

# Ekstremno programiranje (XP)

- Objektno orientiran pristop
- Planiranje → načrtovanje → kodiranje → testiranje

Planiranje: množica uporabniških scenarijev (user stories) → prioriteta, čas razvoja

Načrtovanje: enostavni načrti, prototipi

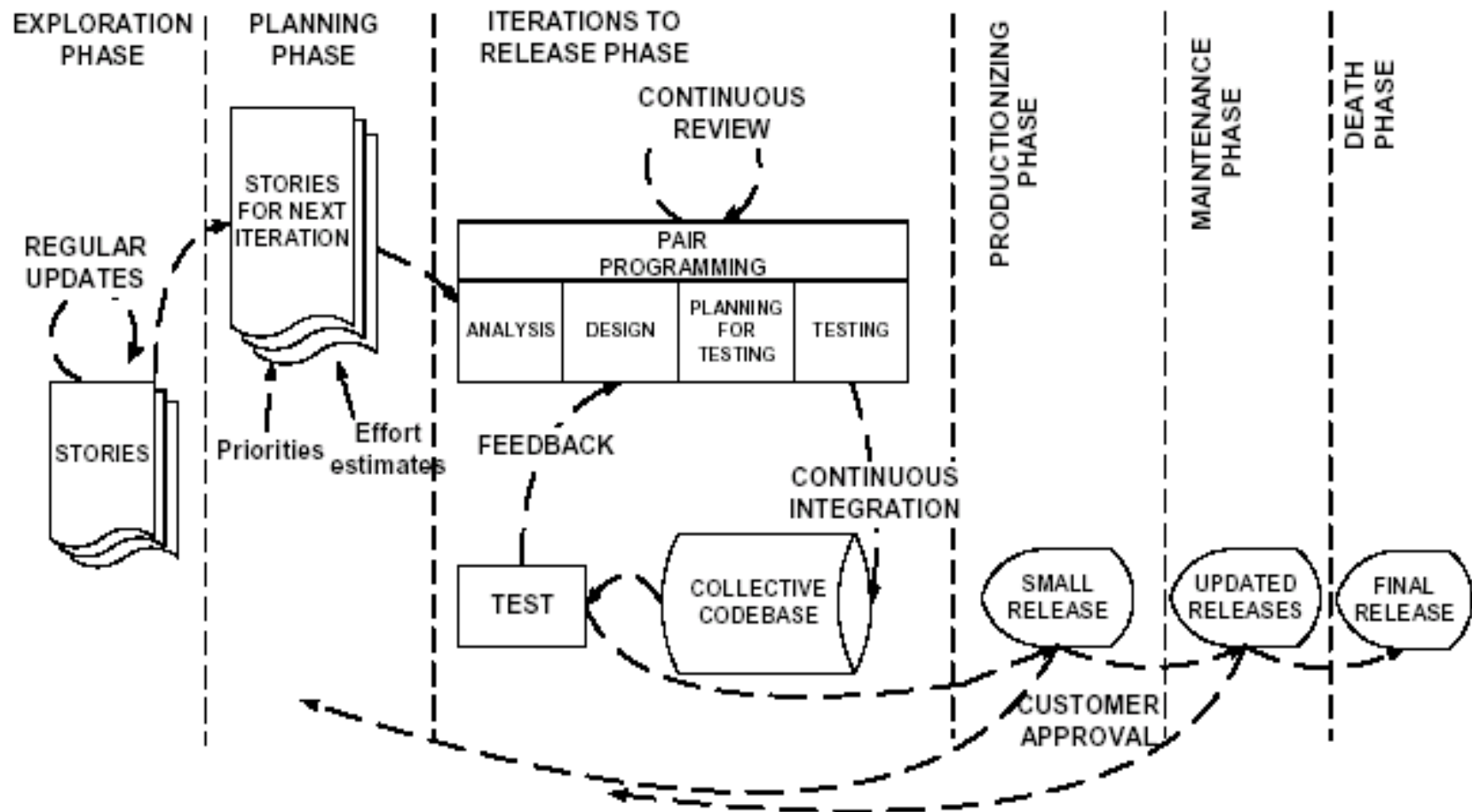
Kodiranje: **unit testi**, programiranje v parih, integracija, refactoring

Testiranje: izvajanje unit testov, testi sprejemljivosti

Nova verzija PO po vsakem uspešnem testiranju

Ocena hitrosti razvoja po 1. verziji

# Življenski cikel XP



# Uporabniški scenariji

- Uporabniški primeri, ki kratko opišejo zahteve za sistem z naravnim jezikom.
- Berljive za naročnika in razvijalca
- Ni nujno da so popolne, obljuba za nadaljnje pogovore.

