

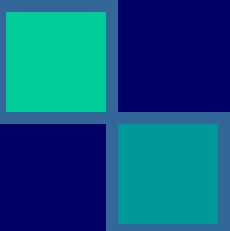

Objektni softverski inženjering



Prof. dr Pere Tumbas
ptumbas@ef.uns.ac.rs

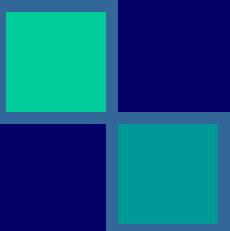



Softver i softverski inženjering

- 
- Softverski proizvod i softver - skup računarskih programa, datoteka i dokumentacije namenjenih realizaciji određenih zadataka (JUS ISO 9127/94).
 - Softverska podrška - rad na održavanju softvera i prateće dokumentacije u funkcionalnom stanju. Pružaju je proizvođač, predstavnik, distributer i dr.
- 

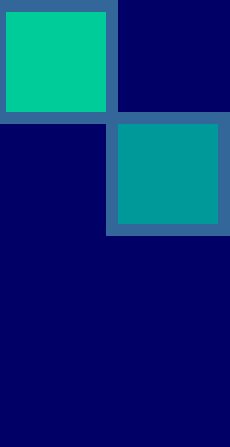



Softver i softverski inženjering

- 
- Softverski inženjering - primena sistematičnog, disciplinovanog i merljivog pristupa razvoju, uvođenju i održavanju softvera.
 - Softverski inženjering - primena inženjeringa, naučnih i matematičkih principa i metoda u ekonomičnoj proizvodnji kvalitetnog softvera.
 - Fritz Bauer, 1968.
- 

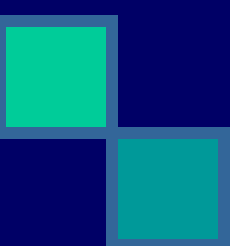



Istorijat razvoja softvera

- **Prva etapa:** proizvođači hardvera za sopstvenu računarsku opremu izrađuju softver, specifičan softver batch orijentacije, sopstvenim kadrovima, niski troškovi, ..
 - **Druga etapa:** opšteupotrebljiv softver, široke namene i on-line orijentacije, softverske kuće, ...
 - **Treća etapa:** softver za distribuirane sisteme, univerzalni softver sa jedinstvenim načinom korišćenja i smanjenim troškovima primene, grafički korisnički interfejs, ...
 - **Četvrta etapa:** softver za primenu na PC povezanim u mrežu, pojava novih tipova informacionih sistema, korisnici i samostalno razvijaju softver, korišćenje Interneta, ...
- 
- 

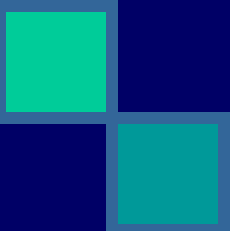



Faktori razvoja softvera:

- 
- Eksplozija informacija,
 - Pojava softverskih kuća,
 - Raznovrsna i brojna primena računara,
 - Potreba za lakoupotrebnim i jeftinim softverom,
 - Brze i stalne promene u društvu,
 - Zahtevi za visokim nivoom kvaliteta softvera,
 - Brza realizacija odnosno kraće vreme dostupnosti,
 - Zahtevi za univerzalnom primenom, ...
- 

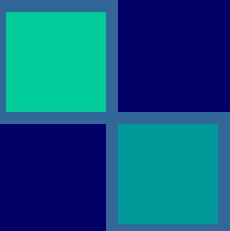


Zahtevi tržišta - uslov industrijske proizvodnje softvera:


- 
- Opšta primenljivost softvera (zadovoljenje različitih potreba),
 - Razvoj softvera uz računarsku podršku i CASE,
 - Realizacija jeftinijeg softvera za korisnika obzirom na veliki broj kupaca,
 - Kraće vreme realizacije softvera,
 - Savremeniji i delotvorniji softveri realizovani od stručnijih i kompetentnijih osoba,
 - Visok nivo kvaliteta softvera, ...
- 



Strukturu softverskog inženjeringa čine:

- Razvojni principi,
 - Metodi,
 - Sredstva (alati) i
 - Procedure (postupci).
- 

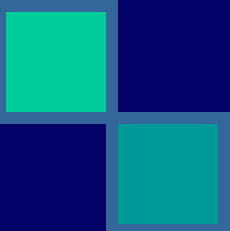

Izbor uslovljen: individualnim i kreativnim postupkom razvoja, različitošću sistema za koji se softver razvija i različitošću okruženja, nepostojanja recepta za izbor.





