitekture neuronske mreže
- Da li se može koristiti transfer znanja?

# Prema potrebi, revidirana su poglavlja „Problem“ i „Rešenje“

- Revidirajte poglavlja tako da ne sadrže nedefinisane koncepte
  - Čitalac može da razume tekst bez potrebe da istražuje druge izvore
- Poglavlje „Teorijske osnove“ ne definiše koncepte koji nisu relevantni za razumevanje rada
- Iz poglavlja "Rešenje" je jasno kako je svaki od zahteva sistema realizovan
- Ako neki zahtev niste ispunili (sasvim ili delimično), ipak navedite zahtev u „Teorijskim osnovama“. Kasnije ćete u „Rezultati/diskusija“ to istaći kao ograničenje svog rešenja