

# 1 CJELOVITI PRISTUP RAZVOJU INFORMACIONIH SISTEMA

Od naučne konferencije održane 1968./69. godine u organizaciji NATO saveza, na kojoj je uveden pojam “softverski inženjering”, razvijen je niz metoda koje se primjenjuju u izgradnji informacionih sistema (IS). Iako su sve one dobre u smislu da olakšavaju provedbu pojedinih faza razvoja IS, pravi sinergijski učinak njihove primjene dolazi do izražaja tek kad se primijene kao dio cjelovite metodologije za razvoj IS. Iako postoji određeni broj ovakvih cjelovitih metodologija, koje su poznate i dokazane u praksi, ne može se tvrditi da je bilo koja od njih pogodna za razvoj svih IS. Izbor metodologije razvoja IS, koja pokriva sve faze njegova razvoja, ovisi o karakteristikama organizacije u kojoj posmatrani IS djeluje. Optimalan pristup razvoju IS bio bi onaj u kojem se vrši izbor metoda, a ne gotovih metodologija, te se izabrane metode uzročno posljedično povezuju u cjelovitu metodologiju razvoja IS. Postoji više poznatih metodologija koje pokrivaju gotovo cijeli životni ciklus razvoja IS.

1960-tih godina dva značajna trenda su karakterizirala informatičku zajednicu. S jedne strane, pojavila su se dovoljno moćni računari (hardver), a s druge strane smo imali široko raspoložive više programske jezike. To je rezultiralo primjenom računara u izgradnji informacionih sistema (IS) za različite tipove organizacija. Veliki postotak ovih informacionih sistema je bio neuspješan, kasnio u razvoju ili premašio predviđeni budžet. Ovi problemi doveli su do stanja koje je široko prepoznato kao “kriza softvera” [IAC]. Kriza je pokazala da je u razvoju softvera za IS potreban metodološki pristup. NATO je 1968. i 1969. godine sponzorisao sastanke na kojima je ovaj problem jasnije definisan i utvrđeni su određeni početni koraci [NATO]. Pojavio se koncept “softversko inženjerstvo”, zasnovan na ideji da bi se u razvoju softvera trebao primijeniti pristup sličan inženjerskom. Iako su krajem 60-tih i početkom 70-tih godina provedena istraživanja u području softverskog inženjerstva, imala su mali uticaj na praktični razvoj softvera. Ipak, razvijeni su najvažniji koncepti u ovom području, kao što su top-down oblikovanje, poboljšanje korak po korak, modularnost i strukturno programiranje. Ovi koncepti prerasli su u metode, te su predstavljali veliki zaokret u pristupu razvoju softvera, te razvoju IS općenito. Vremenom su se profilirala dva značajna pristupa u razvoju IS: strukturni i objektni. Svaki od ovih pristupa razvio je niz metoda koje pokrivaju različite faze životnog ciklusa razvoja IS. Svaka metoda u razvoju IS olakšava projektantima posao, jer na formalan ili poluformalan način omogućava prikupljanje, čuvanje i prezentiranje znanja o organizaciji za koju se IS razvija. Međutim, metode razvoja IS su potpuno iskoristive jedino u slučaju da se primjenjuju u okviru cjelovite metodologije, koja pokriva veći dio životnog ciklusa razvoja IS. Općenito, da bi pristup razvoju IS bio cjelovit, odabrana metodologija razvoja IS mora zadovoljiti tri uslova [Dobr]:

- ◆ definisan je skup metoda koje čine metodologiju s dobro definisanim rezultatima primjene svake metode;
- ◆ ukupni rezultati primjene metodologije mogu se prikazati jednim meta modelom;
- ◆ definisan je slijed uporabe metoda koje čine metodologiju.

## 1.1 Definisanje osnovnih pojmova

**Metodologija** je nauka o cjelokupnosti svih oblika i načina istraživanja pomoću kojih se dolazi do objektivnog i sistemskog naučnog znanja. Metodologija o projektovanju i gradnji informacionih sistema je nauka o razvoju informacionih sistema.

Metodologije detaljnije opisuju faze razvoja i aktivnosti pojedine faze na najnižem potrebnom nivou detalja. Metodologija opisuje veze između pojedinih aktivnosti i propisuje niz izvođenja aktivnosti. Ona određuje kada započeti i kada završiti s aktivnošću te koji podaci (informacije) jesu ulazni u aktivnost a koji su rezultat rada aktivnosti. Za svaku aktivnost je određena metoda i tehnika kojom se aktivnost izvodi te propisuje kvaliteta izlazne dokumentacije.

Pod metodologijom za projektovanje, izgradnju i održavanje podrazumijeva se sveukupnost principa, pravila, metoda i tehnika koje se koriste u postizanju cilja projektovanja, izgradnje i održavanja informacijskog sistema. Metodologije su danas "komercijalni" proizvodi. Njihovi autori su tvrtke i pojedinci koji su svojim radom unaprijedili proces izrade informacijskog sistema.

Metodologija odabire metode, prilagođava ih konačnom cilju, propisuje redoslijed upotrebe metoda, propisuje proces modeliranja od početka do kraja životnog ciklusa informacijskog sistema.

**Metoda** je definisani postupak djelovanja za postizanje određenog cilja na nekom praktičnom ili teorijskom području. Metoda je način istraživanja ili praktičnog postupanja i djelovanja kako bi se došlo do nekog rezultata. Metoda je racionalni postupak kako doći do spoznaje ili znanja, ili kako da se dokaže istina.

Primjenom procesa modeliranja zasnovanog na nekoj metodi kao rezultat dobiva se model. U raznim fazama životnog ciklusa informacijskog sistema primjenjuju se razne metode.

**Model** je pojednostavljena reprezentacija o relevantnim svojstvima sistema. Modeliranje je proces razvoja modela. Model nastaje procesom apstrakcije u kojem se prvo biraju relevantni elementi koje reprezentacija treba da sadrži, a zatim se svakom elementu pridružuju relevantne osobine, koje se žele prikazati u okviru modela.

Model je pojednostavljeni prikaz relevantnih karakteristika sistema. Model je reprezentacija nekih objekata, veza među objektima i atributa objekata. Direktno postoji samo sistem, a svaki naš prikaz sistema je model. Sistem se opisuje pomoću modela - preko modela.

**Modeliranje – proces izrade modela.** Modelirajući sistem danom metodom projektant može izraditi jedan ili više modela sistema. Proces modeliranja je različit za razne metode. Modelirajući odabranom metodom gradi se model danom metodom.

**Projektovanje informacionih sistema** je nalaženje relevantnog modela realnog sistema. Na osnovi projektovanog modela izgrađuje se informacioni sistem. Model informacijskog sistema sastoji se u osnovi od tri modela: model podataka, model procesa i model resursa.

