



SVEUČILIŠTE U ZAGREBU



Fakultet
elektrotehnike i
računarstva

Razvoj komunikacijske programske podrške

Ak. god. 2021/2022

Modeli razvoja programskog proizvoda

Uvodno o pojmovima

- ◆ Životni ciklus programskog proizvoda (definicija prema standardu TL9000)
 - procesi, aktivnosti i zadaci uključeni u koncept, definiciju, razvoj, funkcioniranje i održavanje programskog proizvoda
 - sastoji se od faza (npr. definicija i analiza zahtjeva, dizajn sustava, ..., itd.)
 - Često se još koriste i sinonimi
 - generički proces razvoja programskog proizvoda
 - proces razvoja programskog proizvoda
 - razvoj programskog proizvoda
 - engleski nazivi: *software lifecycle*, *(generic) software development (process)*, *software process*

Uvodno o pojmovima (2)

- ◆ Osnovne skupine procesnih aktivnosti zajedničke svim procesima razvoja bez obzira na primijenjeni model
 - odgovaraju fazama generičkog životnog ciklusa programskog proizvoda (u daljnjem tekstu: PP)
- ◆ Različite organizacije mogu imati međusobno različite procese za razvoj istog tipa PP
- ◆ Ista organizacija može koristiti različite procese za razvoj različitih PP

Životni ciklus programskog proizvoda



Životni ciklus programskog proizvoda

Definicija i analiza zahtjeva

Dizajn sustava

Dizajn programa

Implementacija programa

Testiranje programa

Integracijsko testiranje

Testiranje sustava

Isporuka

Održavanje

Povlačenje iz uporabe

Životni ciklus programskog proizvoda

Definicija i analiza zahtjeva

**Dizajn sustava
Dizajn programa**

**Implementacija programa
Testiranje programa**

**Integracijsko testiranje
Testiranje sustava**

Isporuka

Održavanje

Povlačenje iz uporabe

Procesni modeli

◆ Modeli životnog ciklusa PP

■ *SW development models*

- definiraju slijed aktivnosti koje se izvode pri razvoju i održavanju
- na različite načine kombiniraju različite procesne aktivnosti
- identificiraju redoslijed, ulaze i izlaze iz pojedinih aktivnosti
- ne osiguravaju veliku razinu detalja, nego prikazuju samo ključne aktivnosti i njihove ovisnosti

◆ Definirane su različite norme (*standards*)

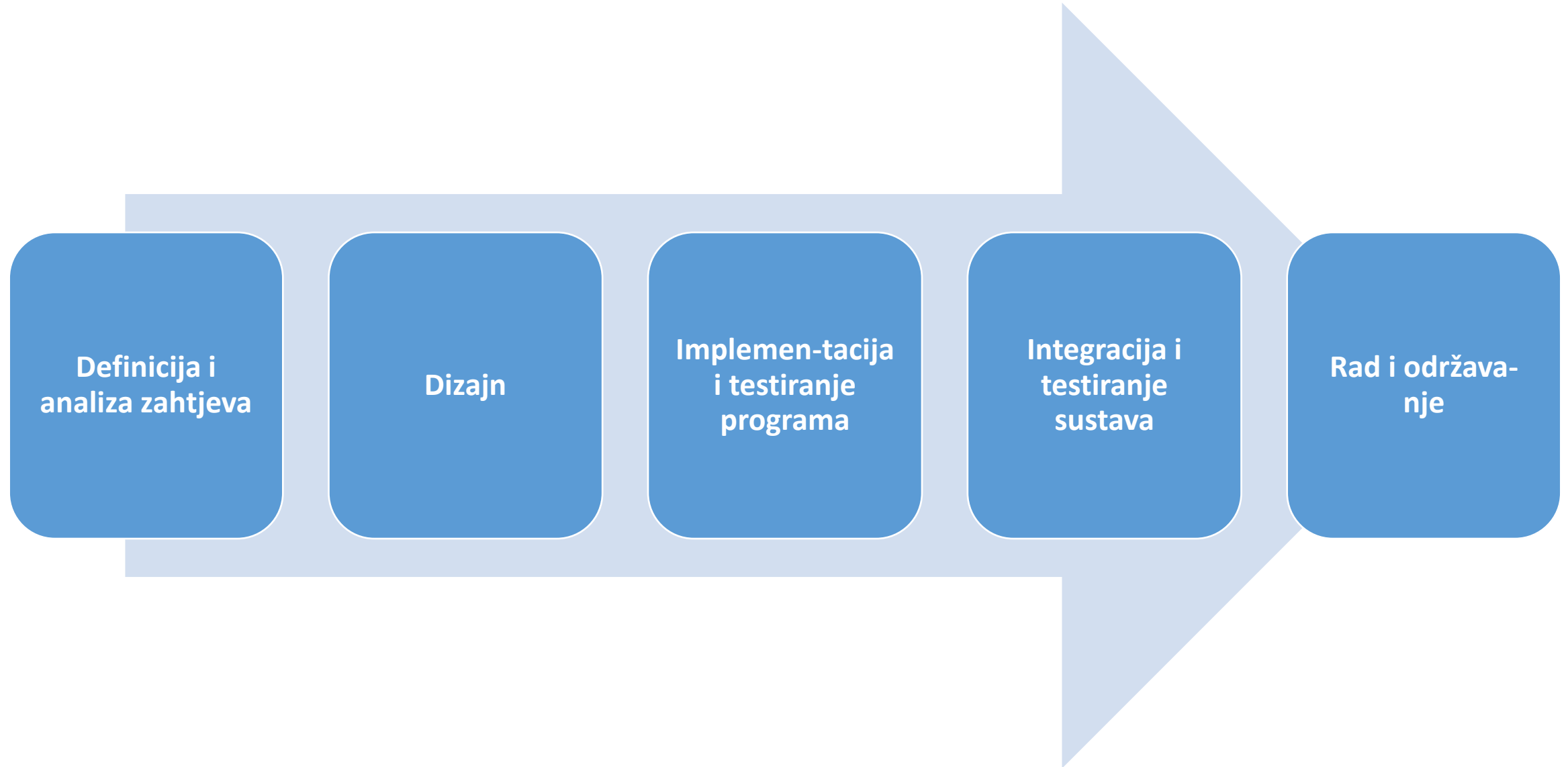
- specificiraju pojedini model (npr. ISO 9000) ili
- definiraju načine procjene i poboljšanja postojećih modela (npr. ISO 15504)

Ad hoc model

- ◆ Code And Fix
- ◆ Nespecificirani zahtjevi
- ◆ Nedefinirana arhitektura
- ◆ Razvije se prototip proizvoda koji se ponovno i ponovno razvija dok korisnik nije zadovoljan



