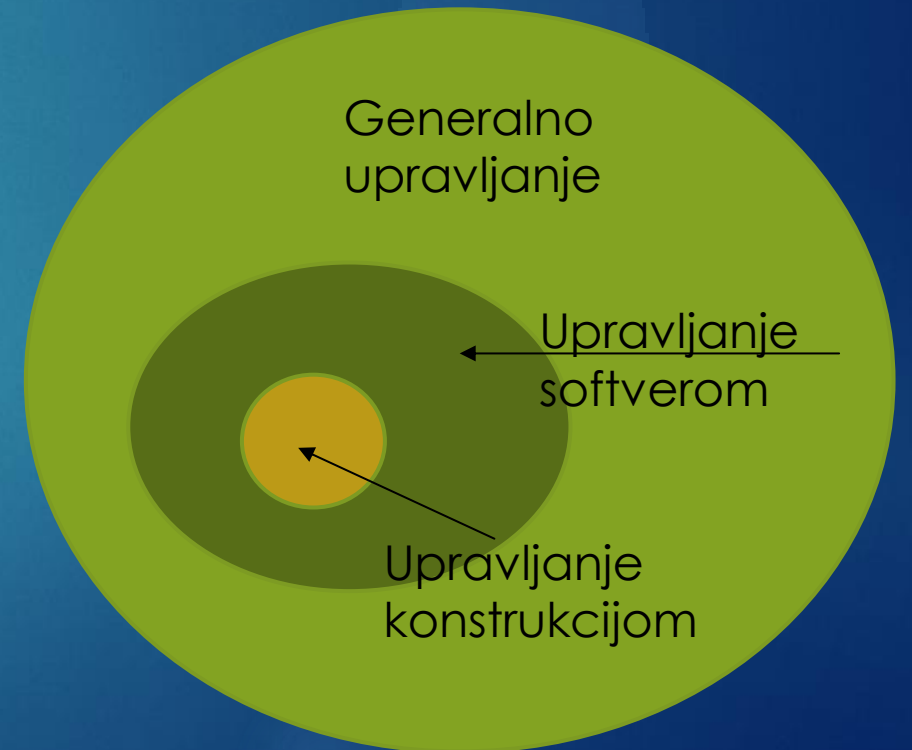




Upravljanje konstrukcijom softvera

SAMOSTALNI PROJEKAT IZ PREDMETA „RAZVOJ SOFTVERA 2”
NEMANAJA SUBOTIĆ

1. Podsticanje dobrog kodiranja
2. Menadžment konfiguracije
3. Procena rasporeda konstruisanja
4. Merenja
5. I programeri su ljudi!
6. Upravljanje menadžerom



1. Podsticanje dobrog kodiranja

- ▶ Utvrđivanje standard
- ▶ Podsticajne tehnike

Podsticajne tehnika dobrog kodiranja

- ▶ Pravilo „rad u paru“
- ▶ Pregled svake linije koda
- ▶ Code sign-offs (potpisivanje od strane seniora)
- ▶ Razgovarajte o dobrim primerima prakse
- ▶ Kodovi nisu privatno vlasništvo
- ▶ Nagrađujte dobre kodove
 1. Zaslužena nagrada
 2. Vredna nagrada
- ▶ Jedan jednostavan standard

2. Upravljanje konfiguracijom

- ▶ Sistematsko obrađivanje izmena
- ▶ **SCM**(„Software Configuration Management“)

Zahtevi i promene dizajna

- ▶ Pratiti sistematsku proceduru
- ▶ Grupno obrađivanje promena
- ▶ Procena svake promene
- ▶ Alarmantan broj promena!
- ▶ Komisija za praćenje promena

Softverske promene u kodu

► Softver za kontrolu verzija

1. Nema uzaludnog, istovremenog rada
2. Jednostavno ažuriranje
3. Pregled svih prijavljenih verzija
4. Spisak svih izmena u verzijama
5. Sigurnost

Ček-lista

Verzije alata

- Da li sistem za kontrolu verzija olakšava upravljanje konfiguracijom?
- Da li sistem za kontrolu verzija olakšavanja tisknog rad?

Plan B

- Blagovremeno bekapujete materijale u projektima?
- Da li ih povremeno šaljete u odvojeno skladište?
- Da li su baš sve informacije od značaja –bekapovane-?
- Da li ste isprobali –bekap- sistemski oporavak?

Ček-lista

Upravljanje konfiguracijom

- Da li vaš softver za upravljanje konfiguracijom olakšava posao programerima?
- Da li timski radite na izmenama ili imate odbor za kontrolu izmena kao više sistematski pristup?
- Da li sistematski procenjujete cenu, vreme, raspored i kvalitet izmena?