**Model podataka** prikazuje stanje sistema preko skupa podataka.

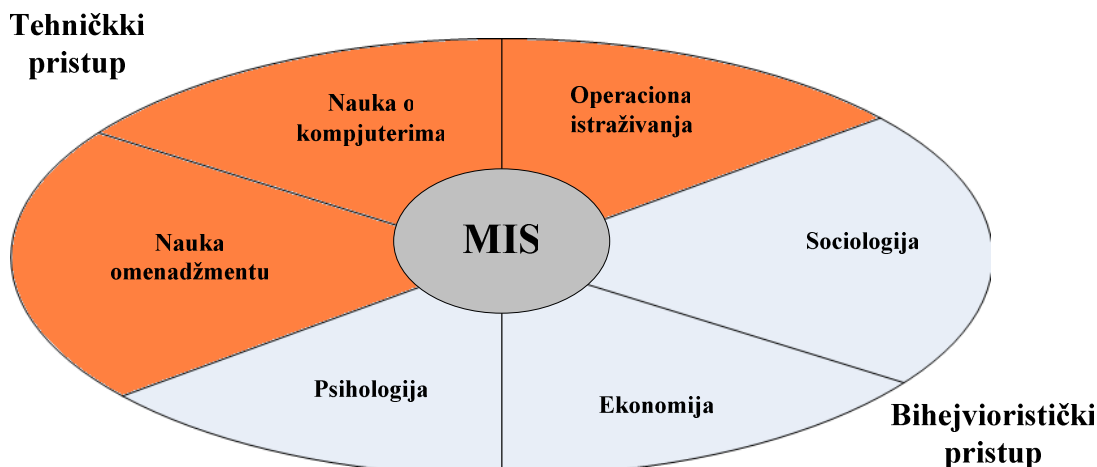
**Model procesa** prikazuje skup procesa koji mijenjaju stanje sistema i skupa procesa pomoću kojih se formiraju izlazi iz sistema.

**Model resursa** prikazuje "procesore" (kadrove, organizacijske jedinice, opremu) sa tačke gledišta njihovih kapaciteta i dinamike korištenja tih kapaciteta.

**Projekt** je imenovana zamisao koja podrazumijeva izvođenje skupa poslova neophodnih za izgradnju informacijskog sistema. Osobe koje projektuju informacione sisteme nazivamo **projektantima**.

## 1.2 pristupi razvoju IS

IS proučavaju višedisciplinirano područje. Ne postoje dominantne teorije u proučavanju informacionih sistema. Slika ilustruje glavne discipline koje pridonose problemima, posljedicama i rješavanjima u proučavanju IS. Općenito, proučavanje IS može biti podjeljeno na tehničke i biheviorističke pristupe. Danas su IS društvenotehnički sistemi. Iako su sastavljeni od mašina, sprava i «teške» fizičke tehnologije, oni zahtjevaju osnovna društvena, organizacijska i intelektualna ulaganja da bi radili kako treba[37].



Slika : Savremena dostignuća u IS

### 1.2.1 Tehnički pristup

Tehnički pristup IS naglašava matematički osnovane modele za proučavanje IS, kao i fizičku tehnologiju i formalne sposobnosti ovih sistema. Discipline koje pridonose tehnološkom pristupu su: *nauka o kompjuterima*, *nauka o menadžmentu*, *operaciona istraživanja*

**Kompjuterska nauka** se brine utemeljivanjem teorija o spojivosti, metodama komuniciranja i metodama o efikasnom pohranjivanju i pristupu podacima.

**Nauka o menadžmentu** naglašava razvoj modela za donošenje odluka i menadžerskih vježbi.

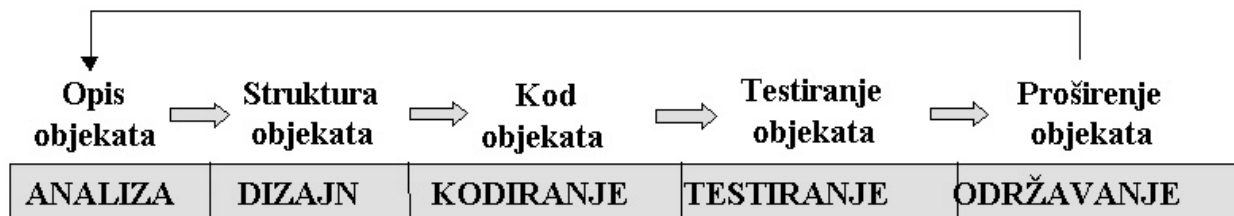
**Operaciono istraživanje** se fokusira na matematičkim tehnikama za optimiziranje odabranih parametara organizacija: prenošenje, inventura i trgovački promet.

### 1.2.2 Bihevioristički pristup

Važan dio IS se odnosi na pitanje ponašanja u razvoju i dugoročnom postojanju IS. Pitanja kao što je strategija poslovne integracije, dizajn, implementacija, korisnost i menadžment ne mogu biti korisno istraženi modelima korištenim u tehnološkom pristupu. Druge biheviorističke discipline doprinose važnim konceptima i metodama. Npr. sociologija proučava IS sa stanovišta kako grupe i organizacije kreiraju razvoj sistema i također, kako sistemi djeluju na pojedice, grupe i organizacije. Psihologija proučava IS sa interesom kako ljudi kao donosioci odluka percipiraju i koriste stvarnu informaciju. Ekonomisti proučavaju IS sa interesom usmjerenim na uticaj koji sistemi imaju na kontrolu i strukturu troškova u okviru firme i okviru tržišta. Bihevioristički pristup ne odbacuje tehnologiju. Zapravo, tehnologija IS je često stimulans za probleme u ponašanju. Ali, fokus ovog pristupa nije generalno na tehničkim rješenjima. Umjesto toga, koncentrira se na promjene u stavovima, politiku menadžmenta i organizacije i ponašanje.

### 1.2.3 Objektno orjentisani pristup

Objektno orjentisana tehnologija (O-O) je jedan od zadnjih pristupa razvoju softvera (slika 6.1.). Kako ona zahtjeva potpuno drugačije promišljanje i pogled na razvoj softvera, može se smatrati paradigmom za sebe. U osnovi, to je programska paradigma, koja može biti primijenjena s razvojnim alatima s kojima je kompatibilna.



Taylor, 1992.

