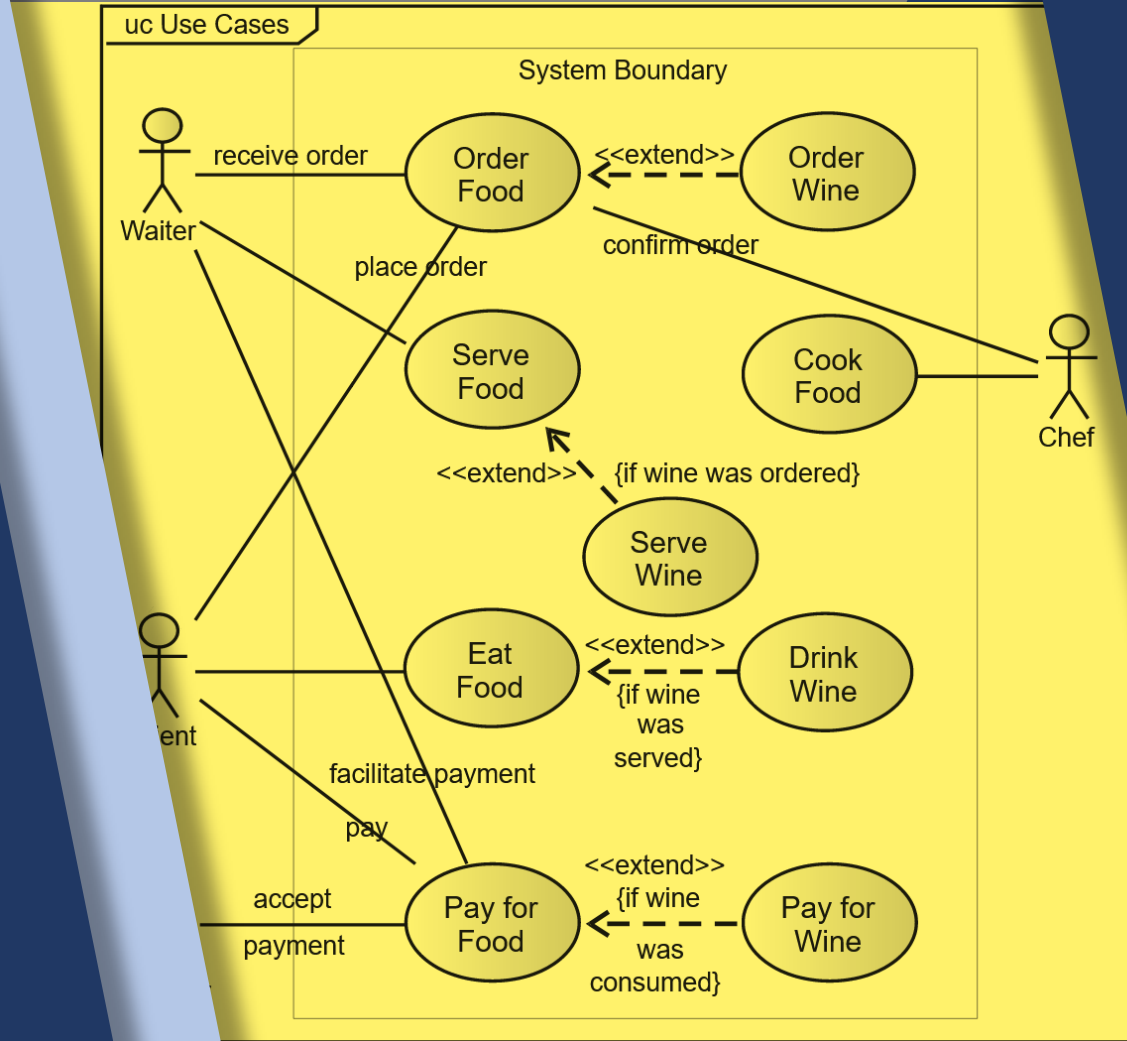


# Specifikacija i modeliranje softvera

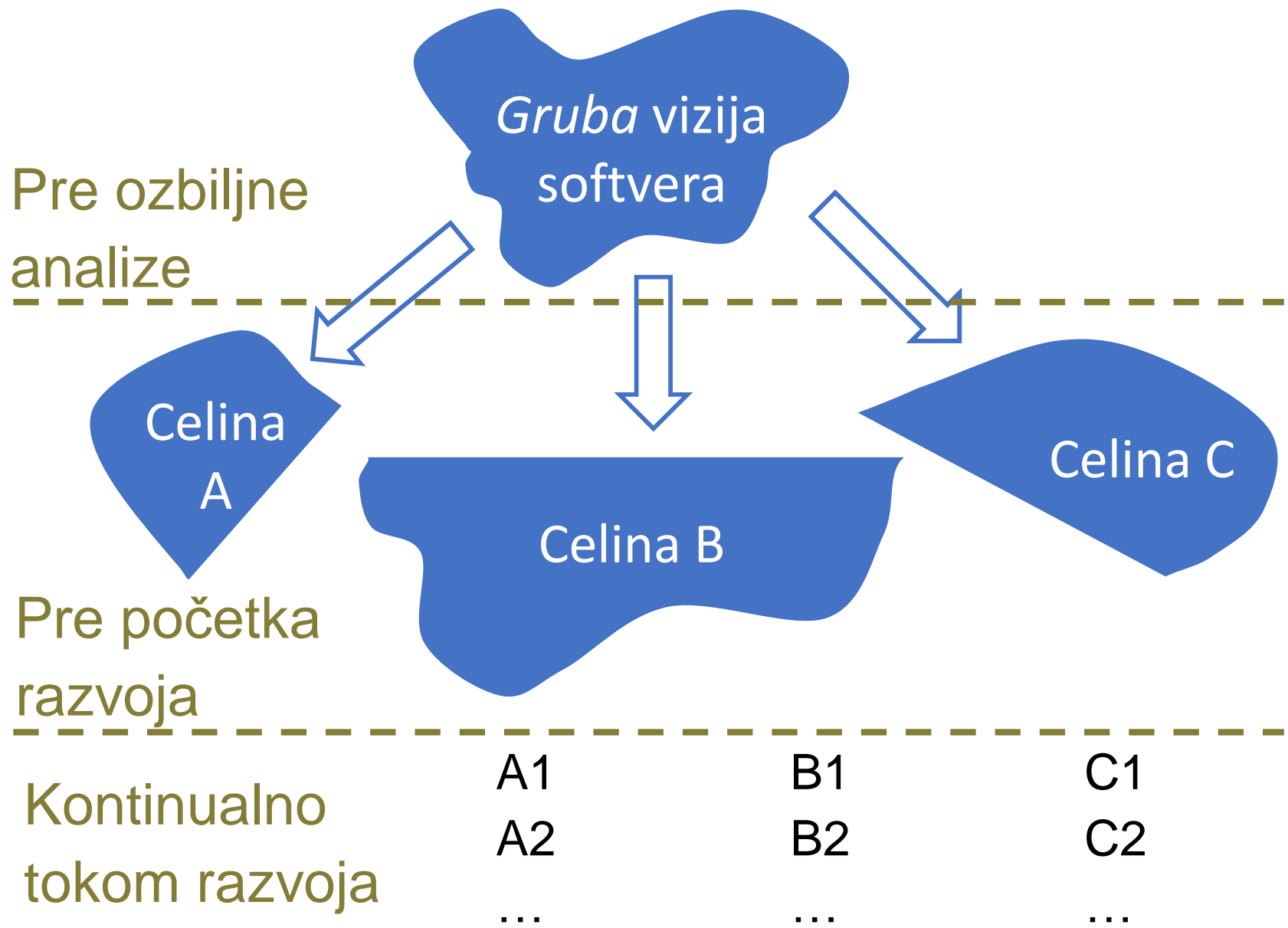
Tehnike razrade  
softverskih zahteva I

Nikola Luburić

nikola.luburic@uns.ac.rs

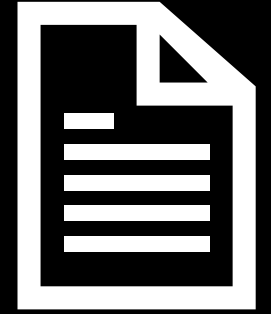


# Kada skupljati zahteve?

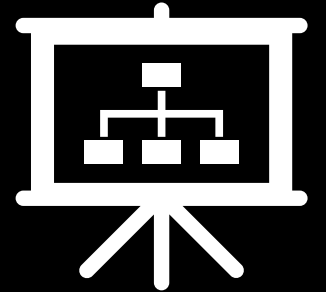


# Tehnike razrade softverskih zahteva I

Kako se pišu  
korisničke priče?



Kako se koriste  
prototipovi?



# Tehnike razrade softverskih zahteva I



Korisnička priča?

*wut?*

**As a <role>  
I want <goal>  
So that <benefit>**

**Acceptance criteria:**

# Tehnike razrade softverskih zahteva I



Korisnička priča?

## Korisnička priča (*User story*)

- ❖ Kratak, jednostavan opis potrebe korisnika
- ❖ Alat za komunikaciju klijenta i razvojnog tima

**As a <role>  
I want <goal>  
So that <benefit>**

Acceptance criteria:

„Kao prijavljeni korisnik,  
želim da delim slike sa svojim pratiocima,  
kako bi bili u toku sa mojim aktivnostima“

Acceptance criteria

- 1) Podržati *drag & drop* unos slike
- 2) Opciono definisanje naslova i tagova za sliku

# Tehnike razrade softverskih zahteva I



Korisnička priča?

## Struktura priče

- 1) Odredi korisnike
- 2) Definiši šta žele
- 3) Opiši njihovu korist
- 4) Dodaj bitne uslove ispunjenosti zahteva

As a <role>  
I want <goal>  
So that <benefit>

Acceptance criteria:

„Kao **prijavljeni korisnik**,  
