

## **1. Šta opisuje dokumentacija zahteva?**

Dokumentacija zahteva opisuje zašto je program potreban i stavlja ga u kontekst koji opisuje kako može pomoći korisniku te kako i što će konačan proizvod raditi.

## **2. Navedite neki primer šta treba da sadrže sistemski zahtevi?**

Sistemski zahtevi izriču ograničenja ili uslove rada samog sistema. Evo šta na primer sistemski zahtev može da sadrži:

- Arhitektura
- Snaga procesora
- Memorija
- Periferni uređaji
- Platforma
- API i upravljački programi
- Internet pretraživač

## **3. Navesti neka pravila prilikom pisanja korisničke dokumentacije.**

Korisnička dokumentacija ne sadrži objašnjenja programskog koda niti opis samog dizajna sistema. Kako korisnička dokumentacija nije namenjena informatičkom osoblju važno je izbegavati tehnički žargon i složene sheme i skice sistema. Ukoliko je programsko rešenje vrlo složeno poželjno je koristiti slike prilikom opisivanja pojedinih koraka. Za ovakve dokumente nije presudno da budu organizovani po nekoj strogoj shemi, ali je ipak važno proizvesti pregledan dokument te teme objasniti na jednostavan način. Isto kao i tehnička dokumentacija, bitno je da korisnička dokumentacija bude uvek ažurirana i sinhronizovana s programskom podrškom.

## **4. Zašto je veoma korisno koristiti alate za upravljanje zahtevima?**

Dokumentovanje programskih rešenja predstavlja velik zadatak koji sadrži mnogo ručnog rada, a to se u projektima često preskače jer oduzima previše vremena. Kako bi se stvaranje i održavanje dokumentacije olakšalo razvijeni su brojni alati koji pomažu u stvaranju pojedinih dokumentacija.

## **5. Koji alat se koristi za izradu dijagrama, navesti neke podržane tipove dijagrama?**

Microsoft Visio se koristi za izradu različitih dijagrama. Podržani su razni tipovi dijagrama i moguće ih je pohraniti u obliku slike (npr. PNG, JPEG).

Neki od podržanih tipova dijagrama za dokumentovanje programskih rešenja su:

- Dijagrami baza podataka
- UML dijagrami
- Dijagram toka podataka
- Mrežni dijagrami

## **6. Koje su faze u razvoju programske dokumentacije?**

U ranim fazama razvoja računarstva programska podrška je bila vezana za računar na kojem je i napravljena. Razvojem modernog računarstva nastale su platforme pomoću kojih se programska podrška mogla prenositi s jednog računara na drugi. Ovakav preokret pokrenuo je naglu informatizaciju i integraciju računara u gotovo sve grane današnjeg modernog društva.

Dokumentacija programske podrške u ranim fazama razvoja računarstva se uglavnom zasnivala na korisničkim uputstvima.

Druga faza razvoja dokumentacije nastala je zajedno sa zajednicom otvorenog koda.

Konačna faza razvoja je nastala korištenjem službenih dokumentacija u komercijalnim programskim proizvodima.

## **7. Navedite neki primer šta treba da sadrže korisnički zahtevi?**

Korisnički zahtevi definišu šta konkretno sistem treba da radi i kako će pomoći krajnjim korisnicima prilikom rada.

Korisnički zahtevi su:

- Funkcionalni zahtevi - ono što sistem treba da radi.
- Nefunkcionalni zahtevi - ograničenja za vrste rešenja koja će udovoljiti funkcionalnim zahtevima.
- Ciljevi dizajna – Na primer za evidenciju korisnika, glavni cilj dizajna mogao bi biti da korisnici mogu lako da pronađu informacije o klijentima.

## **8. Kome je namenjena korisnička dokumentacija i šta je njen cilj?**

Za razliku od tehničke dokumentacije, korisnička dokumentacija je namenjena krajnjim korisnicima sistema. Njen cilj je na jednostavan način objasniti kako koristiti pojedine funkcionalnosti programske podrške.

