

# Osnove programskog inženjerstva. Životni ciklus razvoja programske podrške.

2017/18.01

- Programska oprema/podrška/potpora, softver (engl. *software*)
  - dio računalnog sustava koji nema fizikalnih dimenzija
  - opći pojam za sve vrste programa, programskih jezika itd
  - skup elemenata ili objekata u jedinstvenoj “konfiguraciji” koju čine računalni programi + podaci + dokumentacija
  - svojstva:
    - složenost, podložnost pogreškama,
    - ne troši se, teško mjerljiv,
    - stari, dugo se koristi,
    - lako se kopira (zajedno s pogreškama)
- Primijenjena programska potpora = Računalna aplikacija (engl. *application*)
  - namjenski program, primjenska programska oprema
  - računalom podržano rješenje jednog ili više poslovnih problema ili potreba
- Informacijski sustav = sustav aplikacija za upravljanje ljudskim aktivnostima

## ► Programsko inženjerstvo (software engineering)

- “Software Engineering: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is **the application of engineering to software**.

- ISO/IEC/IEEE 24765:2010 Systems and Software Engineering—Vocabulary

- **sistematičan, discipliniran i mjerljiv pristup razvoju, primjeni i održavanju softvera**

- **primjena inženjerskog pristupa na programsku opremu**

- Programsko inženjerstvo je inženjerska disciplina koja obuhvaća sve aspekte izrade programske opreme. [Sommerville, 2004]

## ► Područje programskog inženjerstva

- poslovi kojima se oblikuje i razvija programska oprema

- sustavna primjena prikladnih alata i tehnika na čitav proces razvoja programske potpore

# Programsko inženjerstvo i srodna područja

4

*“A scientist builds in order to learn; an engineer learns in order to build.” (Fred Brooks)*

## ➤ Steve McConnell: „*Software Engineering is not Computer Science*”

Professional Software Development (originalni isječak na  
[https://www.gamasutra.com/view/feature/131817/software\\_engineering\\_is\\_not\\_.php](https://www.gamasutra.com/view/feature/131817/software_engineering_is_not_.php))

## ➤ [Sommerville, 2004]

- računarska znanost fokusira se na teorijske osnove
- programsko inženjerstvo orijentirano na praktičnu primjenu u razvoju i isporuci programske potpore

## ➤ Petter J. Denning

*„Engineering has been marginalized by the unhealthy belief that engineering is the application of science” - Communications of the ACM, Vol. 60 No. 12, Pages 20-23*

# Vrste aplikacija

# Jednokorisničke, samostalne aplikacije

6

➡ „standalone”, početkom 90-ih, dBase, Clipper, ZIM, ...

The screenshot displays a multi-screen database application interface. At the top, a header bar contains the text "MR" on the left and "25.05.93" on the right. Below this, a "MAIN MENU" screen is visible, listing several options: "Grad", "Os", "Ze", "Maticni broj", "Prezime", "Ime", "Zemlja rodjenj", "Mjesto rodjenj", "Naziv grada", "Zemlja stanova", and "Mjesto stanova". To the right of the menu, a data entry screen is shown for "Grad". It contains fields for "Sifra zemlje" (set to "HR") and "Naziv zem" (set to "Zemlja"). Below these fields, a status bar indicates "1/1". Further down, another screen shows "Postanski Sifra zemlje : HR" and "Naziv zemlje : Republika Hrvatska.....". At the bottom, a screen displays "Naziv grada : Zagreb....." and "Adresa : Srednjaci bb.....". A footer bar at the very bottom lists function keys: "F5-Zoom F6-Window F10-Save ESC".

```
MR 25.05.93
```

```
MAIN MENU
Grad
Os
Ze
Maticni broj
Prezime
Ime
Zemlja rodjenj
Mjesto rodjenj
Naziv grada
Zemlja stanova
Mjesto stanova
```

```
Grad Add
Sifra zemlje : HR
Naziv zem Zemlja 1/1
Postanski Sifra zemlje : HR
Naziv zemlje : Republika Hrvatska.....
F5 ENTER F2-Add F3-Delete F4-Change F8-Browse F9
```

```
Naziv grada : Zagreb.....
Adresa : Srednjaci bb.....
Zemlja rodjenja : AAA
F5-Zoom F6-Window F10-Save ESC
```

# Poslužiteljske aplikacije

7

➡ serverske, 90-ih, Informix, Oracle, ...

| kfertalj (ansi) | PIS - PERSONALNI INFORMACIJSKI SUSTAV | (Sri) 04.12.96 |

OSOBA: [Dohv] Sljed Preth Unos Izmj Ostalo Lista Rasp Zap ...

Postavljanje uvjeta za dohvat zapisa

===== ( LALIĆ MARIJAN ) ===== [(2/3)]===== 120/261 =====

Sprema (71) (Visoka VII/1

Br.svjed. ( ) Dat. (

Zanimanje ( 54131 ) (ORGANIZATORI S

Stamb.stanje ( 3 ) (Privremeni korisnik

Krvna grupa ( )

Zdrav.broj ( )

Ured za obranu ( 0 ) (\*\*\*\*\*

Osobni UPD (33207) (SIN-VSS I S

Početak voj.roka ( )

Vojska voj.roka ( 0 ) (Nepoznata

Razl.prest.voj.roka (

Sudjel. u dom.ratu (D)

Broj dokumenta

Vrsta dokumenta BKA

Mj. troška/pri. 30

Konto 99

Iznos kuna

Dug/Pot D

Datum dokumenta 05

Analitika R

Program

Aktivnost

Pror. glava

Izvor financir.

Pozicija

Opis

Valuta / devize

Orig. broj

BANKOVNI IZVOD

Pregled : radnik

radnik

naziv

90125

AGATIĆ

(-

) NENAD

90737

AGIĆ

(-

) DARKO

00953

AGLIĆ ALJINOVIC

(ANDRIJA

) ANDREA

00914

ALEKSIĆ-MASLAĆ

(KREŠIMIR

) KARMELA

90431

ALIĆ

(-

) MLADEN

90617

AMBRUŠ

(-

) DAVORIN

90731

ANDRASSY

(-

) MLADEN

90037

ANDREIĆ

(-

) VALENTIN

00899

ANDRES

(LADISLAV

) DALIBOR

01300

ANĐELINIĆ

(MARKO

) MATKO

arhk-ff

traz fm

Akt( )

IZBOR

2

3

4TRAŽI

5

6

7NATRAG

8NAPRIJE9

0POMOĆ

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10



## Klijentske aplikacije

8

► “debeli” klijenti, kraj 90-ih, Microsoft Access, Visual Basic, Java, ...

[illegible]

**Vrsta: *Abies alba* Mill.**

---

Autor:

Datum: 
 Thumb: ☐

Tip: 
 Skanirano: ☒
 Javno: ☐

Objekt:

Tehnika:

Pohrana:


Referenca:

Godina:

Naslov:

Lokalitet:

Opaska:





# Mobilne i distribuirane aplikacije (1)

9

➤ Internet, džepne, “tanki” klijenti, *remoting*, 2000-, .NET, J2EE

**Subscribe** 1:14

Subscriber: Pretplatnik  
Server name: vozura  
Virtual directory: repldata3  
Publication: ggg  
Database: ggg  
Username: sa  
Password: \*\*\*\*\*  
Database Path: \\My Documents\\Gg