Slika 6.1. O-O paradigma

O-O tehnologija sadrži četiri ključna elementa:

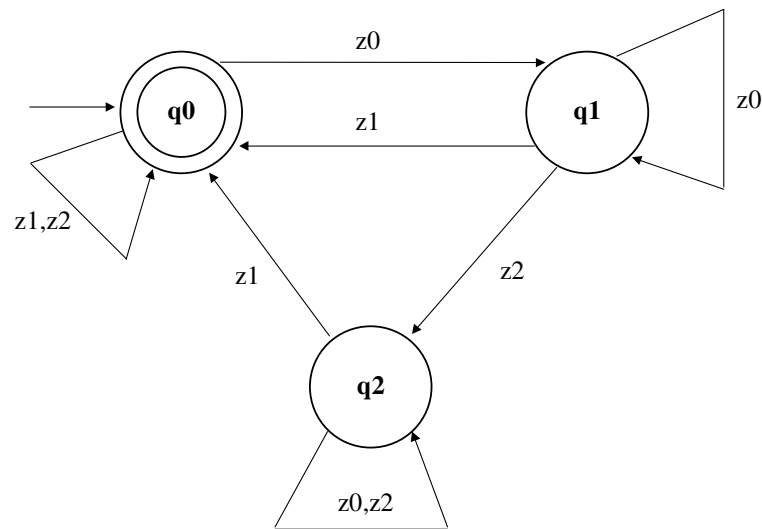
1. **Objekti:** softverski paketi koji su oblikovani i razvijeni s ciljem da odgovaraju entitetima realnog svijeta i sadrže podatke i akcije koji su karakteristični za njima odgovarajuće realne entitete.
2. **Poruke:** komunikacijski mehanizmi uspostavljeni za opis načina na koji objekti zajednički djeluju.
3. **Metode:** akcije koje objekti izvode da bi se zadovoljili funkcionalni zahtjevi problemske domene. Objekti zahtijevaju akcije od drugih objekata putem poruka.
4. **Klase:** predlošci za definisanje familije objekata i svih podataka i akcija koje su im zajedničke, te za osiguranje koncepta nasljeđivanja koji olakšava održavanje O-O softvera.

U kontekstu tradicionalnog životnog ciklusa, objektni pristup osigurava zajednički jezik kroz sve faze razvoja softvera, smanjujući barijere među fazama i unapređujući kompletan proces. Objekti definisani za vrijeme analize zahtjeva postaju sistemni objekti za vrijeme dizajna, koji se onda implementiraju za vrijeme programiranja i na kraju održavaju. Objektna tehnologija se, međutim, ne primjenjuje kroz paradigmu životnog ciklusa, već više kroz paradigmu prototipiranja. Kao i kod primjene svake tehnologije koja je drastično različita od tehnologije na koju smo navikli, mora se platiti određena cijena. Da bi se iskoristile prednosti O-O tehnologije, mora se nabaviti novi O-O razvojni softver, dizajneri se moraju obučiti za novi način promišljanja, moraju se ustrojiti biblioteke novih komponenti, a organizacija mora biti pripremljena za novu tehnologiju.

Iako O-O pristup pokriva sve faze životnog ciklusa razvoja softvera [Mont 94], [Coad 97], on još uvijek nije primjenjiv na sve faze životnog ciklusa razvoja IS. Tako npr. još uvijek nisu razvijene objektno metode koje bi se primjenjivale u početnim fazama strateškog planiranja IS. Bitan razlog zašto je tome tako, jest da se većina složenih organizacija u kojima rade ljudi (ne u potpunosti automatizirane organizacije), na najvišem nivou apstrakcije ne mogu posmatrati kao objekti. Naime, ukoliko cijelu organizaciju posmatramo kao objekt, postavljaju se slijedeća pitanja:

- ♦ Koja su to stanja u kojima se organizacija može naći i kako se vrši prelaz iz jednog stanja u drugo?
- ♦ Koje su to metode (programski dio objekta) koje organizacija koristi?

- ♦ Koji su to atributi (podatkovni dio objekta) koji organizaciju opisuju kao cjelinu?
- ♦ Projektiramo li IS za određenu organizaciju (objekt), kakav je smisao pojma “klasa objekta”?



q0 ... normalno (mirnodopsko) stanje  
 q1 ... stanje mobilizacije  
 q2 ... stanje izvršenja ratne zadaće

početno, završno  
 prijelazno  
 prijelazno

z0 ... prijem zapovijedi za mobilizaciju;  
 z1 ... prijem zapovijedi za demobilizaciju  
 z2 ... prijem zapovijedi za izvršenje zadaće

*Slika 4.2. Dinamička analiza sistema*

Prema tome, u administrativnim tipovima organizacije, početne metode u životnom ciklusu razvoja IS bi trebale biti strukturne. Međutim, postoje tipovi organizacija (obrambene organizacije, zapovjedni brodovi, organizacije za zaštitu od elementarnih nepogoda i sl.), čija poslovna tehnologija se temelji na tzv. standardnim operativnim postupcima i standardiziranoj dokumentaciji. Isto tako, ove organizacije imaju više (konačan broj) stanja funkcioniranja, a prelaze iz stanja u stanje uzrokuju poznati događaji. One u svom ponašanju podsjećaju na automate, pa se njihovo ponašanje može formalno opisati [Dobr 02]. Zahvaljujući navedenim karakteristikama njihove poslovne tehnologije, u strateškom planiranju IS za ove organizacije može se, u određenoj mjeri, primijeniti O-O pristup. Za opis ovih organizacija u fazi strateškog planiranja IS, razvijene su dvije metode [Dobr 98]. Prva je DKS (dinamička klasifikacija sistema), koja omogućava razlikovanje organizacija s vremenski promjenjivim funkcijama, od ostalih klasičnih organizacija.

U osnovi metode DKS leže tri uslova koje organizacija mora ispunjavati da bi pripadala klasi organizacija s vremenski promjenjivim funkcijama. Bez popratnog formalnog aparata, ovi uslovi su:

- ♦ stanja organizacije se međusobno isključuju i organizacija mora biti u bar jednom od svojih stanja,
- ♦ u svakom trenutku u promatranom vremenskom intervalu, organizacija mora biti u tačno jednom od svojih stanja, što je stroži gornji uslov,
- ♦ vjerovatnost istovremenog nastanka događaja koji uzrokuje promjenu stanja organizacije, i same promjene stanja je jednaka vjerovatnosti nastanka događaja.

Zadnji uslov je najvažniji. On znači da organizacije sa stabilnom i standardiziranom poslovnom tehnologijom, koje koriste standardne operativne postupke kao poslovna pravila, te imaju vješte i obučene zaposlenike, djeluju gotovo kao automati. Posljedica toga je da će takva organizacija, kad se desi određeni događaj, sigurno promijeniti svoje stanja.

Druga metoda koja se može koristiti u početnim fazama životnog ciklusa razvoja ovih organizacija, a u osnovi joj leži O-O pristup, je DAS (dinamička analiza sistema).

Ukoliko metoda DKS pokaže da posmatrana organizacija spada u klasu s promjenjivim funkcijama, tada se primjenjuje metoda DAS, koja za rezultat ima model konačnog automata koji opisuje dinamičke karakteristike posmatrane organizacije. Primjer ovakvog konačnog automata za rezervnu jedinicu oružanih snaga prikazan je na slici 4.2.