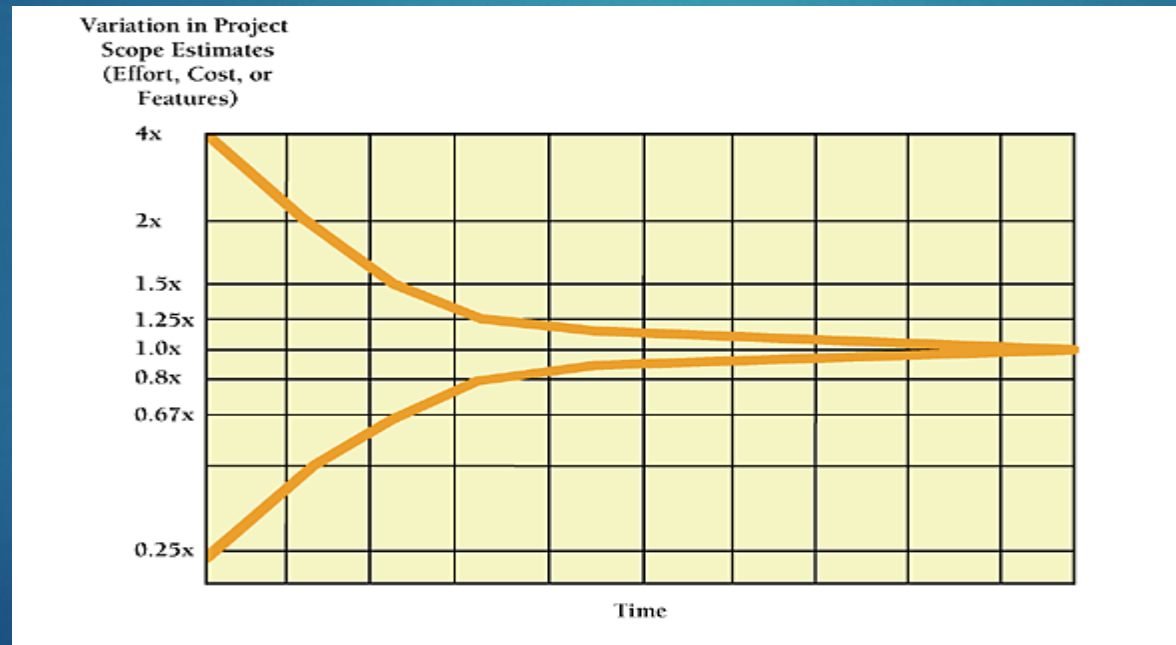
3.Procenjivanje

Pristup proceni

- ▶ Ustanovite ciljeve
- ▶ Odvojite vreme za procene
- ▶ Ustanovite zahteve projekta
- ▶ Procenite detalje
- ▶ Procenite sve različitim metodama, uporedite rezultate
- ▶ Blagovremeno ponovite procene


Procena količine konstrukcije

- ▶ Počnite sa prethodnim iskustvom. Koliko vremena je bilo potrebno u prošlim projektima?
- ▶ Zabeležite nova iskustva za buduće projekte.
- ▶ Treba uspostaviti kontrolu, a ne predviđanje





Šta ako kasnite?

- ▶ Ostanite pozitivni!
 - ▶ Proširite tim
 - ▶ Suzite obim projekta
- 

4.Merenja

- ▶ Svaki merljivi aspekt projekta treba odmeriti
- ▶ Treba biti svestan bočnih efekata
- ▶ Merenje je neizostavno!

Korisna merenja u razvoju softvera

Size

- Total lines of code written
- Total comment lines
- Total number of classes or routines
- Total data declarations
- Total blank lines

Defect Tracking

- Severity of each defect
- Location of each defect (class or routine)
- Origin of each defect (requirements, design, construction, test)
- Way in which each defect is corrected
- Person responsible for each defect
- Number of lines affected by each defect correction
- Work hours spent correcting each defect
- Average time required to find a defect
- Average time required to fix a defect
- Number of attempts made to correct each defect
- Number of new errors resulting from defect correction

Productivity

- Work-hours spent on the project
- Work-hours spent on each class or routine
- Number of times each class or routine changed
- Dollars spent on project
- Dollars spent per line of code
- Dollars spent per defect

Overall Quality

- Total number of defects
- Number of defects in each class or routine
- Average defects per thousand lines of code
- Mean time between failures
- Compiler-detected errors

Maintainability

- Number of public routines on each class
- Number of parameters passed to each routine
- Number of private routines and/or variables on each class
- Number of local variables used by each routine
- Number of routines called by each class or routine
- Number of decision points in each routine
- Control-flow complexity in each routine
- Lines of code in each class or routine
- Lines of comments in each class or routine
- Number of data declarations in each class or routine
- Number of blank lines in each class or routine
- Number of *gotos* in each class or routine
- Number of input or output statements in each class or routine

5. I programeri su ljudi!



100

Activity	Source Code	Business	Personal	Meetings	Training	Mail/Misc. Documents	Technical Manuals	Operating Procedures, Misc.	Program Test	Totals
Talk or listen	4%	17%	7%	3%				1%		32%
Talk with manager		1%								1%
Telephone		2%	1%							3%
Read	14%					2%	2%			18%
Write/record	13%					1%				14%
Away or out		4%	1%	4%	6%					15%
Walking	2%	2%	1%			1%				6%
Miscellaneous	2%	3%	3%			1%		1%	1%	11%
Totals	35%	29%	13%	7%	6%	5%	2%	2%	1%	100%

Source: "Research Studies of Programmers and Programming" (Bairdain 1964, reported in Boehm 1981).