želim da **delim slike sa svojim pratiocima**,  
kako bi **bili u toku sa mojim aktivnostima**“

### Acceptance criteria

- 1) Podržati *drag & drop* unos slike
- 2) Opciono definisanje naslova i tagova za sliku



Korisnička priča?

## Struktura priče

**As a** <role>  
**I want** <goal>  
**So that** <benefit>

Acceptance criteria:

Rola ističe grupu korisnika

- ❖ Dizajn funkcija i interfejsa

Cilj definiše posao

- ❖ Tražena svojstva softvera
- ❖ Razlaže se na zadatke (*Task*)

Korist definiše značaj

- ❖ Dogovaranje prioriteta
- ❖ Šira slika

# Tehnike razrade softverskih zahteva I



Korisnička priča?

## Struktura priče

- ❖ Može da ima slobodniju formu

Izveštaj zauzeća  
operacione sale

Ko je zauzimao  
operacionu salu?

*koja je  
razlika?*



# Tehnike razrade softverskih zahteva I



Korisnička priča?

*ko je ovo?*

Kao korisnik,

Hoću da se prijavim na sistem putem naloga na društvenim mrežama,

Kako bih brzo mogao da pristupim funkcijama sistema

*kako se sve  
ovaj zahtev  
može ispuniti?*

*šta je  
najbolje?*

# Tehnike razrade softverskih zahteva I



Korisnička priča?

Ko je zauzimao operacionu salu?

Kao sekretar,  
Hoću da menjam postojeći raspored,  
Kako bih mogao da uslužim hitne slučajeve

**Z3.1:** Odredi bar dve prednosti i dve mane definisanja zahteva u ovako prosto formi  
3 minuta, trojke

# Tehnike razrade softverskih zahteva I



Korisnička priča?

