



PROGRAMSKO INŽENJERSTVO

Druge laboratorijske
vježbe

Analiza zahtjeva

Poslovni zahtjevi uključuju izjave na visokoj razini o ciljevima i potrebama vašeg projekta

Korisnički zahtjevi pomažu vam pronaći ono što očekujete od određenog rješenja

Zahtjevi sustava opisuju karakteristike proizvoda koje će zadovoljiti vaša očekivanja i poslovne potrebe

- *Funkcionalni zahtjevi* opisuju načine na koje se proizvod mora ponašati
- *Nefunkcionalni zahtjevi* (atributi kvalitete) opisuju opće softverske karakteristike

Zašto je ovo važno?

Fail Story: Cyberpunk 2077

Problem: Igra obećavala konzole prošle generacije, ali nisu analizirali nefunkcionalne zahtjeve (performanse)

Rezultat: Refund skandal, pad dionice za 30%, izgubljeni milijuni

Lekcija: Nefunkcionalni zahtjevi nisu "nice to have" - oni su kritični!

1 sat planiranja = 10 sati manje debugginga

Loši zahtjevi

- Tehnički dug
- Refactoring nakon 6 mjeseci
- Nezadovoljni korisnici
- Burnout tima

Dobri zahtjevi

- Clean implementacija
- Smooth maintenance
- Happy korisnici
- Sretni developeri



Vrste zahtjeva



Poslovni zahtjevi

Visoka razina - što želimo postići?

 **REAL WORLD**

Netflix primjer: "Povećati retention rate korisnika za 20%"



Korisnički zahtjevi

Što korisnici očekuju od rješenja?

 **REAL WORLD**

Netflix primjer: "Želim automatsko preuzimanje sljedeće epizode"



Zahtjevi sustava

Karakteristike koje zadovoljavaju očekivanja

- ▶ **Funkcionalni:** Kako se sustav ponaša
- ▶ **Nefunkcionalni:** Atributi kvalitete (brzina, sigurnost...)



Funkcionalni zahtjevi

Instagram Stories značajka

Poslovni zahtjev: Povećati engagement korisnika

Korisnički zahtjev: Dijeliti sadržaj koji nestaje nakon 24h

Funkcionalni zahtjevi:

- ▶ Upload foto/video (max 15 sek)
- ▶ Dodavanje filtera i stickera
- ▶ Opcija dijeljenja na profil
- ▶ Swipe up za linkove (verified accounti)
- ▶ View count i lista gledalaca
- ▶ Automatsko brisanje nakon 24h

⚡ Nefunkcionalni zahtjevi

Funkcionalni: "Što sustav radi" → **Nefunkcionalni:** "Kako dobro to radi"

TikTok Video Upload

- ▶ **Performanse:** Video processing < 5 sekundi
- ▶ **Skalabilnost:** Podržava 100M+ dnevnih uploada
- ▶ **Pouzdanost:** 99.9% uptime (max 8.76h downtime godišnje)
- ▶ **Upotrebljivost:** Intuitivni UI - average user uploaduje prvi video u < 2min
- ▶ **Sigurnost:** Content moderation AI - flagiranje neprihvatljivog sadržaja u < 30s

🔧 Alati za mjerenje:

Lighthouse

WebPageTest

Sentry

DataDog

K6 Load Testing



Real World: Uber Ride-Sharing

Poslovni zahtjev:

Povezati vozače i putnike za sigurnu, brzu i pouzdanu vožnju

Funkcionalni zahtjevi:

- ▶ Real-time GPS tracking vozača
- ▶ Estimacija cijene prije narudžbe
- ▶ Multiple payment metode
- ▶ Rating sustav (vozači ↔ putnici)
- ▶ Push notifikacije (vozač pronađen, vozač stigao...)
- ▶ Trip history s računima



Real World: Uber Ride-Sharing

Nefunkcionalni zahtjevi:

- ▶ ⚡ **Performanse:** Pronalaženje vozača < 30 sekundi
- ▶ 📍 **Točnost:** ETA accuracy ± 2 minute
- ▶ 🔒 **Sigurnost:** Enkripcija svih transakcija
- ▶ 📈 **Skalabilnost:** 1M+ simultanih korisnika po gradu
- ▶ 🔋 **Efikasnost:** Optimizacija baterije (background tracking)
- ▶ 🌐 **Dostupnost:** Offline mode za osnovne funkcije

🔍 REAL WORLD

Što kad preskočiš NFZ? Uber je 2016. imao problem s baterijom - app je drenirao bateriju vozača. Rezultat: Razočarani vozači, loši reviewi, pad upotrebe.