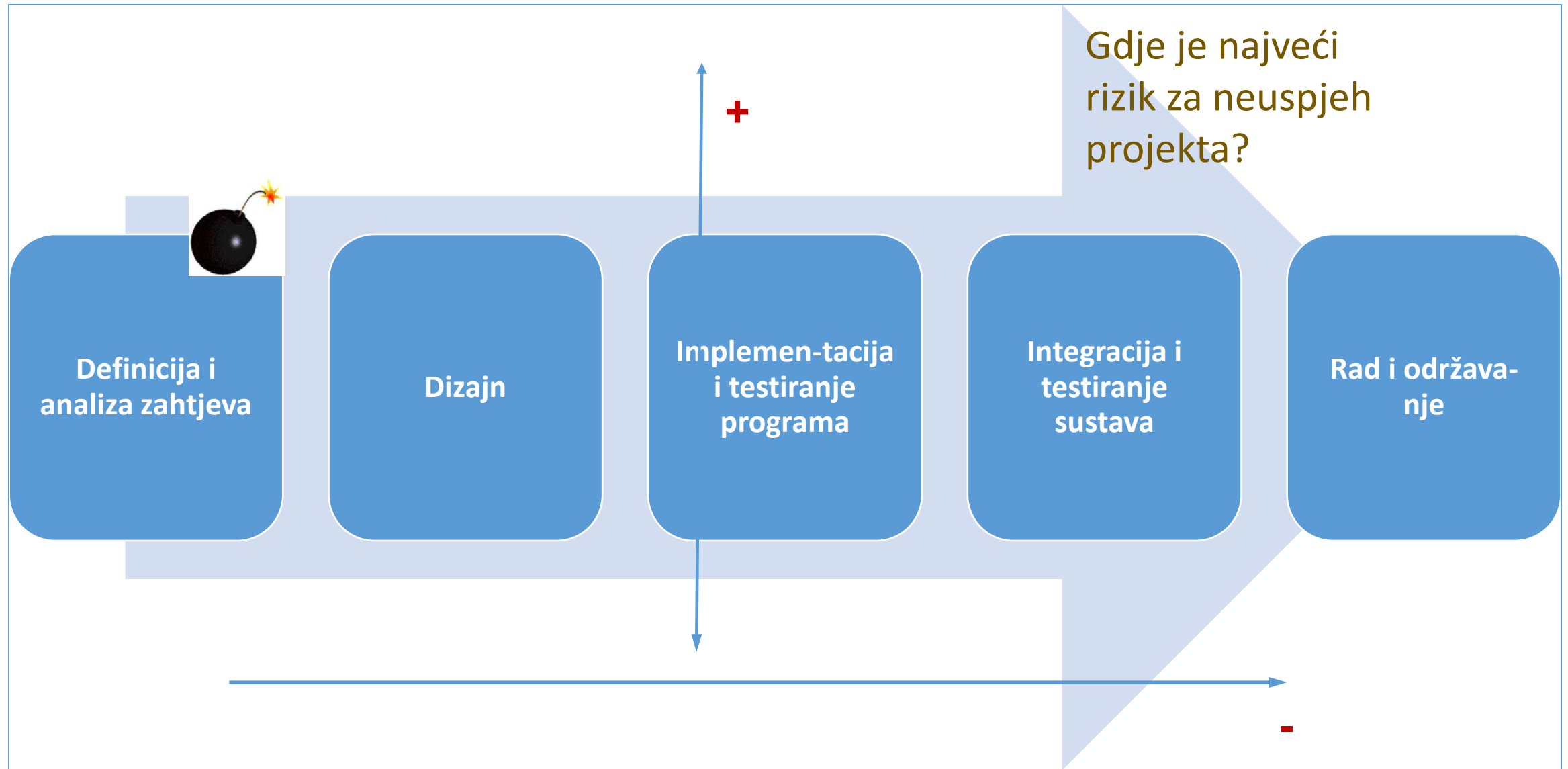
Vodopadni model



Vodopadni model

- ◆ Najstariji sustavni pristup razvoja PP
- ◆ Nastao 60-ih godina - proces razvoja prvi puta postaje vidljiv
- ◆ Aktivnosti se kaskadno odvijaju, slijede jedna drugu
 - ☺ dobra vidljivost - svaka aktivnost definira informacije i uvjete za sljedeću, u pojedinom trenutku se točno zna što se radi
 - ☺ orijentiran prema dokumentaciji
 - ☹ povratak unatrag
 - ☹ dugotrajan proces