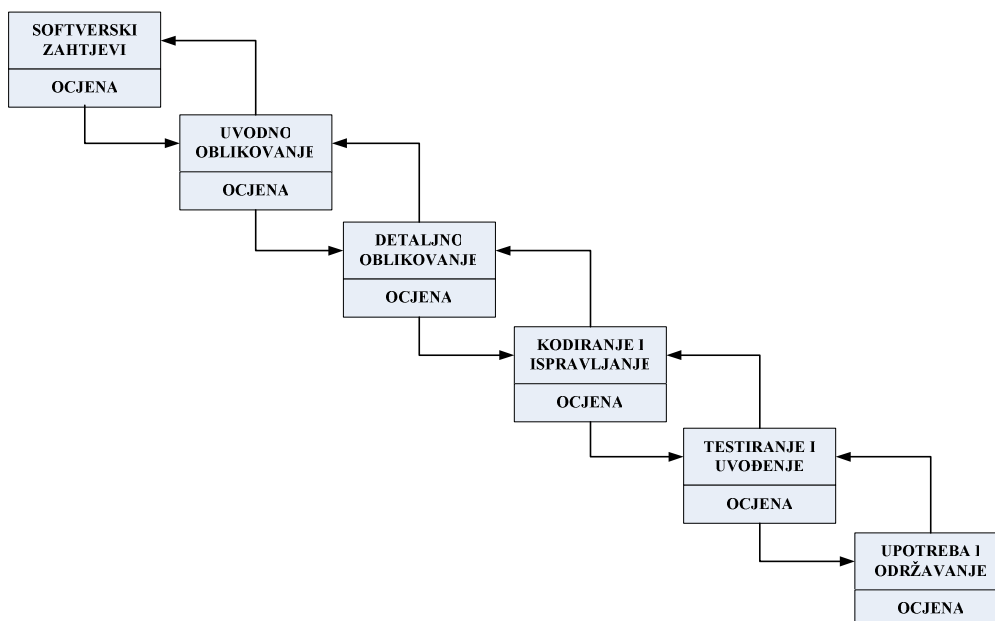
## **2 METODOLOGIJE RAZVOJA IS**

Danas ima mnogo dostupnih metoda i one pokrivaju različita područja informacijskog sistema. Neke tvrtke su razvile njihove lične metode, te iste nisu dostupne široj javnosti. Metodologija PIS (projektovanja informacionih sistema) ima svoje zasebne metode za izučavanje predmeta projektovanja u okviru koga se studiraju i uspoređuju metode. Praktičar, projektant informacionih sistema, se ne bavi metodologijom i rijetko će studirati sve metode. Rješenje za praktičara leži u odabiru jedne metodologije projektovanja informacionih sistema. Metodologije su danas "komercijalni" proizvodi. Metodologija odabire metode, prilagođava ih konačnom cilju, propisuje redoslijed upotrebe metoda, propisuje proces modeliranja od početka do kraja životnog ciklusa informacijskog sistema. Neke metodologije imaju isto ime kao i metode. [45]

Cjeloviti pristup razvoju IS podrazumijeva upotrebu metodologije koja u sebi sadrži metode primjenjive u pojedinim fazama životnog ciklusa razvoja IS, na način da je svaka faza pokrivena s jednom ili više metoda. Ideja cjelovitosti pristupa razvoju IS dovela je do nastanka niza metodologija, kao što su: SPIS (Strategic Planning of Information Systems), IE (Information Engineering), TAFIM (Technical Architecture Framework for Information Management), IDEF (Integrated DEFinition), MIRIS (Metodologija za Razvoj Informacionih Sistema), OOIE (Object Oriented Information Engineering) i sl. Sve ove metodologije nastoje svojim metodama pokriti što više faza životnog ciklusa razvoja IS. Neke od njih koriste slične metode, neke imaju naglasak na višim fazama razvoja IS, a neke na nižim. Neke su podržane CASE alatima u većoj, a neke u manjoj mjeri.

### **2.1 Životni ciklus razvoja IS**

Cjeloviti pristup razvoju IS, kao i metodologije koje podržavaju ovakav pristup, usko su povezani s idejom životnog ciklusa razvoja IS. Ova ideja je prihvaćena iz inženjerske zajednice, koja je 30-tih godina utvrdila da svi proizvodi imaju život konačnog trajanja. Ovaj život počinje sa početnom zamisli i prolazi kroz faze specifikacije, oblikovanja, implementacije, održavanja i zastarijevanja. Za softversko inženjerstvo životni ciklus podrazumijeva niz zadataka na koje mogu biti primijenjene inženjerske metode. Sve ove zadatke zajedno integrišu se u metodologiju životnog ciklusa softvera (slika 4.3.).



Slika 4.3. “Waterfall” model životnog ciklusa softvera

Boehm (1976.) je popularizirao “Waterfall” model životnog ciklusa softvera. Osnovna vrijednost ove metodologije je u tome što je dizajn prepoznat kao posebna i kritična faza u razvoju softvera, jer posljedice prijevremenog kodiranja bez odgovarajuće analize i dizajna mogu biti izuzetno negativne. Proširenjem modela životnog ciklusa softvera dolazimo do životnog ciklusa razvoja IS. Različiti autori navode različite faze životnog ciklusa informacijskog sistema, ali se u njegovom osnovnom toku svi potpuno slažu. Faze životnog ciklusa IS su prema različitim autorima različito definisane, ali u osnovi sadrže slijedeće (Brumec):

- ◆ izrada studija izvodljivosti (procjena spremnosti za razvoj),
- ◆ analiza funkcija organizacijskog sistema,
- ◆ definisanje osnovne arhitektura IS,
- ◆ izrada modela procesa,
- ◆ izrada modela podataka,
- ◆ izrada programskog sistema,
- ◆ uvođenje,
- ◆ procjena efekata.

### 2.1.1 IDEF (Integrated Definition)

IDEF familija metoda (slika 5.1.), ključni je rezultat IICE1 (Information Integration for Concurrent Engineering) projekta. Ova familija metoda predstavlja skup jednostavnih tehnika i standardnih jezika za razvoj IS za podršku složenim organizacijskim sistemima. Treba naglasiti da ove metode pokrivaju veći dio faza životnog ciklusa razvoja IS.

IICE program je zamišljen za razvoj teoretskih temelja, metoda i alata za projektovanje informacioni integrisanih organizacija. Ove metode su dizajnirane za upravljanje informacionim resursima i resursima znanja kao ključnim osloncima kvalitetnih organizacija koje postižu bolje performanse u smislu troškova životnog ciklusa i efikasnosti. Ime IDEF

<sup>1</sup> IICE projekt financira američko ratno zrakoplovstvo (Armstrong Laboratory, Logistics Research Division, Wright Patterson Air Force Base, Ohio), a glavni suradnik je Knowledge Based Systems Inc. (KBSI), Texas.

nastaje iz vazduhoplovnog programa za ICAM (Integrated Computer-Aided Manufacturing), koji je razvio prvu ICAM definiciju ili IDEF metode. Daljnji razvoj IDEF tehnologije podržava sveukupne napore za stvaranjem familije međusobno podudarnih metoda za organizacijsku integraciju. Radi toga se fokus IDEF metoda širi na potpuno upravljanje kvalitetom (TQM), konkurentno inženjerstvo (Concurrent Engineering) i reinženjering poslovnih procesa (BPR). Time familija Integrated DEFinition (IDEF) metoda stiče preduslove za podršku svim fazama životnog ciklusa razvoja IS.

