

---

# Razvoj softvera I

Kvalitet softvera



# Sadržaj



- Kvalitet softvera
  - Performanse
  - Skalabilnost
  - Nadogradnja
  - Sigurnost
  - Dostupnost
  - Integracija

---

*Quality means doing it right even when no one is looking.*



*- Henry Ford*

# Kvalitet softvera



- Kvalite je postao sinonim i jedan od glavnih karakteristika savremenog softvera. Sve faze razvoja su usmjerene u pravcu razvoja kvalitetnog softvera, te zbog toga veliki značaj pripada karakteristikama, odnosno atributima koji omogućavaju mjerenje kvaliteta softvera.
- Prilikom određivanja kvaliteta softvera posmatraju se sljedeće karakteristike:
  - Performanse
  - Skalabilnost
  - Nadogradnja
  - Sigurnost
  - Dostupnost
  - Integracija



# Performanse



- Performanse predstavljaju metriku koja definiše količinu posla koju softver mora izvršiti u određenom vremenskom periodu. U zavisnosti od područja primjene, performanse imaju različit značaj, npr.:
  - vrijeme prijema i obrade signala u robotici od izuzetne važnosti, te mala kašnjenja mogu dovesti do značajnih nepravilnosti u radu
  - aplikacije u bankarskom ili telekomunikacijskom sektoru obrađuju na hiljade transakcija/sekundi, pa su bilo kakva kašnjenja itekako osjetna
- Kvalitet performanse se može izmjeriti sljedećim karakteristikama:
  - Propusnost – količina posla koju softver mora izvršiti u jedinici vremena (broj obrađenih narudžbi, broj bankovnih transakcija i sl.). Prilikom određivanja propusnosti je potrebno voditi računa o odabiru momenta u kojem definisana propusnost treba biti zadovoljena jer postoje vremenski periodu u kojima je određeni softver poprilično opterećen
  - Odziv – vrijeme koje je softveru potrebno da kreira rezultat ili odgovor na određeni zahtjev. Manje vrijeme odziva kreira jedinstven osjećaj i pospješuje efikasnost rada

# Skalabilnost



- Pojam skalabilnost se odnosi na karakteristiku softvera da adekvatno odgovori zahtjevima koji su u određenim momentima veći od očekivanih, a neki od primjera su:
  - Preopterećenje zahtjevima – novi hardverski resursi bi u značajnoj mjeri trebali poboljšati rad softvera u slučaju povećanog opterećenja, a da tom prilikom nije potrebno vršiti bilo kakve promjene u arhitekturi softvera
  - Povećanje količine podataka – odnosi se na ponašanje softvera u momentima kada se količina podataka značajnije poveća, a potrebno je izvršiti njihovu kompleksniju pretragu ili obradu



# Nadogradnja



- Atribut nadogradnje ili proširivosti podrazumijeva mjeru kompleksnosti sa kojom se susreću oni koji su zaduženi za implementaciju novih (ne)funkcionalnih zahtjeva
- U zavisnosti od kompleksnosti projekta, arhitekture softvera, korištenih standarda, te kvaliteta implementacije, nadogradnja može postati izuzetno zahtjevna i skupa aktivnost. Najbolji mehanizam koji osigurava pojednostavljenje buduće nadogradnje svakako je iskustvo i adekvatna procjena na početku projekta
- Prekomjerni inženjering, odnosno nepotrebno raslojavanje projekta (kreiranje robusnih arhitektura projekata) u cilju pripreme za neočekivane promjene svakako može dovesti do nepotrebnog gubljenja vremena i novca. Potrebno je procijeniti koliko fleksibilnosti je dovoljno za trenutne i realne buduće zahtjeve

# Indeks održavanja



- Indeks održavanja predstavlja softversku metriku čija vrijednost predstavlja stepen jednostavnosti održavanja i izmjene izvornog koda, a izračunava se prema sljedećoj formuli:

$$\textbf{Maintainability Index} = \text{MAX}(0, (171 - 5.2 * \log(\text{Halstead Volume}) - 0.23 * (\text{Cyclomatic Complexity}) - 16.2 * \log(\text{Lines of Code})) * 100 / 171)$$

- Halstead-ova vrijednost predstavlja mjeru složenosti programskog modula, sa naglaskom na kalkulativnu/računsku složenost
- Cyclomatic Complexity se dobije mjerenjem broja linearno nezavisnih putanja kroz određeni dio izvornog koda. Što je njihov broj veći, izvorni kod je složeniji



# Sigurnost



- Pojam sigurnosti je izuzetno kompleksan i zahtijeva implementaciju na više nivoa. Neki od najčešćih zahtjeva koje se odnose na sigurnost su:
  - Autentifikacija – verifikacija korisničkog identiteta
  - Autorizacija – provjera privilegija korisnika za pristup određenom dijelu softvera ili resursu
  - Enkripcija – zaštita podataka u svakoj fazi njihovog korištenja
  - Integritet – implementacija mehanizama koji osiguravaju da podaci ili poruke ne mogu biti modificovani od strane malicioznih korisnika



# Dostupnost

- Karakteristika dostupnosti je direktno vezana za pouzdanost softvera, te je jedna od karakteristika koja je lako mjerljiva
- Većina svremenih softvera teži ka stalnoj (100%) dostupnosti, što nije jednostavno ispuniti
- Upravo zbog toga se nastoje eliminisati oni dijelovi koji bi mogli uzrokovati privremenu ili eventualno trajnu nedostupnost
- Vrijeme oporavka je u direktnoj vezi sa dostupnošću jer i u slučaju zastoja u radu određenih dijelova softvera izuzetno je važno za koliko vremena će oni ponovo postati operativni

---

# Integracija



- Karakteristika integracije se odnosi na mogućnost korištenja podataka kojima manipuliše određeni softver u mnogo širem kontekstu, bez značajnijih modifikacija na arhitekturi
- U novije vrijeme, implementacija API-a predstavlja jedan od najefikasniji načina da se podaci učine dostupni drugim akterima poslovnog procesa
- U zavisnosti od potreba i poslovnih relacija, pored API-a, podaci se drugim akterima mogu učiniti dostupnim i integracijom podataka. Pomenuto je moguće postići omogućavanjem direktnog pristupa određenom dijelu ili replici baze podataka (uz adekvatne kontrolne mehanizme), a također i eksportovanjem potrebnih podataka u odgovarajući format (XML, CSV i sl.)

---

KRAJ PREZENTACIJE

