



Poslovno-proizvodni informacioni sistemi

Prof. dr Mirjana Misita



Specifični modeli, okviri i standardi u PPIS

U ovom poglavlju biće prikazani specifični modeli koji se koriste prilikom razvoja, implementacije, ocene i ostalih faza PPIS-a.

U nastavku biće prikazani specifični standardi koji se odnose na PPIS.

Potom, uloge profesionalaca koji se bave PPIS-om i na kraju etika kao sastavni deo PPIS-a.



Modeli za razvoj PPIS-a

Modeli razvoja softvera definišu način organizovanja aktivnosti u životnom ciklusu razvoja sistema. Oni određuju redosled faza razvoja, način upravljanja projektom i pristup testiranju i implementaciji sistema.

Najpoznatiji modeli razvoja softvera su:

- Model vodopada,
- Iterativni model,
- Model rapidnog razvoja aplikacija,
- Model prototipskog razvoja,
- Spiralni model,
- Model zasnovan na komponentama,
- Model unificiranog procesa razvoja,
- Modeli agilnog razvoja, itd.



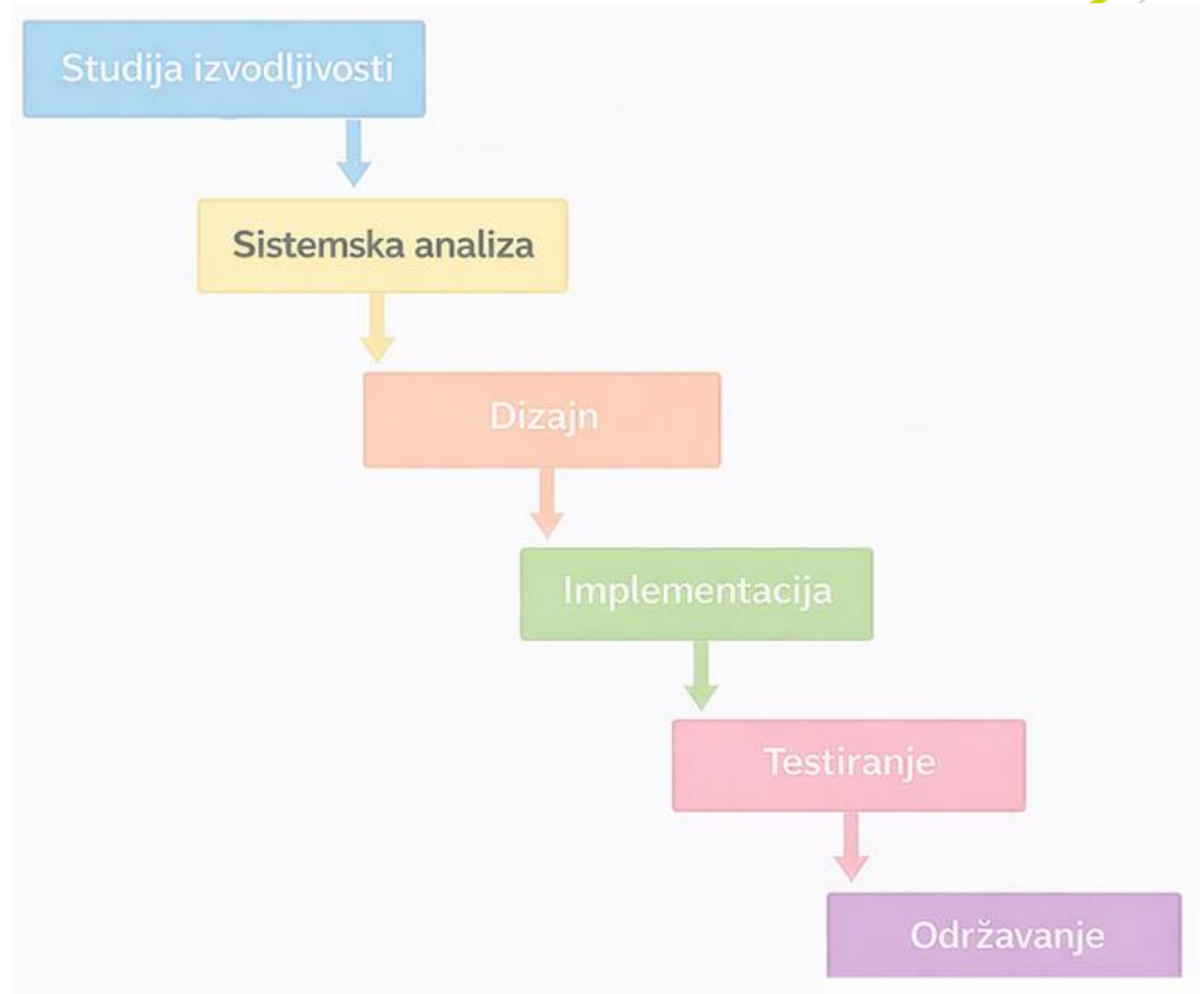
Vodopad model

Najstarija i najpoznatija metodologija koja se koristi za koordinaciju napora u razvoju informacionih sistema poznata je kao „**Waterfall**“ (**model vodopada**).

