

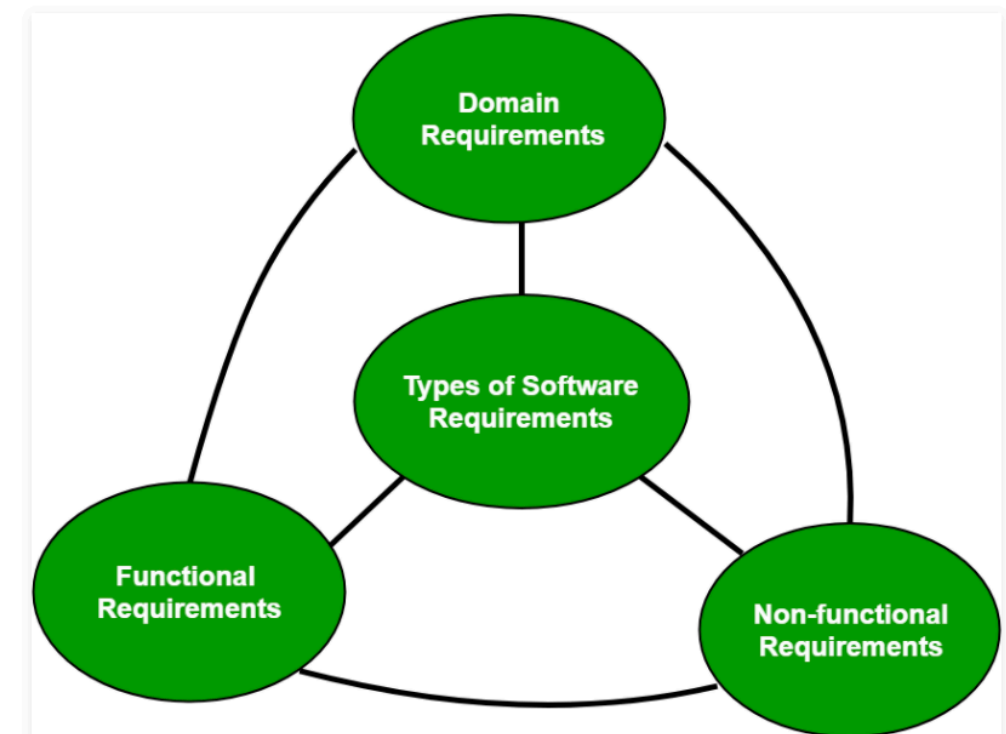
SIMS - VEŽBE 01

---

# FUNKCIONALNI I NEFUNKCIONALNI ZAHTEVI

# KLASIFIKACIJA SOFTVERSKIH ZAHTEVA

- ▶ Standard IEEE 729 softverske zahteve definiše na sledeći način:
  - ▶ Uslov ili mogućnost korisnika da reši neki problem ili postigne određeni cilj
  - ▶ Uslov ili sposobnost koji sistem ili komponenta sistema moraju da ispune ili poseduju da bi udovoljili ugovoru, standardu, specifikaciji ili drugim formalno nametnutim dokumentima
  - ▶ Dokumentovani prikaz stanja ili sposobnosti kao u 1 i 2
- ▶ Sve softverske zahteve možemo podeliti u 3 velike grupe:
  1. Funkcionalni zahtevi
  2. Nefunkcionalni zahtevi
  3. Domenski zahtevi



---

## DOMENSKI ZAHTEVI

- ▶ Zahtevi domena su zahtevi koji su karakteristični za određenu kategoriju ili domen projekata
- ▶ U ovu kategoriju spadaju osnovne funkcije koje sistem određenog domena mora nužno pokazivati
- ▶ Na primer, u akademskom softveru koji održava evidenciju o školi ili fakultetu, funkcionalnost omogućavanja pristupa listi fakulteta i spisku učenika svakog razreda je zahtev domena
- ▶ Ovi zahtevi su stoga identifikovani iz tog modela domena i nisu specifični za korisnika

*“Any Requirement Which Specifies What The System Should Do.”*

---

# FUNKCIONALNI ZAHTEVI

- ▶ Funkcionalni zahtevi definišu osnovno ponašanje sistema
- ▶ U osnovi, oni su ono što sistem mora ili ne sme da radi
- ▶ Mogu se posmatrati kao način na koji sistem reaguje na ulazne podatke tj. mogu biti predstavljeni u obliku ulaznih podataka koji se daju sistemu, izvršenoj operaciji i očekivanom izlazu



