

# **Principi softverskog inženjerstva SI3PSI**

## **Uvod**

# Organizacija kursa

- ▶ mailing lista: si3psi
  - ▶ sajt predmeta: <http://si3psi.etf.rs>
  - ▶ Predavač: Dr Dragan Bojić, docent
  - ▶ Asistent: Dražen Drašković
  - ▶ Kontakt: [bojic@etf.rs](mailto:bojic@etf.rs)
- 
- ▶ Formiranje ocene:
  - ▶ 40% na osnovu timskog projekta koji se radi u toku trajanja nastave
  - ▶ 60% na osnovu kolokvijuma i ispita

# 1. Sadržaj uvodnog predavanja

- a) Oblasti i definicija softverskog inženjerstva
- b) Značaj softverskog inženjerstva u praksi –  
- problemi razvoja softvera
- c) O veličini softvera
- d) Istorija softverskog inženjerstva
- e) Razlike između razvoja softvera u obrazovnom okruženju  
i u praksi
- f) Literatura, članci, radovi, ...

# Oblasti softverskog inženjerstva: podoblasti

- ▶ Modeli životnog ciklusa
- ▶ Softverski procesi (CMM, ISO 9000)
- ▶ Upravljanje konfiguracijom
- ▶ Procena truda
- ▶ Upravljanje projektima
- ▶ Upravljanje kvalitetom
- ▶ “Cleanroom” softversko inženjerstvo
- ▶ Reverzno inženjerstvo
- ▶ Shvatanje programa (razumevanje programa)
- ▶ Softverska metrika
- ▶ Softverska arhitektura
- ▶ Održavanje softvera
- ▶ Inženjerstvo zahteva
- ▶ Specifikacija softvera
- ▶ Ponovno korišćenje
- ▶ Verifikacija programa
- ▶ Dokumentovanje softvera
- ▶ SI “bazirano-na-znanju”
- ▶ Sistematsko testiranje
- ▶ Pravljenje prototipova
- ▶ CASE alati
- ▶ Programski jezici za SI
- ▶ Objektna-orijentisanost (analiza, dizajn, programiranje, testiranje)
- ▶ Ekstremno programiranje
- ▶ Programiranje u parovima

▶ „*Programiranje je relativno mala komponenta oblasti softverskog inženjerstva. Neke druge komponente (uključujući na primer, specifikaciju softvera i projektovanje) su jednako važne.*“