*ko ovo definiše?*

*šta kad ne može?*

## Kvalitet priče

*treniraj*

Artikuliše funkcionalnost „običnim jezikom“  
bez tehničkih detalja => jasno svim učesnicima

**I**ndependent – sloboda u redosledu realizacije

**N**egotiable – nije ugovor, već tema za razgovor

**V**aluable – razlog što uopšte radimo

**E**stimable – neophodna procena za planiranje

**S**mall – dovoljno da nije problem ako ih bacimo

**T**estable – provera ispunjenosti priče

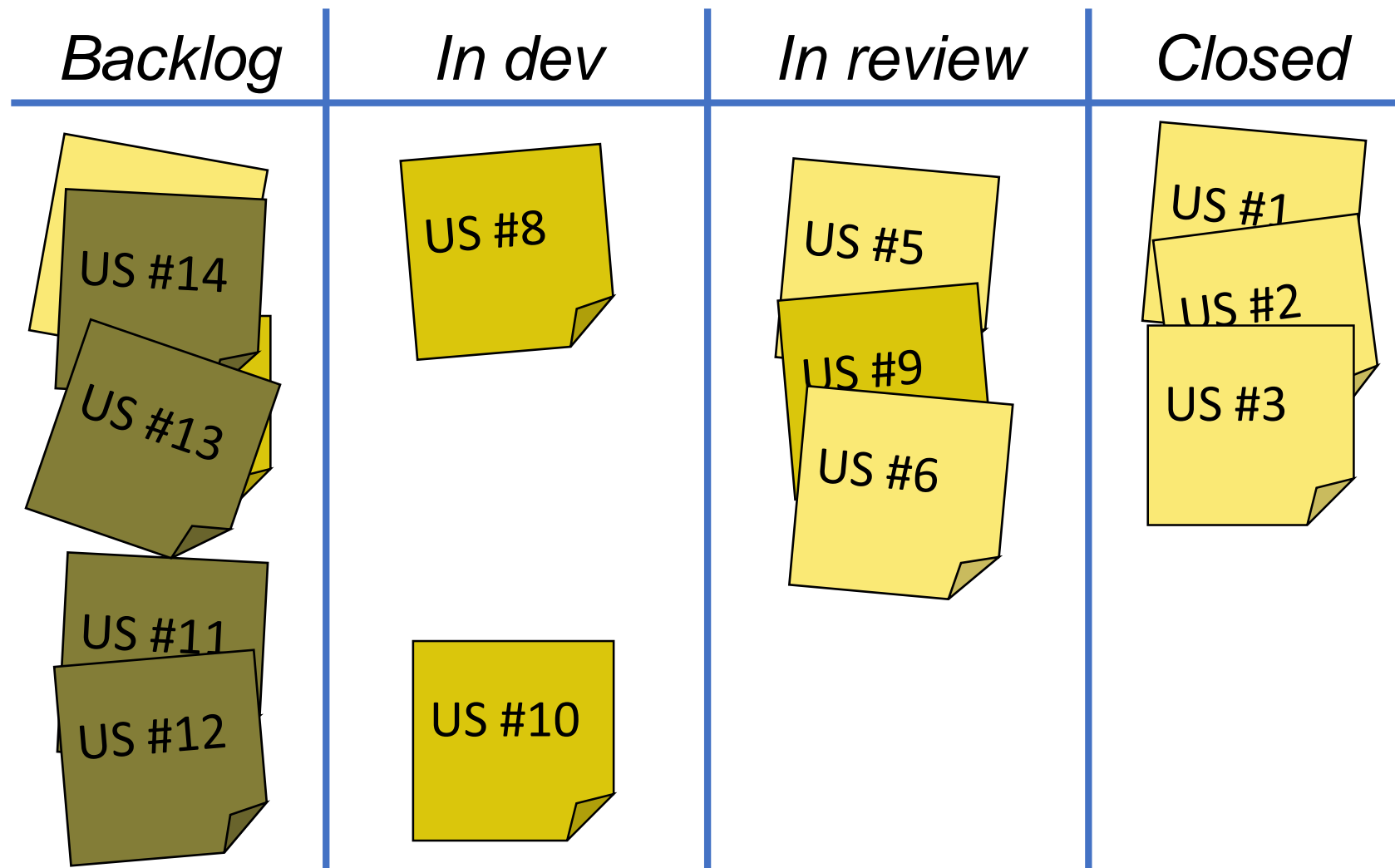
# Tehnike razrade softverskih zahteva I



Korisnička priča?

## Životni cikelus priče

Stanja korisničkih priča



# Tehnike razrade softverskih zahteva I



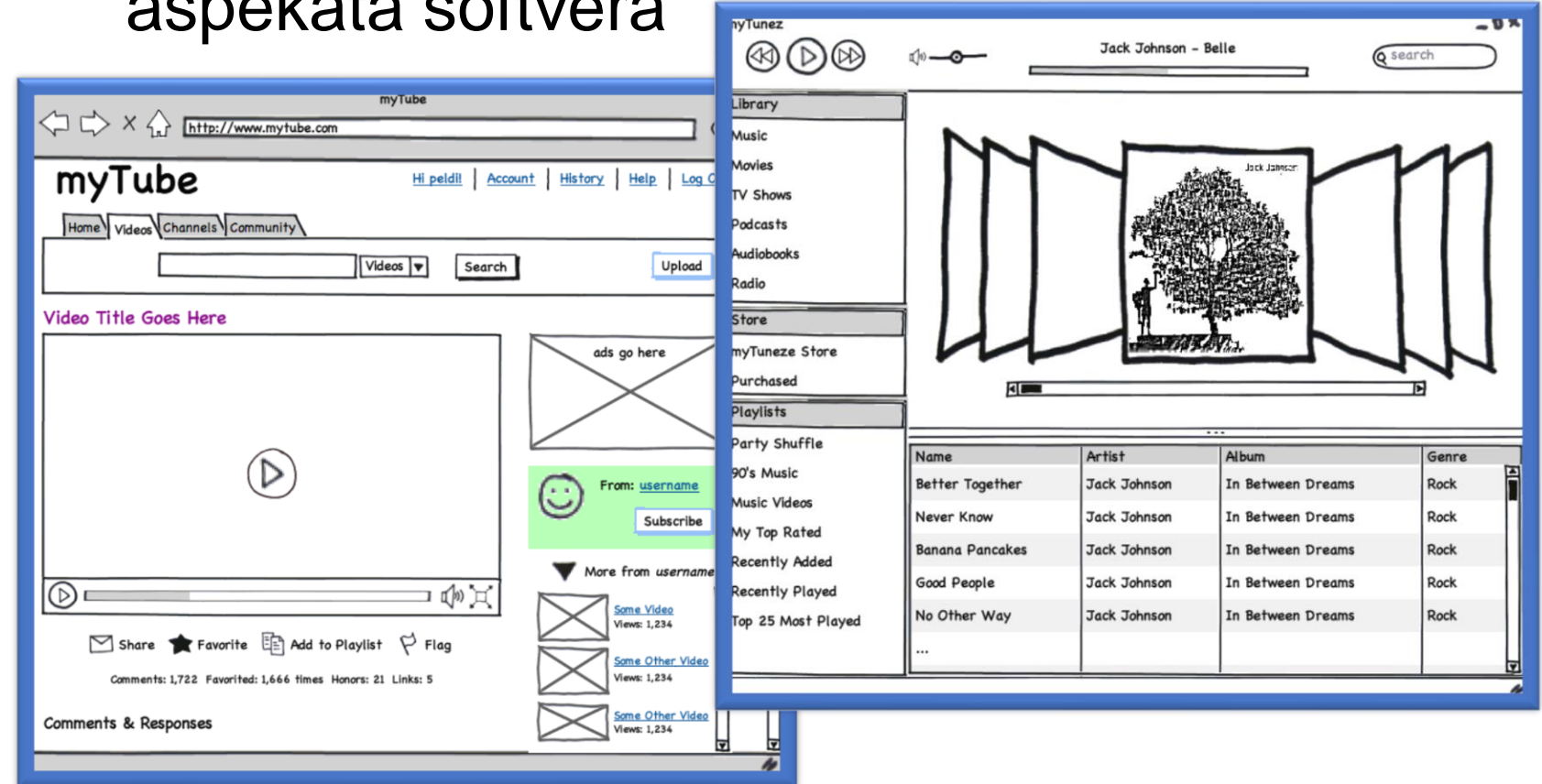
Korisnička priča?