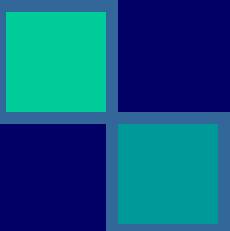

Razvojni principi - određuju način izvršenja rada.

Opšti principi

- 
- **Modelovanje** - koje podrazumeva izbor modela, rukovanje različitim nivoima detaljnosti modela, vezu modela prema stvarnosti, nezavisnost pogleda na celoviti sistem,...
 - **Apstrakcija** - kao rezultat modelovanjem stečenih rezultata, daje specifikaciju modela koja je nedvosmislena, razumljiva, jedinstvena, nezavisna od sistema i lako izmenljiva,
- 



Opšti principi


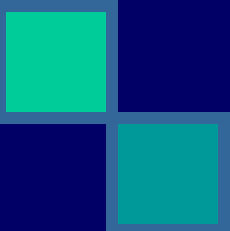
- **Iteracija** - ponovljiv ciklus aktivnosti koji je uzrokovan različitim razlozima: nemogućnost definisanja informacionih zahteva u jednom prolazu usled nepoznavanja ili stalnog usavršavanja zahteva od korisnika, veliki broj neočekivanih situacija, dužina perioda razvoja, potreba za finim usaglašavanjem, doterivanjem i preciziranjem, podizanje nivoa kvaliteta proizvoda,
- 
- 



Opšti principi

- **Arhitektura** - dekomponovanje složene strukture proizvoda na njegove segmente odnosno arhitekturu bilo u analizi i dizajnu, bilo u održavanju.

Razlozi za strukturiranje su: lakša razumljivost složenih sistema, jednostavnije organizovanje razvojnog procesa, lakša ugradnja elemenata u celinu sistema, upravljiviji su kompleksni i složeni sistemi.





Opšti principi


Arhitektura izražava: elemente sistema, njihovu organizaciju, ponašanje strukturnih elemenata, zajedničko ponašanje, način međusobne povezanosti, organizovanje u složenije celine i dr.

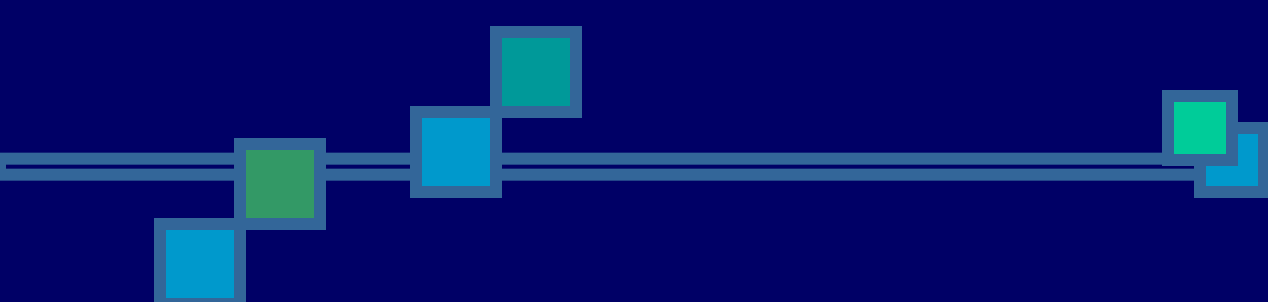
- **Dokumentovanje - , ..**
- 



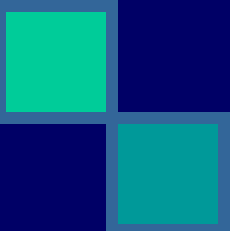

Razvojni principi - određuju način izvršenja rada.