# Softversko inženjerstvo

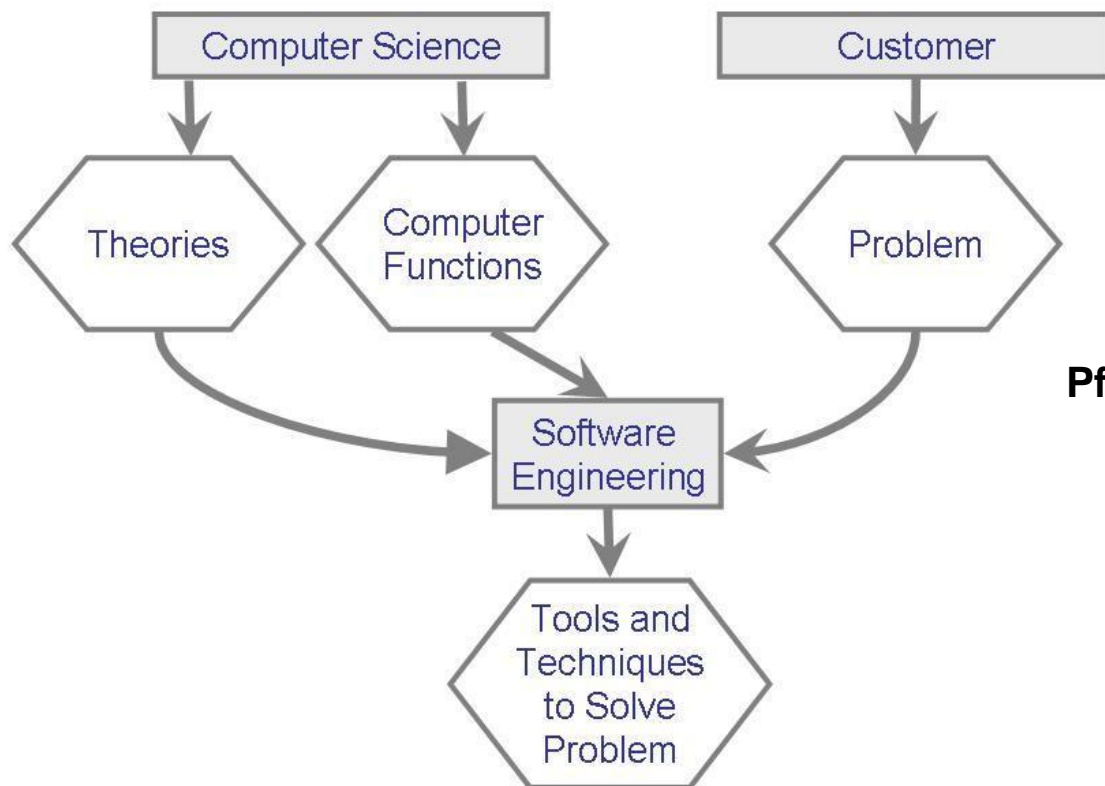
## uvodne definicije

- ▶ *„SE je inženjerska disciplina koja se bavi svim aspektima proizvodnje softvera.*
- ▶ *Inženjeri softvera treba da usvoje sistematski i organizovan pristup radu i upotrebljavaju odgovarajuće alate i tehnike zavisno od problema koji rešavaju, razvojnih ograničenja i raspoloživih resursa “*
- ▶ (Sommerville, 6. izdanje)

## Definicije softverskog inženjerstva (2)

- ▶ „SI je primena sistematskih, disciplinovanih, merljivih pristupa razvoju, rukovanju i održavanju softvera; drugim rečima, primeni inženjerstva na softver“ (IEEE standardni rečnik terminologije softverskog inženjerstva, 1990).
- ▶ „Softversko inženjerstvo je definisani, korak-po-korak proces, koji olakšava specificiranje, projektovanje, implementaciju i testiranje softvera za skup iznetih zahteva, na najbrži i najprofitabilniji mogući način“ (Kehoe: ISO 9000-3, 1995, S. 11).

# Veza softverskog inženjerstva i računarske nauke



Pfleeger, Atlee 4ed

- ▶ Računarska nauka bavi se teorijom i osnovama (apstrakcija); softversko inženjerstvo bavi se praktičnim aspektima razvoja i isporuke korisnog softvera

# Softver

Softver =  
programi, podaci, dokumentacija

## ► osobine softvera:

- nema otpadaka tokom rada softvera
- starenje (softver se stalno ažurira)  
→ posle 10 godina upotrebe i ažuriranja, ne postoji ni jedna originalna linija koda (Pagel, Six p. 35)
- dugotrajna upotreba  
→ u proseku 10 - 15 godina (Wallmüller 90, p. 3)  
i do 30 godina (Hausi Müller)
- jednostavan za kopiranje (→ takođe i grešaka)
- težak za merenje (metrike: kvalitet, kvantitet)
- relativno složen