Prototip?

## Prototipovi

- ❖ Alat za sporazumevanje korisnika i razvojnog tima
- ❖ Alat za procenu kompleksnosti aspekata softvera



# Tehnike razrade softverskih zahteva I



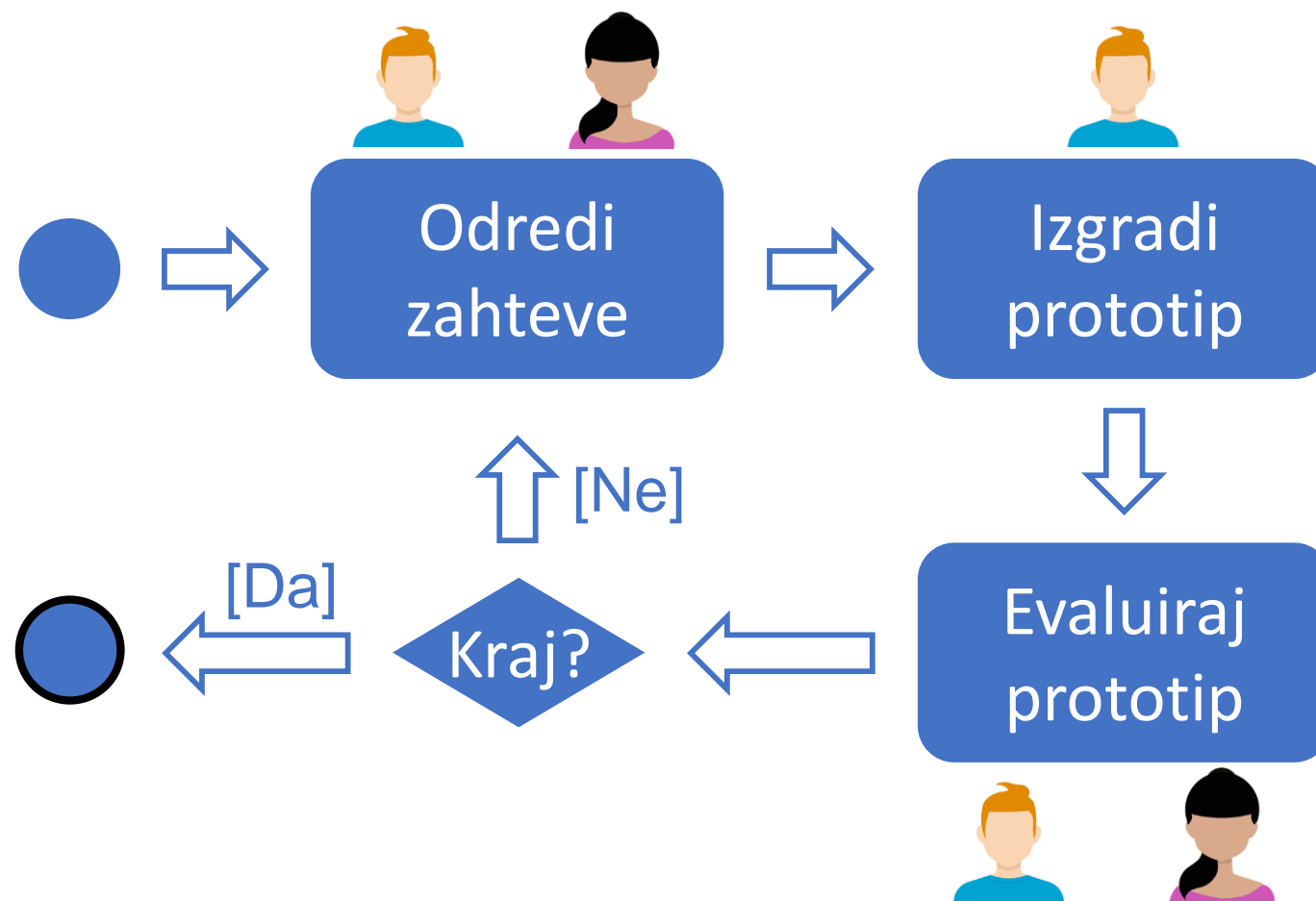
Korisnička priča?



Prototip?

## Pravljenje UI prototipa

- ❖ Programer  saraduje sa korisnikom 
- ❖ Tokom ili nakon prikupljanja zahteva



# Tehnike razrade softverskih zahteva I



Korisnička priča?