Model se zasniva na generičkim fazama životnog ciklusa koje vode programere od početne studije izvodljivosti do održavanja završenog sistema. Svaka faza modela postaje ulaz za narednu fazu (videti Prilog 25). Ove faze su opisane na sledeći način:

- **Studija izvodljivosti** - Proučavanje postojećih sistema radi jasnog razumevanja problema i utvrđivanja ciljeva projekta.
- **Sistemska analiza** - Analizira informacije krajnjih korisnika i utvrđuje nedostatke postojećih sistema. Podaci prikupljeni u ovoj fazi koriste se kao deo novog sistema.
- **Dizajn** - Karakteristike i funkcije sistema detaljno se opisuju, uključujući raspored ekrana, poslovna pravila, dijagrame procesa, programske instrukcije i drugu prateću dokumentaciju.
- **Implementacija** - Dizajni se prevode u programski kod korišćenjem programskih jezika.
- **Testiranje** - Sistem se postavlja u testno okruženje radi provere grešaka i nedostataka.
- **Održavanje** - Sistem se uvodi u produkciju u realnom poslovnom okruženju. Počinju da se javljaju izmene, ispravke i dopune, a sistem postepeno evoluira.

Glavna prednost ovog modela je jednostavna struktura i jasno definisane faze razvoja, dok je njegova osnovna slabost ograničena mogućnost prilagođavanja promenama zahteva tokom razvoja sistema.



Slika 2.3. Waterfall – vodopad metodologija



Iterativni model

- Osnova iterativnog razvoja softvera je **razvoj u više ponovljenih ciklusa (iteracija)**.
- U **svakoj iteraciji se ponavljaju sve faze razvoja koje postoje i kod modela vodopada** (Larman, 2003).
- U svakoj iteraciji se **popravlja i doraduje urađeno, dodaju nove mogućnosti i funkcionalnosti softverskog rešenja** koje se razvija.
- Od plana razvoja zavisi koliko će se u konkretnoj iteraciji raditi na doradi i usavršavanju ranije urađenih delova sistema, a koliko na izradi novih komponenti sistema koje se dodaju prethodnom rešenju. Posle implementacije svake nove verzije sistema koriste se povratne informacije od korisnika i na osnovu toga sagledavaju planira se dalji razvoj.
- Model iterativnog razvoja je kreiran tako da se lakše prevaziđe problem detaljnog i pouzdanog definisanja zahteva na početku razvoja, kao i činjenicu da nije moguće predvideti sve probleme u radu pre početka korišćenja i da se novi zahtevi često definišu tek kada je sistem u upotrebi.
- U početnoj iteraciji se najpre razvija neki inicijalni skup funkcionalnosti. Definisanje osnovnih funkcionalnosti softvera omogućava da se već prva verzija rešenja može primeniti u radu.
- Dizajn osnovne arhitekture sistema se obično realizuje u prvoj iteraciji. Nove funkcionalnosti koje se dodaju trenutno razvijenom proizvodu predstavljaju inkrement prethodnog sistema pa se ovaj način razvoja često naziva i **iterativno-inkrementalni**.



Model rapidnog razvoja aplikacija

Model rapidnog razvoja aplikacija (Rapid Application Development - RAD) je inkrementalni model procesa razvoja. Softverski proizvod se razvija u kratkim razvojnim ciklusima, obično u trajanju od 2 do 3 meseca.

Faze razvoja: analiza, dizajn, implementacija i testiranje su naglašeno sabijene u kratke iterativne razvojne cikluse. Proizvod se razvija sa malim timovima. Karakteristike modela su: mali razvojni timovi, razvoj sa kratkim iterativnim razvojnim ciklusima, direktna i efektivna komunikacija članova razvojnog tima i jasna definisanost projektnih ciljeva. Takav pristup omogućava paralelni rad na modelu podataka, modelu procesa, pa i izradu prototipa koji se odmah potom proveravaju, pa sve ponovo u novoj iteraciji. U ovakvom načinu razvoja maksimalno se koriste CASE alati, generatori koda i slični napredni pomoćni softveri. Veoma važno za ovakav razvoj je aktivno uključivanje i saradnja sa svim stejkholderima. Testiranje koje je sastavni deo svake faze razvoja, a koje na svakom inkrementu sprovode članovi razvojnog tima i isto tako i sami korisnici, obezbeđuje da brzi razvoj bude i proveren i potvrđen.

Rapidni razvoj aplikacija ima za cilj znatno brži razvoj i veći kvalitet nego u tradicionalnom modelu vodopada. Ključni ciljevi RAD koncepta su povećana brzina razvoja uz visok kvalitet izlaznog proizvoda i kontrolisani troškovi.

U okviru ovog koncepta prepoznaju se četiri osnovne komponente:

1. Metodologija. Koriste se formalizovane najnaprednije tehnike razvoja, vodi se računa da se sve aktivnosti sprovedu ispravno uz potpuno dokumentovanje;
2. Upravljanje. Projektom se tako upravlja da se obezbedi brzi razvoj, za brzo definisanje korisničkih zahteva koriste se sredstva u formi praktikuma, a u svim fazama RAD razvoja koriste se tehnike vremenskog upravljanja;
3. Ljudski faktor. Članovi razvojnog tima moraju biti dobro obučeni, treba da budu potpuno upoznati sa metodologijom, kao i da napredno koriste softverske alate;
4. Alati. Automatizacija razvoja sistema postiže se upotrebom CASE alata, korišćenjem softvera za automatsku proveru ispravnosti baza podataka, generisanje koda, izradu prototipova i slično, sve to sa ciljem efikasnog iterativnog razvojnog postupka;



