

# Različni modeli procesov razvoja PO (1/2)

- Z modelom procesa razvoja PO se v razvoj PO vnese določena struktura, stabilnost, kontrola, organizacija.
- Model procesa definira aktivnost, akcije, delovne naloge in njihove rezultate (programi, podatkovne strukture, dokumenti)
- Modeli se lahko prilagodijo specifičnosti posameznega projekta
- Proces vodi ekipo programerjev skozi aktivnosti, ki so organizirane v tok (linearen, inkrementalen, evolucijski)

# Različni modeli procesov razvoja PO (2/2)

Generičen proces razvoja PO: **komunikacija, planiranje, modeliranje, razvoj, inštalacija**

Procesi se razlikujejo po:

- aktivnostih
- koliko pozornosti namenimo posameznim aktivnostim
- toku procesa

Skupine:

- Klasični model
- Iterativni modeli
- Evolucijski modeli
- Posebni modeli
- Agilni modeli



# Klasični model razvoja PO

Komunikacija → planiranje → modeliranje → razvoj → inštalacija

- linearen potek
- primeren za reševanje zelo dobro definiranih problemov
- Težave:
  - Realni projekti zelo redko sledijo linearnemu poteku
  - Naročniki so zelo redko sposobni eksplicitno navesti svoje zahteve
  - Naročniki morajo biti potrpežljivi, saj rezultati niso vidni do pozne faze projekta
  - Lahko pride do blokirane stanja

# Inkrementalni model razvoja PO

Komunikacija → planiranje → modeliranje → razvoj → inštalacija → prva verzija

Komunikacija → planiranje → modeliranje → razvoj → inštalacija → druga verzija

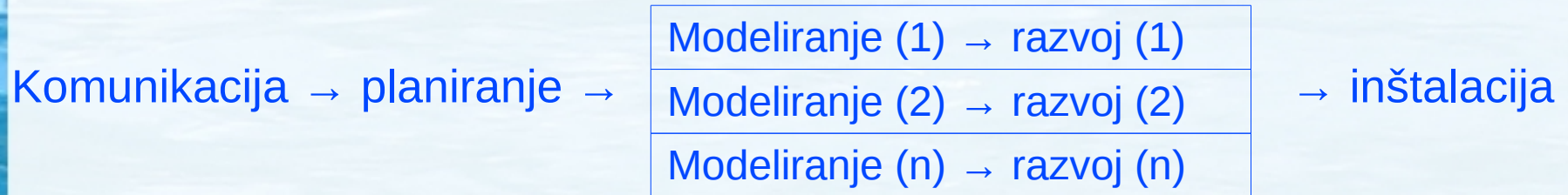
...

Komunikacija → planiranje → modeliranje → razvoj → inštalacija → končna verzija

- Gre za klasični model razvoja PO, ki je definiran na iterativen način
- Prva verzija: ključna funkcionalnost!
- Na osnovi uporabniških izkušenj s prvo verzijo → plan za drugo verzijo, itd.
- Lažje dodajanje programerjev na projekt, večja kontrola nad tveganji
- Težave: ponavljanje aktivnosti



# Hitri model razvoja PO (RAD)



- Max 60 – 90 dni za modeliranje in razvoj
- Predpogoj: dobro razumevanje zahtev uporabnika in skalabilnost projekta
- Planiranje je ključna aktivnost
- Modeliranje: operacije, pod. strukture, procesi
- Razvoj: ponovna uporaba komponent, avtomatsko generiranje kode
- Težave:
  - zahteva dovolj programerjev
  - zahteva zelo motivirane programerje
  - zahteva modularizacijo projekta
  - ni primeren za projekte z velikimi tehničnimi tveganji

## Razvoj PO s prototipi (1/2)

- Komunikacija → planiranje → modeliranje → razvoj prototipa → inštalacija -> povratne informacije od uporabnika → planiranje → modeliranje → razvoj → inštalacija
- Uporaba:
  - znane so osnovne zahteve, ne pa vsi detajli (vhodni podatki, izhodni podatki, podrobnosti glede procesiranja podatkov, itd)
  - programerji niso gotovi glede uporabe algoritmov, glede primernosti sistema, glede interakcije med človekom in programom, itd.
- Samostojen proces ali del drugega procesa
- Razvoj prototipa mora biti hiter
- Prvi prototip (GUI ali ključna funkcionalnost) → pomoč pri natančnejšem opredeljevanju zahtev



## Razvoj PO s prototipi (2/2)

Prototip se lahko zavrže ali razvije v končni produkt

Težave:

- Prevelika pričakovanja naročnikov
- Neoptimalna implementacija lahko ostane v kodi tudi po evoluciji prototipa v delujočo aplikacijo

Pravila igre je treba določiti takoj na začetku!