Osim toga, IDEF familija metoda (IDEFmetodologija) namijenjena je za osiguranje ravnoteže između metoda za specijalne namjene (opisi specifičnih dijelova organizacije), koje imaju uski krug primjene i tzv. “super metoda” koje nastoje uključiti sve organizacijske tipove i biti široko primjenjive. Ova ravnoteža se održava osiguravanjem posebnih mehanizama za integriranje rezultata pojedinih metoda iz IDEF familije.

IDEF metodologija proglašena je 1994. g. federalnim standardom za razvoj IS u SAD [NISO 93], [NIS1 93]. Od svih metoda u IDEF familiji najpoznatije su IDEF02 (modeliranje procesa) [Marc 93], IDEF1X (modeliranje podataka) [Bruc 92], IDEF3 (opis toka procesa), te IDEF4 (objektno-orjentisani dizajn).

GODINA	METODA	OPIS
1962.	Petri nets	C.A.Petri je definisao osnove Petrijevih mreža.
1963.	Teorija IS	B. Langafors je definisao osnove teorije informacionih sistema.
1968.	Software engineering	Na NATO savjetovanju u Garmischu definisana je nova disciplina.
1970.	Relacijska metoda	E.F. Codd je definisao osnove teorije relacijskih baza podataka.
1974.	HIPO	IBM je razvio metodu HIPO (Hierarchy, Input, Process, Output).
1975.	JSP	M. Jackson je objavio metodologiju.
1976.	HOS	M. Hamilton i S.Zeldin su razvili metodologiju HOS (Higher Order Software).
1976.	ER	P.P. Chen objavio metodu Entiteta i veza koja je danas najčešće korištena za modeliranje podataka.
1977.	SD	L.Constantin i E. Yourdon definisali metodologiju SD (Structured Design).
1977.	PSL/PSA	Razvijen prvi CASE alat i metodologija na Univerzitetu u Michigenu u okviru projekta ISDOS.
1978.	SA	T. De Marco je objavio metodologiju strukturne analize SA (Structured Analysis).

---

<sup>2</sup> IDEF0 metoda spada u strukturne metode i temelji se na poznatoj SADT (Structured Analysis and Design Technique) metodi.



1978.	BSP	IBM je razvio metodologiju strategijskog planiranja BSP (Business System Planning).
1979.	SSA	C. Gane i T. Sarson objavili strukturnu analizu sistema SSA (Structured Systems Analysis).
1979.	IDEF	<ul style="list-style-type: none"> <li>◆ IDEF0 za općenite funkcije, probleme i informacije,</li> <li>◆ IDEF1X za podatke,</li> <li>◆ IDEF2 za dinamiku,</li> <li>◆ IDEF3 za modeliranje procesa, događaja, dijagrama prelaza stanja,</li> <li>◆ IDEF4 za OO projektovanje,</li> <li>◆ IDEF5 za ontološke opise stvarnosti kreirane od strane čovjeka (vojska, privreda, školstvo, ...) i to za razumijevanje, projektovanje, upravljanje, organiziranje,...</li> </ul>
1981.	ISAC	M. Lundeberg je završio metodologiju ISAC (Information Systems Work and Analysis of Change). Razvoj je započeo 1971. godine.
1981.	Data Analysis	I. Palmer objavio metodologiju Data Analysis.
1981.	Information Engineering	J. Martin i C. Finkelstein objavili metodologiju informacionog inženjeringa.
1981.	SSADM	Metodologija SSADM (Structured Systems Analysis and Design Method) kao dobre prakse projektovanja započela je 1981. Verzija 3. objavljena je 1986. a verzija 4. 1990. godine uključila je formalno i prototipiranje.
1982.	USE.IT	Izrađen prvi efikasan CASE alat USE.IT na osnovi metodologije HOS, koji je prevodio opis u izvršni kod.
1983.	JSD	M. Jackson je objavio metodologiju JSD (Jackson System Development)
1989.	CASE Savjetovanje	Godine 1989. Održano je prvo savjetovanju "CASE" u Opatiji. Savjetovanje je nastavak "Savjetovanje za informatičku djelatnost" u organizaciji Privredne komore Rijeka koje je održano prvi put 1984.
1989.	Euromethod	Ona omogućuje otvoreno tržište u inženjerstvu IS preko cijele Evrope i omogućuje da se različite metode korištene u Evropskoj zajednici usporede, harmoniziraju, iskoriste trenutne sposobnosti i omogući zajednički cilj za budući razvoj ovih metoda. Eurometoda je razvijena iz slijedećih metodologija: SSADM (Velika Britanija); MERISE (Francuska); DAFNE (Italija); IE (Amerika); MEIN (Španjolska); SDM (Nizozemska); VORGEHENSMODELL (Njemačka).
1990.	OOA	P. Coad i E. Yourdon objavili cjelovitu metodologiju objektno orjentisanog projektovanja IS OOP (Object-oriented Analysis).

1991.	OMT	Rumbaugh, Blaha, Premelani, Eddy i Lorensen definisali su tehnologiji objektnog modeliranja OMT.
1992.	IDEA	Započeo razvoj evropske objektno orjentisane metodologije. IDEA je završena 1996.

Tabela 1 Važniji događaji koji u razvoju metodologija za projektovanje i razvoj IS. [7]

Jedna od poznatijih komercijalno dostupnih metodologija je SSADM (Structured Systems Analysis and Design Method). U SSADM su uključene metode ERM, SSA i druge prilagođene i izmijenjene metodologije. Metodologija pokriva sve faze razvoja projekta i osigurana je kvaliteta procesa razvoja u svim fazama. Postoje desetine CASE (Computer Aided Software Engineering) alata koji podržavaju ovu metodologiju.

NAZIV METODOLOGIJE	TVRKA	ZEMLJA
AD/Cycle (Application Development Cycle)	IBM	SAD
BSP (Business Systems Planning)	IBM	SAD
CASE*Method	Oracle	SAD
IEM (Information Engineering Methodology)	James Martin Associates	Velika Britanija
JSD/JSP (Jackson System Development/Programming)	Michael Jackson Systems LTD	Velika Britanija
PSL/PSA (Problem Statement Language / Problem Statement Analyser)	University of Michigan	SAD
SA/SD (Structured Analysis / Structured Design)	Yourdon Inc.	SAD
SASS (Structured Analysis and System Specification)	Yourdon i DeMarco	SAD
SSA (Structured System Analysis)	Exxon	SAD
SSADM (Structured System Analysis and Design Method)	CCTA	Velika Britanija
MIRIS (Metodologija za razvoj informacionih sistema)	RIS	Hrvatska
Yourdon/OO (Yourdon/Object Oriented)	Yourdon Inc.	SAD

Tabela 2 Neke od komercijalno dostupnih metodologija razvoja IS[6]

## 2.2 Metodologija MIRIS

Metodologija MIRIS je skup metoda namjenjen za projektovanje i izgradnju informacionih sistema (IS). MIRIS je kratica od (MetodologIja za Razvoj Informacionog Sistema). Sa stajališta alata za razvoj IS metodologija MIRIS primjenjiva je: [5]

- za klasičan razvoj IS jezicima treće generacije nad datotekama ili bazama podataka,
- za razvoj jezicima četvrte generacije (4GL) nad relacijskim bazama podataka,
- za razvoj generatorima aplikacija i CASE alatima.