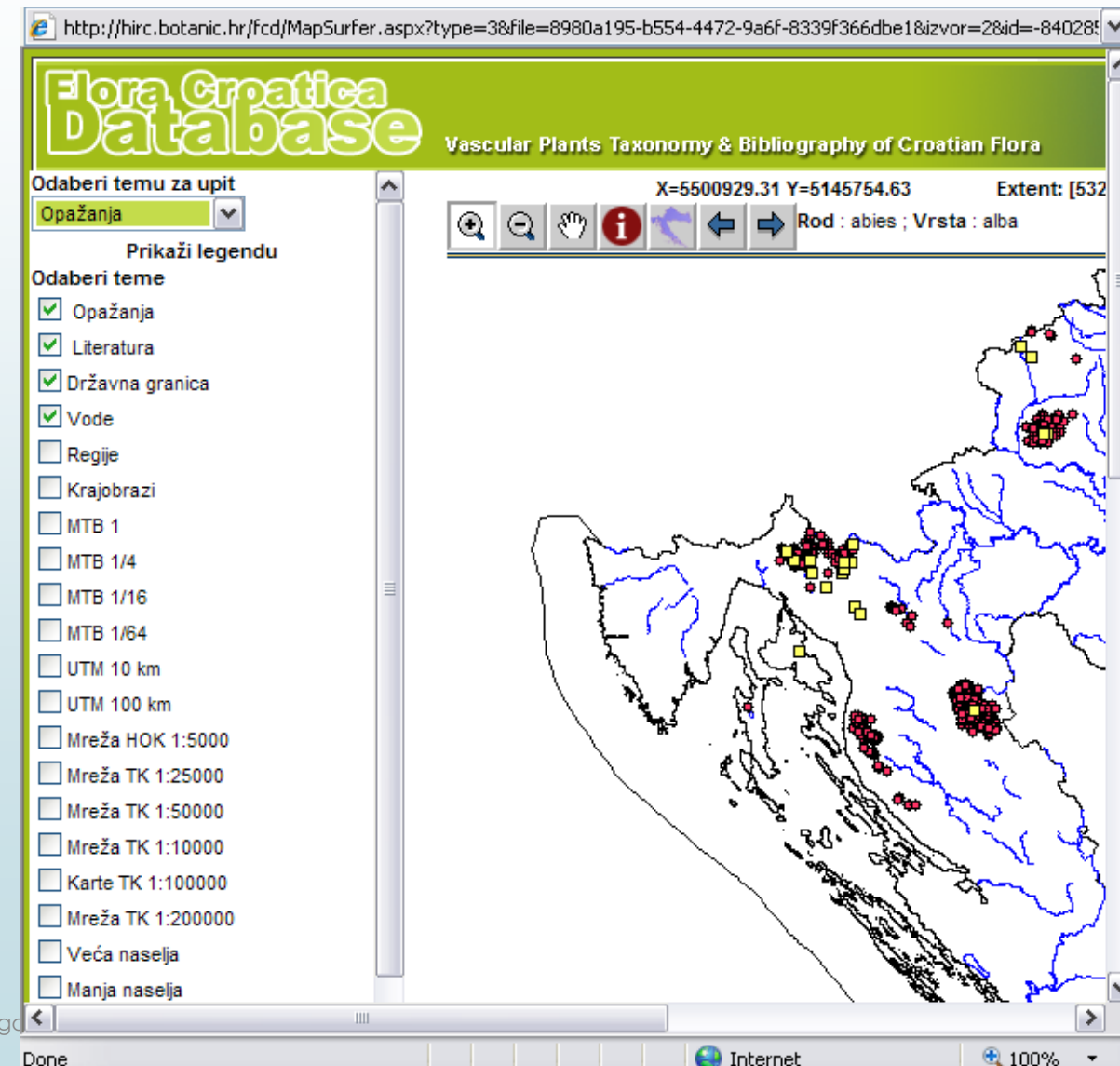
**Poslovi** 7:44

Osoba: ☐ ozura  
Datum: ☐ 7.8.06  
Projekt: ☐ !AktivanJavan  
Posao: ☐ backup

RbrPosa	Osoba	DatPosao	KratP
254662	anaj	21.2.04	[VER
254663	duje	14.2.04	111
254664	tošo	19.2.04	111
254665	anaj	19.2.04	111
254666	anaj	21.2.04	ACI
254667	duje	21.2.04	111
254668	tošo	12.2.04	111

Sati Sum 31975,65

View



# Mobilne i distribuirane aplikacije (2)

10

➡ Web servisi + mobilne, web aplikacije, jQuery

## Operations at <http://hirc.botanic.hr/services/Herbar.svc>

Count	<a href="#">GET</a>	Vraća ukupni broj podataka u herbaru koji zadovoljavaju traženi filter. Za postavke filtera pogledaj
Count/{IdZbirke}	<a href="#">GET</a>	Vraća ukupni broj podataka u herbarskoj zbirci koji zadovoljavaju traženi filter. Za postavke filtera pogledaj
Sabiraci	<a href="#">GET</a>	Vraća popis sabirača koje su zavedene u FCD-u koji počinju određenim nazivom (može se ispisati po
Search	<a href="#">GET</a>	Vraća popis herbara koji zadovoljavaju određeni filter (može biti prazan). Opcionalno rezultat može biti oblikovan po imesvojte, datumsabiranja, inventarnibroj, zbirka, idslike, tdwg, porodica, tip. Filter za search može biti: godina, idherbara, idzbirke, inventarni broj (*), nagib opisnalazista (riječi odvojene razmakom). Primjer: Toni/saslikom=true/opisnalazista=otok Vis/idzbirke=14/godina=2012
SearchCollection/{IdZbirke}	<a href="#">GET</a>	Pretražuje herbarsku zbirku. Vidi Search za detalje.
TDWG	<a href="#">GET</a>	Vraća popis država po TDWG-u. Rezultat je lista parova (oznaka države i naziv države)
TipoviPrimjeraka	<a href="#">GET</a>	Vraća popis tipova primjeraka herbara. Rezultat je lista parova (oznaka tipa i naziv tipa)
Zbirke	<a href="#">GET</a>	Vraća popis zbirke koje su zavedene u FCD-u. Rezultat je lista parova (oznaka zbirke i naziv zbirke)

Opazanja   Herbar   Korisne poveznice   Prikaži praznu kartu

kako koristiti bazu

Prijava korisnika

➔ ➔ ➔ 1.3.2013. 12:22:32

Pretraga

Rezultati pretrage

Kriterij pretrage: Godina sabiranja=2012, Naziv zbirke=CNHM Herbarium of Croatian Natural History Museum, TDWG=Croatia

12

Otisi herbarsku etiketu

Zapisa po stranici 25 Ukupno rezultata: 42

	Id herbara	Slika	Porodica	Ime svojte	Sabirač	God.	Tip	Naziv zbirke	Država (TDWG)	Ozn. koord.
<input type="checkbox"/>	<a href="#">31493</a>		Rosaceae	Amelanchier ovalis Medik.	Vrbek, Mirjana	2012		CNHM Herbarium of Croatian Natural History Museum	Croatia	
<input type="checkbox"/>	<a href="#">31492</a>		Boraginaceae	Lithospermum purpureocaeruleum L.	Vrbek, Mirjana	2012		CNHM Herbarium of Croatian Natural History Museum	Croatia	



# Debeli klijenti i višeslojne aplikacije (1)

11

➡ Ovisno o namjeni i debeli klijenti i višeslojne aplikacije

The screenshot displays the Prologis ERP - Razvoj application interface. The main window is titled "IT produkt katalog" and contains a sidebar with navigation options like "Indikatori proizvoda", "Tipovi naplate", and "Indikatori". The main area shows a table of products with columns for "Šifra proizvoda", "Šifra SAP", "Šifra AFA", "Naziv proizvoda", "Naziv nadproiz...", "Šifra nadproiz...", and "Tip naplate". A specific product, "APP-0001", is selected, showing details like "Tibco" and "IT EAI".

Overlaid on the right is a form titled "Prologis ERP - Razvoj" for creating an invoice. The form includes fields for "Šifra sheme knjiženja" (702000), "Vrsta entiteta" (702 Obračun PDV-a), "Kontni plan" (1 RRIF), "Grupa pozicije" (4 obračun PDV i PDV-K), "Vr. poreznog dokumenta" (14 Ulazni račun, R1), and "Vr. plaćanja virmanom" (0 Ne plaća se). The "Opis" field is set to "Obračun PDV". There are also checkboxes for "Ima ograničenu primjenu" and "Aktivan".

At the bottom, a table titled "Dovucite stupac kako bi ste grupirali podatke po tom stupcu" lists various columns and their corresponding data types, including "Rbr", "Vrsta pozi...", "Ds", "Ps", "P", "K", "Z", "O", "Algoritam", "Kontrola", "Rbrl", and "U".

# Debeli klijenti i višeslojne aplikacije (2)

12

- Aplikacije s različitim izvorima podataka (OPC server, baza podataka, konfiguracijske datoteke, ...)

Control Recipe Editor (CRE)

## 01 : 005 - Milling

Batch ID: 0      Process cell: 03 Brewhouse  
Order ID: 0      Recipe Category: 01 Production  
Ctrl Num: 0      Master Recipe: 01 Montel red  
Batch spare: 0

17:48:28 - Malt Quantity -3 → 1 OK  
17:48:32 - Mill current - a : Value is not float  
17:48:35 - Mill current - a : Value is not float

6-1056 Milling - Local malt intake

#	Id	Name level	Value	Unit	PasswordLevel	Min-Max
1	1	Control time	3.00	min	10	Unlimited
2	24	Mill current	a	A	10	Unlimited
3	32	Malt Quantity	1.00	kg	10	Unlimited
4	33	Amount of Malt in line	5.00	kg	10	Unlimited
5	35	Transfer Malt from:	1: Malt intake		10	
6	31	Fixed speed inlet rotary valve	12.00	%	10	Unlimited
7	37	Speed line rotary valve	54.00	%	10	Unlimited
8	43	Transfer from:	1: Second silo		10	

Version: 3.0.0.6218    User: montel    Password level: 100    Group: Development    Database: MASTER    OPC Server: opc.tcp://161.53.71.18:4845



## ➤ Model procesa

- Općenito: Plan razvoja, koji navodi opće postupke razvoja programskog proizvoda.
- Preciznije: Definicija koja kaže koje aktivnosti treba obaviti, tko ih treba obaviti i u kojoj ulozi; kojim redoslijedom, koji će proizvodi biti razvijeni i kako ih vrednovati.