Vodopadni model



Vodopadni model

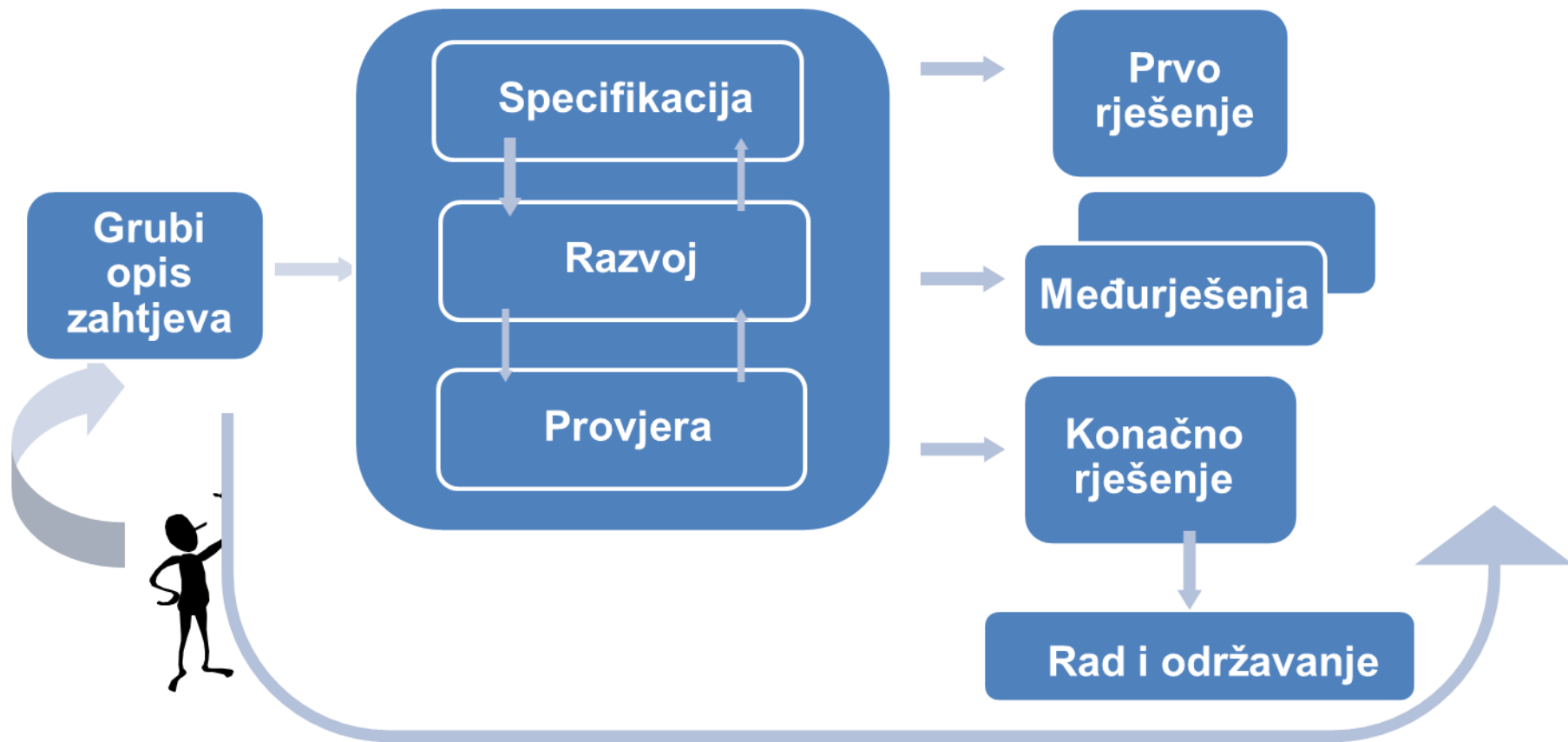
◆ Primjena

- veliki projekti
- novi sustavi (malo iskustva, konceptijski dobro objašnjeno)
- “dobri” zahtjevi

◆ Ne primijeniti

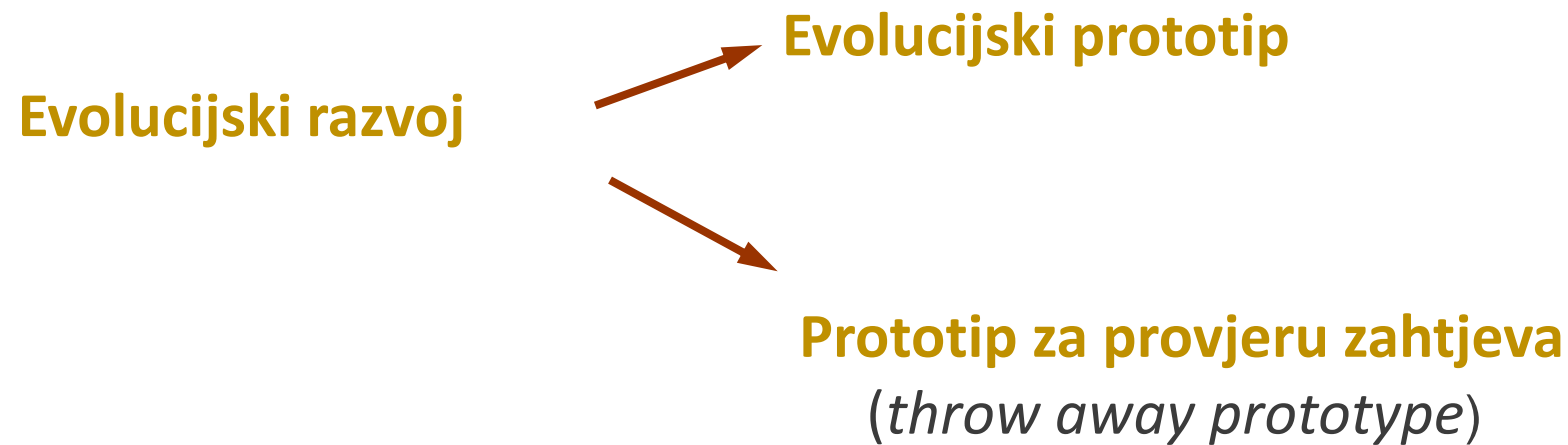
- mali projekti
- evolucija postojećeg sustava
- “loši” zahtjevi (očekuju se problemi u projektu vezani uz zahtjeve, npr. česte promjene, nedovoljna i loša specifikacija, neistražena domena i sl.)

Evolucijski model



Evolucijski model

- 😊 adekvatno trajanje (kreće se odmah u posao, tijekom rada se rješavaju problemi)
- 😞 loša vidljivost
- 😞 opasnost isporuke rješenja “bez arhitekture”



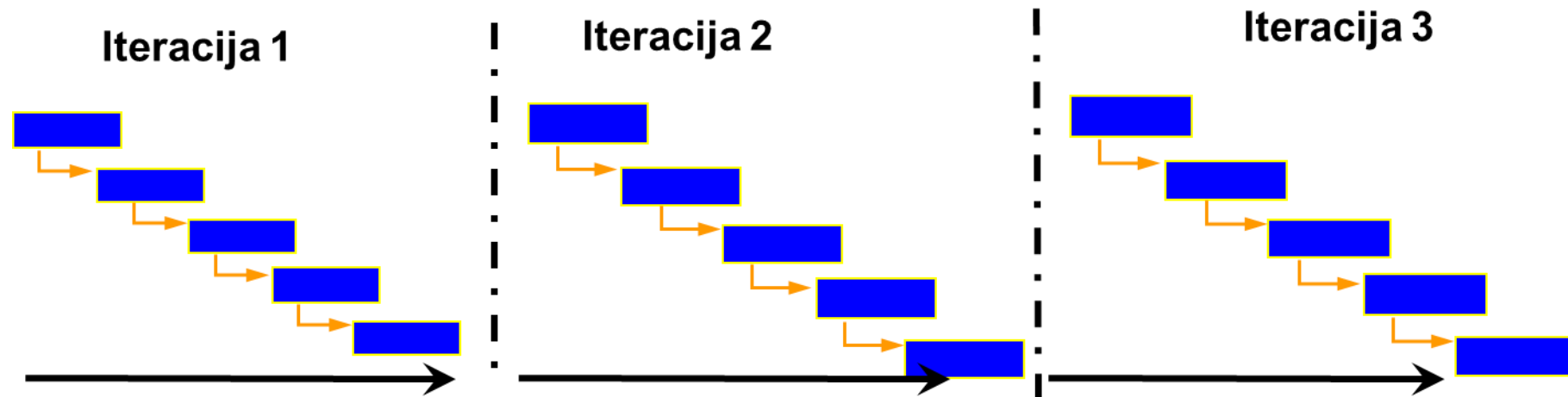
Evolucijski model

◆ Primjena

- manji projekti
- dio sustava
- slabi zahtjevi
- vidljiva funkcionalnost (grafička sučelja)
- motivirani tim