- ▶ Funkcionalni zahtevi obično definišu IF/THEN ponašanje i uključuju proračune, unos podataka i poslovne procese
- ▶ Funkcionalni zahtevi su karakteristike koje omogućavaju sistemu da funkcioniše onako kako je zamišljeno. Drugim rečima, **ako funkcionalni zahtevi nisu ispunjeni, sistem neće funkcionisati**. Funkcionalni zahtevi su karakteristike proizvoda i usredsređeni su na zahteve korisnika

---

# FUNKCIONALNI ZAHTEVI

- ▶ Sve ove funkcionalnosti moraju biti nužno ugrađene u sistem kao deo ugovora. To su u osnovi zahtevi koje je naveo korisnik i koje se mogu videti direktno u konačnom proizvodu, za razliku od nefunkcionalnih zahteva
- ▶ Na primer, u sistemu upravljanja bolnicom, lekar treba da bude u mogućnosti da preuzme informacije svojih pacijenata. Svaki funkcionalni zahtev na visokom nivou može da uključuje nekoliko interakcija ili dijaloga između sistema i spoljnog sveta. Da bi se tačno opisali funkcionalni zahtevi, moraju se nabrojati svi scenariji.

---

# FUNKCIONALNI ZAHTEVI

- ▶ Funkcionalni zahtevi se bave sledećim pitanjima:
  - ▶ Pravila poslovanja
  - ▶ Korekcije transakcija, prilagođavanja i otkazivanja
  - ▶ Administrativne funkcije
  - ▶ Autentikacija
  - ▶ Nivoi ovlašćenja - autorizacija
  - ▶ Spoljni interfejsi
  - ▶ Zahtevi za sertifikaciju
  - ▶ Izveštavanje
  - ▶ Istorijski podaci
  - ▶ Pravni ili regulatorni zahtevi

---

# ZADATAK

- ▶ Zamislite da pravite softver za elektronski dnevnik za škole u Srbiji. Navedite:
  - ▶ 2 funkcionalna zahteva
  - ▶ 2 domenska zahteva

---

# ZADATAK

- ▶ Funkcionalni zahtevi (šta sistem radi):
  - ▶ Nastavnik može da unese ocenu za učenika samo za predmet koji predaje
  - ▶ Roditelj može da pregleda izostanke i ocene svog deteta putem korisničkog naloga
- ▶ Domenski zahtevi (pravila specifična za školstvo u Srbiji):
  - ▶ Ocene moraju biti unete u skladu sa zakonskom skalom: od 1 do 5, gde je 5 najviša ocena
  - ▶ Prikaz ukupnog uspeha učenika mora da koristi zvanične kategorije: odličan, vrlo dobar, dobar, dovoljan, nedovoljan – u skladu sa Pravilnikom o ocenjivanju



*“Any Requirement That Specifies How The System Performs A Certain Function.”*

---

# NEFUNKCIONALNI ZAHTEVI

- ▶ Iako funkcionalni zahtevi definišu šta sistem radi ili ne sme da radi, nefunkcionalni zahtevi određuju kako sistem to treba da radi
- ▶ Nefunkcionalni zahtevi ne utiču na osnovnu funkcionalnost sistema (otuda i naziv, nefunkcionalni zahtevi). Čak i ako nefunkcionalni zahtevi nisu ispunjeni, sistem će i dalje ispunjavati svoju osnovnu namenu.
- ▶ Ako će sistem i dalje raditi bez zadovoljavanja nefunkcionalnih zahteva, zašto su oni važni? Odgovor je upotrebljivost
  - ▶ Nefunkcionalni zahtevi definišu ponašanje sistema, funkcije i opšte karakteristike koje utiču na korisničko iskustvo
  - ▶ Koliko su nefunkcionalni zahtevi definisani i izvršeni, određuje koliko je sistem lak za upotrebu i koristi se za procenu performansi sistema.
  - ▶ Nefunkcionalni zahtevi su svojstva proizvoda i usredsređeni su na očekivanja korisnika
- ▶ PRIMER - ukoliko je funkcionalni zahtev da sistem učitava veb stranicu nakon što neko klikne na dugme, trebalo bi da postoji povezan nefunkcionalni zahtev koji određuje koliko brzo se veb stranica mora učitati. Bez toga su korisničko iskustvo i percepcija kvaliteta u opasnosti ako su prisiljeni predugo čekati, iako su funkcionalni zahtevi u potpunosti ispunjeni.