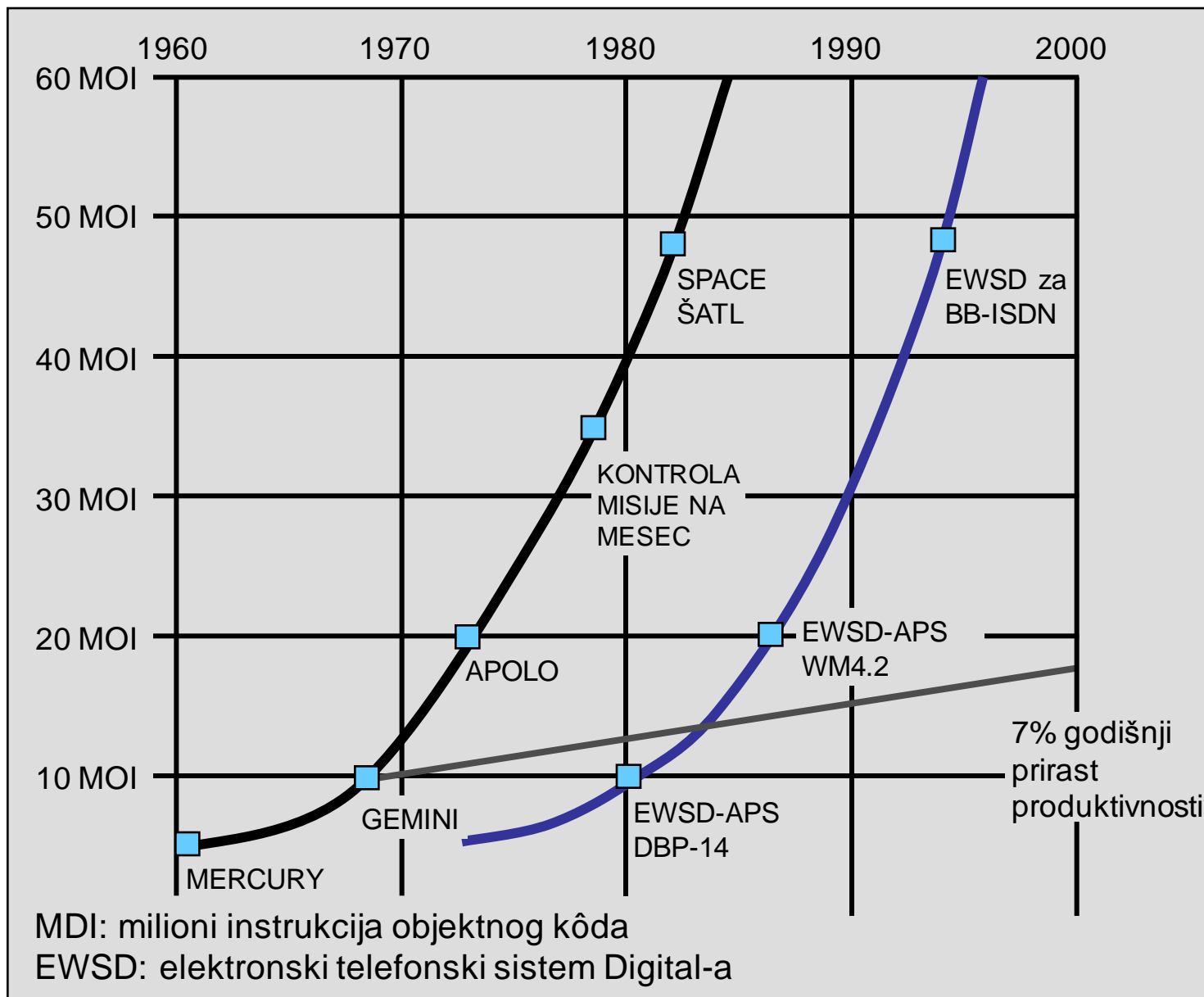


# Složenost softvera

- ▶ *Prepoznatljive karakteristike softvera industrijske snage je da je izuzetno složeno, ako ne i nemoguće, za pojedinca da shvati sve veze u njegovom projektovanju. Prosto rečeno, složenost ovakvih sistema prevazilazi intelektualni kapacitet čoveka.*

(Booch, Objektno-orijentisana analiza i projektovanje, s. 4)

# Povećana složenost softvera



# Posledice složenosti

**Softver je ...**

- sklon greškama
    - ponaša se drugačije od očekivanog
      - skuplji nego što je planirano
        - završen prekasno
          - neprimenljiv
- veoma često: propast projekta