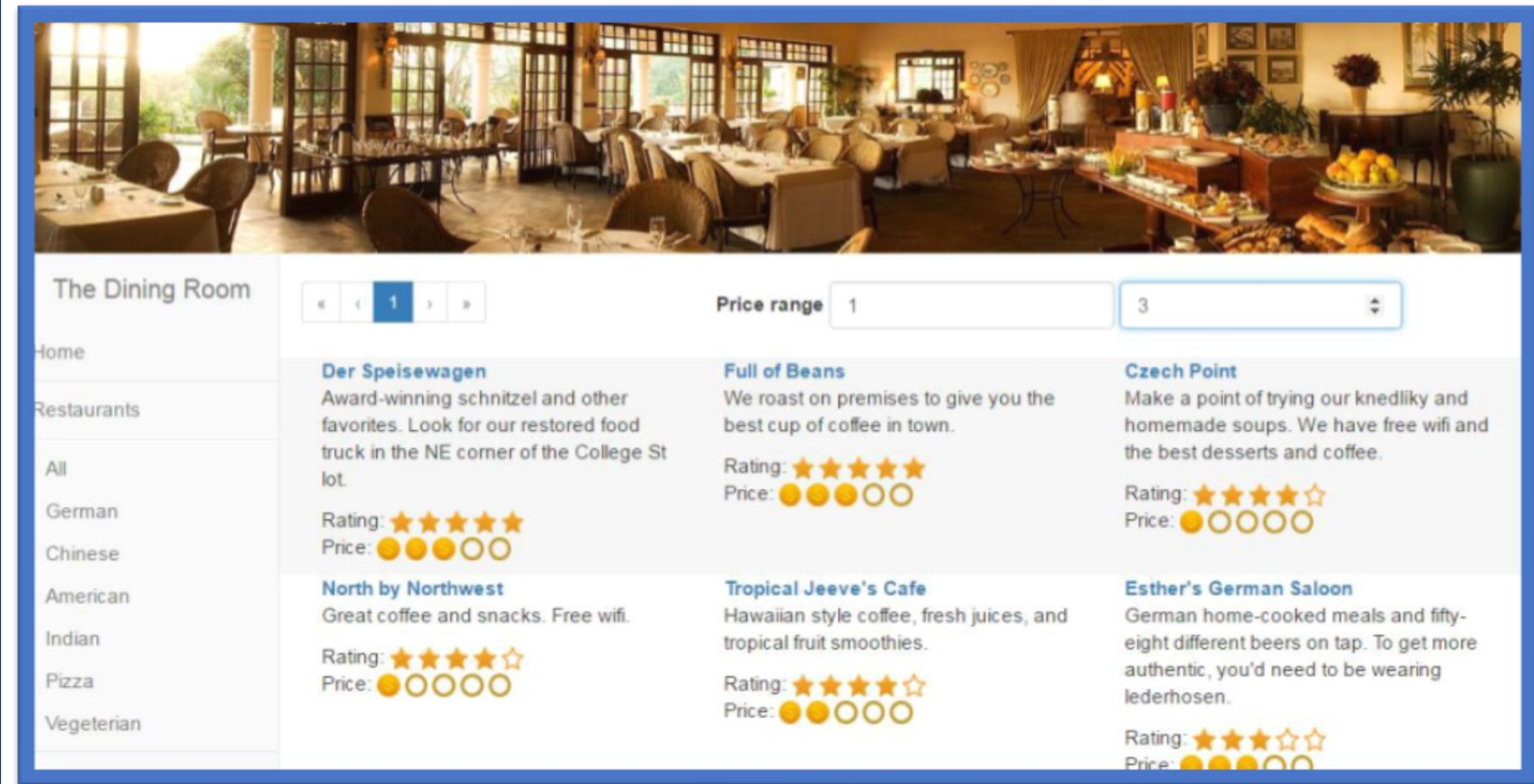
Prototip?

## Vrste UI prototipa

### Evolutivni

- ❖ Konstantan razvoj, evoluirá u konačno rešenje
- ❖ Komunikacija nad „nečim što radi“

*prednosti i mane?*





# Tehnike razrade softverskih zahteva I



Korisnička priča?



Prototip?

## Vrste UI prototipa

### Odbaciv (*throw-away*)

- ❖ Kratkog veka, odbacuje se
- ❖ Validacija zahteva i provera pretpostavki

*prednosti i mane?*

**Student Information** Help

Student Number: 789-567-234

FirstName:

Middle:

Surname:

Salutation:  ☐

Date first Enrolled: June 14 2003

Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to CS	Fall 2003	A+	Passed
CSC 200 Intro to AI	Fall 2003	A	Passed
CSC 203 Advanced AI	Spring 2004	-	Enrolled

**Add a Seminar** Help

Seminar Number:

Name:

Results

Seminar	Term	Seats Avail	Professor
CSC 250 Agile Techniques	Fall 2004	4	Smithy J.
CSC 300 Agile EUP	Spring 2003	17	Jones S.
CSC 310 Agile Database techniques	Spring 2004	0	Johnson H.

Course description:

CSC 310 Agile Database Techniques

This Course describes evolutionary development strategies for data oriented development. See [www.agiledata.org](http://www.agiledata.org) for details.

This Course currently has 39 people waitlisted for it.