Model prototipskog razvoja

- Model prototipskog razvoja je iterativan model.
- Po ovom modelu razvoja najpre se **razvija početni-inicijalni model softverskog proizvoda**. Ovaj početni model softvera obično sadrži **elemente korisničkog interfejsa kako bi korisnici koji su uključeni u razvojni proces od samog početka mogli da iskažu svoje mišljenje i zahteve**.
- Definisana funkcionalna specifikacija budućeg softvera je u ovom modelu razvoja početna osnova za početak razvoja. Prototip se inicijalno razvija, testira i kasnije po potrebi doraduje.
- **Prototip predstavlja osnovu, šablon za razvoj kompletnog proizvoda.**
- Ovakav pristup je praktičan kada se na početku **ne poznaju dovoljno zahtevi korisnika i kada se očekuje postupno definisanje kompletnih zahteva**. Koristi se i kada je potrebno izvesti simulaciju rada proizvoda da bi budući korisnik stekao uvid u očekivani način rada i mogućnosti softvera, kao i kada razvojni tim želi proveriti neki pristup ili karakteristiku sistema.
- Najpre se sprovodi prikupljanje najvažnijih zahteva korisnika. U saradnji članova razvojnog tima i samih korisnika definišu se **ciljevi razvoja proizvoda**. Sprovodi se pojednostavljeni dizajn sa pažnjom na elemente softvera koji direktno komuniciraju sa korisnikom.
- Tako kreirani prototip se modifikuje iterativno sve dok prototip ne zadovolji zahteve korisnika.
- Prototip služi i dizajneru kao sredstvo za bolje razumevanje zahteva, i kao sredstvo komunikacije sa budućim korisnicima. U suštini prototipski razvoj služi i kao mehanizam za identifikovanje korisničkih zahteva.

Spiralni model



Spiralni model predstavlja kombinaciju modela vodopada i prototipskog modela razvoja.

Koristi se za velike i kompleksne projekte. Spiralni model (Boehm, 1986) je predstavljen široj publici 1988. godine. Ovaj model spaja dobre osobine modela vodopada i modela prototipskog razvoja. Kao **specifičnost sadrži aktivnosti analize rizika**. Posebno je preporučen za **razvoj skupih i važnih projekata**.

Vizuelna predstava modela je u obliku spirale. Model predviđa **četiri faze razvoja koje se ponavljaju u ciklusima**:

1. **faza – planiranje.** U ovoj fazi kroz komunikaciju članova razvojnog tima i korisnika realizuje se definisanje ciljeva, određivanje projektnih alternativa i identifikovanje ograničenja procesu razvoja,
2. **faza - analiza alternativa i rizika.** Ova faza sadrži aktivnosti analize alternativa i identifikovanja rizika u upravljanju projektom,
3. **faza – inženjering.** Faza inženjeringa je produktivna faza, u kojoj se radi na razvoju novih elemenata proizvoda i na unapređenju ranije realizovanih delova sistema, testiranju, dokumentovanju i instaliranju rešenja kod korisnika,
4. **faza - procena korisnika.** Faza procene realizovanih rezultata razvoja softverskog proizvoda.

Spirala razvoja u ovom modelu sastoji se višestrukih uzastopnih ciklusa razvoja. Osnovna ideja modela je da se određeni redosled koraka razvoja i održavanja ponavlja.

U prvom-početnom ciklusu spirale, slično modelu vodopada najpre se prikupljaju zahtevi i planira razvoj. Na osnovu prikupljenih početnih zahteva obavlja se analiza rizika zahteva.

Ako se ustanovi određeni nivo neizvesnosti u rešavanju zahteva moguće je pristupiti sigurnijem prototipskom razvoju ili nekoj vrsti simulacije. Posle donošenja odluke o daljem razvoju u početnom kao i u svakom ciklusu spirale, sledi faza inženjeringa sa odabranim modelom razvoja softvera, modelom vodopada, modelom prototipskog razvoja, ili bilo kojim drugim odabranim.

Veoma važno za model spirale, na kraju svakog ciklusa razvoja, korisnik ocenjuje proizvod i daje sugestije za njegovu modifikaciju. Na osnovu povratne sprege od korisnika, priprema se faza planiranja novog ciklusa razvoja i analiza rizika. Na osnovu analize rizika koji je deo svakog ciklusa može se doneti odluka o prekidu daljeg razvoja ili o nastavku.



Model zasnovan na komponentama

- Osnovna ideja ovog modela je uspostavljanje softverskog proizvoda kao organizovanog skupa **pojedinačnih komponenti**.
- Konfigurisanje, grupisanje, specijalizovanje i dokumentovanje komponenti su neke od **osnovnih aktivnosti u ovom pristupu**.
- Osobine komponenti zavise od njihove veličine, složenosti i funkcionalnih mogućnosti. Uobičajeni pristup organizovanju komponenti je eksploatacija sličnosti.
- Razvoj softvera uz ponovno korišćenje komponenti, ako se pravilno i racionalno primenjuje, **skraćuje vreme izrade softvera, povećava robustnost sistema, podiže kvalitet proizvoda zahvaljujući višestrukoj proverbi pojedinačnih komponenti, smanjuje troškove razvoja i značajno olakšava održavanje proizvoda**.

Npr. pod „komponentama” se uglavnom misli na gotove ili polugotove softverske module koji se mogu ponovo koristiti u različitim sistemima.

- To mogu biti: login modul, modul za rad sa bazom, UI komponente (dugmad, forme, tabele), API servisi, biblioteke i framework delovi, mikroservisi, itd., zavisno da li su vezani za frotend ili backend. (Frontend je deo softvera koji korisnik vidi i sa kojim direktno komunicira, Backend je deo softvera koji radi “iza scene” i obrađuje logiku i podatke).