# Primeri softverskih problema

- ▶ **Aerodrom u Denveru:** nije mogao biti pušten u rad (nije otvoren) 1994 – softver za sistem za transport prtljaga nije radio
- ▶ **Siemens:** računovodstveni softver za lekove razvijen za fondove zdravstvenog osiguranja nije bio gotov na vreme (sledstveni troškovi od 1 milijarde DEM - Berliner Zeitung 22.05.96)
- ▶ **Deutsche Telekom:** pogrešan proračun cene telefonskih impulsa za 1.1.96 (softverska greška: bez praznične tarife – šteta: stotine miliona DEM)
- ▶ **F18 ratni avion:** tokom vežbe 1983 sa novorazvijenim softverom – nakon prelaska ekvatora, avion se okrenuo naglavačke. Razlog je bila greška u znaku operacije (+ - ) (Wallmüller 1990, s. 1).

# Primeri softverskih problema - nastavak

- ▶ **Prva svemirska raketa za Veneru:** 1979 nije pronašla metu: U Fortran programu, upotrebljena je decimalna tačka, umesto zareza. Gubitak: više stotina miliona dolara (Wallmüller 1990, s. 1).
- ▶ **Therac 25 – računarski kontrolisani uređaj za terapijsku radijaciju:** između juna 1985. i januara 1987. 6 ljudi je predozirano (5 od njih je kasnije umrlo) kao posledica nedostajuće softverske sigurnosne brave koja bi trebala da spreči predoziranje (Berry, D.M., “Myths and realities of software development”).

# Veličina softvera

- ▶ Klasifikacija softvera u zavisnosti od njegove veličine:  
mali: do 2000 linija koda  
srednji: 2000 – 100.000  
veliki: 100 000 – 1.000.000  
veoma veliki: > 1 Mil.  
(A. Macro, s. 70)
- ▶ prosečna veličina softvera u svakoj od 100 najvećih kompanija u SAD:  
35 Mil. linija koda  
(Hausi Müller, s. 3-12)
- ▶ recimo, SAD, Ministarstvo odbrane:  
1,4 milijarda linija koda  
(raspoređeni u 1.700 centara podataka)  
troškovi rukovanja ovim sistemima:  
9 milijardi dolara godišnje  
(CACM, 5, 1994, s. 26)

# Istorijat SE

- ▶ *softverska kriza:*  
pojam od sredine 60-ih
- ▶ *sredinom 60-ih*  
cene softvera postaju veće nego cene hardvera  
(slika)
- ▶ *NATO konferencije 1968, 69:*  
definicija pojma softverskog inženjerstva  
→ velike prednosti  
ali: i danas imamo kriza softvera
- ▶ *cene softvera (širok svet)*  
1985: 140 milijardi dolara  
1995: 435  
(Yourdon)

# Troškovi hardvera i razvoj softvera

Year	Hardware Cost Index
1960	698
1970	75
1980	10.5
2000	0.5
2001	0.4
2002	0.4
2003	0.3
2004	0.3