---

# NEFUNKCIONALNI ZAHTEVI

- ▶ Nefunkcionalni zahtevi su u osnovi ograničenja kvaliteta koja sistem mora da zadovolji u skladu sa projektnim ugovorom. Prioritet ili stepen primene ovih faktora varira od jednog projekta do drugog. Takođe se nazivaju i zahtevi za ponašanje. U osnovi se bave pitanjima poput:
  - ▶ Prenosivost
  - ▶ Sigurnost
  - ▶ Održavanje
  - ▶ Pouzdanost
  - ▶ Prilagodljivost
  - ▶ Performanse
  - ▶ Mogućnost ponovne upotrebe
  - ▶ Fleksibilnost

---

# NEFUNKCIONALNI ZAHTEVI

- ▶ NFR su klasifikovani u sledeće tipove:
  - ▶ Ograničenja interfejsa
  - ▶ Ograničenja performansi: vreme odziva, sigurnost, prostor za skladištenje itd.
  - ▶ Operativna ograničenja
  - ▶ Ograničenja životnog ciklusa: upravljivost, prenosivost itd.
  - ▶ Ekonomska ograničenja

# RAZLIKE IZMEĐU ZAHTEVA

Funkcionalni	Nefunkcionalni
Definišu sistem ili njegovu komponentu	Definišu atribut kvaliteta sistema
Precizira: „Šta sistem treba da uradi?“	Precizira: „Kako sistem treba da ispunjava funkcionalne zahteve?“
Određuje ih korisnik	Specificiraju ih tehničke role - arhitekta, tehnički vođe i programeri softvera
Obavezno je ispuniti ove zahteve	Nije obavezno ispunjavanje ovih zahteva
Definisano na nivou komponente	Primenjeno na čitav sistem
Pomaže vam da verifikujete funkcionalnost softvera	Pomaže vam da verifikujete performanse softvera
Funkcionalna testiranja poput sistema, integracije, end to end, API testiranja itd	Nefunkcionalno testiranje kao što su performanse, stres, upotrebljivost, bezbednosno testiranje itd

---

# KAKO SE PRIKUPLJAJU?

- ▶ Jedan od najboljih načina za prikupljanje zahteva je okupljanje svih zainteresovanih strana za vođenu **brainstorming** sesiju
- ▶ Imajte na umu da u najvećem broju slučajeva naručioci projekta nisu korisnici i njegovi korisnici
- ▶ Uključite predstavnike korisnika u tim, koji su jedan od najboljih izvora za nefunkcionalne zahteve
- ▶ Oni su ti koji će vam reći stvari poput
  - ▶ *Ako bih mogao da unesem podatke ovim redosledom ...*
  - ▶ *Ako bih mogao da tim podacima pristupam i preko prečice ...*
  - ▶ *Ovo je nešto što inače jako retko koristimo ...*

---

# KAKO SE PRIKUPLJAJU?

- ▶ Dok razmišljate o funkcionalnim zahtevima, uzmite u obzir sledeće grupe:
  - ▶ **Poslovni zahtevi** - obično sadrže krajnji cilj, kao što je sistem za naručivanje, mrežni katalog ili fizički proizvod. Takođe može uključivati stvari kao što su tokovi odobrenja i nivoe autorizacije.
  - ▶ **Administrativne funkcije** - rutinske stvari koje će sistem raditi, poput izveštavanja.
  - ▶ **Zahtevi korisnika** - to je ono što korisnik sistema može da uradi, kao što je naručivanje ili pregledanje mrežnog kataloga.
  - ▶ **Sistemske zahtevi** - to su stvari poput softverskih i hardverskih specifikacija, sistemskih odgovora ili sistemskih radnji.

---

# KAKO SE PRIKUPLJAJU?

- ▶ Nakon definisanja funkcionalnih zahteva, razmislite o nefunkcionalnim zahtevima, kao što su:
  - ▶ **Upotrebljivost (usability)** - fokusira se na izgled korisničkog interfejsa i na to kako ljudi komuniciraju s njim. Koje su boje ekrani? Koliko su velika dugmad
  - ▶ **Pouzdanost / raspoloživost (reliability / availability)** - koji su zahtevi za vremenom rada aplikacije? Da li treba da funkcioniše 24/7/365?
  - ▶ **Prilagodljivost (scalability)** - kako potrebe rastu, može li sistem to da podnese? Za fizičke instalacije ovo uključuje rezervni hardver ili prostor za njegovu buduću instalaciju.
  - ▶ **Performanse (performance)** - kakav treba da bude odziv aplikacije?
  - ▶ **Podrška (supportability)** - da li se pruža interna podrška ili je potrebna daljinska dostupnost spoljnih resursa?
  - ▶ **Sigurnost (security)** - koji su bezbednosni zahtevi, kako za fizičku instalaciju, tako i iz cyber perspektive?