Model unificiranog procesa razvoja

- Model unificiranog procesa razvoja (Unified Process – UP) predstavili su 1999. godine Ivar Jacobson, Grady Booch i James Rumbaugh. Model kombinuje iterativni i inkrementalni pristup razvoju softvera i zasniva se na intenzivnoj upotrebi UML jezika za modelovanje.
- Osnovna ideja modela jeste da se softver razvija postepeno, kroz više iteracija, pri čemu se u svakoj iteraciji realizuje deo funkcionalnosti sistema uz stalno testiranje i unapređivanje.
- Podrazumeva **intenzivno korišćenje UML-a, njihovog zajedničkog objedinjenog jezika modelovanja**. Model je našao široku primenu, omogućio je unapređenje produktivnosti posebno na velikim projektima.
- Posebna vrednost modela je što najviše zahvaljujući UML-u pruža svim učesnicima u razvoju, jedinstven i zajednički skup metoda i jezika za sprovođenje procesa razvoja. Samim tim komunikacija među članovima tima za razvoj je podignuta na visok nivo, svi imaju pristup i razumevanje potrebnim informacijama u obliku baza znanja, projektnim obrascima, raznim uputstvima i preporukama i slično.
- **Ovaj model prikazuje proces razvoja iz vremenske i strukturne perspektive**. Posmatrano iz vremenske perspektive model opisuje proces kroz životni ciklus razvoja softvera - dinamička dimenzija modela. Posmatrano iz strukturne perspektive model opisuju aktivnosti i efekte aktivnosti kroz različite uloge - statička dimenzija modela.
- Posmatrano iz vremenske perspektive predviđa četiri faze razvoja: započinjanje, razrada, konstrukcija i tranzicija. Ove faze se svaka pojedinačno izvršavaju u nekoliko iteracija, pri čemu tačan broj iteracija po fazama zavisi od konkretnog procesa razvoja. Iteracija svakako uključuje aktivnosti analize, dizajna, implementacije i testiranja.



Agilni modeli razvoja

Agilni razvoj softvera je zasnovan na iterativnom i inkrementalnom modelu razvoja. Osnovna ideja agilnog razvoja jeste da se softver razvija postepeno kroz **kratke iteracije (sprintovi)** pri čemu se nakon svake **iteracije isporučuje funkcionalan deo sistema**. Na osnovu povratnih informacija korisnika vrše se izmene i unapređenja u narednim ciklusima razvoja.

Agilni modeli posebnu pažnju posvećuju:

- komunikaciji unutar tima,
- saradnji sa korisnicima,
- brzom povratnoj informaciji,
- kontinuiranom testiranju i unapređivanju sistema,
- prilagodljivosti promenama zahteva.

Agilni razvoj posebno je pogodan za složene i promenljive projekte, dok je manje efikasan u krutim i strogo hijerarhijskim organizacijama.



Agilni model se najčešće koristi za upravljanje projektima, prvenstveno tako što projekte deli na faze ili sprintove.

Ovi sprintovi obično obuhvataju planiranje, dizajniranje, razvoj, testiranje, implementaciju i pregled.

Nakon svakog sprinta, projektni tim analizira napredak i vrši prilagođavanja pre prelaska na sledeći sprint.

Agilni model može biti koristan u fazi kada profesionalci planiraju i razvijaju informacioni sistem. Postoji više verzija Agilnog okvira koje se koriste u upravljanju projektima, uključujući Kanban, Lean i Scrum.

Model razvoja	Osnovna ideja	Prednosti	Nedostaci	Pogodan za
Model vodopada	Razvoj kroz strogo definisane uzastopne faze	Jednostavan i pregledan	Teško prilagođavanje promenama	Manje i jasno definisane projekte
Iterativni model	Softver se razvija postepeno kroz više iteracija	Rano otkrivanje grešaka	Potrebno dobro planiranje iteracija	Projekte sa mogućim izmenama zahteva
Model rapidnog razvoja aplikacija (RAD)	Brza izrada softvera uz intenzivno uključivanje korisnika	Kratko vreme razvoja	Manje pogodan za velike sisteme	Manje i srednje projekte
Model prototipskog razvoja	Izrada prototipa radi boljeg razumevanja zahteva	Lakše definisanje zahteva korisnika	Mogućnost loše organizacije sistema	Projekte sa nejasnim zahtevima
Spiralni model	Razvoj kroz iteracije sa analizom rizika	Dobra kontrola rizika	Složen i skup za realizaciju	Velike i rizične projekte
Model zasnovan na komponentama	Softver se sklapa od gotovih ili polugotovih komponenti	Brži razvoj i lakše održavanje	Zavisnost od kompatibilnosti komponenti	Sisteme sa mogućnošću ponovne upotrebe modula
Model unificiranog procesa razvoja (UP)	Iterativni razvoj uz UML i jasno definisane faze	Dobra organizacija velikih projekata	Veća složenost procesa	Velike i kompleksne sisteme
Agilni modeli razvoja	Fleksibilan razvoj kroz kratke iteracije i stalnu komunikaciju	Brzo prilagođavanje promenama	Teže funkcionisanje u krutim organizacijama	Dinamične i promenljive projekte



Specifični okviri u PPIS

U oblasti poslovno-proizvodnih informacionih sistema razvijeni su različiti okviri koji pružaju smernice za upravljanje informacionim sistemima, organizaciju IT resursa i usklađivanje informacionih tehnologija sa poslovnim ciljevima organizacije.

- COBIT
- ITIL
- McKinsey 7S
- SFIA
- Zachman

- Navedni okviri služe za organizaciju, upravljanje i unapređenje rada informacionih sistema i IT okruženja u organizaciji — ali svaki iz drugog ugla.