Year	PC Software Titles
1980	90
1986	150
1991	9,000
1996	19,000
2003	42,000

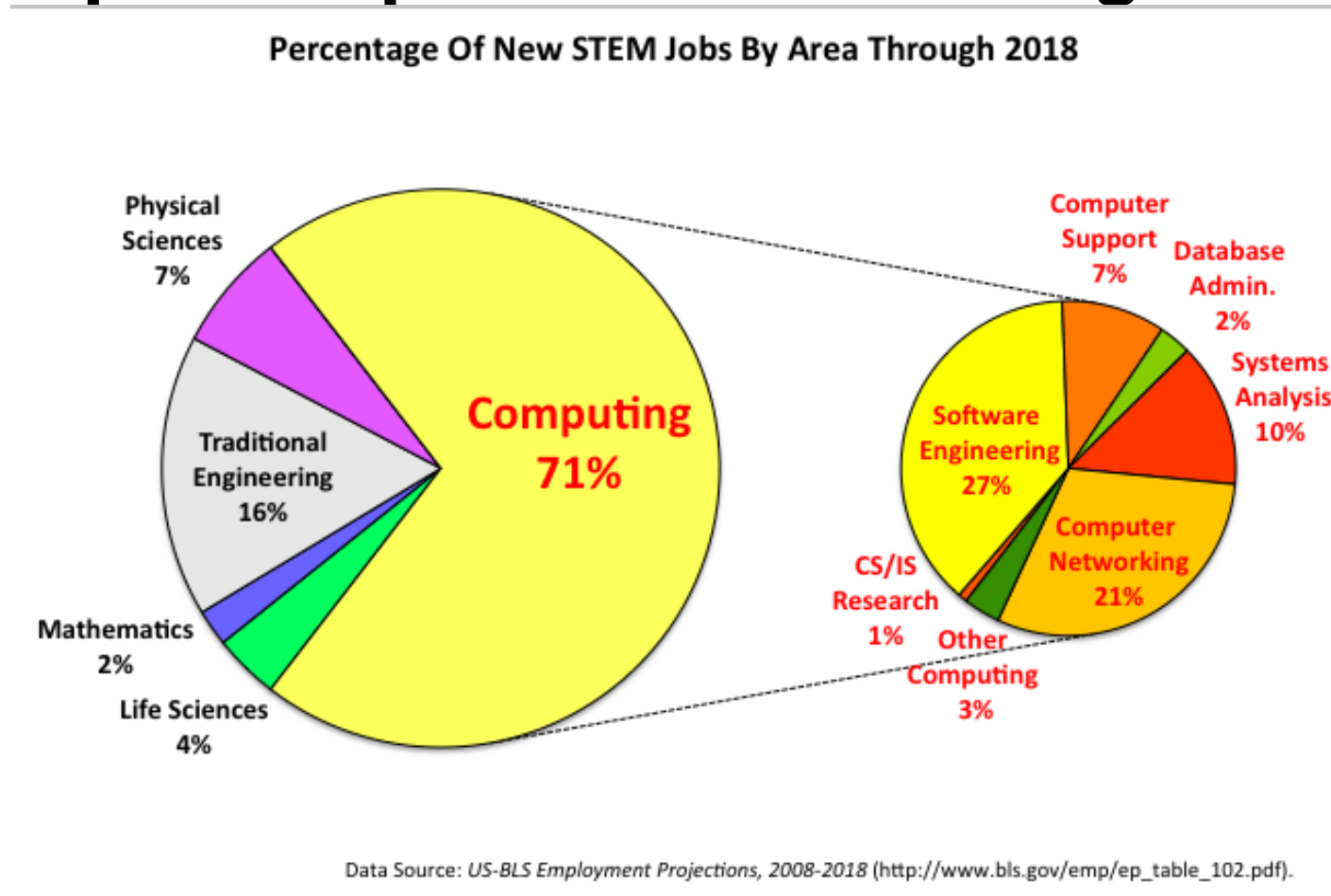
The Software Encyclopedia 1980 -  
2003, R.R. Bowker

Jorgenson, Dale W. and Kevin Stiroh, "Raising the Speed Limit: US Economic Growth in the Information Age." Measuring and Sustaining The New Economy: National Academy Press 2001.

Cena hardvera dramatično padala s vremenom, tako da je i fiksni trošak razvoja softvera opao. Broj softverskih naslova povećavao se stopom jednakom stopi pojeftinjenja hardvera