Spot the Missing Requirement!

Scenario: Food Delivery App (Glovo/Wolt style)

Evo liste zahtjeva. Koja kritična kategorija nedostaje?

Definirani zahtjevi:

- ▶ ☒ Browse restorani po lokaciji
- ▶ ☒ Add items to cart
- ▶ ☒ Multiple payment options
- ▶ ☒ Real-time order tracking
- ▶ ☒ Rating system
- ▶ ☒ Push notifications

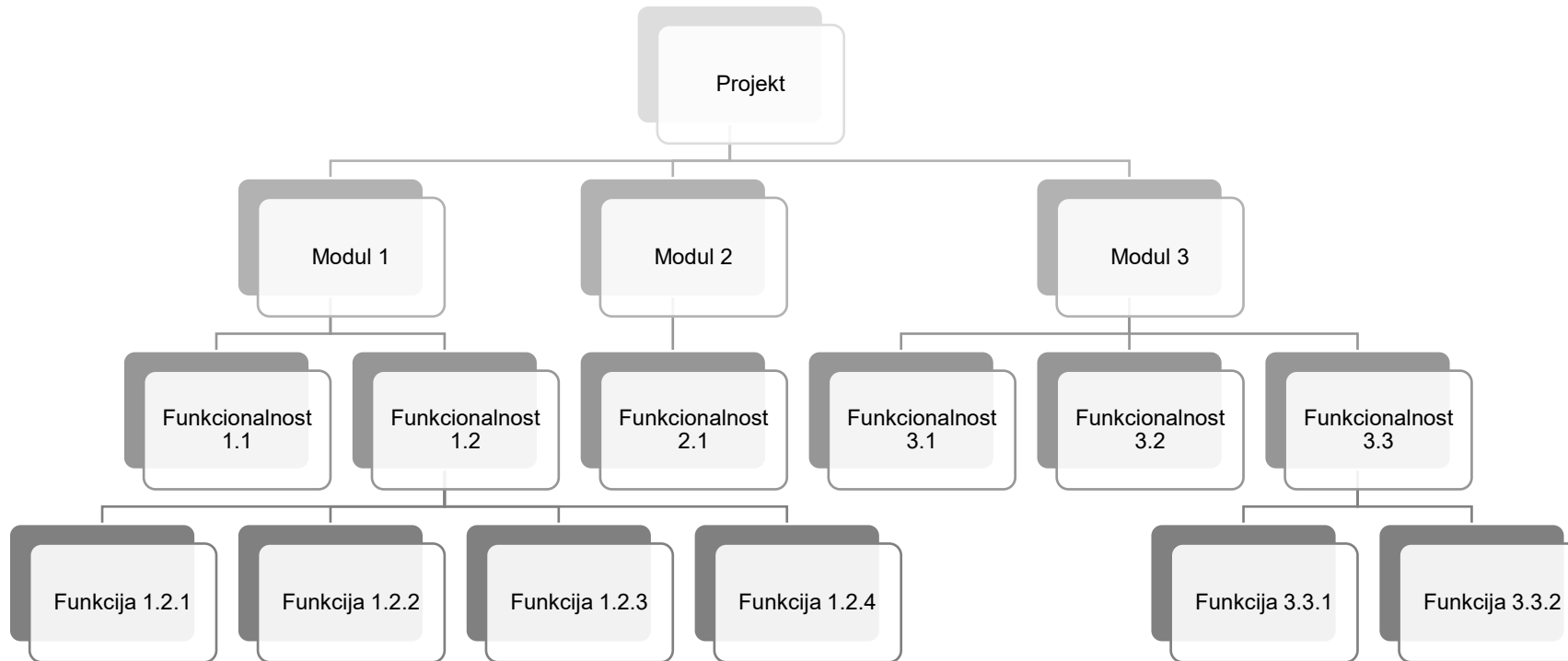
? Što nedostaje?