Okvir	Čemu služi
COBIT	Za kontrolu i upravljanje IT sistemima u organizaciji. Pomaže da IT podržava poslovne ciljeve, smanji rizike i poveća bezbednost.
ITIL	Za organizaciju i unapređenje IT usluga i podrške korisnicima (help desk, incidenti, održavanje sistema itd.).
McKinsey 7S	Za analizu i unapređenje organizacije, menadžmenta i usklađenosti svih delova kompanije.
SFIA	Za definisanje i procenu znanja, veština i kompetencija zaposlenih u IT sektoru.
Zachman Framework	Za planiranje i organizaciju arhitekture informacionog sistema i celokupne strukture organizacije.



COBIT okvir

COBIT (Control Objectives for Information and Related Technologies) okvir omogućava razvoj, upravljanje i održavanje informacionih sistema u organizaciji.

Ovaj okvir definiše pet osnovnih oblasti procesa koje obuhvataju ključne aktivnosti potrebne za efikasno funkcionisanje informacionih sistema.

Te oblasti su:

- procena, usmeravanje i nadzor,
- usklađivanje, planiranje i organizovanje,
- izgradnja, nabavka i implementacija,
- isporuka usluga i podrška,
- praćenje, evaluacija i procena.



ITIL okvir

Information Technology Infrastructure Library (ITIL) predstavlja okvir najboljih praksi za upravljanje IT uslugama u organizaciji.

Osnovna ideja ITIL-a jeste da IT sektor ne treba samo da „održava računare i servere“, već da **organizovano pruža kvalitetne IT usluge korisnicima i podržava poslovanje firme.**

ITIL definiše:

- kako se rešavaju incidenti,
- kako se upravlja promenama sistema,
- kako se organizuje tehnička podrška,
- kako se održavaju servisi,
- kako se prati kvalitet IT usluga.



McKinsey 7S okvir

Naziv **McKinsey 7S okvir** potiče od činjenice da model obuhvata **sedam ključnih elemenata organizacije**, čija imena na engleskom jeziku počinju slovom **S**. Ovaj koncept razvili su konsultanti kompanije McKinsey početkom 1980-ih godina kao alat za analizu i unapređenje organizacione efikasnosti.

Osnovna ideja modela jeste da uspešnost organizacije zavisi od usklađenosti sedam međusobno povezanih elemenata („7S“):

- Strategy (strategija),
- Structure (struktura),
- Systems (sistemi),
- Style (stil upravljanja),
- Staff (zaposleni),
- Skills (veštine),
- Shared Values (zajedničke vrednosti).

Elementi modela dele se na tvrde i meke elemente. Tvrđi elementi obuhvataju strategiju, strukturu i sisteme. Oni su formalno definisani, dokumentovani i lakše merljivi. Meki elementi obuhvataju stil upravljanja, zaposlene, veštine i zajedničke vrednosti. Oni se odnose na organizacionu kulturu, međuljudske odnose, kompetencije zaposlenih i način funkcionisanja organizacije.










SFIA okvir

- **SFIA** (Skills Framework for the Information Age) predstavlja sveobuhvatan okvir veština i kompetencija za IT industriju. Razvila ga je i nadgleda SFIA Foundation.
- Organizacije širom sveta koriste SFIA da mapiraju znanje i ekspertizu potrebnu za svaku ulogu u organizaciji.
- Osnovna svrha SFIA okvira jeste da organizacijama pomogne u planiranju i upravljanju ljudskim resursima u oblasti informacionih sistema.
- Korišćenjem ovog okvira organizacije mogu jasno definisati koje veštine i znanja su potrebni za određene IT uloge, proceniti postojeće kompetencije zaposlenih i planirati njihov profesionalni razvoj.

SFIA NIVOI ODGOVORNOSTI I STRUČNOSTI



NIVO	NAZIV NIVOVA	OPIS	KLJUČNE KARAKTERISTIKE	PRIMERI ULOGA
1	 PRAĆENJE INSTRUKCIJA	Radi pod neposrednim nadzorom. Izvršava jednostavne zadatke po jasnim instrukcijama.	<ul style="list-style-type: none"> • Radi po jasnim uputstvima • Ograničena autonomija • Fokus na ličnom učenju 	Pripravnik, pomoćni tehničar, operater unosa podataka
2	 ASISTIRANJE I PODRŠKA	Pružna podršku i radi uz blizak nadzor. Obavlja rutinske zadatke i pomaže drugima.	<ul style="list-style-type: none"> • Radi uz povremeni nadzor • Pruža podršku timu • Počinje da koristi osnovno znanje 	IT podrška (junior), tehničar, administrator sistema
3	 PRIMENA ZNANJA I VEŠTINA	Primjenjuje stečeno znanje i veštine za obavljanje zadataka. Radi samostalno u okviru jasno definisanih uputstava.	<ul style="list-style-type: none"> • Samostalno obavlja zadatke • Primjenjuje znanje i veštine • Rešava uobičajene probleme 	Programer, sistem inženjer, analitičar, specijalista
4	 OMOGUĆAVANJE I PODRŠKA PROCESIMA	Ima širu odgovornost za obavljanje zadataka i podršku procesima. Može nadzirati druge.	<ul style="list-style-type: none"> • Samostalno planira i prioritizuje • Podržava procese i druge ljude • Preuzima odgovornost za kvalitet 	Voda tima, senior inženjer, koordinator projekta
5	 OBEZBEĐIVANJE I SAVETOVANJE	Pružna stručne savete i osigurava efikasnost rešenja. Ima značajan uticaj na rezultate u oblasti svog rada.	<ul style="list-style-type: none"> • Pruža stručne savete • Utice na odlučivanje • Odgovoran za isporuku rešenja 	IT konsultant, arhitekta rešenja, menadžer projekta
6	 INICIRANJE I UTICAJ	Ima značajan uticaj na strategije i rezultate. Inicira i vodi promene. Odgovoran je za oblasti rada.	<ul style="list-style-type: none"> • Formuliše i implementira planove • Vodi inicijative i promene • Ima značajan uticaj 	IT menadžer, menadžer programa, rukovodilac odeljenja
7	 POSTAVLJANJE STRATEGIJE, INSPIRISANJE I MOBILISANJE ORGANIZACIJE	Odgovoran za definisanje strategija i vođenje organizacije. Ima najširi uticaj i odgovornost.	<ul style="list-style-type: none"> • Definiše strategije i pravac • Inspiriše i mobilizuje organizaciju • Odgovoran za dugoročne rezultate 	CIO, direktor, izvršni rukovodilac, strateški lider