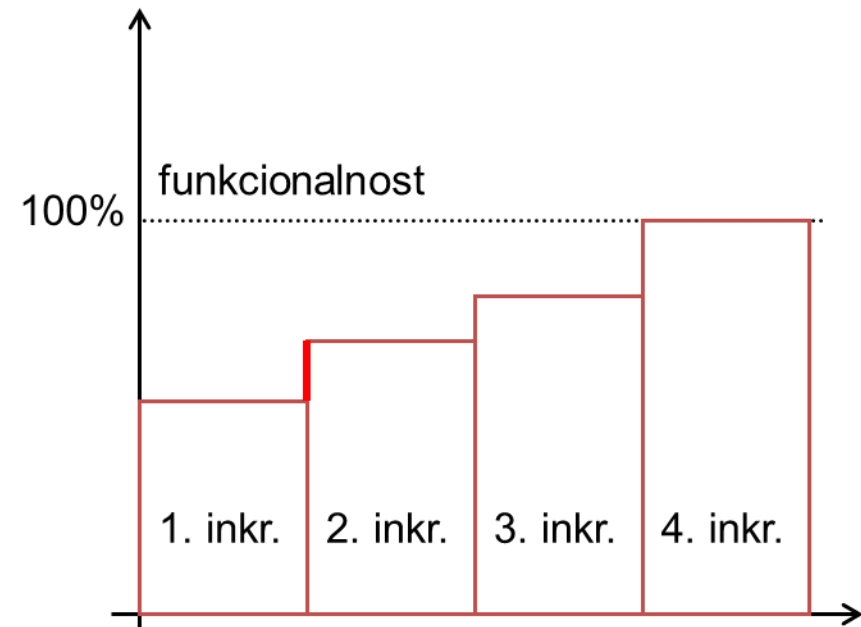
Iterativni model



Iterativni model

- ◆ kombinira elemente vodopadnog modela s iterativnim pristupom evolucijskog
- ◆ prvi inkrement – **jezgreni proizvod**

- 😊 stalna provjera
- 😊 povratno djelovanje
- 😞 rano definiranje arhitekture
- 😊 fleksibilan odnos s naručiteljem