A) Accessibility features (screen reader support)

B) Offline mode za basic funkcionalnosti ☒

C) Multi-language support

Funkcionalna dekompozicija





Razbijanje kompleksnog sustava na manje, upravljive dijelove

Spotify - Projekt struktura

PROJEKT: Spotify

Modul 1: Music Streaming

- Funkcionalnost 1.1: Playback Control
 - Play/Pause/Skip
 - Volume Control
 - Repeat/Shuffle
- Funkcionalnost 1.2: Queue Management
 - Add to Queue
 - Reorder Queue
 - Clear Queue

Modul 2: Discovery & Recommendations

- Funkcionalnost 2.1: Discover Weekly
- Funkcionalnost 2.2: Daily Mix
- Funkcionalnost 2.3: Search & Browse

Modul 3: Social Features

- Funkcionalnost 3.1: Follow Friends
- Funkcionalnost 3.2: Collaborative Playlists
- Funkcionalnost 3.3: Share Songs

Use Cases (Slučajevi korištenja)

Skup interakcija između sustava i korisnika za postizanje određenog cilja



Aktor

Korisnik sustava ili grupa korisnika


 REAL WORLD

Primjer: Customer, Admin, Payment Gateway (eksterni sustav)



Cilj

Finalni uspješan završetak procesa


 REAL WORLD

Primjer: "Successfully order food and receive it"



Sustav (Scenarij)

Proces i koraci potrebni da bi postigli cilj

 REAL WORLD

Main Flow: Happy path - sve ide po planu

Alternative Flows: Što ako... (payment fails, item out of stock...)

Use Case: Join Voice Channel

Aktor: Discord User

Cilj: Započeti voice chat sa prijateljima

Preduvjeti: User je logiran, ima microphone permissions

Main Flow (Happy Path):

1. User klikne na voice channel
2. Sustav provjerava permissions
3. Sustav ustanovljava WebRTC konekciju
4. Sustav prikazuje usera u voice channel listi
5. Audio stream započinje

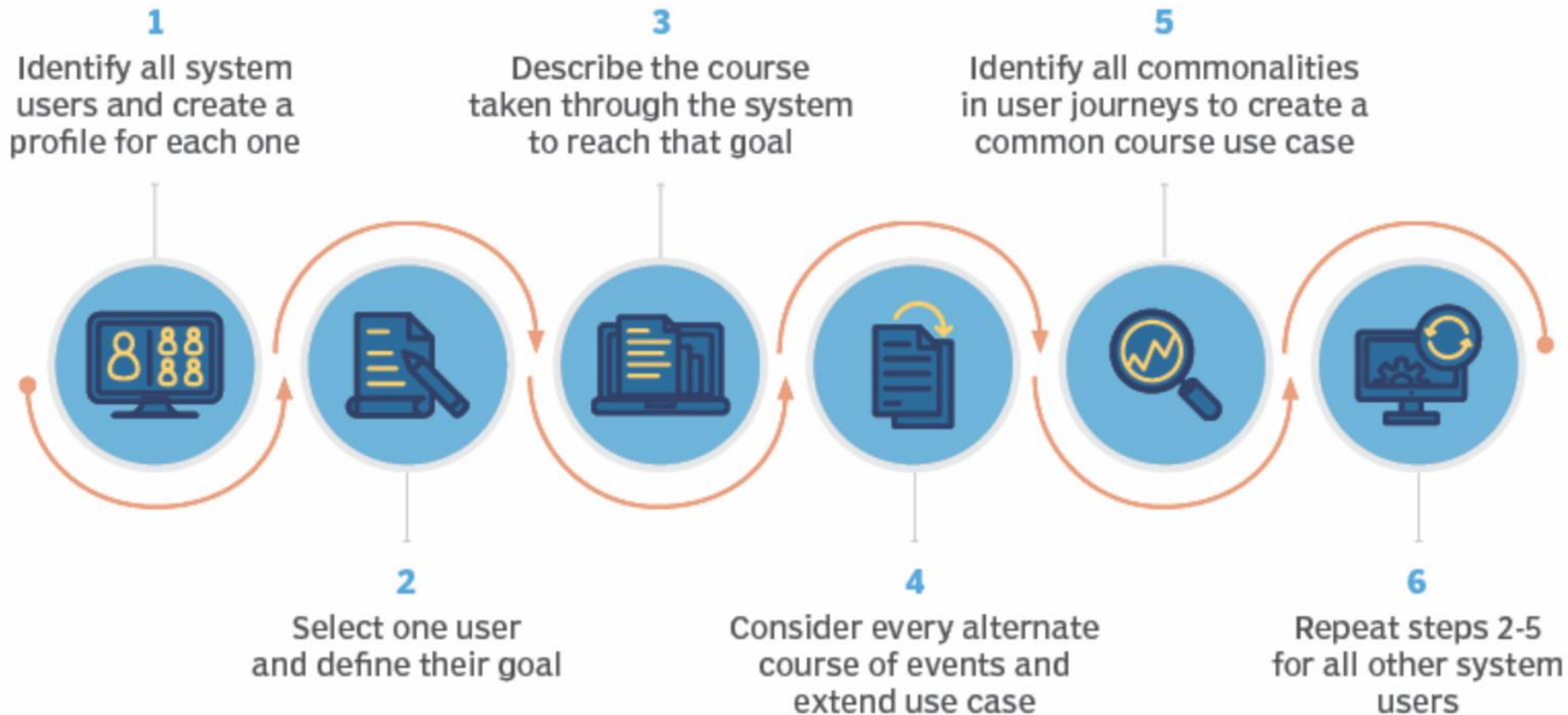
Alternative Flow 1: No Microphone Permission