Zachman-ov okvir

Zachman okvir predstavlja jedan od najpoznatijih modela za organizovanje i opisivanje arhitekture informacionih sistema u organizaciji. Ovaj okvir koristi matricu dimenzija 6×6 kroz koju se analiziraju ključna pitanja: šta, kako, gde, ko, kada i zašto.

Tabela 2.1. Matrica 6x6 za Zachaman-ov okvir

	Šta (Podaci)	Kako (Funkcija)	Kada (Vreme)	Ko (Ljudi)	Gde (Lokacija)	Zašto (Motivacija)
Opseg (Kontekstualni)	Lista stvari	Lista procesa	Lista događaja	Lista organizacija	Lista lokacija	Lista ciljeva
Model preduzeća (Konceptualni)	Semantički model	Model poslovnog procesa	Glavni raspored	Model toka rada	Logistička mreža	Poslovni plan
Model sistema (Logički)	Logički model podataka	Arhitektura aplikacije	Struktura obrade	Arhitektura korisničkog interfejsa	Arhitektura distribuiranog sistema	Model poslovnih pravila
Tehnološki model (Fizički)	Fizički model podataka	Dizajn sistema	Struktura kontrole	Arhitektura prezentacije	Tehnološka arhitektura	Dizajn pravila
Implementacija (Detaljni)	Definicija podataka	Programi	Definicija vremena	Arhitektura bezbednosti	Arhitektura mreže	Specifikacija pravila
Funkcionalno preduzeće	Upotreblijivi podaci	Funkcionalna funkcija	Operativna mreža	Funkcionalna organizacija	Implementirani raspored	Operativna strategija


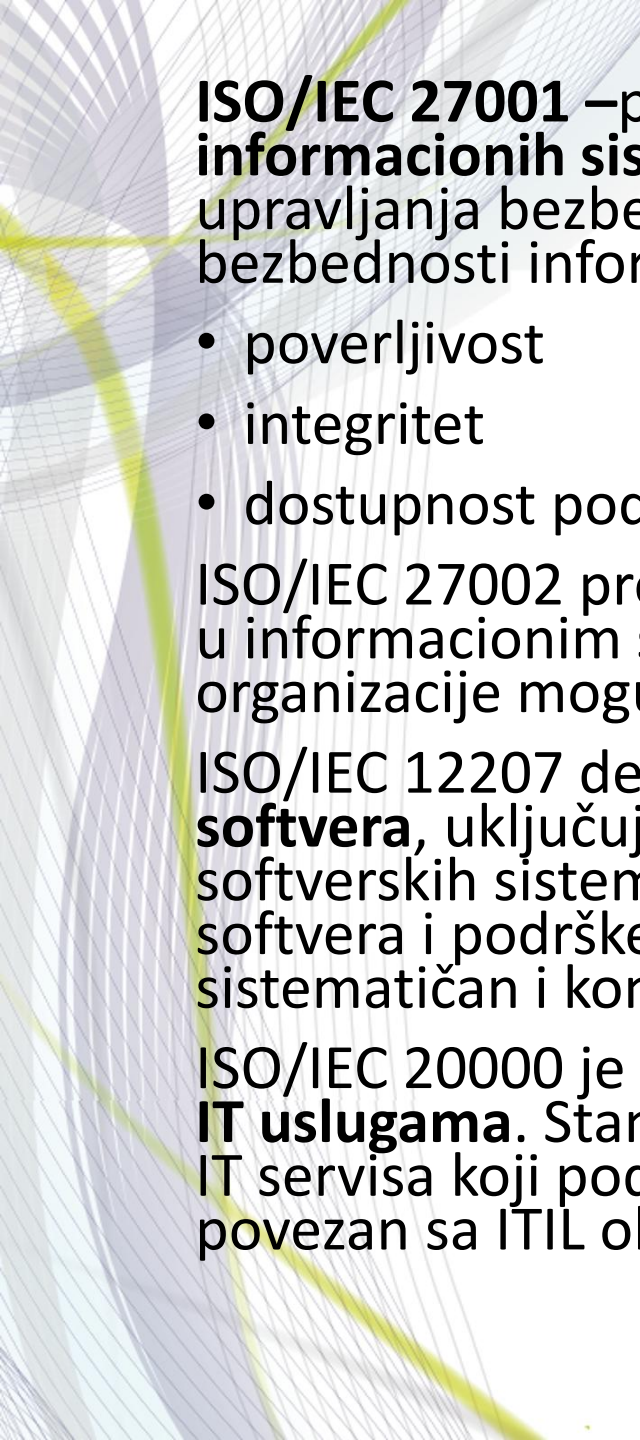


- Ovaj okvir je razvio **John Zachman** 1987. godine sa ciljem da obezbedi sistematičan pristup analizi, projektovanju i upravljanju složenim informacionim sistemima. Zachman okvir pomaže organizacijama da jasno strukturiraju sve elemente informacionog sistema i njihove međusobne odnose.
- Osnovu ovog okvira čini **matrica dimenzija 6 × 6**, koja omogućava posmatranje organizacije iz različitih perspektiva i kroz različite aspekte sistema. Kolone matrice predstavljaju ključna pitanja koja se postavljaju prilikom analize sistema: **šta, kako, gde, ko, kada i zašto**. Ova pitanja se odnose na podatke, procese, lokacije, učesnike, vremenske događaje i poslovne ciljeve sistema.