# Perspektive posla u softverskog industriji



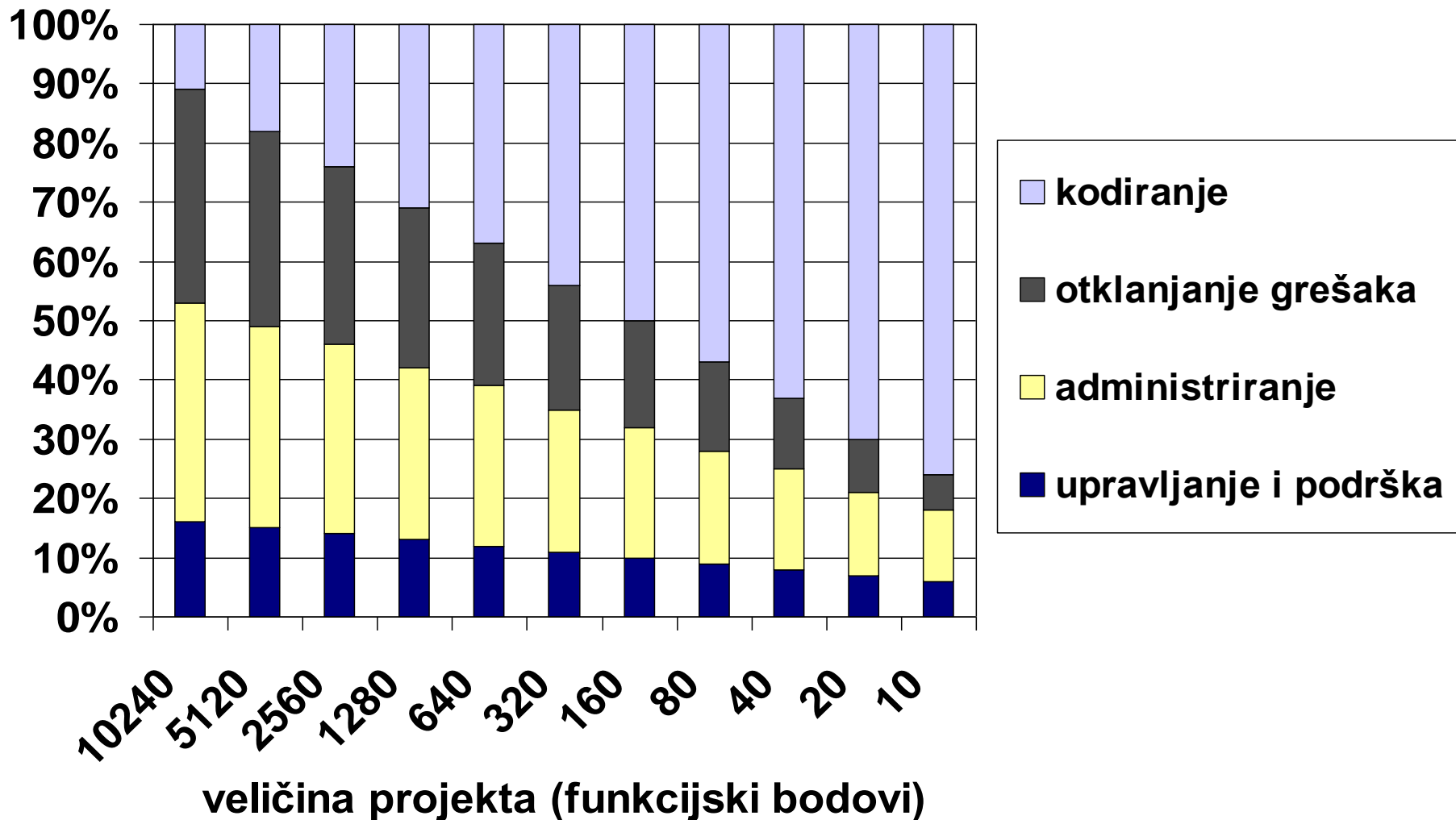
- ▶ STEM: Science, Technology, Engineering, Matematics
- ▶ Procene američkog biroa za statistiku rada (US-BLS) za 2008-18.