# Tehnike razrade softverskih zahteva I

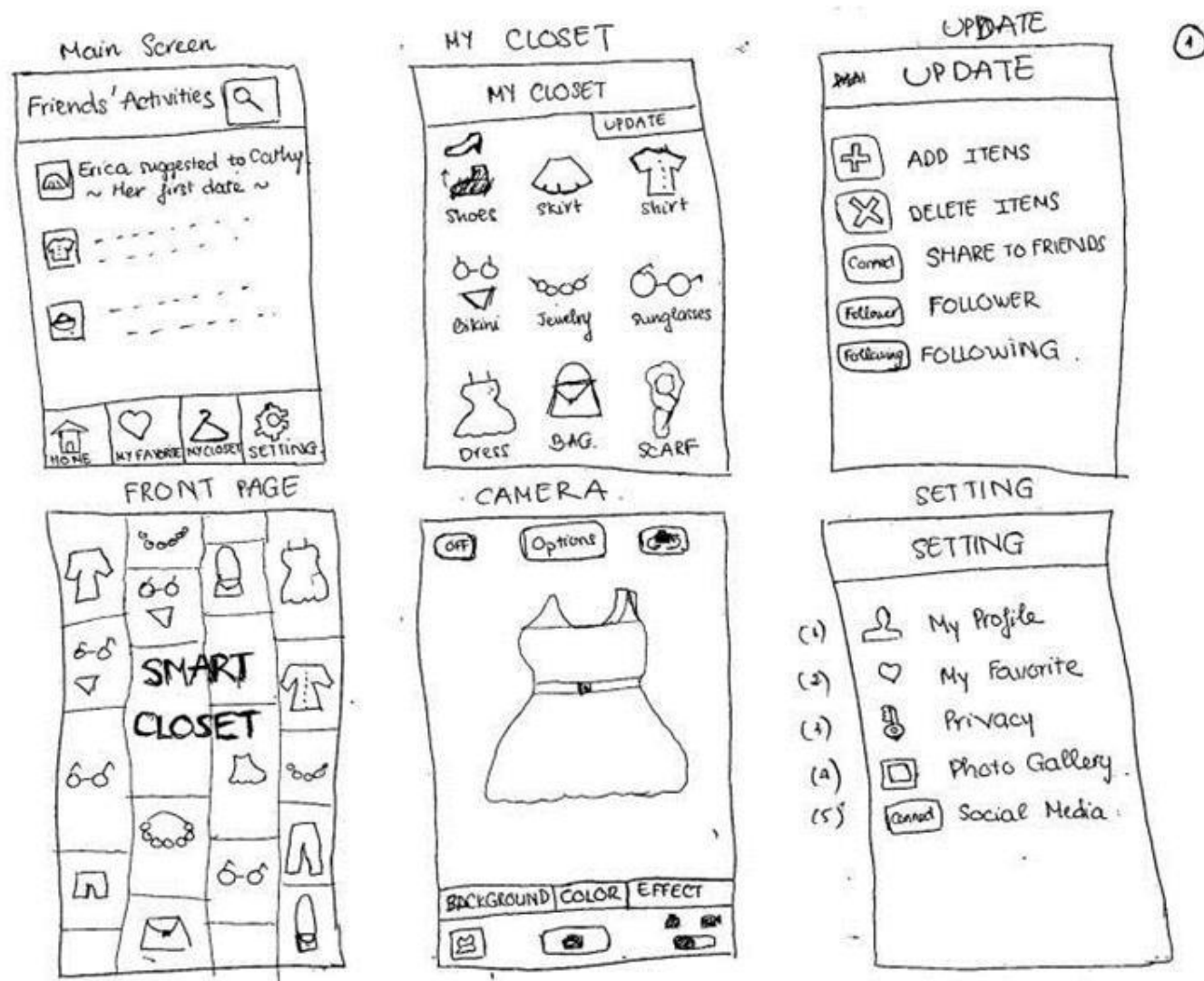


Korisnička priča?



Prototip?

## Primeri UI prototipa Ilustracija ekrana – papir i olovka



# Tehnike razrade softverskih zahteva I



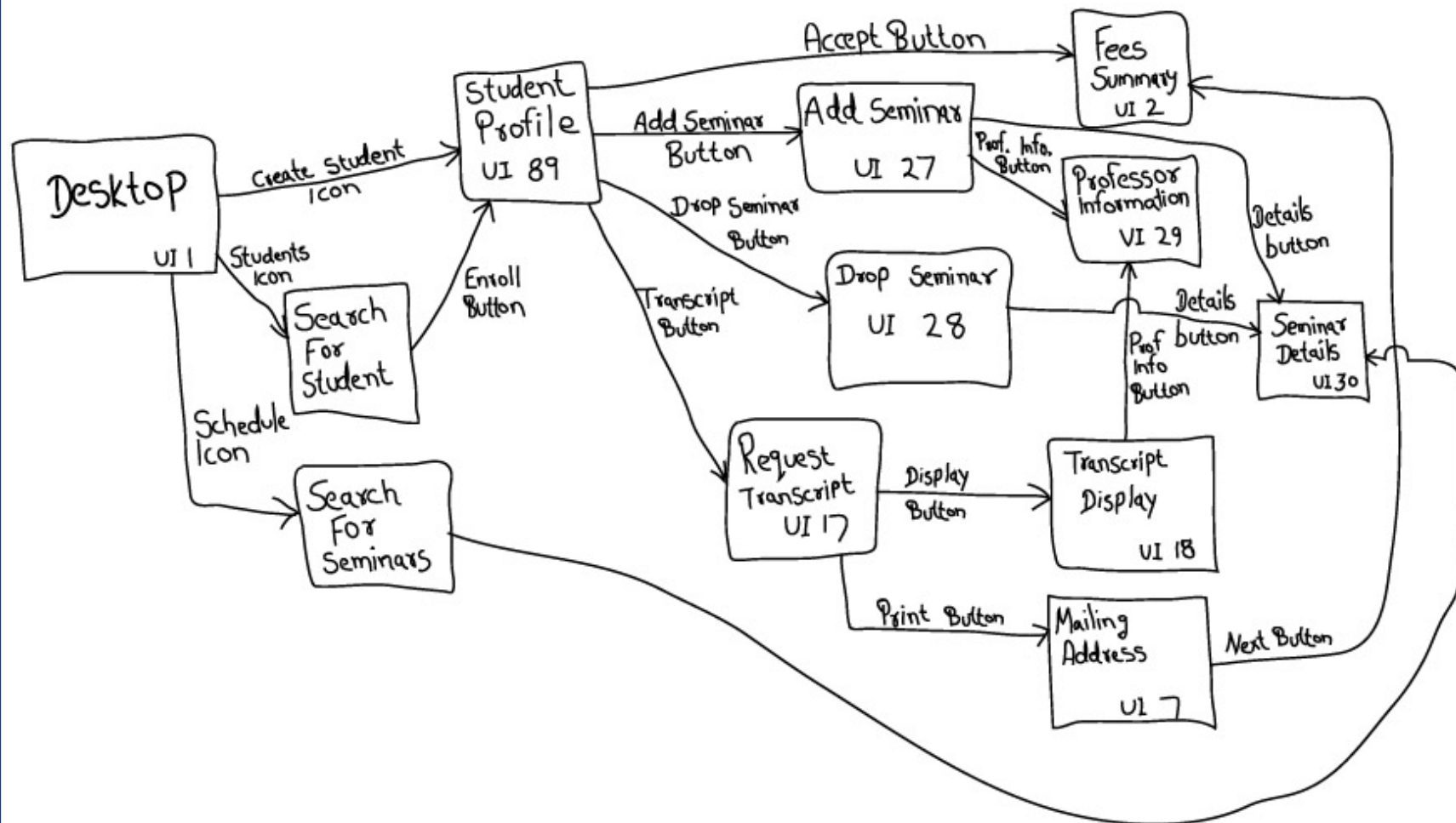
Korisnička priča?