1. Na koraku 2, sustav detektuje missing permission
2. Sustav prikazuje permission request dialog
3. User odobrava/odbija
4. Ako odobri → nastavak main flow
5. Ako odbije → user joinuje bez audio (listen-only mode)

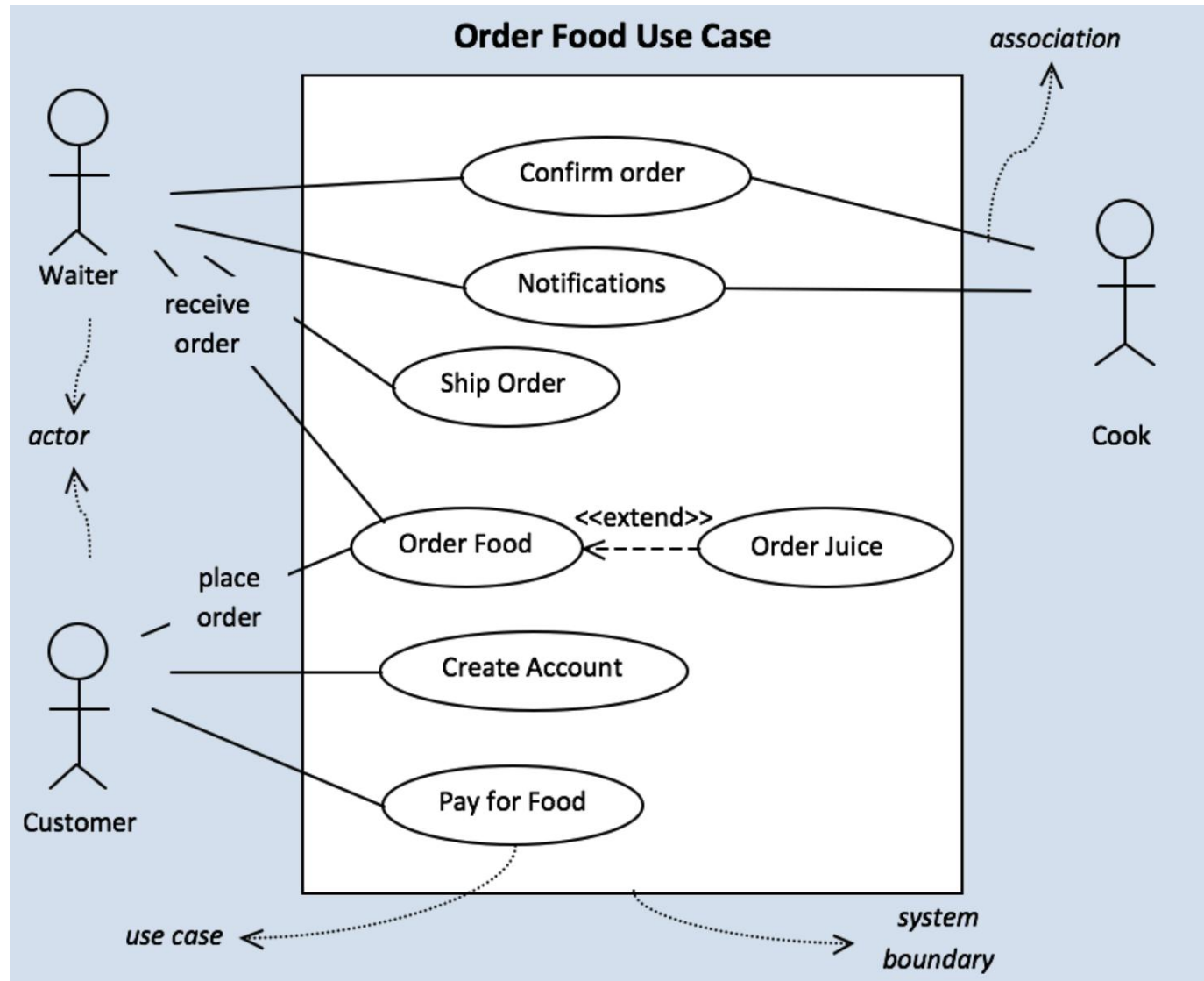
Alternative Flow 2: Voice Channel Full

1. Na koraku 2, sustav detektuje channel je pun (user limit reached)
2. Sustav prikazuje error message
3. Use case ends

Slučajevi korištenja



UML diagram – UseCase



Za kreiranje dijagrama:

Draw.io (diagrams.net)

- ✓ Free, browser-based
- ✓ UML templates
- ✓ Izvoz u PNG/SVG/PDF

→ app.diagrams.net

Miro

- ✓ Collaborative whiteboard
- ✓ Real-time suradnja
- ✓ Templates za sve

→ miro.com

Za dokumentaciju zahtjeva:

Notion

All-in-one workspace

Confluence

Enterprise wiki

GitHub Wiki

Markdown docs

Testing & Monitoring:

Lighthouse

WebPageTest

Sentry

Datadog

Postman

K6

JMeter

REAL WORLD

 **Pro tip:** Većina firmi koristi kombinaciju ovih alata. Naučite barem Draw.io i Notion - bit će vam u CVu!



Projektni zadatak

Vaš zadatak:

Kompletna analiza zahtjeva za vaš projektni sustav

1 Analiza zahtjeva