# Studentski razvoj softvera



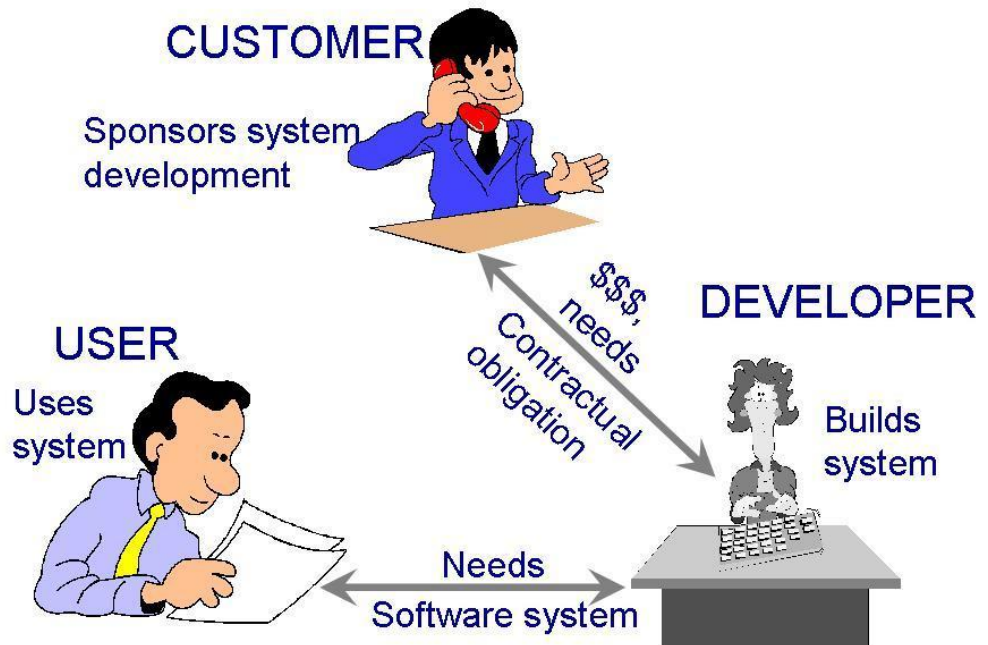
„Uobičajeni“ studentski pogled na životni vek softvera