Prototip?

## Primeri UI prototipa

Navigacija kroz aplikaciju – marker i tabla



# Tehnike razrade softverskih zahteva I



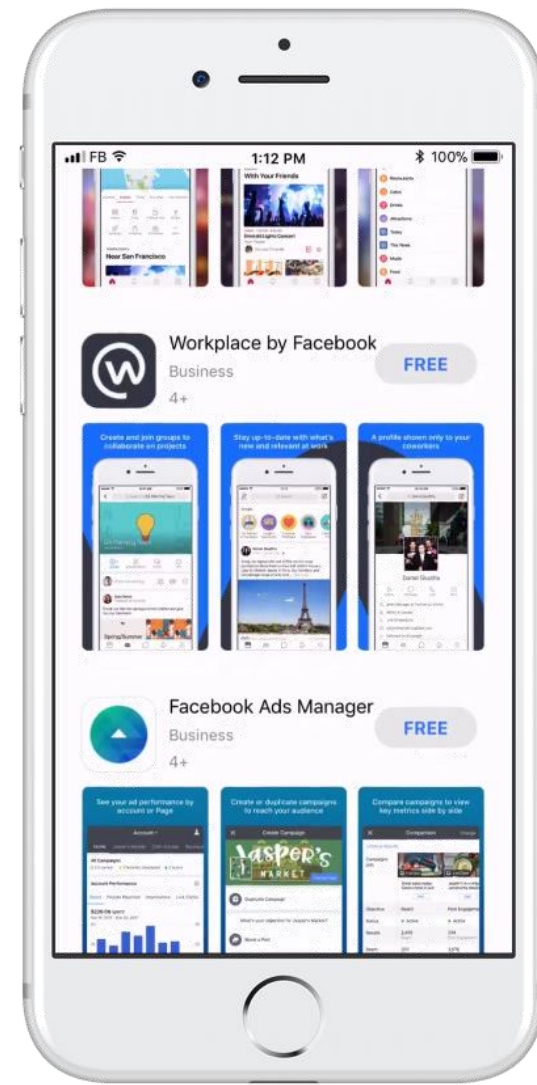
Korisnička priča?



Prototip?

## Primeri UI prototipa

*High-fidelity* prototip - [origami.design](https://origami.design)



# Tehnike razrade softverskih zahteva I



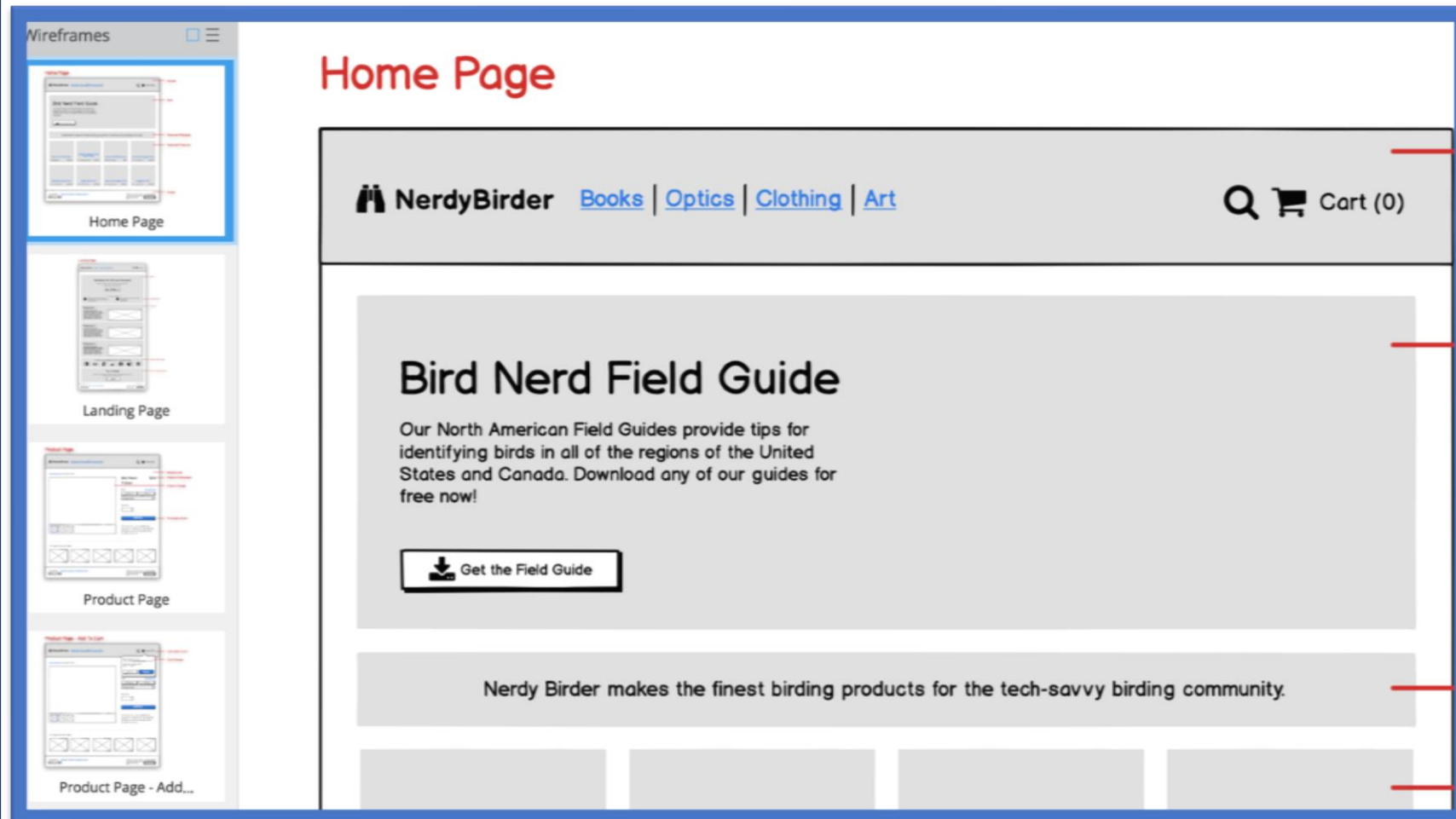
Korisnička priča?