Metodologija MIRIS pokriva:

- Sve faze razvoja IS,
- Projektovanje IS je centralni problem koji se razrješava,
- Kontrolu izgradnje.

Metodologija MIRIS omogućava izuzetno dobar proces modeliranja, dobaru povezanost među metodama, traži dobru pripremu za intervjuisanje, daje mogućnost proširene metode za modeliranje podataka, pojednostavljen proces prevođenja u relacijsku shemu baze podataka. Skraćeno je vrijeme rada projektanta, ali je povećan rad korisnika i članova projektnog tima koji poznaju sistem.

### 2.2.1 Koraci u razvoju metodologije MIRIS

Kod nastanka i razvoja metodologija nameće se osnovno pitanje kako sačiniti model organizacijskog sistema te kako taj model što više približiti stvarnosti.

Istraživanja na području semantičkih modela podataka, programskih specifikacija, baza podataka i programskih jezika, "inteligentnih" sistema, teže ali ne uspijevaju iznaći metode koje direktno i prirodno odgovaraju ljudskom "poimanju" stvarnosti. Inteligentni IS trebaju omogućiti čuvanje i obradu znanja.

Pri crtanju različitih vrsta dijagrama koriste se temeljni principi organizacije znanja kao: klasifikacija, agregacija, generalizacija i hijerarhija. Baze podataka sadrže činjenice korisne samo za pretraživanje tipa pitanje - odgovor. Programi sadrže činjenice korisne samo za vrijeme izvedbe procedure[21].

Modeli nisu u stanju izvoditi zaključivanja na osnovi ugrađenih činjenica bez interakcije s vanjskim svijetom.

Zabranjena je nepotpunost opisa sistema (postupan rast i razvoj opisa i aplikacije tokom korištenja). Traži se potpunost i sveobuhvatnost rješenja odmah od prvog dana stvarnog rada. Sve dok se potpuno ne testira programski proizvod se ne smije potpuno koristiti. Projektant ima viziju algoritma i dizajnira model "jednom zauvijek". A zapravo ne može se pretpostaviti kako je bilo koji naš model potpun opis svijeta koji više ne treba širiti učeći činjenice iz stvarnosti. Oдавde potječe i problem održavanja softvera.

Ne postoji zajednički stav o glavnim neriješenim problemima u prezentaciji znanja ali je prisutno spajanje različitih pristupa.

Koraci razvoja su prikazani u sljedećoj tablici, s naglaskom na događaje vezane uz metodologiju MIRIS. Nije moguće navesti sva značajna otkrića vezana za metodologiju razvoja informacionih sistema.

GODINA	METODA	OPIS
1984.	MIRIS	M. Pavlić započeo istraživanje i primjenu metodologije PIS.
1986.	MIRIS	Objavljen prvi rad o projektovanju informacionih sistema primjenom metode DTP i EV na III. Savjetovanju za informatičku djelatnost u organizaciji Privredne komore Rijeka u Opatiji.

1987.	MIRIS	Rad pod nazivom " <b>Metodologija projektovanja informacionih sistema</b> ", objavljen je na IV. Savjetovanju za informatičku djelatnost u organizaciji Privredne komore Rijeka u Opatiji.
1989.	MIRIS	Definiše "Projektni zadatak", na VI. Savjetovanju za informatičku djelatnost, u organizaciji Privredne komore Rijeka u Opatiji.
1990.	MIRIS	Objavljena knjiga "Sistem analiza i modeliranje podataka - Projektovanje informacionih sistema", Naučna knjiga.
1995.	MIRIS	Objavljena metodologija MIRIS, na savjetovanju CASE 7 u Opatiji.
1996.	MIRIS	Objavljena knjiga "Razvoj informacionih sistema", o metodologiji MIRIS.
1996.	MIRIS	I. Zamlić se priključio razvoju metodologije i značajno utjecao na metodu APP za projektovanje arhitekture programskog proizvoda.
1999.	MIRIS V2	Objavljena verzija 2 metodologije MIRIS.

*Tabela 3 Važnji događaji u razvoju metodologije MIRIS za projektovanje i razvoj IS [5]*

### 2.2.2 Faze i aktivnosti metodologije MIRIS

U sljedećim tabelama ukratko su prikazane faze životnog ciklusa IS prema metodologiji MIRIS.

Prema ovoj metodologiji cjelokupni razvojni put sastoji se od 6 faza (u verziji MIRIS 1 definisao je 7 faza). Općenito se cjelokupni ciklus razvoja informacionog sistema može podijeliti na dvije grupe faza i to:

- ◆ logičkog oblikovanja (projektovanja IS) i
- ◆ fizičkog oblikovanja (izgradnja IS).

Pod projektovanjem podrazumijevamo sve intelektualne aktivnosti koje prethode fizičkoj izgradnji sistema na računaru. Cilj projektovanja informacionih sistema je načiniti nacрте na osnovi kojih će se razviti sistem.

Logičko se oblikovanje sastoji od faza strateškog planiranja, glavnog projekta i izvedbenog projekta, a fizičko oblikovanje od faza izvedbe programske podrške, uvođenja i primjene te održavanja.

Faze su sastavljene od aktivnosti. U tablici su navedene kratice i opisi aktivnosti. [5]

<b>LOGIČKO OBLIKOVANJE – PROJEKTOVANJE</b>		
<b>Faza 1: STRATEŠKO PLANIRANJE IS (SP)</b>		
<b>Kratica</b>	<b>Opis aktivnosti</b>	<b>Metoda</b>
1.1 Analiza	Definisanje i obuka tima, dekompozicija i opis poslovnih procesa, popis grupa podataka i dokumentacije te kretanje podataka kroz sistem	Analiza SP
1.2 Podsystemi	Određivanje podsistema i veza	Sinteza SP
1.3 Prioriteti	Određivanje prioriteta	Ekspertiza