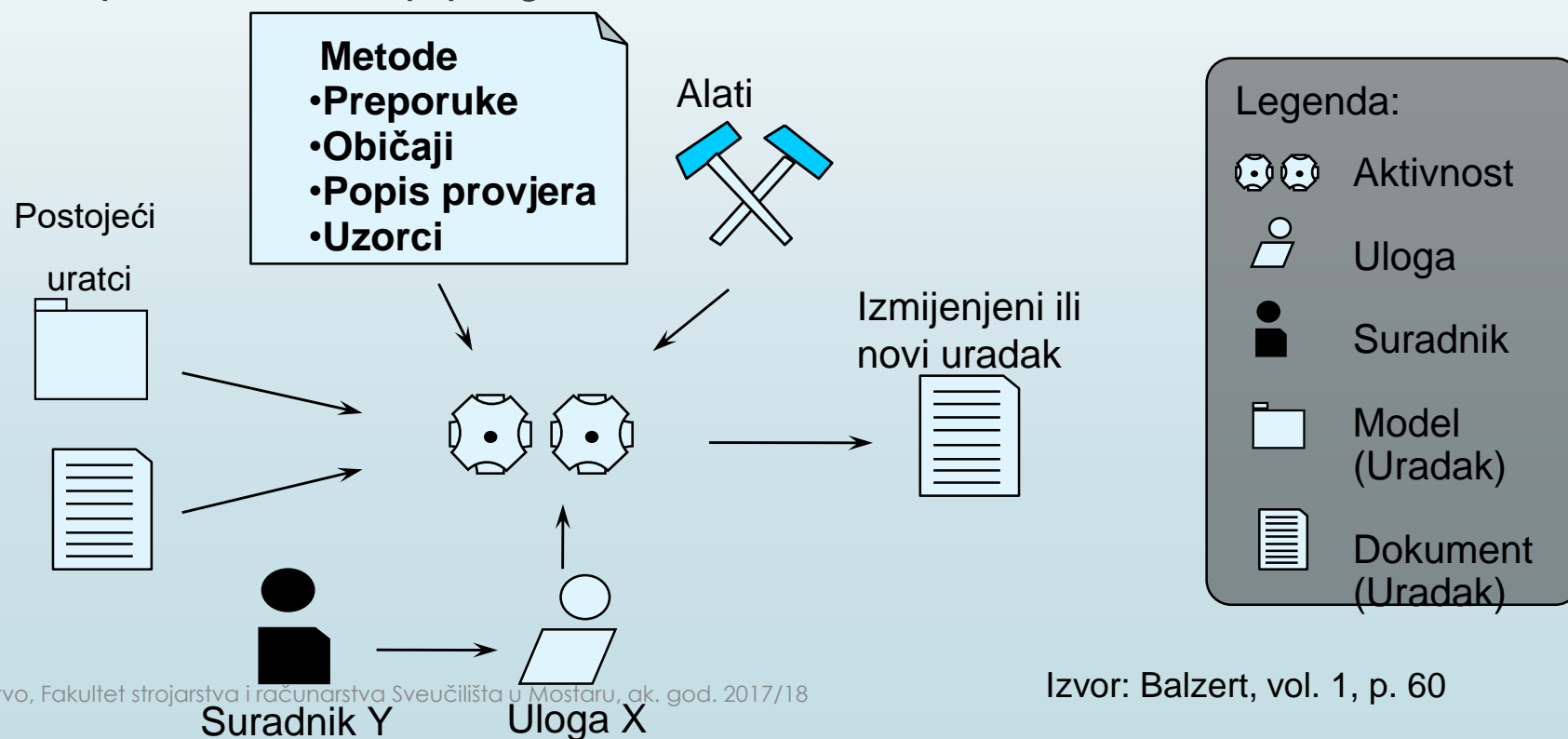
## ➤ SDLC = software/systems development life-cycle

- model razvojnog procesa - unaprijed propisan proces razvoja
- definira faze i zadatke (aktivnosti) koje treba obaviti tijekom razvoja
- ciklus sigurava “kontrolne točke” za praćenje napretka, procjenu postignutih rezultata i donošenje odluka o daljnjim koracima

# Aktivnosti procesa

14

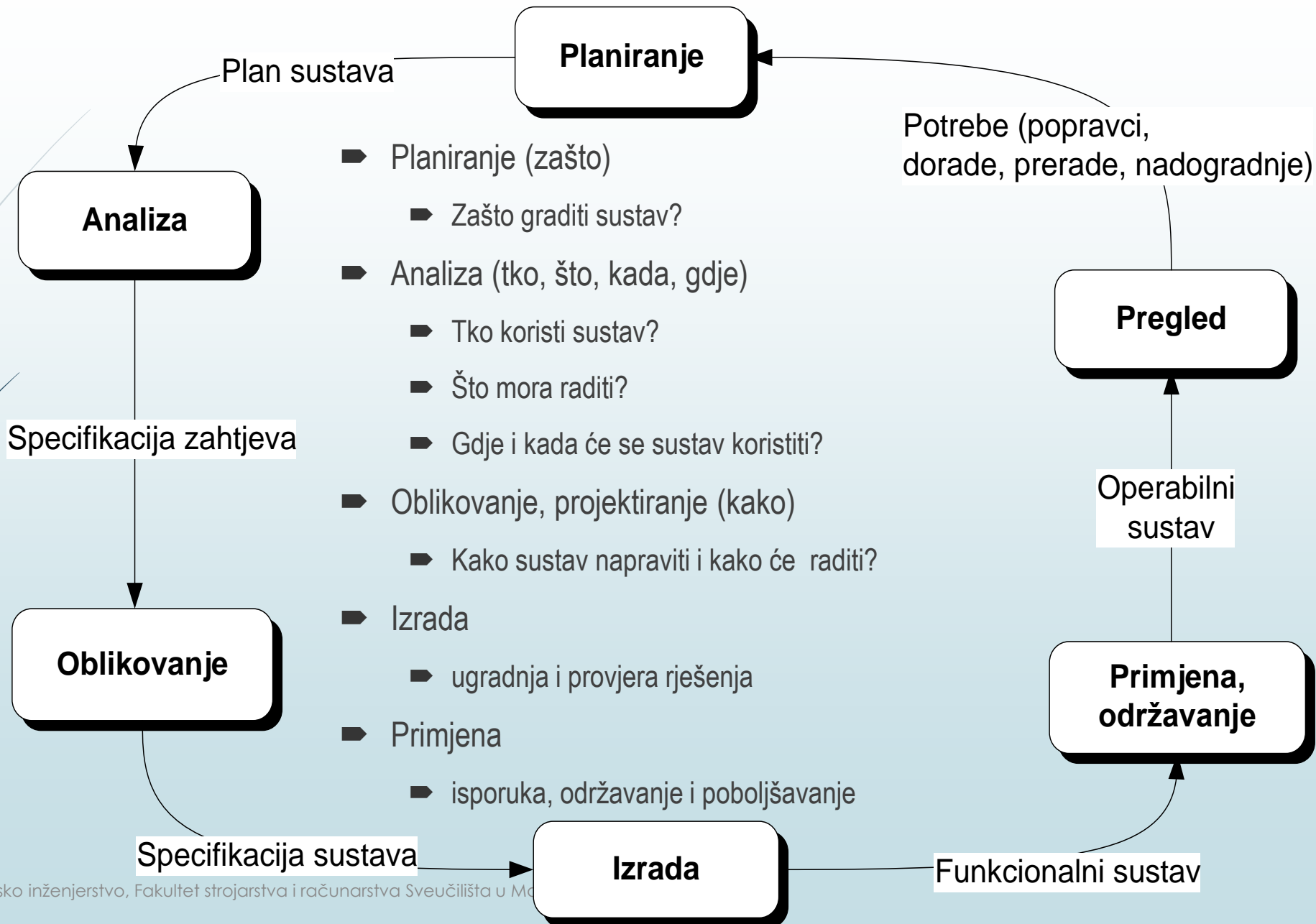
- Aktivnost - podproces u modelu procesa
- Uloga - suradnik koji obavlja određeni posao, npr. voditelj projekta, arhitekt / specijalist za projektiranje, programer, administrator BP
- Programski uradak - dokument, model ili program
- Programski proizvod - skup programskih uradaka