# Distribucija aktivnosti projekta u funkciji veličine projekta

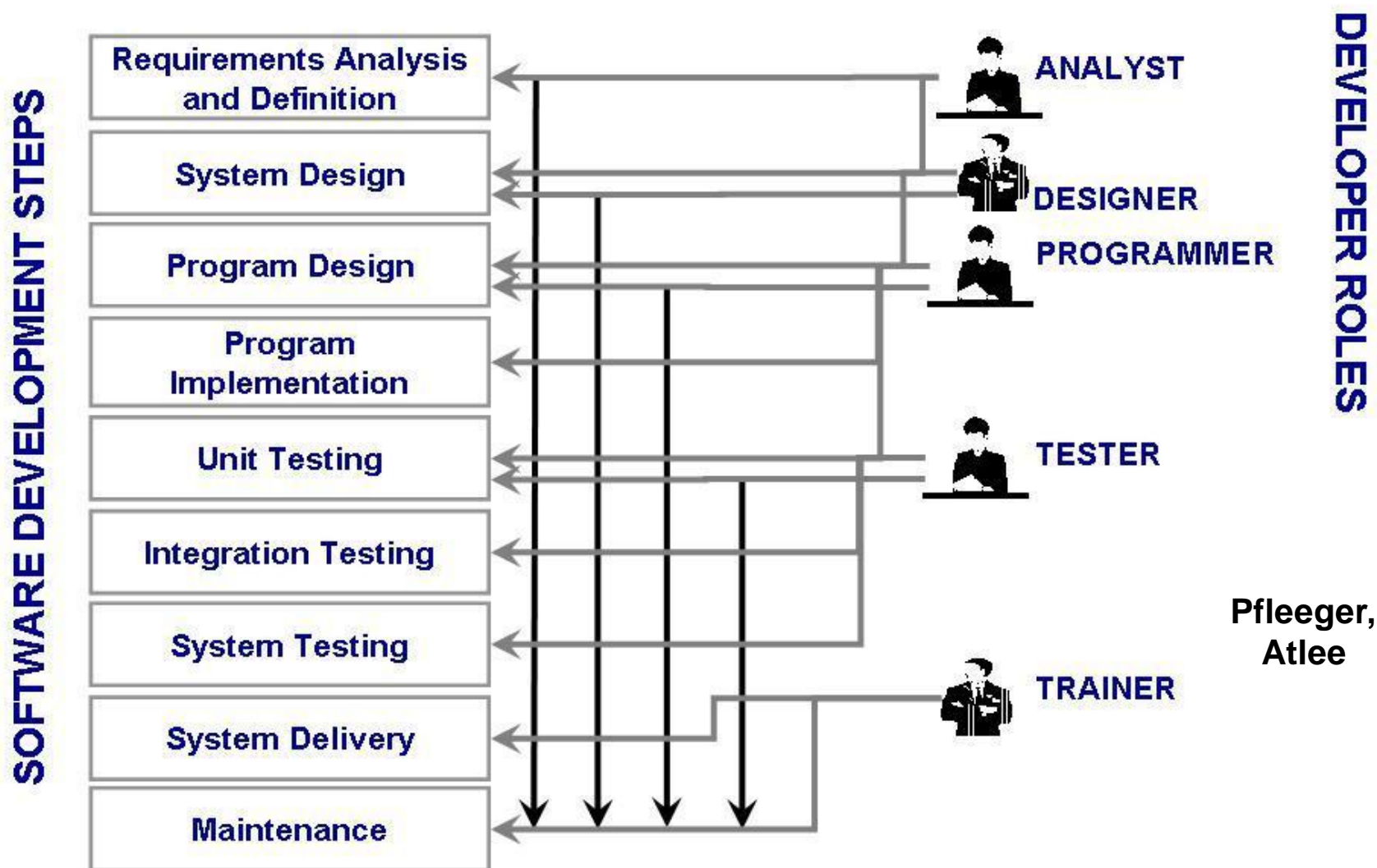


# Razvoj softvera u praksi

- ▶ pravi kupac
  - ▶ pre programiranja: pregovori sa kupcem u cilju raščišćavanja zahteva
  - ▶ česte promene želja kupaca
  - ▶ nesporazumi
- 
- ▶ složeni zadaci—složen program
  - ▶ softver podložen greškama
  - ▶ strogi rokovi
  - ▶ timski rad
- 
- ▶ velika količina dokumenata: videti sledeće slajdove
  - ▶ programiranje čini samo 20 % svih aktivnosti projekta
  - ▶ dugogodišnje korišćenje (10 ... 25 godina)
  - ▶ izmene softvera nakon isporuke (nove želje, novi hardver ...)



# Razvoj softvera je timska aktivnost



Svaki član tima ima svoju ulogu (ne mora biti 1:1 odnos)

# Softverski dokumenti

(Sommerville, Softversko inženjerstvo, 5-to izd., s. 17)

<b>Activnost</b>	<b>Izlazni dokumenti</b>
Analiza zahteva	Studija izvodljivosti Skica zahteva
Definisanje zahteva	Dokumenat zahteva
Specifikacija zahteva	Funkcionalna specifikacija Plan prijemnog testiranja Skica uputstva za upotrebu
Dizajn arhitekture	Specifikacija arhitekture Plan prijemnog testiranja
Dizajn interfejsa	Specifikacija interfejsa Plan integracionog testiranja
Detaljni dizajn	Specifikacija dizajna Plan testiranja elemenata
Kodiranje	Programski kod
Testiranje elemenata	Izveštaj o testiranju elemenata
Testiranje modula	Izveštaj o testiranju modula
Integraciono testiranje	Izveštaj o testiranju integracije Konačno uputstvo za upotrebu
Testiranje sistema	Izveštaj o testiranju sistema
Prijemni test	Završni sistem i dokumentacija

# Literatura

## ► Udžbenici:

- I. Sommerville: Software Engineering (792 strane), 9 izdanja Addison-Wesley (2011)
- R. Pressman: Software Engineering, A Practitioner's Approach, European Adaptation (928 strana), McGraw-Hill, 7 izdanja, 2010
- S.L. Pfleeger, J.M. Atlee, Softversko inženjerstvo, teorija i praksa, prevod trećeg izdanja, CET, 2006. (na engleskom 4 izdanja, 2010, Prentice Hall, 792 strane)