1.4 Resursi	Definisanje cjelovite infrastrukture	Ekspertiza
1.5 Plan	Planiranje glavnih projekata i strateških aktivnosti	Gantogram
<b>Faza 2: GLAVNI PROJEKT (GP)</b>		
2.1 PZ	Izrada projektnog zadatka	PZ
2.2 DTP	Intervjuiranje, raščlanjivanje i modeliranje procesa	SAS
2.3 Procesi GP	Analiza procesa, problema i prijedloga poboljšanja	SAS
2.4 Podaci GP	Opisivanje podataka	SAS
2.5 Plan GP	Planiranje izvedbenih projekata	SAS
2.6 Resursi GP	Definisanje modela resursa glavnog projekta	SAS
<b>Faza 3: IZVEDBENI PROJEKT (IP)</b>		
3.1 DEV	Intervjuiranje i modeliranje podataka	EV
3.2 Prevođenje	Prevođenje modela podataka u shemu BP	Prevođenje
3.3 Arhitektura IP	Definisanje arhitekture programskog proizvoda	APP
3.4 Operacije IP	Projektovanje operacija nad shemom BP	Logika programir.
<b>FIZIČKO OBLIKOVANJE – IZGRADNJA</b>		
<b>Faza 4: IZVEDBA PROGRAMSKOG PROIZVODA (IPP)</b>		
<b>Kratika</b>	<b>Opis aktivnosti</b>	<b>Metoda</b>
4.1 Fizička BP	Projektovanje fizičke baze podataka	Optimiziranje
4.2 Shema BP	Prijavljivanje sheme BP u rječnik podataka	Izvedba
4.3 Prototip	Izvedba prototipa s generisanjem aplikativnog stabla	Izvedba
4.4 Programiranje	4GL ili 3GL programiranje	Izvedba
4.5 Dokumentiranje	Pisanje uputa i objašnjenja programskih rješenja	Izvedba
4.6 Testiranje	Punjenje testnih podataka u BP i testiranje	Testiranje
<b>Faza 5: UVOĐENJE I PRIMJENA (UIP)</b>		
5.1 Obuka	Obuka korisnika	Obuka
5.2 Unos	Unos početnih stanja baze podataka	Uvođenje
5.3 Validacija	Testiranje zadovoljavanja korisničkih zahtjeva	Uvođenje
5.4 Sistem pomoći	Pisanje korisničkog sistema pomoći	Uvođenje
5.5 Paralelan rad	Optimiziranje i paralelan rad starog i novog sistema	Uvođenje

5.6 Preuzimanje	Završno testiranje – preuzimanje	Uvođenje
5.7 Primjena	Primjena programskog proizvoda (korisnici, administrator BP)	Uvođenje
<b>Faza 6: ODRŽAVANJE (ODR)</b>		
	To je izvođenje ranijih aktivnosti radi: uvođenje novih poslovnih procesa, izmjene postojećih poslovnih procesa i otklanjanja grešaka	Ranije navedene metode

*Tabela 4 Kratice i opisi aktivnost metodologije MIRIS.*

Najznačajnije odlike metodologije MIRIS su: [7]

- ◆ Metodologija odabire metode i nije potrebno izučavanje i lutanje,
- ◆ Detaljnije se opisuju faze i aktivnosti,
- ◆ Posao se odvija korak po korak,
- ◆ Definiše se svaki korak,
- ◆ Povezuju se aktivnosti tako da izlaz iz jedne postaje ulaz u drugu,
- ◆ Stvara se zajednički rječnik podataka za sve članove tima,
- ◆ Obučava se proces projektovanja kao najvažniji dio metodologije,
- ◆ Propisuju se standardne metode,
- ◆ Osigurava se kvaliteta poduhvata,
- ◆ Omogućava se planiranje i upravljanje razvojem,
- ◆ Centralizira se projektna dokumentacija,
- ◆ Osigurava se prenos znanja na informatičare.

Izgradnja informacionog sistema je skup aktivnosti koji ima za cilj kompjuterizirati procese definisane projektom i reorganizirati sistem na informatičkoj tehnologiji. Izgradnja informacionog sistema ne znači samo izgradnju softvera.

U fazi projektovanja skraćeno je vrijeme rada projektanta, ali je povećan rad korisnika i članova projektnog tima koji poznaju sistem. Uvedu se skup obrazaca koje popunjavaju naručiocu studije. Po toj metodi korisnici: definišu ciljeve, navode probleme, opisuju željeno stanje, ocjenjuju postojeći IS, zahtijevaju i predlažu poboljšanja te opisuju poslovanje. Dužina izrade strategijskih studija je produžena zato što korisnicima treba vremena za popunjavanje obrazaca. Cilj je smanjenje rada projektanta uz podizanje kvalitete produženje vremena i povećanog učešća korisnika u kreaciji opisa sistema. Ovakve studije postaju korisne i dobar ulaz u fazu glavnog projekta.

Glavni projekt je faza koja slijedi po izradi studije. Iz studije (koja se radi jednom u više godina) slijedi niz glavnih projekata te se prema studiji pristupa izradi glavnih projekata za svaki pojedini podsistem. Glavnih projekata ima onoliko koliko u studiji ima podsistema odnosno onoliko koliko organizacijski sistem ima informacionih podsistema. U glavnom projektu se primjenjuju metode za modeliranje procesa i ocjenjuju potrebni resursi.

U MIRIS\_u fazi Glavnog projekta je definisan skup obrazaca za pripremu korisnika prije intervjua. Po pripremi olakšano je modeliranje procesa i izrada dijagrama toka podataka. Tako se intervjuisanje ne izvodi dok se korisnik ne pripremi i ne prikupi i ne popiše svu raspoloživu dokumentaciju. Tada je intervjuisanje kratko i efikasno. Proces izrade modela procesa (crtanja dijagrama dekompozicije i dijagrama toka podataka) je poboljšán. I dalje se istražuju mogućnosti unapređenja procesa modeliranje s ciljem skraćénja vremena rada projektanta i dizanja nivoa kvalitete opisa. Problem kod opisa je njegova mala korisnost za

sljedeće faze modeliranja podataka. Temeljno pitanje na koje treba odgovoriti je kako opisivati dokumente i podatke na njima a da se prikupljeno znanje pretoči u model podataka. Mogu se koristiti različiti CASE alati za crtanje modela.

Izvedbeni projekt (ili detaljan projekt) je podprojekt i dio nekog glavnog projekta u kome se pristupa projektovanju dijela sistema detaljno. Rezultat je model koji detaljno opisuje dio sistema i koji je dobra osnova za fizički dizajn. Osnovna metoda pri detaljnom projektovanju je EV (metoda entiteta i veza).

U metodologiji MIRIS u fazi Izvedbenog projekta učinjene su izmjene u konceptima metoda EV (entitetske veze) te u proširenju skupa s nekoliko novih koncepata za jasnije isticanje veze među entitetima sistema. Olakšano je prevođenje u relacijski model, odnosno EV je projektiran tako da olakša sljedeću fazu a ne izgubi u snazi organizacije znanja o sistemu. Proces izrade modela je poboljšan.

U Fizičko oblikovanje – izgradnju uključene su tri faze umjesto ranije četiri. Spojene su dvije faze (prototipski razvoj i razvoj 3GL) u jednu fazu nazvanu Izvedba programskog proizvoda (IPP).