# Životni ciklus programske potpore

15



# Faze životnog ciklusa

16

## ➤ Planiranje

- Utvrđivanje ciljeva (poslovne koristi)
- Analiza izvedivosti
- Izrada plana rada
- Ekipiranje projekta
- Upravljanje projektom

## ➤ Analiza

- Prikupljanje informacija
- Modeliranje procesa
- Modeliranje podataka
- Specifikacija zahtjeva

## ➤ Projektiranje, oblikovanje

- Dizajn arhitekture
- Dizajn baze podataka i datoteka
- Dizajn sučelja
- Dizajn programa

## ➤ Izrada, ugradnja (implementacija)

- Konstrukcija
- Testiranje
- Instalacija

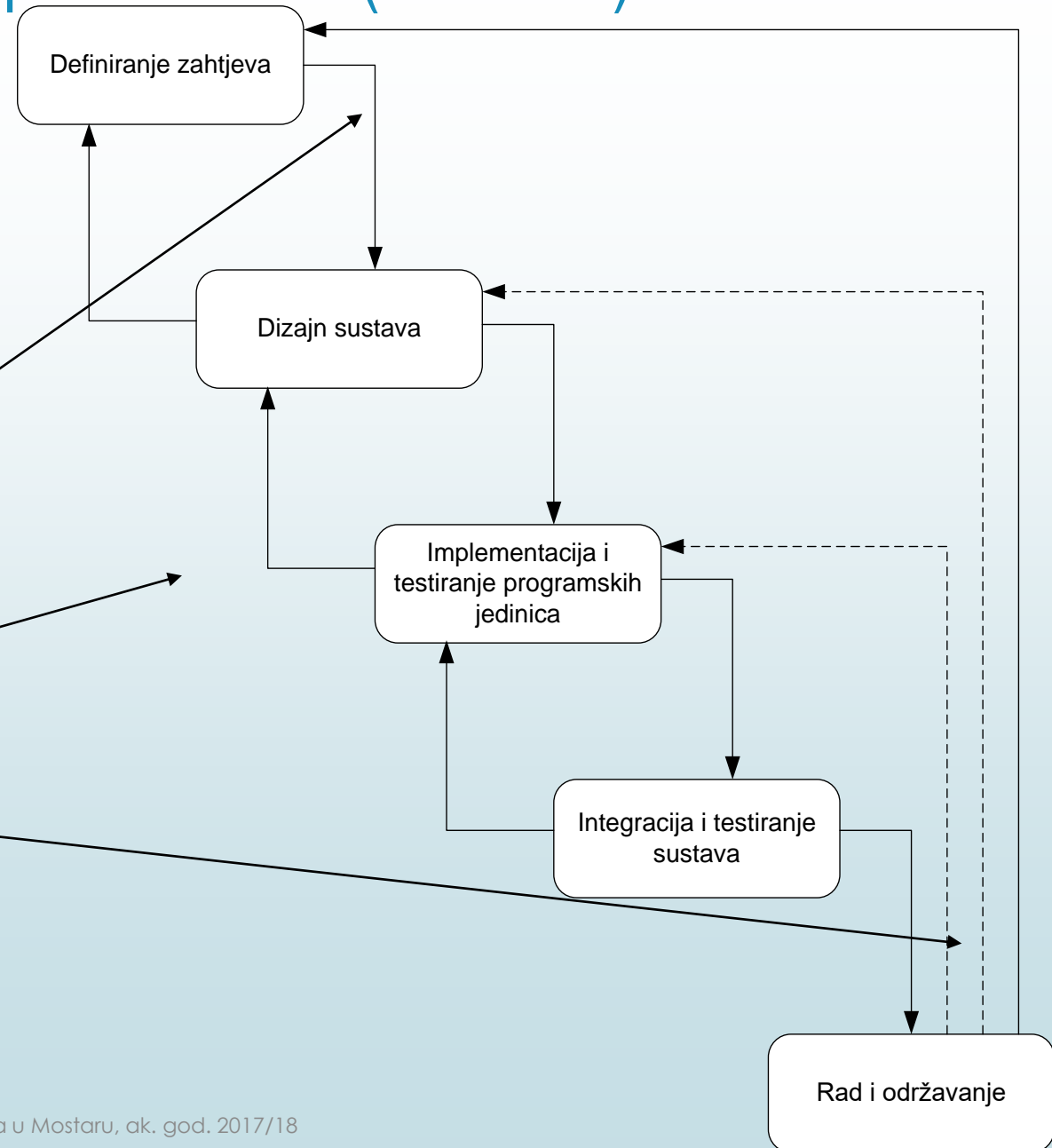
## ➤ Primjena

- Rad
- Održavanje

# Vodopadni model (waterfall)

## ► Varijante

- klasični
- pseudostrukturni
- radikalni



# Vodopadni model razvoja

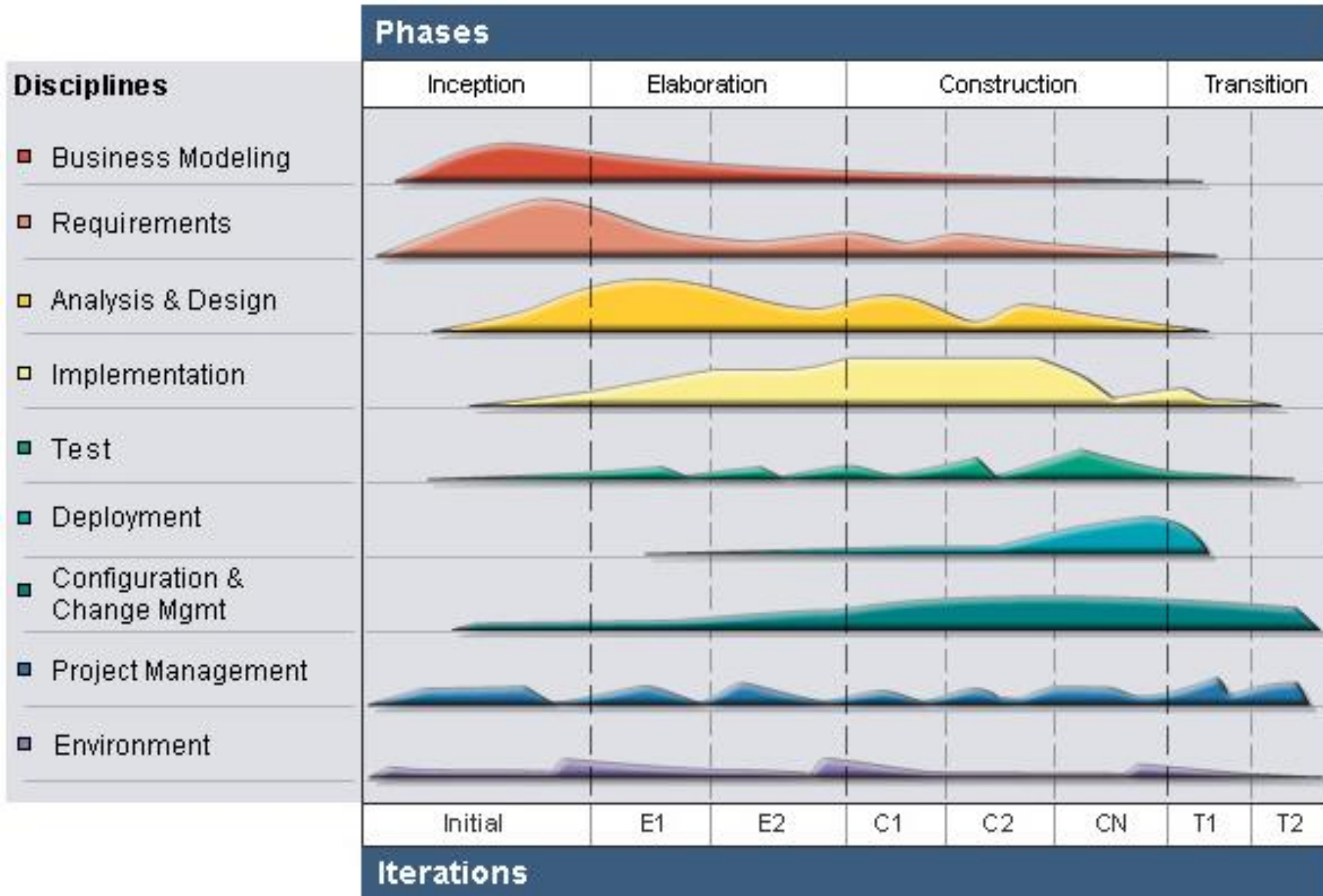
18

- Definiranje zahtjeva (*requirements analysis and definition*)
  - definira funkcionalnost programske potpore prema zahtjevima korisnika.
- Dizajn sustava (*system and software design*)
  - definira cjelokupnu arhitekturu programske podrške
  - grubi model sustava razrađuje se u detaljni opis izvedbe
- Ugradnja i testiranje jedinica (*implementation and unit testing*)
  - faza kodiranja tijekom koje se zahtjevi prevode u programski kod
  - programske jedinice zasebno se testiraju provjerom naspram specifikacije
- Integracija i testiranje sustava (*integration and system testing*)
  - programske jedinice povezuju se u cjelinu
  - provjerava se odgovara li programska potpora zahtjevima korisnika
- Primjena i održavanje (*deployment/operation and maintenance*)
  - započinje predajom sustava korisnicima na uporabu
  - tijekom održavanja uklanjaju se naknadno uočene neispravnosti te se sustav proširuje i poboljšava prema potrebama