Industrijski standardi

- ISO/IEC 27001 je međunarodni standard za bezbednost informacionih sistema, fokusiran na dostupnost, integritet i poverljivost podataka.
- ISO/IEC 12207 – životni ciklus softvera
- ISO/IEC 20000 – upravljanje IT uslugama
- ISO 9001 – sistem upravljanja kvalitetom
- ISO 22301 – upravljanje kontinuitetom poslovanja
- ISA-95 (IEC 62264)
- ISO 22400 – performanse proizvodnih sistema,
- i drugi.



ISO/IEC 27001 –predstavlja međunarodni standard za **upravljanje bezbednošću informacionih sistema**. Standard definiše zahteve za uspostavljanje sistema upravljanja bezbednošću informacija (ISMS). Fokusira se na tri osnovna principa bezbednosti informacija:

- poverljivost
- integritet
- dostupnost podataka.

ISO/IEC 27002 predstavlja skup smernica za implementaciju bezbednosnih kontrola u informacionim sistemima. Standard definiše različite bezbednosne mere koje organizacije mogu primeniti kako bi zaštitile svoje informacione sisteme.

ISO/IEC 12207 definiše procese koji se koriste tokom **životnog ciklusa razvoja softvera**, uključujući planiranje, razvoj, implementaciju, testiranje i održavanje softverskih sistema. Standard obuhvata procese upravljanja projektom, inženjeringa softvera i podrške razvoju sistema. Njegova primena omogućava organizacijama sistematičan i kontrolisan razvoj softverskih rešenja.

ISO/IEC 20000 je međunarodni standard koji definiše zahteve za **sistem upravljanja IT uslugama**. Standard obuhvata procese planiranja, implementacije i unapređenja IT servisa koji podržavaju poslovne procese organizacije. Ovaj standard je usko povezan sa ITIL okvirom i koristi se za standardizaciju upravljanja IT servisima.



- ISO 9001 predstavlja međunarodni standard za **upravljanje kvalitetom u organizacijama**. Standard definiše principe upravljanja kvalitetom koji se mogu primeniti i u razvoju i održavanju informacionih sistema. Njegova primena doprinosi standardizaciji procesa, povećanju efikasnosti organizacije i unapređenju kvaliteta proizvoda i usluga.
- ISA-95 predstavlja međunarodni standard za integraciju **poslovnih i proizvodnih informacionih sistema**. Standard definiše modele razmene informacija između sistema za planiranje poslovanja (ERP) i sistema za upravljanje proizvodnjom (MES). Ovaj standard omogućava bolju integraciju proizvodnih procesa i poslovnog upravljanja.
- ISO 22400 definiše **ključne indikatore performansi (KPI)** za merenje efikasnosti proizvodnih operacija. Ovaj standard se koristi u sistemima za upravljanje proizvodnjom kako bi se pratila produktivnost, kvalitet i efikasnost proizvodnih procesa.



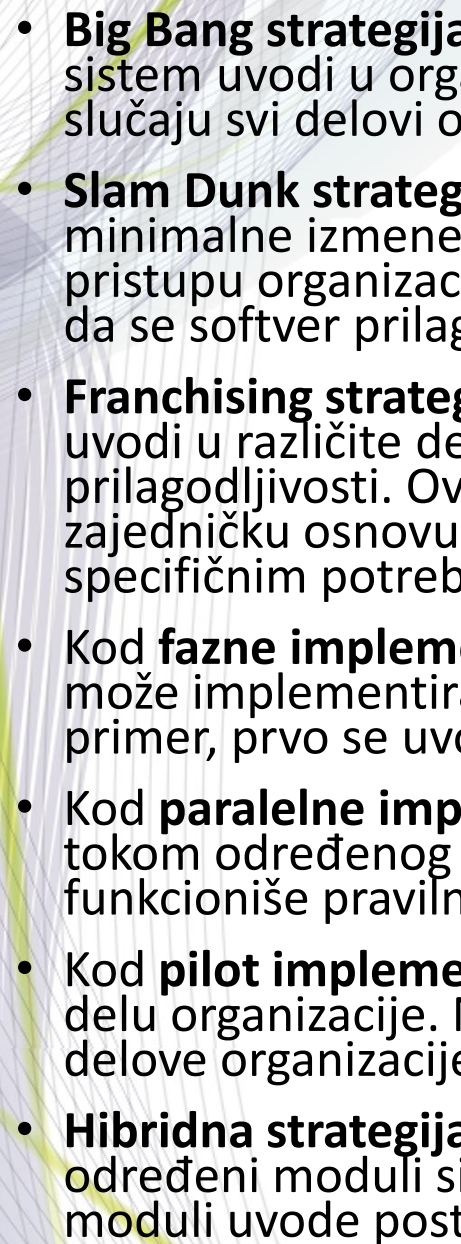

Strategije implementacije poslovno-proizvodnih informacijskih sistema (PPIS)

- Implementacija poslovno-proizvodnih informacijskih sistema predstavlja proces uvođenja novog informacijskog sistema u organizaciju i njegovog povezivanja sa postojećim poslovnim i proizvodnim procesima. Ovaj proces je često složen i zahteva pažljivo planiranje kako bi se smanjili rizici, izbegli prekidi u radu organizacije i obezbedila uspešna primena sistema.
- U praksi se koriste različite strategije implementacije informacijskih sistema, među kojima najpoznatije klasične strategije su: **Big Bang**, **Slam Dunk** i **Franchising** strategija.



