

•
•
•
Kateri pristop izbrati?

Kateri pristop (oziroma kombinacijo) izberemo?

Tradicionalni pristop (življenjski cikel):

- **Prednosti:**

- zahteva (prisili) da razvijalci in (uporabniki) dosledno sledijo korakom, ki jih predpisuje uporabljena metodologija
- razvite aplikacije (bolje) ustrezajo standardom za kvaliteto kot pri drugih pristopih
- maj verjetno je, da izpustimo pomembne zadeve pri analizi zahtev

Kateri pristop izbrati?

- **Problemi:**

- Generira zelo obsežno dokumentacijo v kateri se še posebej uporabniki “zgubijo”
- Uporabniki pogosto niso pripravljeni (in tudi ne sposobni) preučiti specifikacije, ki jih morajo odobriti
- Predolgo časa mine od ideje za aplikacijo do delujoče aplikacije
- Uporabniki pogosto ne znajo in ne morejo dovolj precizno opisati svojih zahtev za aplikacijo

Kateri pristop izbrati?

Prototipni pristop:

- **Prednosti:**
 - Pomaga razjasniti zahteve uporabnikov preden vložimo veliko časa in navora v razvoj aplikacije
 - Pomaga preveriti ustreznost zasnove aplikacije
 - Pomaga motivirati uporabnike za sodelovanje pri razvoju in premagati odpore do aplikacije (računalnika)
 - Lahko na enostaven način razvijemo (običajno) del končen aplikacije

Kateri pristop izbrati?

- **Problemi:**

- sistemska analiza se lahko obrne v napačno smer (k manj bistvenim zahtevam)
- uporabnik včasih noče opustiti prototipa, kar zavira razvoj končne aplikacije
- je lahko zelo zahteven za razvijalce (programerje)
- včasih se je težko odločiti kdaj je prototip zaključen

• Kateri pristop izbrati?
•
•

Računalništvo uporabnikov

- **Prednosti**

- izognemo se formalnemu postopku in s tem povezanim zakasnitvam
- uporabnik ima popoln nadzor nad aplikacijo ter jo lahko razvija in spreminja glede na svoje potrebe
- uporabnik najbolje pozna probleme in priložnosti na svojem strokovnem področju
- ni potrebno uvajanje aplikacije in usposabljanje uporabnikov za delo z aplikacijo

Kateri pristop izbrati?

- **Problemi:**

- ker so uporabniki “amaterji” pri razvoju aplikacij, so pogosto aplikacije slabe kvalitete (slabo testirane, nerodne zasnove, neustrezne podatkovne strukture, nestrukturirana koda, ...)
- uporabnik po nekem času pogosto ni več sposoben vzdrževati aplikacije
- pomanjkljiva zaščita in varnost podatkov
- nastanek izoliranih in “privatnih” baz podatkov in informacijskih sistemov

⋮
Kateri pristop izbrati?
⋮

Kombiniranje pristopov

Opisane pristope in tehnike pogosto kombiniramo.

Nekateri najbolj pogosti načini:

- v okviru tradicionalnega pristopa uporabimo prototip za specifikacijo funkcionalnosti
- uporabimo enostaven in poceni programski paket kot prototip, na osnovu katerega opredelimo zahteve za celovito rešitev (aplikacijo)
- tradicionalni pristop in računalništvo uporabnikov: jedro aplikacije razvijejo informatiki po tradicionalnem pristopi, uporabniki razvijejo zanje specifične aplikacije