Primeri:

Sistem izpiše seznam opravljenih del in ur za izbrani projekt po posameznih mesecih in razvijalcih.

Unit test?



# Unit testi

- Namenjeni odkrivanju napak v posameznih metodah
- Unit testi za trivialne metode (getter/setter metode, ipd)?
- Sestava: inicializacija, izvajanje metode/evaluacija rezultatov -> uspešno/neuspešno
- Sekvenčenje testov, kjer je to smiselno!
- Priporočila:
  - Ko se test zaključi neuspešno, takoj najdi in popravi napako
  - Skrbno hrani vse unit teste
  - Koda naj bo napisana tako, da je enostavno napisati unit test zanjo
  - Predelaj kodo tako, da bo bolj primerna za unit teste

<http://www.testingfaqs.org/t-unit.html>

JUnit – platforma za razvoj avtomatiziranih unit testov v Javi

# Metode dinamičnih sistemov

- Iterativni postopek
- Študija osnovnih zahtev → študija poslovne logike → funkcionalno modeliranje → načrtovanje in razvoj → implementacija

Študija osnovnih zahtev: primernost projekta

Študija poslovne logike: Funkcionalne zahteve, osnovna arhitektura

Funkcionalno modeliranje: množica evolucijskih prototipov

Načrtovanje in razvoj: revizija prototipov, razvoj prototipov

Implementacija: postavitve zadnje verzije v produkcijsko okolje

Možno kombiniranje z XP.

# Scrum

- Zbiranje zahtev → analiza → načrtovanje → evolucija → inštalacija
- Majhne razvojne skupine (5-9 ljudi)
- Šprinti (seznam delovnih nalog za posamezno iteracijo) se lahko kreirajo in spreminjajo po potrebi (do 30 dni dela)
- Poudarek na vzorcih, ki so se izkazali za uspešne:
  - Dnevnik zahtev s prioritetami
  - Šprinti: delovne naloge za posamezne zahteve
  - Sestanki članov ekipe: kaj si naredil od zadnjega sestanka? Na kake težave si naletel? Kaj planiraš narediti do naslednjega sestanka
  - Demo verzije: dodane funkcionalnosti



# Funkcionalno orientiran razvoj (FDD)

- Funkcionalnost: funkcija, pomembna za naročnika, ki jo je mogoče implementirati v manj kot dveh tednih
- Naročniki jih lažje opišejo, lahko jih organiziramo hierarhično
- Funkcionalnost predstavlja izboljšavo PO, zato lahko novo verzijo lansiramo vsake dva tedna
- Planiranje, načrtovanje, spremljanje razvoja je pogojeno s funkcionalnostmi, ne pa z delovnimi nalogami
- Definicija funkcionalnosti: <akcija> <rezultat> <za | od | v | s strani> <objekt>
- Grupiranje funkcionalnosti: <akcija> <objekt>
- Izdelava splošnega modela → seznam funkcionalnosti → planiranje na osnovi funkcionalnosti → načrtovanje na osnovi funkcionalnosti → razvoj na osnovi funkcionalnost

## Faktorji razvoja PO

- Ljudje (motivacija, timsko delo, kadrovanje)
- Procesi
- Produkti
- Tehnologija (orodja)



# Klasične napake (ljudje)

- Pomanjkanje motivacije
- Zaposlovanje “problematičnih” ljudi
- Dodajanje ljudi na projekt, ki je v zamudi
- Glasne, prenapolnjene pisarne
- Nesoglasja med razvojniki in naročniki
- Nerealistična pričakovanja
- Pomanjkanje sodelovanja z uporabniki

# Klasične napake (proces)

- Preoptimistični načrti
- Ne zadostno načrtovanje
- Opuščanje načrtovanja pod pritiski
- Opuščanje zagotavljanja kvalitete
- Ne zadosten nadzor
- “code-like-hell” programiranje

## Klasične napake (produkt)

- Več funkcionalnosti, kot je potrebno
- Spreminjanje funkcionalnosti med projektom (več kot 25%)
- Raziskovalno usmerjen razvoj



# Klasične napake (tehnologija)

- Preveč zanašanja na najnovejše produkte
- Zamenjava orodij med projektom
- Pomanjkanje avtomatske kontrole verzij