Strategije implementacije PPIS

Strategija	Osnovna ideja
Big Bang	sistem se uvodi odjednom u celoj organizaciji
Slam Dunk	implementacija ograničena na osnovne procese
Franchising	implementacija po organizacionim jedinicama
Fazna implementacija	postepeno uvođenje sistema po modulima
Paralelna implementacija	istovremeni rad starog i novog sistema
Pilot implementacija	testiranje sistema u jednoj jedinici
Hibridna implementacija	kombinacija više strategija

- 
- 
- **Big Bang strategija** predstavlja pristup implementaciji informacionog sistema u kojem se novi sistem uvodi u organizaciju odjednom, zamenjujući postojeći sistem u jednom trenutku. U ovom slučaju svi delovi organizacije počinju istovremeno da koriste novi informacioni sistem.
 - **Slam Dunk strategija** podrazumeva implementaciju informacionog sistema u organizaciji uz minimalne izmene postojećih poslovnih procesa. U specifičnim potrebama organizacije. ovom pristupu organizacija se prilagođava standardnim funkcionalnostima softverskog rešenja, umesto da se softver prilagođava
 - **Franchising strategija** predstavlja pristup implementaciji informacionog sistema u kojem se sistem uvodi u različite delove organizacije na način koji omogućava određeni nivo lokalne prilagodljivosti. Ovaj pristup omogućava različitim poslovnim jedinicama ili odeljenjima da koriste zajedničku osnovu informacionog sistema, ali uz određene modifikacije koje odgovaraju njihovim specifičnim potrebama.
 - Kod **fazne implementacije** informacioni sistem se uvodi postepeno, odnosno u više faza. Sistem se može implementirati po modulima, po organizacionim jedinicama ili po poslovnim funkcijama. Na primer, prvo se uvodi finansijski modul, zatim modul za proizvodnju, a potom modul za logistiku.
 - Kod **paralelne implementacije** novi informacioni sistem radi istovremeno sa starim sistemom tokom određenog vremenskog perioda. Na taj način organizacija može proveriti da li novi sistem funkcioniše pravilno, dok stari sistem ostaje kao rezervna opcija.
 - Kod **pilot implementacije** novi sistem se prvo uvodi u jednoj organizacionoj jedinici ili jednom delu organizacije. Nakon testiranja i eventualnih korekcija, sistem se postepeno uvodi i u ostale delove organizacije
 - **Hibridna strategija** predstavlja kombinaciju različitih strategija implementacije. Na primer, određeni moduli sistema mogu biti implementirani odjednom (Big Bang pristup), dok se drugi moduli uvode postepeno kroz faznu implementaciju.



Karakteristike i uloge profesionalaca u PPIS

Profesionalci u PPIS imaju različite odgovornosti i funkcije u organizaciji. One uključuju sledeće:

- Glavni informacioni direktor (CIO) uspostavlja i održava ukupne informacione sisteme organizacije. Odgovornosti CIO-a uključuju obezbeđivanje da sistemi budu usklađeni sa zakonskim zahtevima i da ostali koji su uključeni u informacione sisteme organizacije obavljaju svoje poslove kompetentno.
- Upravljanje PPIS-om podrazumeva upravljanje ljudima, tehnologijom i procedurama potrebnim za pretvaranje podataka u informacije. Ovo uključuje čišćenje, izdvajanje, integraciju, kategorizaciju, označavanje i organizovanje podataka.
- Upravljanje bazama podataka razvija procedure za organizovanje, manipulaciju i preuzimanje podataka koji su skladišteni u računarskim bazama podataka.
- Analiza, dizajn i razvoj sistema ispituju potrebe sistema organizacije i projektuju i razvijaju sistem kako bi te potrebe bile zadovoljene.



- Upravljanje bezbednosnim rizicima PPIS upravlja rizicima koji prete informacionom sistemu organizacije.
- Bezbednost preduzeća, privatnost podataka i upravljanje rizicima fokusiraju se na pretnje, kao što su povrede bezbednosti podataka, sajber napadi i rizici po privatnost podataka, koje mogu ugroziti podatke i informacione sisteme organizacije.
- Računarstvo u oblaku fokusira se na način na koji organizacija koristi informacione sisteme u oblaku za svrhe kao što su skladištenje i obrada podataka.
- Analitika podataka i modeliranje transformišu sirove podatke u korisne informacije i analiziraju te podatke kako bi obezbedili informacije korisne za donošenje odluka u organizaciji i druge operacije.
- Upravljanje projektima u PPIS koristi korake upravljanja projektima — pokretanje, planiranje, izvršenje, praćenje, kontrolu i zatvaranje — kako bi upravljalo IS projektima.



Etika kao sastavni deo PPIS

- Svaki profesionalac u PPIS verovatno će morati da upravlja osetljivim podacima, a nepravilno rukovanje njima može negativno uticati na poslovanje organizacija, kao i na živote pojedinaца.
- Sajber bezbednost je prioritet jer širom sveta hakeri neprestano traže organizacije sa ranjivim sistemima koji se mogu iskoristiti radi finansijske dobiti i drugih kriminalnih aktivnosti.
- Profesionalci u PPIS moraju razumeti rizike informacionih sistema i primenjivati etičko ponašanje kako bi upravljali tim rizicima. Važno je imati na umu da svaki deo informacionih sistema mora biti vođen etičkim pristupom, uz razumevanje njegove važnosti i svest da profesionalci u IS imaju obavezu da učine sve što mogu kako bi zaštitili podatke.