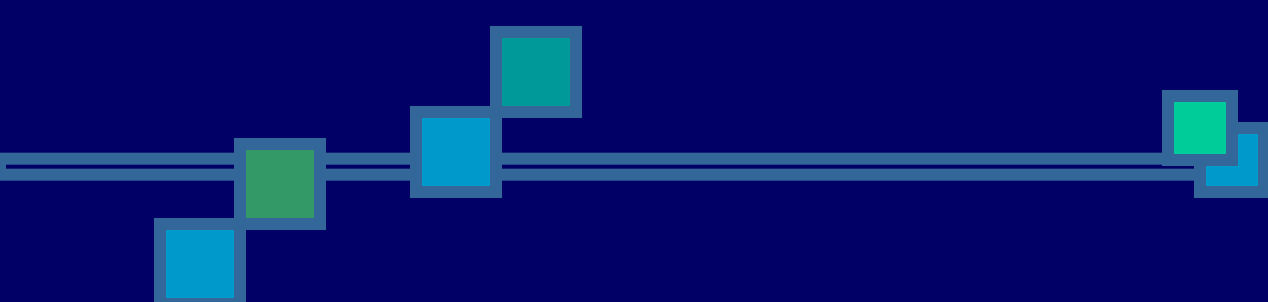
Metodološki koraci - definišu redosled izvršenja aktivnosti, njihove ulaze i izlaze.

- Preskriptivni modeli (model vodopada, ikrementalni model),
 - Razvojni modeli (model prototipskog razvoja, spiralni model),
 - Specijalizovani modeli,
 - Model sjedinjenog procesa,
 - Modeli agilnog razvoja.
- 



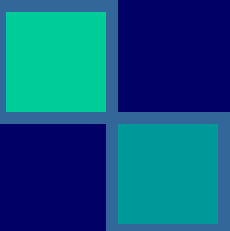
Metodi - propisani sistematski načini u izvršavanju pojedinih zadataka razvoja softvera.

- Mogu se kombinovati i ne postoji najbolji.
 - Strukturiraju se po različitim kriterijumima a najčešće prema fazama razvoja informacionog sistema.
 - Pravilan izbor podrazumeva minimalne troškove i visoki kvalitet proizvoda.
- 
- 




Alati - su automatizovana ili poluautomatizovana podrška metodima.

- CASE tehnologije - posebna vrsta alata.




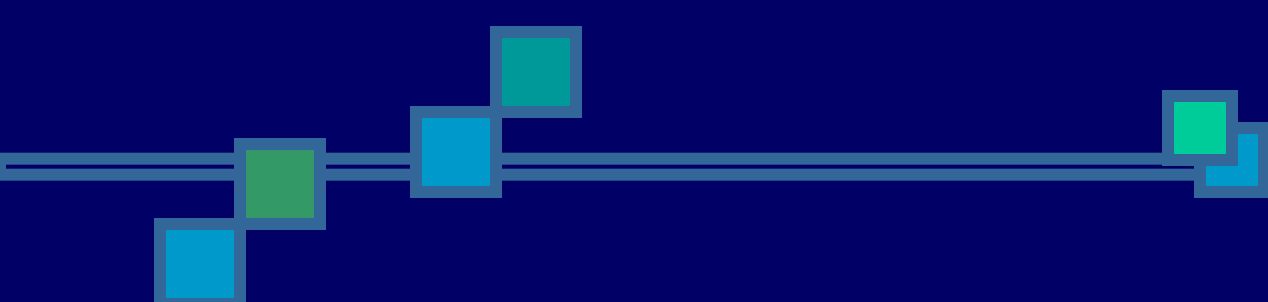
Procedure – su sistemi metoda i alata u realizaciji razvoja informacionog sistema.

- Opređeljuje ih - priroda problema i očekivanje korisnika.
 - Definišu - redosled metoda/koraka, rezultate - izlaze koraka, kontrole radi obezbeđenja kvaliteta realizacije koraka.
- 



Metodologije - sačinjavaju brojne komponente koje definišu:

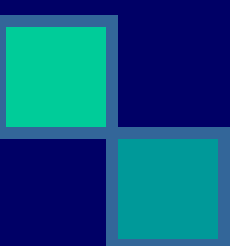

- Kako projekat dekomponovati na faze?
 - Koji zadaci se rešavaju u kojim aktivnostima?
 - Koji se izlazi dobijaju u svakoj fazi?
 - Kada i pod kojim uslovima se oni dobijaju?
 - Koja ograničenja se moraju uzeti u obzir?
 - Koje osobe će biti angažovane?
 - Kako će se rukovoditi projektom i kako će se isti kontrolisati?
 - Koja se sredstva podrške mogu primeniti?
- 