# Spiralni model razvoja PO

- Komunikacija → planiranje → modeliranje → razvoj → inštalacija → nazaj na prvo aktivnost
- Vsak prehod preko faze planiranja lahko prinese izboljšavo načrta
- Rezultat razvoja je lahko različen glede na iteracijo:
  - Izboljšan koncept
  - Prototip
  - Delujoč program z osnovno funkcionalnostjo
  - Delujoč program z dodatnimi funkcionalnostmi
- Primeren za razvoj velikih in zahtevnih projektov, kjer se zahteve s časom spreminjajo
- Težave: zahteva dosledno upravljanje s tveganji, naročnike je težko prepričati v ta pristop



# Sočasni model razvoja PO

- Vsaka aktivnost, akcija in delovna naloga procesa je lahko v enem izmed stanj: se ne izvaja, v teku, čakanje na spremembe, revizija, pod drobnogledom, ustavljena, opravljena.
- Dogodki v procesu razvoja vplivajo na spremembo stanja posameznih enot procesa
- Natačna slika stanja celotnega procesa
- Primeren za projekte, kjer sodeluje več skupin ljudi

# Komponentni model razvoja PO

- Posebni modeli imajo mnoge karakteristike konvencionalnih modelov razvoja PO
- Specifični projekti (komponente za ponovno uporabo)
- Teži se k razvoju aplikacije z že narejenimi komponentami
- Prva akcija v modeliranju in razvoju je iskanje in evaluacija primernih komponent
- Integracija komponent in testiranje



# Razvoj PO s formalnimi metodami

- Skupek aktivnosti, ki programsko opremo definira v smislu formalnih matematičnih specifikacij
- S pomočjo matematične analize se lahko hitro odkrije napake v načrtovanju
- Težave: razvoj formalnih specifikacij je časovno zahteven, malo programerjev je usposobljenih za tak razvoj, uporabniki težko izrazijo svoje zahteve na tak način
- Uporablja se za razvoj projektov, kjer ima vsaka napaka hude posledice:
  - PO v pilotski kabini letal
  - PO v različnih medicinskih inštrumentih

# Razvoj PO z miniaturnimi mejniki

- Za dokončanje delovne naloge največ 1-2 dni (miniaturni mejnik)
- Vsak mejnik označimo z 1 (opravljen) ali 0 (ni v celoti opravljen)
- Omogoča večji nadzor nad tokom projekta
- Odpravlja nenadzorovane zamude projekta
- Dajejo občutek stalnega napredka
- Primeren pristop za projekte:
  - ki so zašli v časovno stisko,
  - za katere je težko oceniti, koliko časa bodo trajali



# Agilni modeli razvoja PO

**Agilnost je sposobnost hitre reakcije na spremembe!**

Osnovne predpostavke:

- Ljudje in komunikacija med njimi > procesi in tehnologija
- Delujoča PO > izčrpna dokumentacija
- Sodelovanje z uporabniki > pogajanja glede pogodb
- Reakcija na spremembe zahtev > sledenje načrtu

# Agilni modeli razvoja PO

Osnovni principi:

- 1) Zadovoljstvo naročnikov in uporabnikov ima najvišjo prioriteto
- 2) Ne brani se sprememb zahtev, tudi če do njih pride pozno v razvoju
- 3) Uporabnikom pogosto dostavi delovne verzije programov (na vsakih nekaj tednov/mesecev)
- 4) Naročniki in razvijalci naj sodelujejo na dnevni bazi
- 5) Projekte dodeli motiviranim razvijalcem, daj jim na voljo okolje in podporo, ki jo potrebujejo ter jim zaupaj
- 6) Najbolj učinkovita metoda pridobivanja informacij znotraj razvijalske ekipe je odkrit pogovor iz oči v oči
- 7) Delujoča PO je osnovno merilo uspeha
- 8) Tempo razvoja mora biti tak, da ga lahko razvijalci, naročniki in uporabniki zdržijo poljubno časa



# Agilni modeli razvoja PO

Osnovni principi:

- 9) Nenehno je treba kontrolirati kvaliteto napisane kode; dober načrt povečuje agilnost
- 10) Enostavnost – umetnost maksimizacije neopranavljenega dela – je ključnega pomena
- 11) Najboljše arhitekture, načrti in zahteve izhajajo iz razvijalskih ekip, ki so sposobne samoorganizacije
- 12) V regularnih intervalih naj se razvijalske ekipe posvetujejo, kako bi postale bolj učinkovite in naj svoje delo temu prilagodijo

# Agilni modeli razvoja PO

Značilnosti:

- zgodnje in pogoste inštalacije delujoče programske opreme
- Majhne, zelo motivirane ekipe razvijalcev
- Neformalne metode
- Enake aktivnosti kot pri klasičnih modelih procesa, vendar z manj poudarka na planiranju in modeliranju in več na razvoju in komunikaciji
- Predstavlja resno alternativo običajnim modelom procesov razvoja PO zaradi sposobnosti prilagajanja spremembam
- Veliko komunikacije!