- ▶ Definirajte funkcionalne zahtjeve sustava (min 10)
- ▶ Funkcionalna dekompozicija prvog nivoa (moduli)
- ▶ Dekompozicija svakog modula do razine 3
- ▶ Min 1 nefunkcionalni zahtjev po modulu



2 Use Cases

- ▶ Use case dijagram po modulu
- ▶ Detaljni opis min 3 use case-a (main + alternative flows)
- ▶ Jasno definirani akteri i ciljevi

3 Format predaje

- ▶ Excel/Google Sheets za funkcionalnu dekompoziciju
- ▶ Draw.io dijagrami (izvoz u PNG + .drawio fajl)
- ▶ PDF dokument sa svim elementima



Recap & Pro Tips

✓ Do's

- ▶ Pričaj s korisnicima prije kodiranja
- ▶ Dokumentiraj alternative flows
- ▶ Definiraj success metrics
- ▶ Think about edge cases
- ▶ Uključi stakeholders rano

✗ Don'ts

- ▶ "Idemo prvo kodirat, pa vidimo"
- ▶ Ignoriraj nefunkcionalne zahtjeve
- ▶ "Svi znaju što treba napraviti"
- ▶ Pretpostavi umjesto da pitaš
- ▶ Zaboravi na dokumentaciju

Recap & Pro Tips



"Nije bug, to je undocumented feature!"

- Svaki developer koji nije napravio analizu zahtjeva

Final Words

Dobra analiza zahtjeva nije overhead - to je investicija.

Svaki sat potrošen ovdje = 10 sati manje debugginga kasnije.

Make it count! 

Hvala na pažnji!

