

•
•
•
•
•
•
•
•
•
•
•
•

RAZVOJ PROGRAMSKIH APLIKACIJ (RPA)

•
•



Predavatelj: doc. dr. Matej Gomboši

matej.gombosi@uni-mb.si

•
•
• Poglavitni poudarek pri predmetu: **Metode, pristop k razvoju (enostavnih) aplikacij**

Uporabniška rešitev (aplikacija):

- računalniška podpora izbranemu poslovnemu postopku ali poslovni funkciji,
- lahko vključuje tudi neavtomatizirane funkcije
- razvoj in uvajanje aplikacije običajno zahteva tudi organizacijske spremembe



Življenjski cikel razvoja aplikacije (*System Development Life Cycle - SDLC*)

- Uporabniške rešitve načrtujemo in razvijamo na nek sistematičen način.
- “Klasični” pristop: življenjski cikel razvoja, ki je močno vplival na oblikovanju metod razvoja informacijskih sistemov

http://en.wikipedia.org/wiki/Systems_Development_Life_Cycle



•
•
•

- ***Življenjski cikel razvoja aplikacije (SDLC):***
razčlenitev razvojnega procesa na zaporedne korake ali faze; preden se lahko začne naslednja, mora biti predhodna zaključena

•
•
•

Življenjski cikel razvoja aplikacije (*System Development Life Cycle - SDLC*)

- Razčlenjen na različne načine, odvisno od avtorja (knjige), oziroma internih standardov v posameznih okoljih, pa tudi z vidika posameznih funkcij in vlog v razvoju. Faze:

1. **Analiza problema**

- Precizno definiramo problem oz. področje, ki ga pokriva aplikacija, napišemo programske specifikacije: opis vhodov, obdelave, izhodov in uporabniških vmesnikov

⋮

2. **Zasnova programa**

Izdelamo podrobno podatkovno strukturo (podatkovni model), uporabniške vmesnike in logiko programov z uporabo orodij kot so entitetni diagrami, akcijski diagrami, psevdo koda, diagrami strukture objektov, diagrami tokov podatkov, ... in razčlenimo funkcije programa v module, algoritme za obdelavo podatkov, ...

3. **Kodiranje**

“Pretvorimo” zasnovo v delujočo aplikacijo: v izbranem programskem jeziku ali orodju izdelamo vmesnike in programsko kodo, vključno z interno dokumentacijo: komentarji, ki razložijo pomen in delovanje posameznih ukazov in segmentov programa

⋮

-
-
-

4. **Testiranje in popravljanje napak**

S testiranjem najdi čim več napak

5. **Dokumentiranje aplikacije**

Preveri in popravi in integriraj interno dokumentacijo, izdelaj uporabnikovo dokumentacijo

6. **Vzdrževanje**

Omogoči usposabljanje in podporo končnim uporabnikom; popravljalj napake, ki se pokažejo med uporabo aplikacije, prilagaja program spremenjenim oz. novim zahtevam uporabnikov