Prototip?

## Primeri UI prototipa

*Wireframes* - [balsamiq.com](https://balsamiq.com), [framebox.org](https://framebox.org)



*šta je ovo konkretno?*

Physician data

Equipment data

*zašto posebno?*

Local data

Hospital Service

Room data

*kakve dileme prezentuje?*

*radi previše?*

Secretary UI

Timetable Service

Patient UI

Manager UI

Patient Service

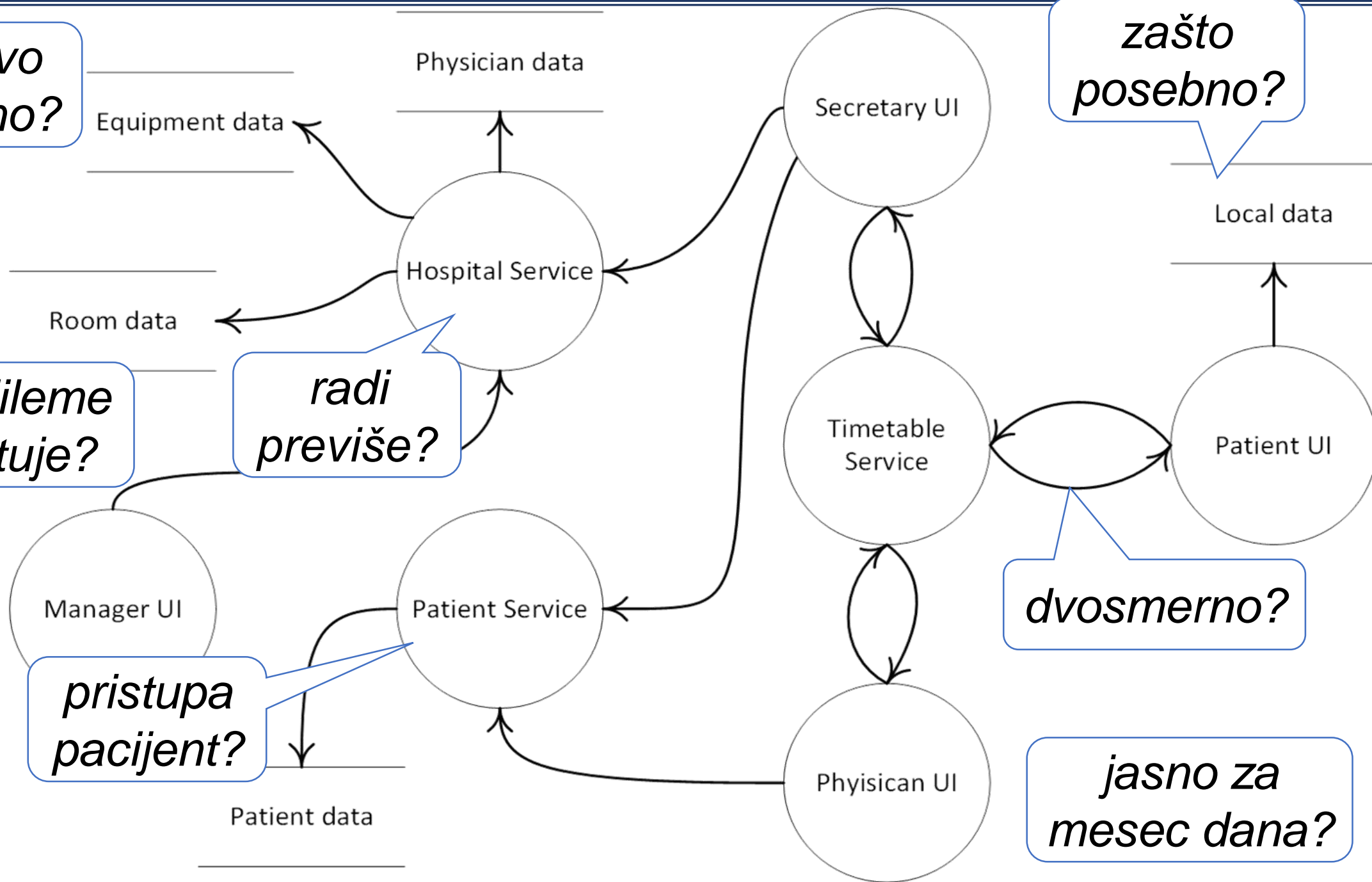
*dvosmerno?*

*pristupa pacijent?*

Patient data

Physician UI

*jasno za mesec dana?*



# Tehnike razrade softverskih zahteva I



Korisnička priča?



Prototip?

## Prototipovi nisu samo za UI

### Arhitektura sistema

Podaci zaštićeni kroz čitav njihov tok?

Odgovornosti glavnih celina jasne i prikladne?

Kanali između celina dobro definisani?

- Sprega između celina minimizovana?
- Ograničenja interfejsa definisana?

...

# Tehnike razrade softverskih zahteva I



Korisnička priča?



Prototip?

## Prototip nas uči

Arhitektura sistema

Ponašanje eksterne  
komponente

Performanse

Korisnički interfejs

Nova funkcionalnost

Struktura i sadržaj  
podataka

Z2.2: Pročitaj *Pragmatic Programmer* temu 13 i definiši kako izgledaju i kako koristimo navedene vrste prototipova



# Tehnike razrade softverskih zahteva I



Korisnička priča?



Prototip?

## Prednosti i mane prototipa

### Zahtevi

- ❖ Jasniji zahtevi, *UI design* (HCI)
- ❖ Nepotpuna analiza zahteva

### Vreme

- ❖ Brže prihvatanje i odbacivanje ideja
- ❖ Razvoj prototipa zahteva vreme

### Komunikacija

- ❖ Diskusija sa korisnikom i unutar tima
- ❖ Nije pogodan za dokumentaciju