## **9. Šta olakšava pisanje tehničke dokumentacije kod programskih rešenja i zašto?**

Kako bi se stvaranje i održavanje dokumentacije olakšalo razvijeni su brojni alati koji pomažu u stvaranju pojedinih dokumentacija

**10. Šta je Doxygen i kako se u njemu vrši dokumentovanje?**

Doxygen je alat za proizvodnju tehničke dokumentacije za programske jezike C/C++, C#, Java, PHP, Python i druge. Dokumentovanje se zasniva na pisanju posebno označenih komentara u izvornom kodu. Na temelju komentara se proizvodi dokumentacija u HTML, PDF ili LaTeX obliku koji je moguće uključiti u tehničkoj dokumentaciji kao opis funkcija.

**11. Da li je zaista neophodno uvek pisati tehničku dokumentaciju programskih rešenja ili je moguće bez nje?**

U većini današnjih projekata zahtevi se brzo menjaju i potrebno je brzo reagovati kako projekat ne bi ispaštao na kvalitetu. Ovakvo dinamičko okruženje često uzrokuje zanemarivanje dokumentacije. Dokumentacija programskog projekta ima višestruke uloge. Na primer, može služiti kao komunikacijski medij između članova razvojnog tima ili kao priručnik koji će korisnici moći da koriste prilikom rada i administracije sistema. Takođe, može poslužiti kao jedinstvena zbirka informacija razvojnim inženjerima koji održavaju sistem. Često se koristi kao izvor informacija menadžmentu koji će na temelju tih informacija donositi odluke vezane uz budžet, poslovni plan i daljnju razradu rasporeda razvoja projekta.

**12. Šta se opisuje sistemskim zahtevima?**

Sistemskim zahtevima se opisuju tehnička svojstva sistema tehničke dokumentacije.

**13. Tehnička dokumentacija se deli na?**

Tehnička dokumentacija se deli na dokumentaciju zahteva, tehničku dokumentaciju i korisničku dokumentaciju.

**14. Kako se dobija dokumentacija izvornog koda?**

Razvijeni su razni alati koji se zasnivaju na posebno oblikovanim komentarima u izvornom kodu. Na temelju komentara se proizvodi dokumentacija u HTML, PDF, LaTeX ili XML (u zavisnosti od alata).

**15. Koji su glavni motivi za izradu kvalitetne tehničke dokumentacije?**

Korisnici kvalitetne tehničke dokumentacije nadilazi potrebe novih članova razvojnog tima. Prilikom implementacije velikih i složenih sistema rešavaju se razni problemi i prepreke. Kada ne bi postojala kvalitetna dokumentacija bilo bi nezamislivo shvatati i prisećati se posebnosti pojedinih rešenih problema.

**16. Koja je uloga dokumentacije kod programskih projekata?**

Velika i složena programska rešenja vremenom postala su zahtevna za održavanje. Dodatno, složena programska rešenja postala su zavisna od programera koji su ih razvili jer je obuka novih programera predstavljala velik trošak. Naime, osim znanja o svim korišćenim tehnologijama i alatima, novi programeri morali su razumeti način razmišljanja programera koji je razvio program. Bez kvalitetne programske dokumentacije novi programeri nisu mogli održavati postojeći projekat i nadograđivati ga.

**17. Šta definišu korisnički zahtevi?**

Korisnički zahtevi definišu šta konkretno sistem treba da radi i kako će pomoći krajnjim korisnicima prilikom rada.

**18. Šta predstavlja tehnička dokumentacija?**

Tehnička dokumentacija predstavlja jedinstveni jezik komunikacije između onoga ko razvija programsku podršku i ostalih stručnjaka koji imaju interes za projektom.

**19. Koji načini se koriste prilikom dokumentovanja baza podataka?**

Dokumentacija baze podataka se ističe dijagramima koji opisuju pojedine tablice i njihove zavisnosti i opis pojedinih atributa.

**20. Navesti jedan alat za upravljanje zahtevima i ukratko ga opisati?**

Rational DOORS je komercijalni alat organizacije IBM za prikupljanje i održavanje zahteva. Stvoren je s ciljem olakšavanja procesa upravljanja zahtevima prilikom razvoja programske podrške.

Alat pruža web interfejs putem kojeg je moguće pratiti razvoj zahteva. Svaku izmenu zahteva moguće je napraviti pomoću jednostavnog korisničkog interfejsa ili putem složenijeg interfejsa koji omogućuje izmenu dodatnih opcija. Svakom članu je moguće ograničiti nivo pristupa kako bi se osigurao visok nivo sigurnosti i ispravnosti. Čitav sistem je dostupan preko web interfejsa koji olakšava pristup sistemu članovima razvojnog tima.