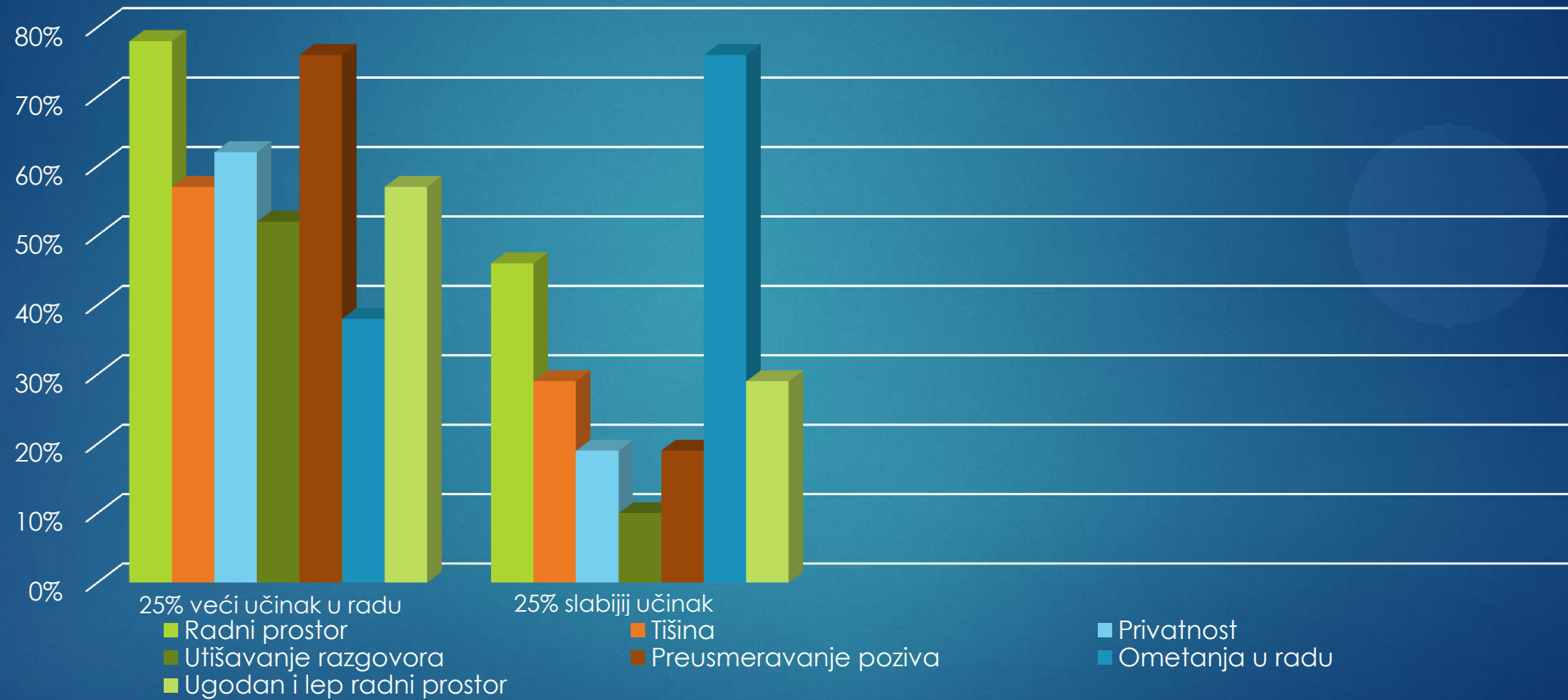
Različitosti

- ▶ Razlike u učinku i kvalitetu
- ▶ Lične razlačitosti
- ▶ Timske razlike

Okruženje




Faktor okruženja



6. Upravljanje menadžerom

- ▶ Tehnički kompetentni menadžeri su retki
- ▶ Pripremite ideje za brainstorming
- ▶ Podučavajte svog menadžera
- ▶ Radite svoj posao na pravi način
- ▶ Nađite drugi posao



*„Programeri i menadžeri su ljudi, i najbolje rade
kada su najbolje tretirani.“*

*Stiv Mek Konel,
„Kopletan kod“*

HVALA NA PAŽNJI!