- Unified software development process (UDP)
  - izvorno Objectory
  - danas IBM Rational Unified Process (RUP)
- Iterativni i inkrementalni razvoj
  - softver se razvija i objavljuje po dijelovima
  - glavne faze obavljaju se kroz niz iteracija
    - svaka iteracija obavlja se standardnim životnim ciklusom koji uključuje analizu, oblikovanje, ugradnju i provjeru
    - rezultat iteracije je proizvod završne kakvoće (production-quality), provjeren i integriran, koji zadovoljava podskup ukupnih zahtjeva
    - isporuke mogu biti interne ili prema korisnicima
- RUP sadrži niz "predložaka" razvojnih procesa (roadmaps) za različite modele razvoja i tipove projekata

# Faze i koraci razvoja

20

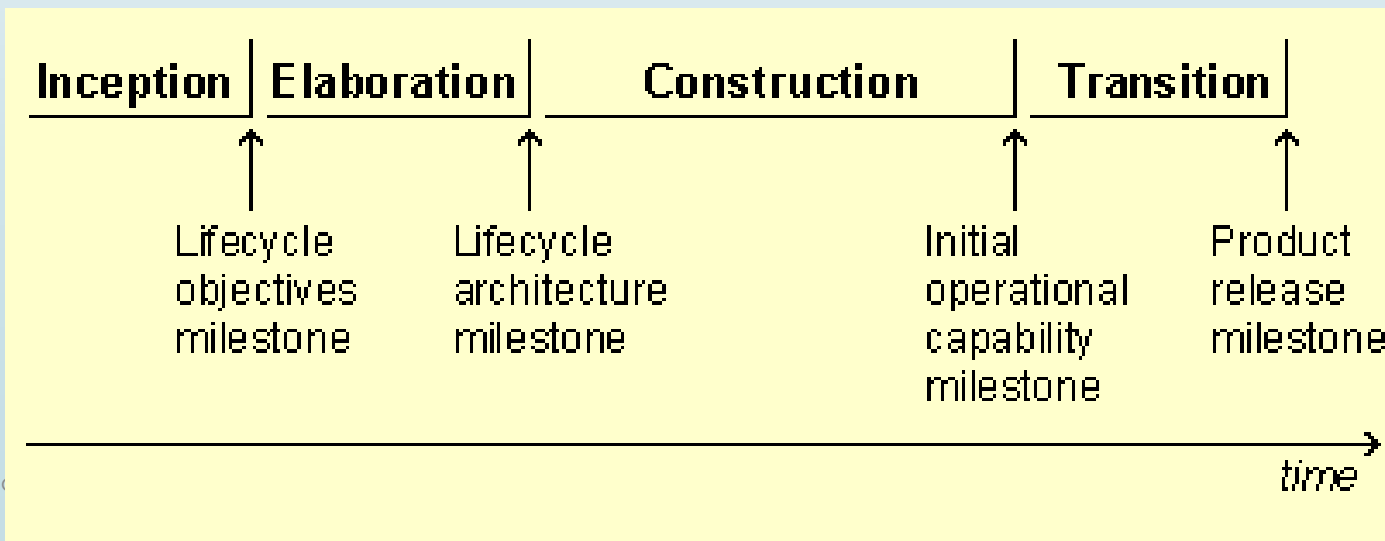




# Faze, discipline i kontrolne točke

21

- Horizontalna os predstavlja vrijeme i aspekte životnog ciklusa
  - cikluse, faze, iteracije i kontrolne točke
- Vertikalna os predstavlja discipline - logički grupirane aktivnosti
  - u ranijim fazama naglasak je na poslovnom modeliranju i zahtjevima
  - u kasnijima na implementaciji, testiranju i ugradnji te upravljanju izmjenama i konfiguracijom
  - disciplina upravljanja projektom ujednačenog je intenziteta.
- Osnovicu životnog ciklusa čine četiri slijedne faze od kojih svaka završava velikom kontrolnom točkom



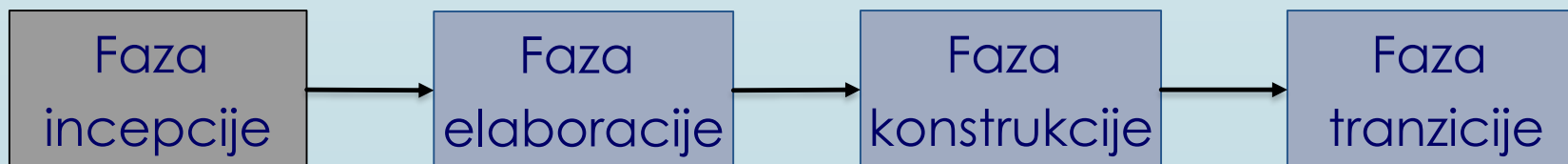
- Poslovno modeliranje (Business Modeling)
  - Identificira poslovni kontekst sustava i oblik organizacije
  - Definiraju se ciljevi i okvirna funkcionalnost, poslovna pravila i sl.
- Requirements (Zahtjevi)
  - Definira kako saznati i prikupiti želje te ih pretvoriti u skup zahtjeva
- Analiza i dizajn (Analysis & Design)
  - Definira pretvorbu zahtjeva u dizajn
  - Analiza usmjerena na logički pogled i funkcionalne zahtjeve
  - Dizajn usmjeren na fizički pogled i nefunkcionalne zahtjeve
- Implementacija (Implementation)
  - Kako razviti, organizirati, testirati i integrirati komponente
- Provjera (Test)
  - Kako testirati i procijeniti kvalitetu rješenja
- Uvođenje u primjenu (Deployment)
  - Aktivnosti potrebne da sustav bude dostupan krajnjim korisnicima

- Upravljanje konfiguracijom i promjenama (Configuration & Change Management)
  - kako kontrolirati i sinkronizirati evoluciju skupa komponenti i isporuka
- Upravljanje projektom (Project Management)
  - planiranje projekta, upravljanje rizicima, praćenje napretka i metrika
- Okolina (Environment)
  - organizira dijelove metodologije, procese i alate kao okruženje timu
- Životni ciklus iteracije: mini-vodopad (Mini-Waterfall)
  - usitnjeni standardni životni ciklus razvitka
  - zasnovan i vođen na slučajevima korištenja

# Glavne faze razvoja - Počinjanje

24

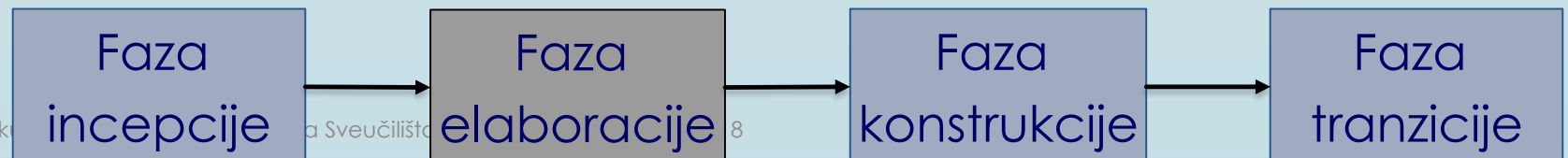
- Formuliranje opsega projekta
  - opis problemskog konteksta te najvažnijih zahtjeva i ograničenja
  - prikupljanje najvažnijih zahtjeva (10% detaljno)
  - preporuča se istaknuti i kritične scenarije korištenja (UC scenariji)
- Inicijalna procjena ukupnog troška, vremena i rizika
- Planiranje i priprema poslovnog slučaja
- Priprema okruženja za projekt
  - Procjena projekta i organizacije, odabir alata, razvojnih okruženja i procesa
- Izrada prijedloga arhitekture
  - demonstrirati izvedivost simulacijom, inicijalnim prototipom i sl.