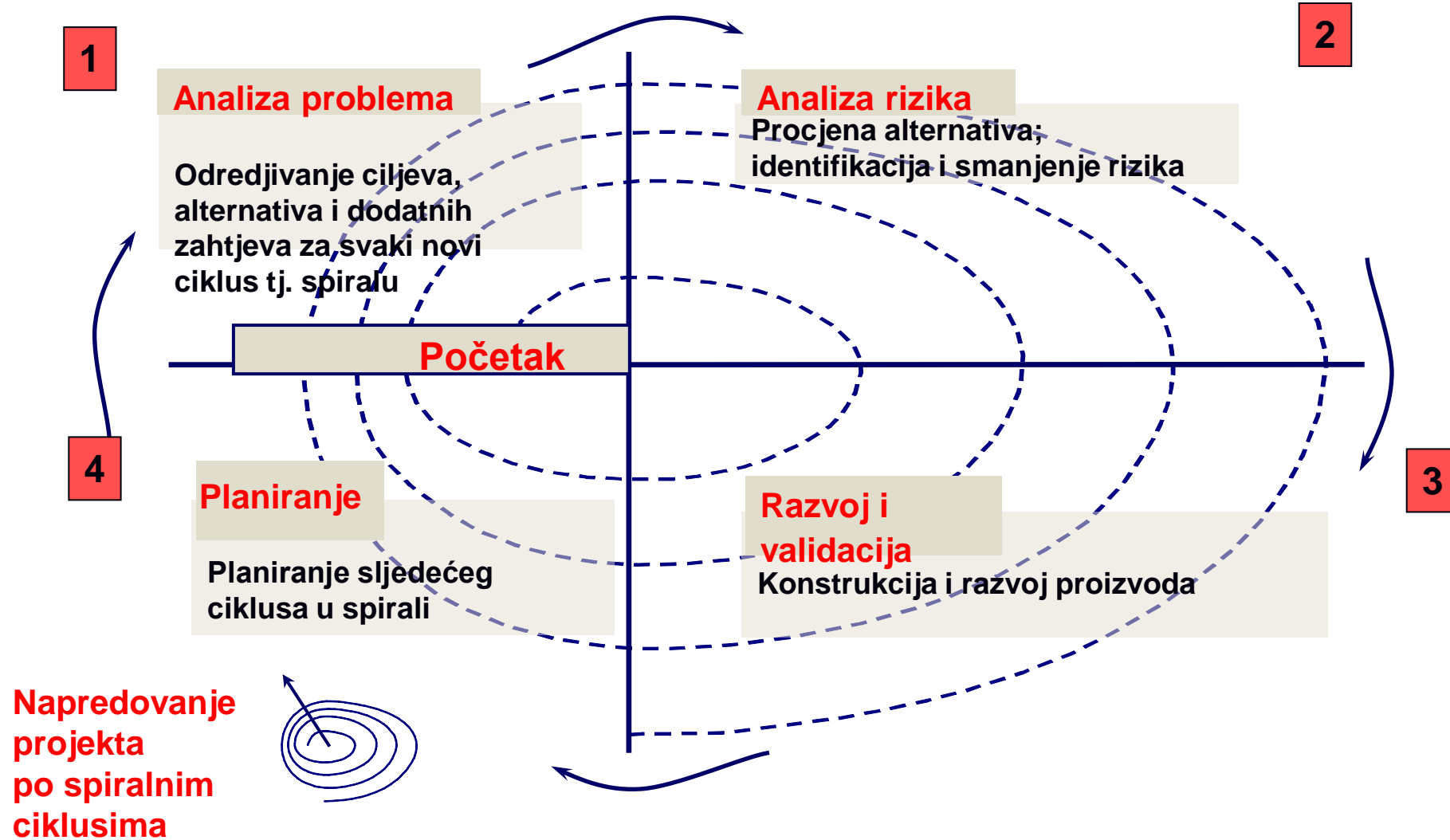
Spiralni model

- ◆ 1985. godine, autor Barry Boehm

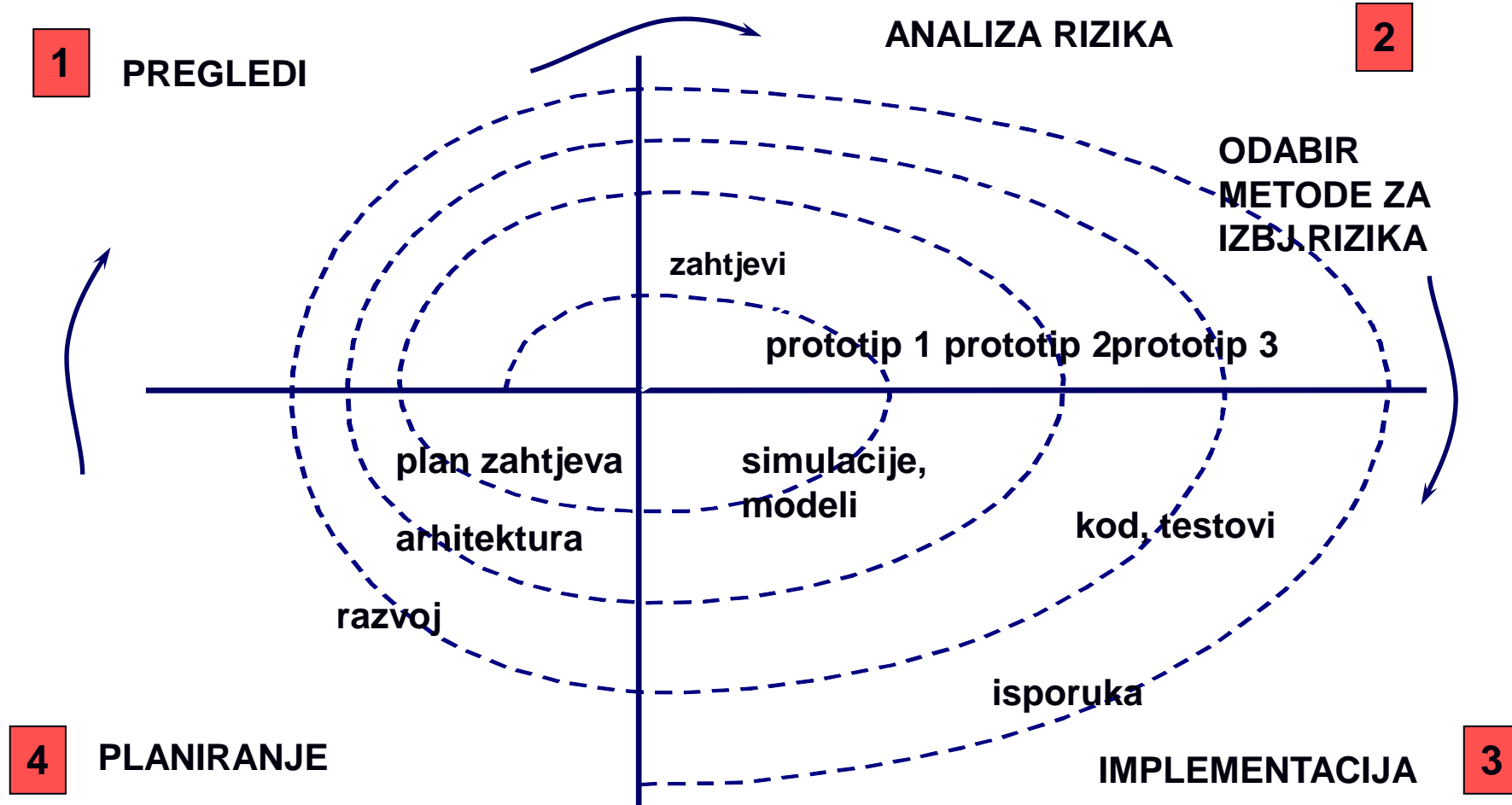


- ◆ Razvoj projekta ide po fazama koje se definiraju s ciljem identifikacije i smanjenja rizika neuspjeha
- ◆ Model ima oblik spirale
 - u jednoj spirali razvija se dio PP ili povećava njegova funkcionalnost
- ◆ Iterativni pristup razvoju PP
- ◆ Nadogradnja evolucijskog modela

Spiralni model (2)



Spiralni model (3)



Spiralni model (4)

- ◆ Na početku svake faze provodi se procjena rizika
- ◆ Nastoji se utvrditi moguće rizike i razriješiti ih prije nastavka (uklanjanjem ili svođenjem na najmanju moguću mjeru)
- ◆ U slučaju da je rizik preveliki, projekt se prekida
- ◆ Primjeri rizika
 - Rizik da isporučeni PP neće odgovarati stvarnim zahtjevima
 - izrada prototipa kao dio faze određivanja zahtjeva
 - Rizik da će cijena izrade premašiti korist ostvarenu uporabom
 - provođenje analize troškova-koristi (*cost-benefit*) prije provođenja svake pojedine faze

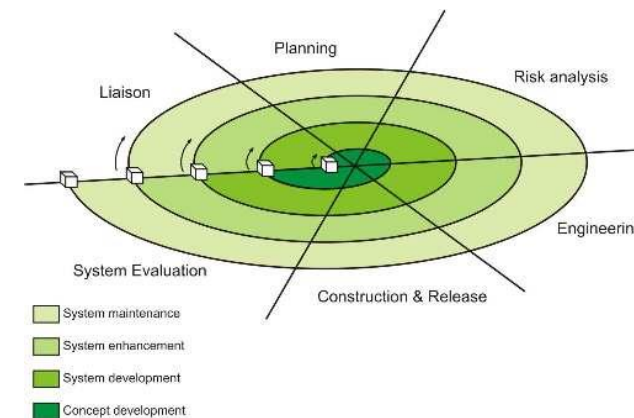
Spiralni model (5)

- 😊 dobra kontrola rizika
- 😊 moguće uklopiti u već postojeći model razvoja
- 😊 dobro prilagođen velikim projekt
- 😞 skup
- 😞 ponekad dugotrajan
- 😞 zahtijeva veliko stručno znanje prilikom procjene rizika – visoko domensko i upravljačko znanje

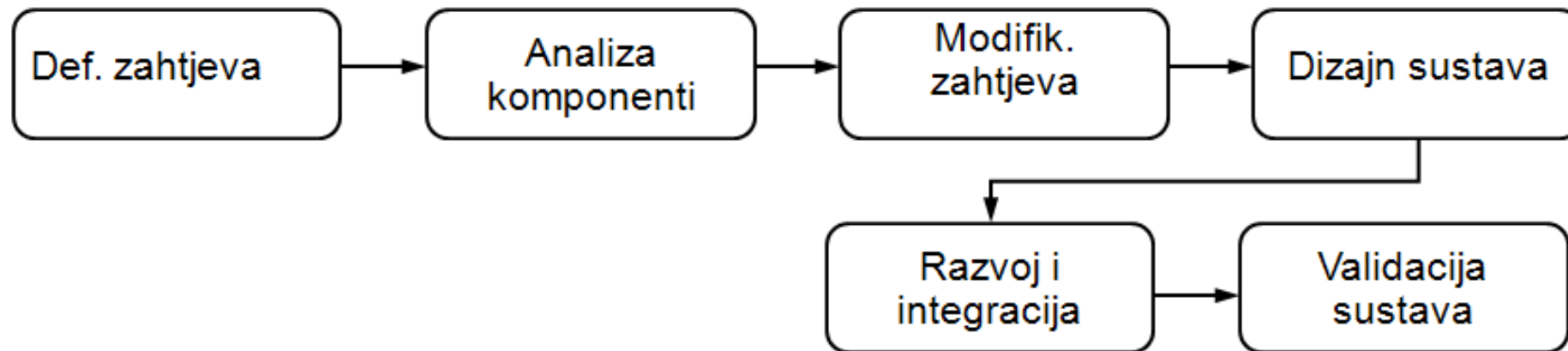
Spiralni model (5)