---

## KAKO SE PRIKUPLJAJU?

- ▶ Iako je idealno prvo definisati sve funkcionalne zahteve, a zatim preći na nefunkcionalne zahteve, stvarnost je takva da će ljudi razmišljati o obe vrste zahteva tokom brainstorming sesija
- ▶ Voditelj sastanka mora biti dobro upućen u razliku kako bi ih mogao klasifikovati kako diskusija odmiče.
- ▶ Tokom faze pokretanja projekta, treba biti pažljiv čak i prilikom usputnih razgovora sa korisnicima i zainteresovanim stranama. Možda ćete otkriti da iznose zahteve kojih se nisu setili tokom početnog prikupljanja zahteva.



---

# KAKO SE PIŠU?

- ▶ Funkcionalni i nefunkcionalni zahtevi mogu se napisati na nekoliko različitih načina. Evo nekoliko najčešćih, mada je sve što efikasno prenosi informacije svima koji su uključeni sasvim prihvatljivo
  1. Specifikacija zahteva
  2. Struktura raščlanjenog rada
  3. Korisničke priče
  4. Slučajevi upotrebe

---

# KAKO SE PIŠU?

1. Najčešći način pisanja funkcionalnih i nefunkcionalnih zahteva je dokument o **specifikaciji zahteva**.
  - ▶ Ovo je jednostavno pisani opis potrebnih funkcija
  - ▶ Navodi cilj projekta i uključuje pregled projekta kako bi se pružio kontekst, zajedno sa svim ograničenjima i pretpostavkama
  - ▶ Iako je većina dokumenta o specifikaciji zahteva narativna, dobra je ideja uključiti vizuelne prikaze zahteva kako bi se pomoglo netehničkim zainteresovanim stranama da razumeju opseg
2. Usko povezana sa dokumentom o specifikaciji zahteva je **struktura raščlanjenog rada ili WBS (Work Breakdown Structure)**.
  - ▶ Na ovaj način se razlaže čitav proces na njegove komponente, tako što se zahtevi „razlažu“ na njihove pojedinačne elemente sve dok se ne mogu dalje raščlaniti

---

# KAKO SE PIŠU?

## 3. Treći pristup su **korisničke priče (user stories)**

- ▶ Ovo su opis funkcionalnosti iz perspektive krajnjeg korisnika i tačno opisuju šta žele da sistem radi
- ▶ U njemu se efektivno navodi „Kao <tip korisnika>, želim <cilj> tako da <razlog>«
- ▶ Jedna prednost korisničkih priča je ta što im za pisanje nije potrebno puno tehničkog znanja
- ▶ Korisničke priče se takođe mogu koristiti kao prethodnica dokumenta o specifikaciji zahteva pomažući u definisanju korisničkih potreba

## 4. **Slučajevi upotrebe (use case)** slični su korisničkim pričama jer nije potrebno tehničko znanje

- ▶ Slučajevi upotrebe detaljno opisuju šta korisnik radi dok izvršava zadatak
- ▶ Slučaj upotrebe može biti „proizvod za kupovinu“ i sa stanovišta korisnika opisuje svaki korak u procesu kupovine

---

# PRIMERI FUNKCIONALNIH I NEFUNKCIONALNIH ZAHTEVA

Funkcionalni	Nefunkcionalni
Kada posetilac stranice otvori nalog, server će poslati poruku dobrodošlice	Kada šalje e-poruke dobrodošlice, server ih mora poslati u roku od 10 minuta od registracije
Kada se status porudžbine promeni u ispunjenje, lokalni štampač će odštampati listić za pakovanje	Kada se štampaju paketi za pakovanje, oni moraju biti sa obe strane listova belog papira A5 formata
Sistem mora omogućiti korisniku da popuni i preda obrazac usluge	Kada se obrazac zatraži od servera, on se mora učitati u toku 1 sekunde. Kada se pritisne dugme za slanje, slanje mora da se završi u roku od 2 sekunde