# Glavne faze razvoja – Elaboracija (razrada)

25

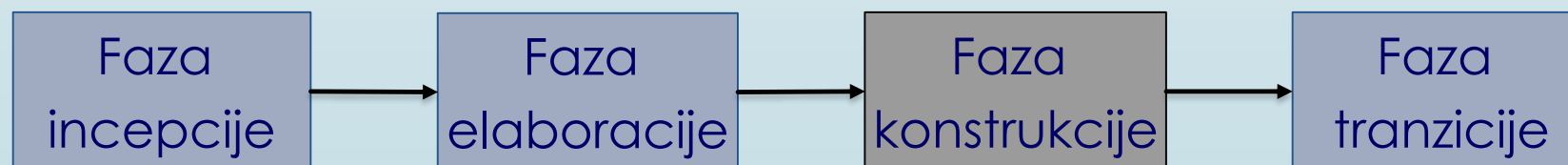
- Definiranje, validacija i zacrtavanje arhitekture
- Osiguranje da su arhitektura, zahtjevi i planovi stabilni, a rizici ublaženi
  - tako da se može pouzdano odrediti trošak i završetak projekta
- **Prikupljanje detaljnih zahtjeva (80%)**
- Ažuriranje vizije projekta
  - razumijevanjem kritičnih UC koji su ujedno i nositelji najvećih rizika
- Izrada plana iteracija za fazu konstrukcije
- Dorada razvojnog procesa i uspostava razvojnog okruženja
  - uključujući proces, alate i podršku za automatizaciju
- Dorada arhitekture i odabir komponenti
  - procjena potencijalnih komponenti - cijena i trajanje naredne faze



# Glavne faze razvoja – Konstrukcija (izgradnja)

26

- Upravljanje resursima, kontrola projekta i optimizacija procesa
  - paralelni razvoj nekoliko razvojnih timova s ciljem ubrzanja razvoja
- Završetak iterativnog i inkrementalnog razvoja konačnog proizvoda
  - provjera prihvatljivosti
  - podrazumijeva dovršetak analize, dizajna, razvoja i testiranja
- Procjena razvijenih isporuka naspram definirane Vizije projekta
- Provjera da li su programska podrška, lokacije i korisnici spremni za beta isporuku

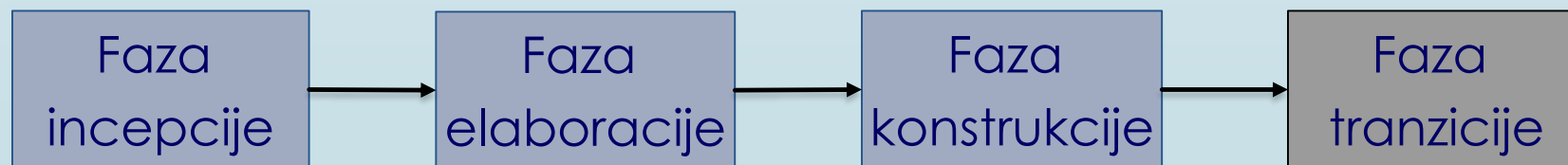




# Glavne faze razvoja – Tranzicija (prijelaz)

27

- Izvršavanje planova uvođenja u primjenu
- Dovršavanje korisničke dokumentacije i uputa
- Poduka krajnjih korisnika i održavatelja
- Testiranje programskog rješenja na lokaciji isporuke
- Izrada isporuke (release) konačnog programskog rješenja
- Omogućavanje proizvoda dostupnim svim krajnjim korisnicima
- Prikupljanje povratne informacije od krajnjih korisnika
- Fino podešavanje rješenja (popravak manjih pogrešaka, poboljšanje performanci) na temelju povratne informacije



# Agilni postupci razvoja

Ekstremno programiranje i Scrum kao neki od primjera agilnih postupaka razvoja

# Proglas agilnosti

29

- agilitas (lat.) - svojstvo brzine, okretnosti, hitrosti, lakoće, radinosti
- Manifest agilnosti (objava, proglas)
  - Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas
  - skijalište Snowbird, Utah, 2001.
- *Pojedinci i interakcije* važniji od procesa i alata
- *Softver koji radi* važniji od sveobuhvatne dokumentacije
- *Suradnja s naručiteljem/korisnikom* važnija od pregovora o ugovoru
- *Odziv na promjenu* važniji od praćenja plana

# Načela (principi) agilnosti

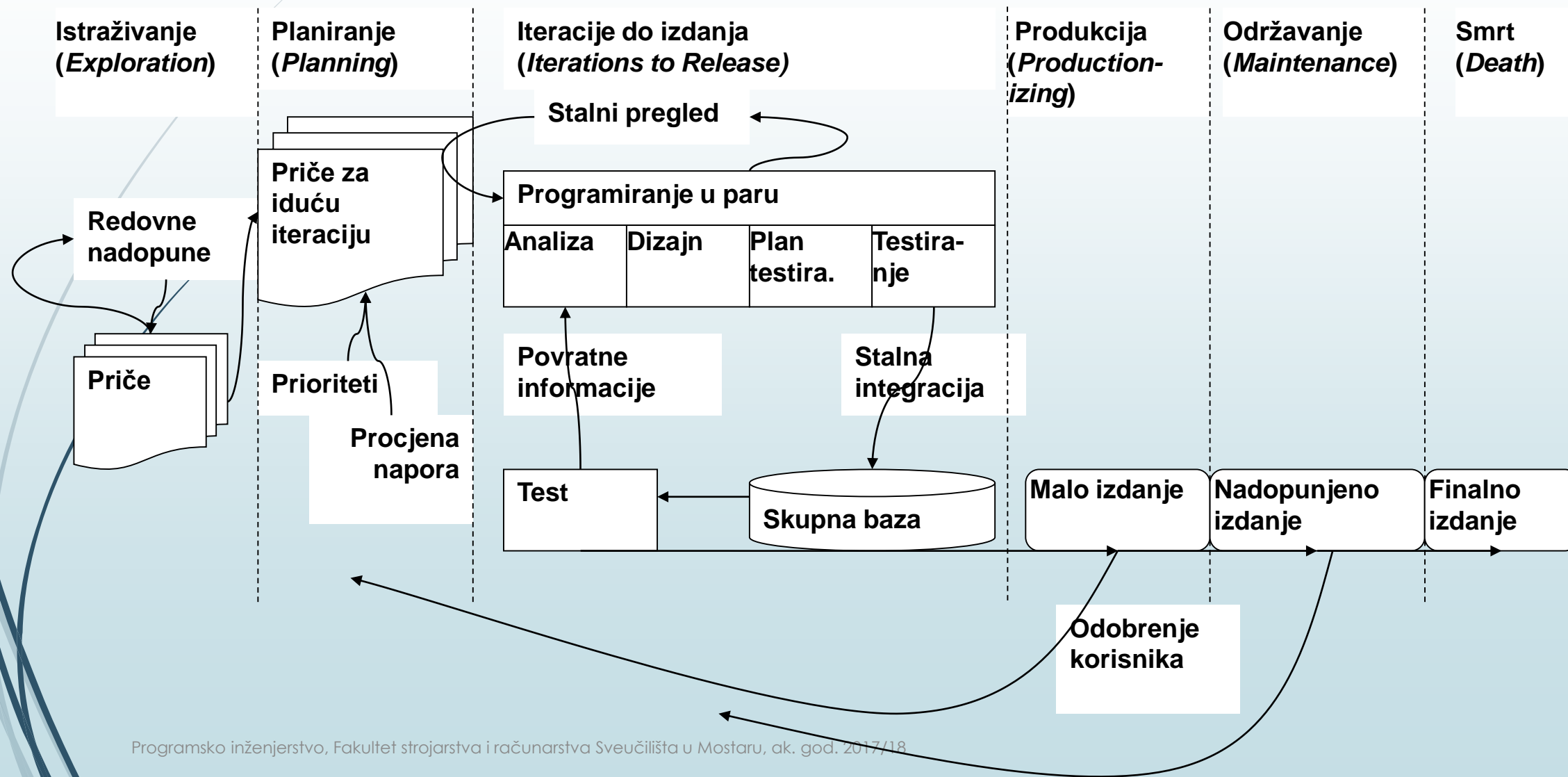
30

- Zadovoljstvo korisnika ranim i kontinuiranim isporukama softvera
- Promjene zahtjeva se željno prihvaćaju, čak i u kasnoj fazi razvoja
- Česta i što ranija isporuka softvera koji radi – 2x/mj do 1x/nekoliko mj
- Česta (dnevna) suradnja „poslovnjaka“ i razvojnika
- Motiviranje pojedinaca za rad u projektu – okruženje, povjerenje
- Usmena komunikacija - najbolja metoda za razmjenu informacija
- Glavna mjera napretka - softver koji radi
- Održivi razvoj - sponzori, tim i korisnici - održavati stalni tempo
- Kontinuirana pažnja na tehničku izvrsnost i dobar dizajn
- Jednostavnost je nužna
- Najbolje arhitekture, zahtjevi i dizajn - iz samoorganiziranih timova
- Tim spoznaje kako postati efektivniji, a zatim se često prilagođava