◆ Primjena

- Projekti u kojima se očekuju različiti problemi s tehnologijom, ljudima, tržištem i okolinom
- Interni projekti (naručitelj i izvođač iz iste organizacije)
- Izgradnja velikih sustava (troškovi izrade malih sustava su mali)
- Projekti u kojima provođenje analize rizika ne predstavlja preveliki relativni trošak (npr. za projekte iznad 25 kUSD)



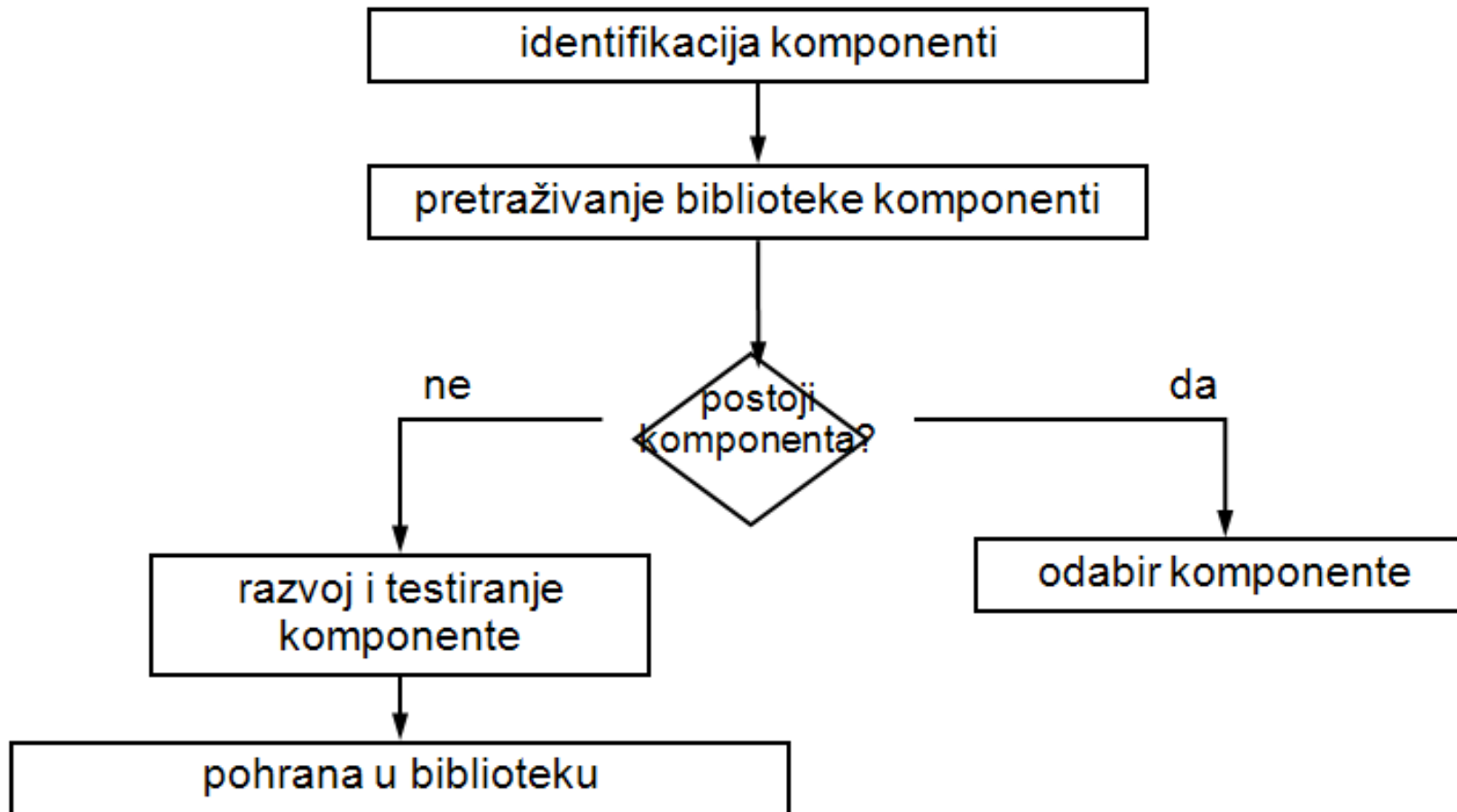
Komponentni razvoj



Komponentni razvoj

- ◆ *Component Based Development* (CBD)
- ◆ nastao na temeljima OO programiranja
- ◆ Osnovna ideja
 - dobro dizajnirane i implementirane objektno orijentirane klase mogu se više puta koristiti u različitim aplikacijama i arhitekturama
- ◆ **Komponenta**
 - dio programa koji se može izvršiti na logičkom ili fizičkom nivou
 - jedno ili više sučelja
 - jedinstveni identifikator
 - enkapsulira podatke irelevantne za poziv komponente

Komponentni razvoj



Komponentni razvoj

◆ razvoj je integracija

😊 brzi rezultati

😊 brza demonstracija sustava

😊 uštede vremena i troškova

😊 povećanje produktivnosti

😞 nestaju generička znanja

◆ Primjena

- ugradbeni sustavi

- telekomunikacijski sustavi

Korisnički usmjeren razvoj programskog proizvoda

- ◆ Metoda razvoja progr. proizvoda koja se zasniva na paradigmi „*User Centered Design*” (UCD)
 - pri čemu riječ „*design*” ovdje označava vanjski izgled i logiku funkcioniranja programskog proizvoda (a ne arhitekturu)