- 
- **STRADIS**
 - **BSP**
 - **Yourdon Systems Method**
 - **SADT**
 - **ISAC**
 - **James Martin IE**
 - **Gane and Sarson**
 - **SSADM**
 - **De Marco**
 - **RAD**

- **SDM - Pandata**
 - **Lorensen**
 - **Martin and Odell**
 - **RUP**
 - **XP**
 - **SCRUM**
 - **FDD**
 - **Crystal**
 - **.....**
- 




Značaj softverskog inženjeringa

- 
- Sistematski pristup razvoju, implementaciji, održavanju softvera kroz životni ciklus,
 - Različito kombinovanje metoda, alata i procedura.
- 

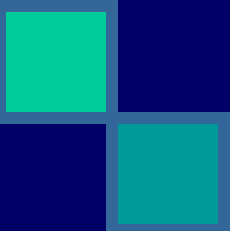



Ciljevi softverskog inženjeringa

- Izrada pouzdanog softvera,
 - Izrada softvera prema originalnim specifikacijama,
 - Izrada softvera prilagodljivog promenama,
 - Izrada softvera koji se može ponovo upotrebiti,
 - Izrada softvera koji zadovoljava korisnika,
 - Razvoj izveden na vreme i efektivno sa aspekta troškova.
 - Razvoj koji zadovoljava korisničke zahteve i
 - Razvoj koji obezbeđuje kvalitetan proizvod.
- 




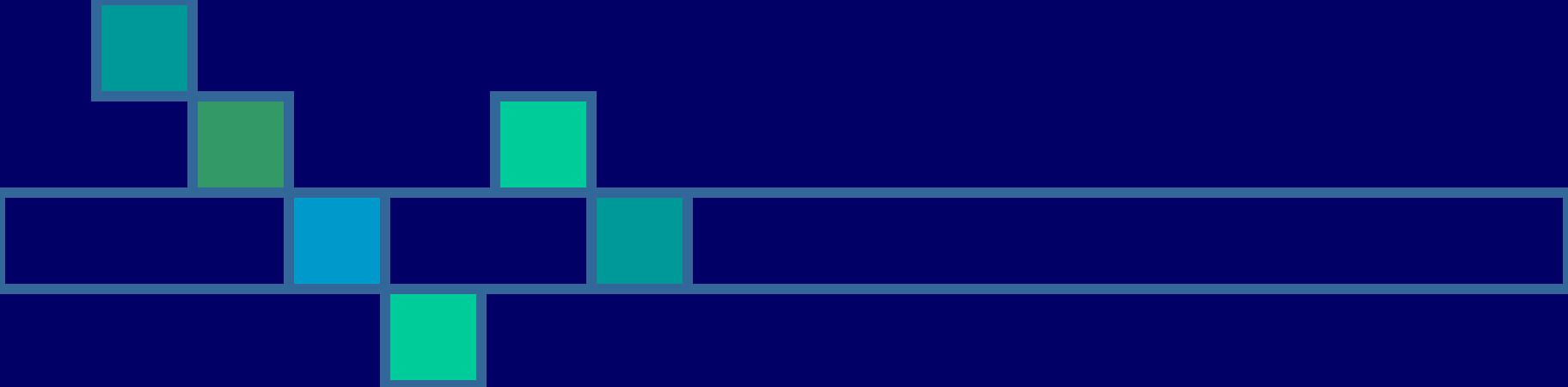
Karakteristike softverskog inženjeringa

- *Jedinstvenost i kompleksnost*
 - *Maksimum principa i ciljeva softverskog inženjeringa,*
 - Funkcionalnost (korektno i tačno),
 - Performantnost (vreme pristupa, protok, brzina,..)
 - Temeljnost u analizi zahteva,
 - Upotreba metodologija i CASE alata,
 - Razvoj odgovarajuće dokumentacije za operativni rad i održavanje,
 - Sposobnost prilagođavanja promenama,
 - Dobro definisani korisnički interfejs,
 - Efektivnost,....
- 
- 



Razvoj softvera u praksi

- Korisnik se bavi svojim svakodnevnim poslom, govori žargonski, ne veruje u promene,
 - Česte izmene korisničkih zahteva,
 - Kompleksan posao,
 - Veliki broj grešaka u testiranju, dosta vremena i novca za njegovu ispravku,
 - Timski rad, koji nije familijaran softver inženjerima,
 - Veliki obim dokumentacije prati složenost posla, dr.
- 



Hvala na pažnji!