# Ekstremno programiranje

31

➡ Životni ciklus:



# Faze ekstremnog programiranja (1/3)

32

## ➡ Istraživanje

- ➡ Korisnici bilježe svoje priče na kartice
- ➡ Svaka kartica sadrži jednu mogućnost programa.
- ➡ Projektni tim se pobliže upoznaje s alatima, tehnologijom i postupcima projekta.
- ➡ Radi se prototip sustava za testiranje tehnologije i varijanti arhitekture sustava.
- ➡ Faza istraživanja traje nekoliko tjedana do nekoliko mjeseci

## ➡ Planiranje

- ➡ Postavlja prioritete na korisničke priče (tj. svojstva programskog rješenja)
- ➡ Planira se doseg prvog malog izdanja i vrijeme za pojedinu karticu
- ➡ Zatim se određuje cjelokupni vremenski raspored.
- ➡ Rok za izdavanje prvog malog izdanja obično je unutar dva mjeseca.
- ➡ Faza planiranja traje nekoliko dana.



# Faze ekstremnog programiranja (2/3)

33

## ➤ Iteracije do izdanja (razvoj)

- Uključuje nekoliko iteracija sustava prije prvog izdanja.
- Vremenski raspored iz faze planiranja se razlaže u više iteracija
- Pojedina iteracija traje jedan do četiri tjedna.
- Prva iteracija stvara takav sustav koji obuhvaća cijelu arhitekturu ciljanog sustava.
- Klijent određuje kartice koje će se koristiti pri svakoj narednoj iteraciji.
- Testovi prihvatljivosti izvode se na kraju svake iteracije.
- Na kraju posljednje iteracije, sustav je spreman za produkciju.

## ➤ Produkcija

- Dodatno testiranje i provjera performansi sustava prije isporuke klijentu.
- Razrješenje primjedbi na sustav te odlučivanje da li će se riješiti u ovom izdanju.
- Iteracije trajanja tri do najviše tjedan dana.
- Zakašnjele nove ideje i prijedlozi se dokumentiraju i njihova implementacija odgađa.

## ➡ Održavanje

- ➡ nakon što je prvo izdanje pušteno u produkciju
- ➡ XP projekt mora istovremeno održavati softver u primjeni i proizvoditi nove iteracije
- ➡ Zbog toga se brzina implementacije smanjuje
- ➡ Održavanje može zahtijevati nove članove projektnog tima i promjenu strukture tima.

## ➡ Faza smrti je blizu kada klijent nema više novih kartica s pričama

- ➡ Podrazumijeva se da sustav zadovoljava sve zahtjeve (npr. pouzdanost i stabilnost).
- ➡ Vrijeme u XP projektu da se konačno napiše sva korisnička dokumentacija budući da više nema promjena na arhitekturi, dizajnu i kodu sustava.
- ➡ Smrt može nastupiti i kada sustav ne ispunjava sva korisnička očekivanja, ili ako postane preskup za daljnji razvoj.

- **Komunikacija (communication)**
  - Verbalna i elektronička, učestala / stalna, svih dionika
- **Jednostavnost (simplicity)**
  - Najjednostavniji mogući dizajn (KISS), samo što je potrebno i zatraženo
  - kontinuiranim refaktoriranjem i minimizacijom dokumentacije
- **Povratne informacije (feedback)**
  - Od korisnika i unutar tima – što ranije te češće
- **Hrabrost (courage)**
  - Akcije i (teške, nepopularne) odluke
  - npr. odbacivanje dijelova (YAGNI), ili veće promjene u kasnoj fazi projekta
- **Uvažavanje (respect)**
  - svatko doprinosi timu; i mali doprinos je vrijedan
  - izbjegavanje promjena koje bi onesposobile aktualnu verziju ili usporile napredak ostalih

- Priče (korisničke priče) - Stories (User Stories) != Use Case
  - kratki opis funkcionalnosti, nekoliko rečenica
  - toliko da se može odrediti napor
  - „A promise to have a conversation” (Cockburn)
- Tjedni ciklus (Weekly Cycle)
  - razvoj u tjednim ciklusima, tjedan započinje sastankom izbora priča
  - tjedan ne mora započeti u ponedjeljak
- Kvartalni ciklus (Quarterly Cycle)
  - grublje, na dulje staze, razvoj se planira kvartalno („rolling wave”)
- Rezerva (Slack)
  - Zadaci niskog prioriteta koji mogu biti odbačeni ako projekt kasni

- Smještaj ekipe (Sit Together)
  - Kolocirana ekipa, otvoreni prostor
- Cjelovitost i zajedništvo ekipe (Whole Team)
  - Cjelovitost kompetencija, osjećaj pripadnosti
- Informativno radno okruženje (Informative Workspace)
  - Ploče, „visible wall graphs”
- Energičan rad (Energised Work)
  - Odmorni programeri – produktivnost – ograničeni prekovremeni rad
- Programiranje u paru (Pair Programming)
  - Vodič (driver) i promatrač (observer, navigator)
  - Zamjena uloga ali i partnera

- Inkrementalni dizajn (Incremental Design)
  - Nema velikog oblikovanja unaprijed (BDUF – big design up-front)
  - dizajn kao kontinuirani proces malih koraka – refaktoriranje
- Test prije programiranja (Test-First Programming)
  - Testovi trebaju biti napisani prije kodiranja te automatizirani
- Desetminutna gradnja (Ten-Minute Build)
  - sustav se mora moći kompilirati i testirati unutar 10 minuta
  - da bi mogao postići odgovarajuću povratnu informaciju (feedback)
- Kontinuirana integracija (Continuous Integration)
  - Svakih nekoliko sati ili nakon promjena → dnevno

- Upravitelj (manager, big boss)
  - Upravljanje ekipom, rješavanje problema # team lead
- Trener (coach)
  - Savjet, nadzor, kontrola (issue control) # tech lead
- Programer, razvojniki (developer)
  - kodiranje, pisanje testova, refaktoriranje
- Tester
  - Izrada i izvođenje testova, održavanje alata za testiranje
- Klijent, korisnik (customer)
  - piše priče i testove prihvatljivosti (!?), određuje prioritete
- Druge uloge
  - Tracker, Trainer, Doomsayer, ...
  - Product Manager, Domain Expert, Interaction Designer, Business Analyst ...



# Scrum

40

- Naziv potječe iz ragbija, gdje se formira skup igrača (engl. *scrum*) za početak igre nakon prekida
- jednostavni upravljački okvir za inkrementalni razvoj
  - nije definiran proces, ne bavi se tehnikalijama
- iteracija = sprint
- rezultat sprinta – potencijalno isporučivi inkrement proizvoda (engl. *shippable*)



- Ekipa (Scrum Development Team)
  - jedna ili više ekipa od po 7 plus/minus 2 člana
  - svestrani članovi (cross-functional)
  - samoorganizirajuća ekipa (self-organizing)
- Vlasnik proizvoda (Product Owner)
  - zadužen za plan, prioritete, troškove i povrat investicije
- Majstor (Scrum Master)
  - brine o procesu, koordinira, ali ne donosi poslovne ni tehničke odluke
- Klijent
  - održava Product Backlog
- Uprava (Management)
  - odlučuje o doseg, promjenama, standardima, ...