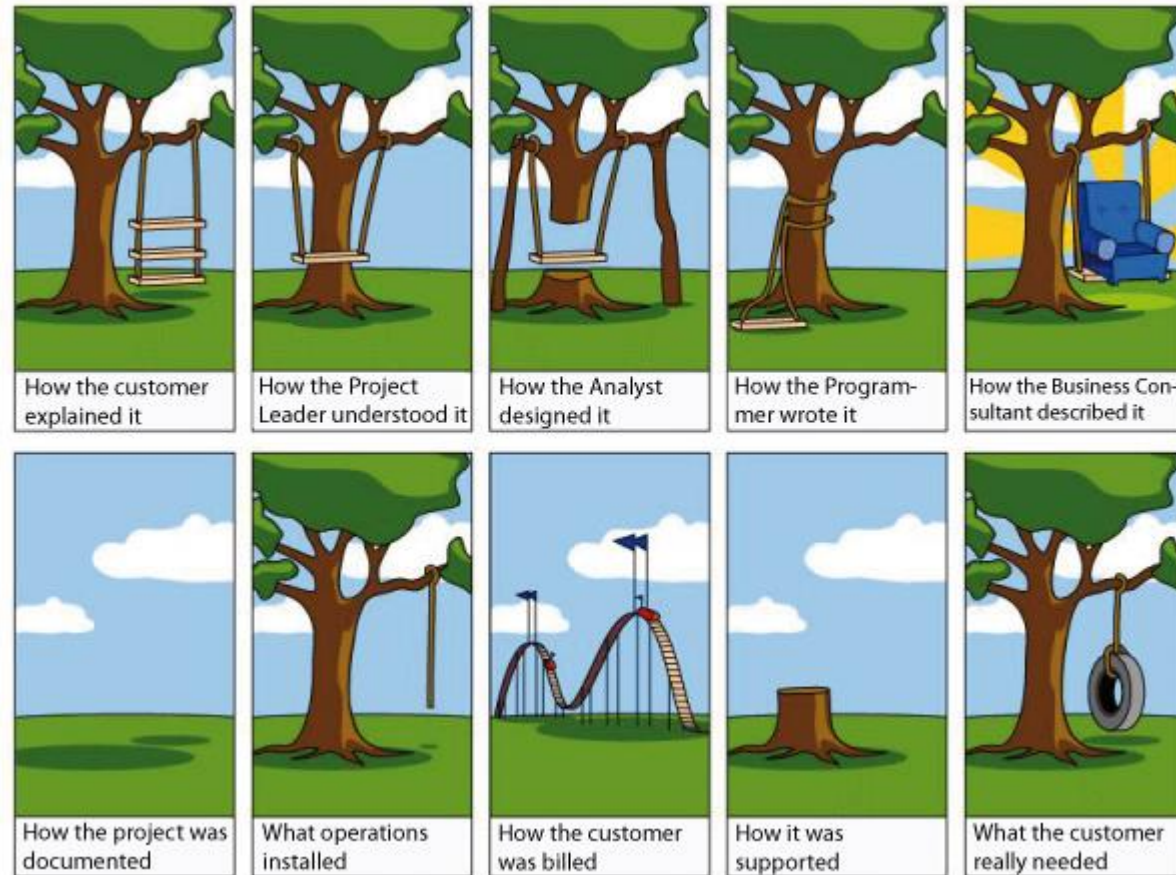
System Centered Design

- What can I easily build on this platform?
- What can I create from the available tools?
- What do I as a programmer find interesting?



http://marjan.fesb.hr/~mcagalj/HCI_12/Lectures/Lecture_4.ppt

Ovako nekako



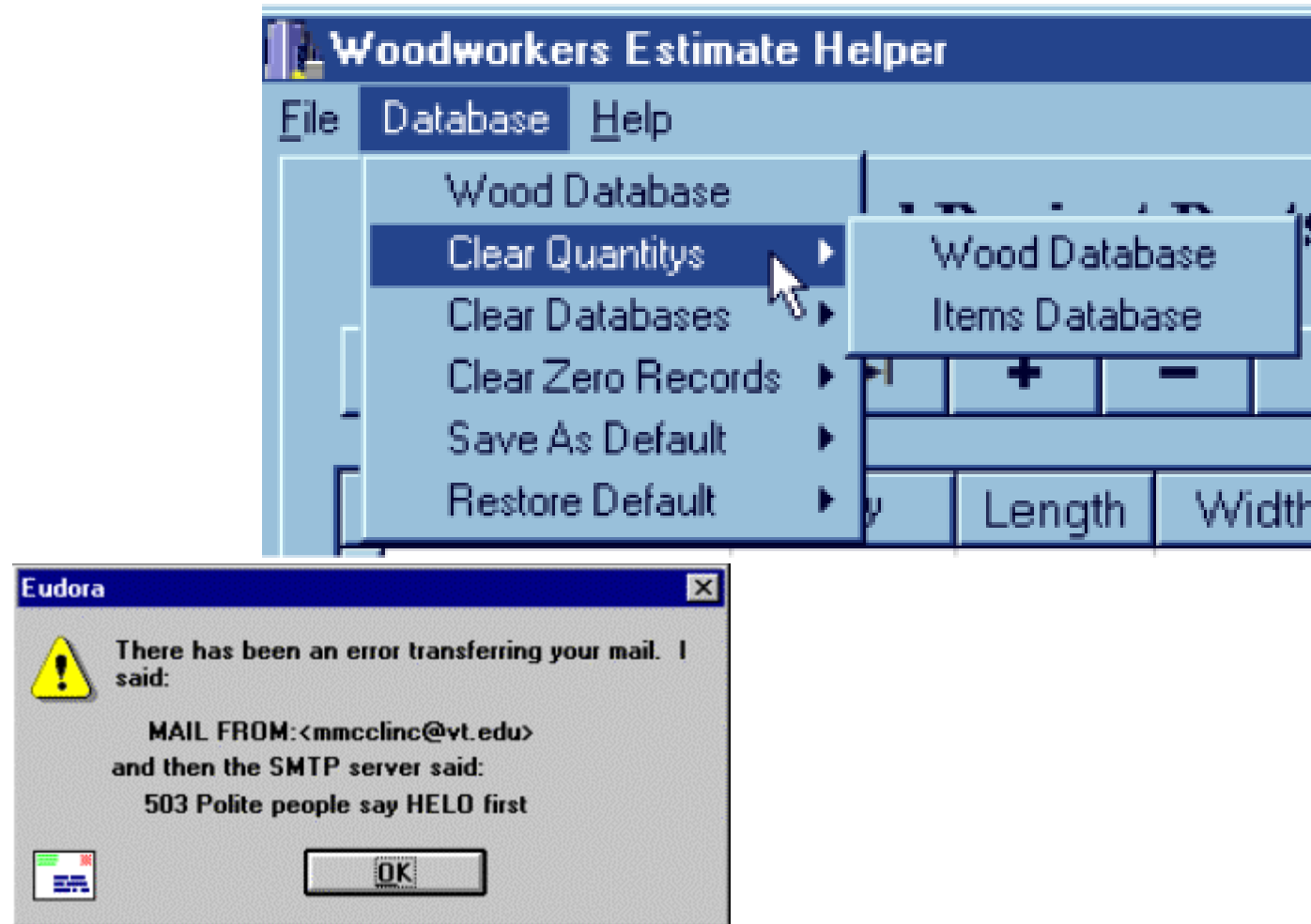
Korisnički usmjeren razvoj programskog proizvoda

- ◆ Izgled i logika programskog proizvoda se zasnivaju na korisnikovima
 - mogućnostima i potrebama
 - kontekstu uporabe
 - poslu
 - aktivnostima

BEAR IN
MIND, WE'LL
NEVER MEET
A CUSTOMER
OURSELVES.