## ➤ Prije igre (pre-game)

- podfaze: Planiranje i Dizajn/Arhitektura
- izrađuje se radna lista proizvoda (Product Backlog - PB)
  - u PB se konstantno zapisuju zahtjevi, procjene napora i prioriteti

## ➤ Razvoj / "igra" (development / game)

- razvoj iterativnim ciklusima, takozvanim sprintovima
- sprintovi - okvirno jednakog trajanja, 30 dana (prema knjizi)
  - tjedan do tri u praksi
- sprint ima sve faze klasičnog ciklusa (RUP ima mini-vodopad !)
  - zahtjeve, analizu, dizajn, evoluciju, test i isporuku
- tri do osam sprintova dok sustav ne bude spreman za distribuciju

## ➤ Poslije igre (post-game)

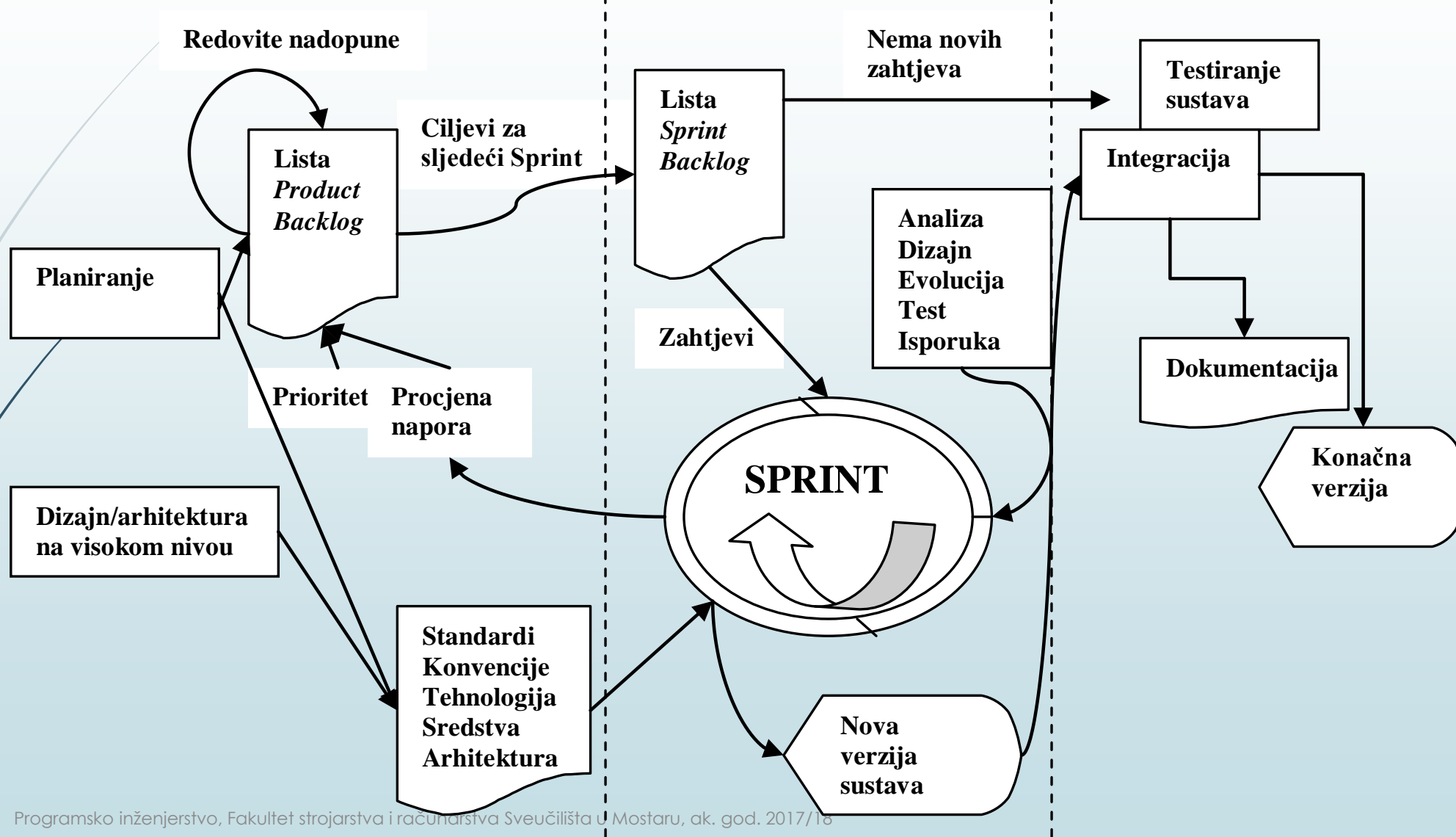
- priprema sustav za izdanje kroz integraciju, testiranje i druge aktivnosti

# Scrum - Proces

## Prije igre

## Razvoj (Igra)

## Poslije igre



## ➤ Product Backlog

- neizvršen rad, preostali posao
- lista poželjne funkcionalnosti
- vidljiva svim dionicima
- svatko može dodati elemente

## ➤ Product Backlog Item - element

- definira "ŠTO", najčešće kao korisnička priča
- ima kriterij prihvatljivosti, definiciju "dovršenosti"
- sadrži više zadataka
- poslovnu vrijednost odredi Vlasnik
- napor procijeni Ekipa

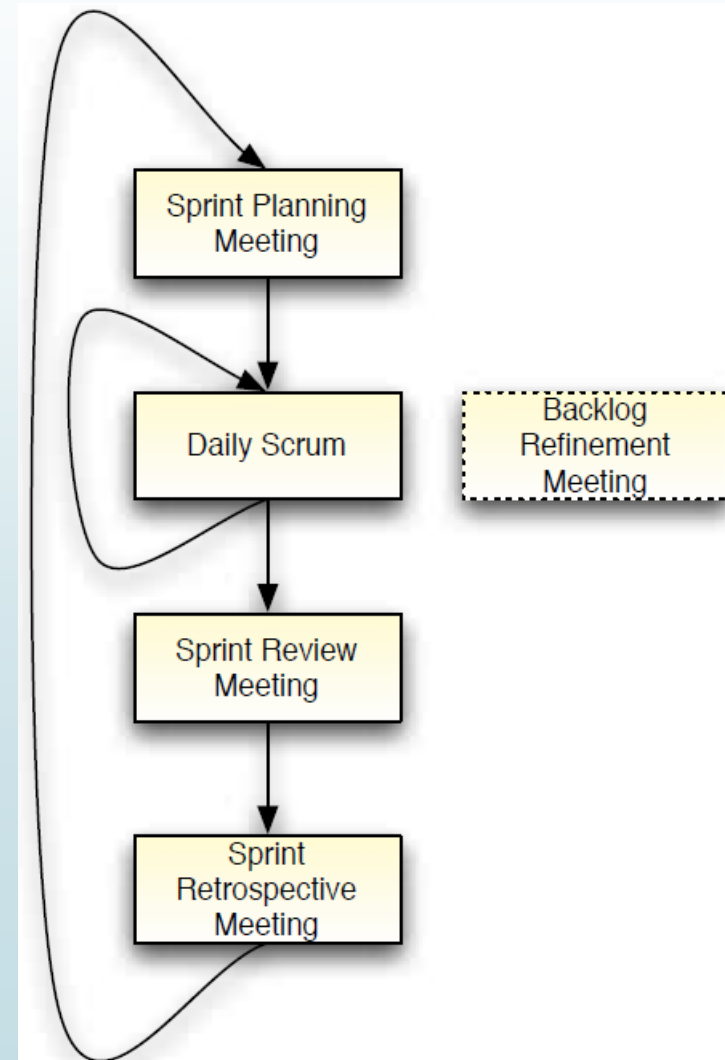
## ➤ Sprint Backlog

- popis zadataka i statusa
- ažuriran tokom sprinta
- vidljiv Ekipi

## ➤ Zadatak sprinta (Sprint Task)

- "KAKO" za PBI "ŠTO"
- dan posla ili manje
- preostali napor procjenjuje se dnevno u satima

- Planiranje sprinta
  - 1. na početku iteracije – koji PB elementi idu u sprint
  - 2. tim dekomponira PB elemente u listu zadataka
  - 30d sprint planira se max 8 sati
- Dnevni Scrum
  - 15min, članovi tima međusobno
  - "standup meeting" – dojam užurbanosti
- Pregled sprinta
  - demonstracija inkrementa na kraju sprinta
  - Vlasnik deklarira "dovršeno"
  - ostalo ide u naredni sprint
- Retrospektiva sprinta
  - samoanaliza procesa
- Pročišćavanje preostalog posla
  - podjela, procjena, prioriteti ...



# Reference

46

- <https://www.computer.org/portal/web/swebok/swebokv3>
- [https://pascal.computer.org/sev\\_display/index.action](https://pascal.computer.org/sev_display/index.action)
- <http://www.rspa.com/>
- <https://www-01.ibm.com/software/hr/rational/>
- <http://www.extremeprogramming.org/>
- <http://www.agilemodeling.com/>
- <https://www.scrumalliance.org/>
- <http://pm-training.net/agile-development-methodology-wiki/>