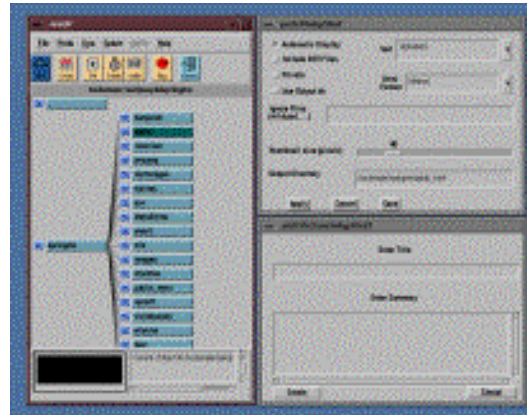
Drastični primjer



SMTP Error: 503 This mail server requires authentication

Korisnički usmjeren razvoj programskog proizvoda

◆ Model korisničkog sučelja



◆ Model korisničkog sučelja

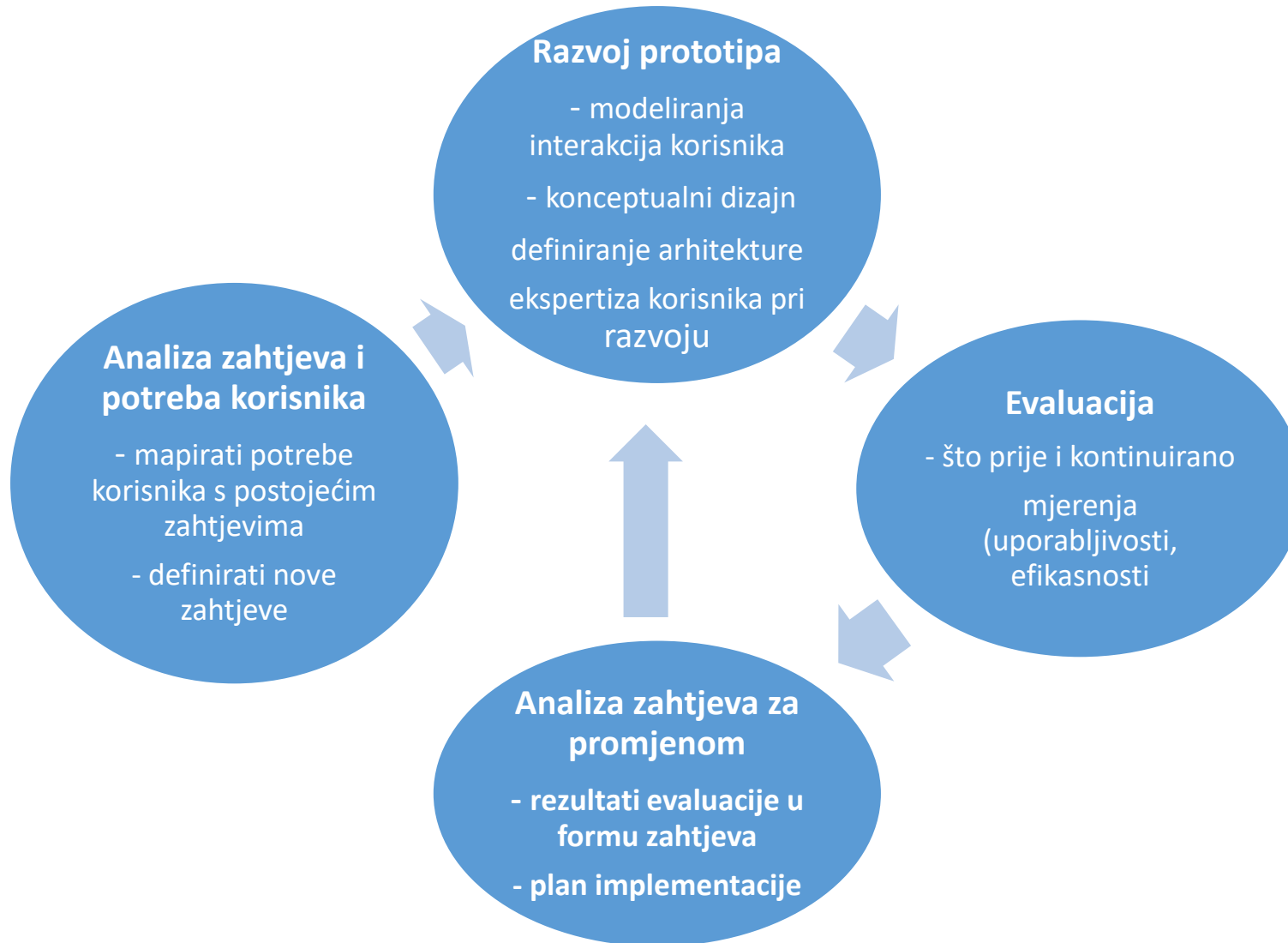


Korisnički usmjeren razvoj programskog proizvoda

ISO 13407 standard *Human-centred design processes for interactive systems*

- ◆ poboljšanje uporabivosti i korisnosti
 - ◆ identifikacija korisnika i njihovih ciljeva
 - ◆ povezivanje dobivenih podataka o korisnicima s razvojem usluge
-
- ◆ <https://www.usabilitytest.com/usabilitynet/tools-13407stds>

Korisnički usmjeren razvoj programskog proizvoda



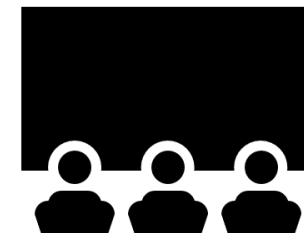
Pogledajte video....

Principles of Human-Centered Design (Don Norman)

◆ https://www.youtube.com/watch?v=rmM0kRf8Dbk&ab_channel=NNgroup

Software Development Lifecycle in 9 minutes!

◆ https://www.youtube.com/watch?v=i-QyW8D3ei0&ab_channel=Testaholic



Samostalni rad...

- ◆ Navedite kategorije korisnika mobilne aplikacije vezane uz pametno parkiralište.
- ◆ Navedite tri specifične potrebe korisnika mobilne aplikacije vezane uz pametno parkiralište (te potrebe mogu imati korisnici svih prethodno navedenih kategorija) koje bi pri razvoju trebalo uzeti u obzir, ako želimo da je web sjedište prilagođeno mogućnostima što većeg broja korisnika.
- ◆ Odaberite jednu potrebu iz prethodne točke i formulirajte je u programski zahtjev (ili više njih).



Dodatna literatura

- ◆ Dizajn usmjeren korisnicima, Hrvatsko dizajnersko društvo, <https://dizajn.hr/blog/dizajn-usmjeren-korisnicima/>
- ◆ Don Norman, Human-Centered Design Considered Harmful, [http://www.jnd.org/dn.mss/human-centered design considered harmful.html](http://www.jnd.org/dn.mss/human-centered_design_considered_harmful.html)