Izvedba ima za cilj načiniti gotov programski proizvod. Ako se pri tome koriste CASE alati i generatori onda se MIRIS prilagođava odabranom alatu i načinjeni modeli u ranijoj fazi predstavljaju inicijalnu verziju projektne dokumentacije a daljnje održavanje dokumentacije izvodi se u CASE alatu.

Uvođenje i primjena je faza u kojoj do izražaja dolazi menadžmet koji treba aktivirati organizaciju na promjenu ponašanja i primjenu gotovog rješenja. Sva odgovornost za uvođenje ja na menadžmentu. Ovu fazu ne mogu izvesti informatičari, ali u njoj trebaju tijesno sarađivati u obuci korisnika, optimizanju i doradi programa. Aktivnosti ove faze nije potrebno opisivati.

Ova faza često može potrajati duže od faze Izvedbe programa.

Održavanje je faza sastavljena od izvođenja ranijih aktivnosti iz svih potrebnih faza radi:

- ◆ uvođenje novih poslovnih procesa,
- ◆ izmjene postojećih poslovnih procesa i
- ◆ otklanjanja grešaka.

Informacioni sistem se razvija i mijenja tokom vremena. Sve spoznaje o promjeni moraju se ugraditi u programsku podršku ali i u dokumentaciju koja je nastala u ranijim fazama. U protivnom će dokumentacija opisivati prošlo stanje sistema i postati neupotrebljiva . Problem dokumentacije značajno olakšava korištenje CASE alata.

## **2.3 Ključni faktori u razvoju IS i nedostaci u razvoju IS metodologijom MIRIS**

Cilj svake metodologije i metode razvoja informacionog sistema je izgraditi i uvesti uspješan sistem čiji se uspjeh može pripisati izvršenju ciljeva sistema i produktivnosti, odnosno sistem treba zadovoljiti utvrđene zahtjeve, biti razvijan na vrijeme i unutar planiranih finansijskih sredstava. Da bi se otkrio nedostatak jedne ili više metoda u postizanju navedenih ciljeva, treba istraživati prepreke postizanja kvalitete i produktivnosti.

Ako metodologije trebaju raditi efektivno unutar organizacije, potrebno je više od odluke o tome koju ćemo usvojiti. Sve su metodologije pogodne u odgovarajućoj okolini. Također, sve mogu zakazati ako okolnosti rade protiv njih. Sljedeće karakteristike su ključni faktori koji



trebaju biti uvaženi za uspješno uvođenje metodologije bilo koje metodologije pa i metodologije MIRIS.

Kvaliteta informacionog sistema procjenjuje se prema tome da li sistem zadovoljava tačno određene zahtjeve. Poteškoće mogu biti sljedeće vrste:

- ◆ Menadžerska obaveza (pristajanje) od samog vrha, vođenje menadžmenta koji razumije koristi koje se mogu očekivati i ulaganje u vrijeme i novac koji je potrebno poduzeti da bi se to učinilo.
- ◆ U početku projekta treba više vremena posvetiti ranim fazama projektovanja. Ovo predstavlja šok za mnoge menadžere, za koje nikakav posao nije napravljen dok se ne izlista linije koda s računara.
- ◆ Učenje i vježbanje je bitno na početku projekta, da bi se prevladalo teškoće učenja, ali treba postojati stalna obaveza samoučenja.
- ◆ Korištenje raznih drugih metoda za područja koja ne pokriva kupljena metodologija.
- ◆ Procjena produktivnosti i usporedba s drugim projektima (i prošlim i budućim). Informacija o tome treba biti sačuvana. Pretjerivanje s kvalitetom opasno je i to često zna upropastiti cijeli projekat.
- ◆ Metodologija mijenja kulturu organizacije i ljude uključene na projektu. Razni su efekti koji mogu obuhvaćati strah od promjena, nesigurnost, čak i neprijateljstvo. Ljudi moraju biti ohrabreni i usmjereni na to da će efekti dobijeni razvojem IS ići njima u korist.
- ◆ Poželjna je podrška dobrog CASE alata ili više njih. Postoji nekoliko aspekta projekta koji će imati koristi od podrške CASE alata. Crtanje modela je osnovna korist. Upravljanje projektom, analiza rizika i kontrola konfiguracije je omogućena nekim CASE alatima i predstavlja napredne koristi. Bolje je ne koristiti CASE alat ako on nije stabilan proizvod. Posebno je teško istovremeno započeti uvoditi CASE alat i metodologiju.
- ◆ Planiranje projekta treba biti jasno utemeljeno na fazama životnog ciklusa metodologije.
- ◆ Suradnja s korisnicima je važna i treba im objasniti faze metodologije. Analitičar treba uzeti vrijeme da korisnike stvarno uključi i osigura njihovo razumijevanje o tome šta se događa.
- ◆ Metodologije i CASE alati stalno se usavršavaju i napreduju. Nikako nije sigurno izabrati metodologiju i alat i onda se opustiti i ne pratiti njihovo unapređenje i razvoj.
- ◆ Mogućnost pomoći projektantima u korištenju metodologije. Potrebno je od iskusnih projekatara učiti te razviti dovoljno modela na "živim" projektima. Često se dobri CASE alati i programski jezici odbace jer se pokušala izgradnja na nedovršenim i nepouzdanim modelima. Tada se CASE alat koristi na kraju projekta kao alat za dokumentaciju.

Ako menadžment koji ima savršeno jasan cilj želi da izgradi softversku podršku koja će davati određene informacije i koja će automatizovati najznačajnije poslovne procese mora imati potrebne i jasne informacije. Naravno, pretpostavka je da je savršeno jasno koje su to informacije i koji su to procesi, kao i da je u stanju da precizno definiše kako se do tih informacija dolazi i kako se ti procesi odvijaju. Ako zna krajnji cilj ali još uvijek ne zna kako i doći do njega.

Glavne nepoznanice na tom putu su:

- ◆ faktor jasnoća
- ◆ faktor resursi
- ◆ faktor saradnja



- ◆ faktor novac
- ◆ faktor vrijeme

Klasični menadžment prepoznaje tri faktora: resurse, novac i vrijeme. Ali su veoma bitna i ostala dva. Nijedan od ta dva faktora nije mrljiv, bar na prvi pogled. Takođe ne može se kupiti ili prebaciti iz drugog sektora. A ipak, IS neće uspjeti ako i ti faktori nisu zadovoljeni.

### **2.3.1 Faktor jasnoća**

Razvoj IS će imati nekoliko faza i treba ga inkrementalno izgrađivati. To je jasno (jer se IS ne može implementirati odjednom). Treba imati u vidu koje faza u razvoju IS će biti. Treba imati viziju šta će biti na kraju pojedine faze. Treba pretpostaviti i to šta se neće dobiti na kraju pojedine faze i na kraju cijelog projekta.

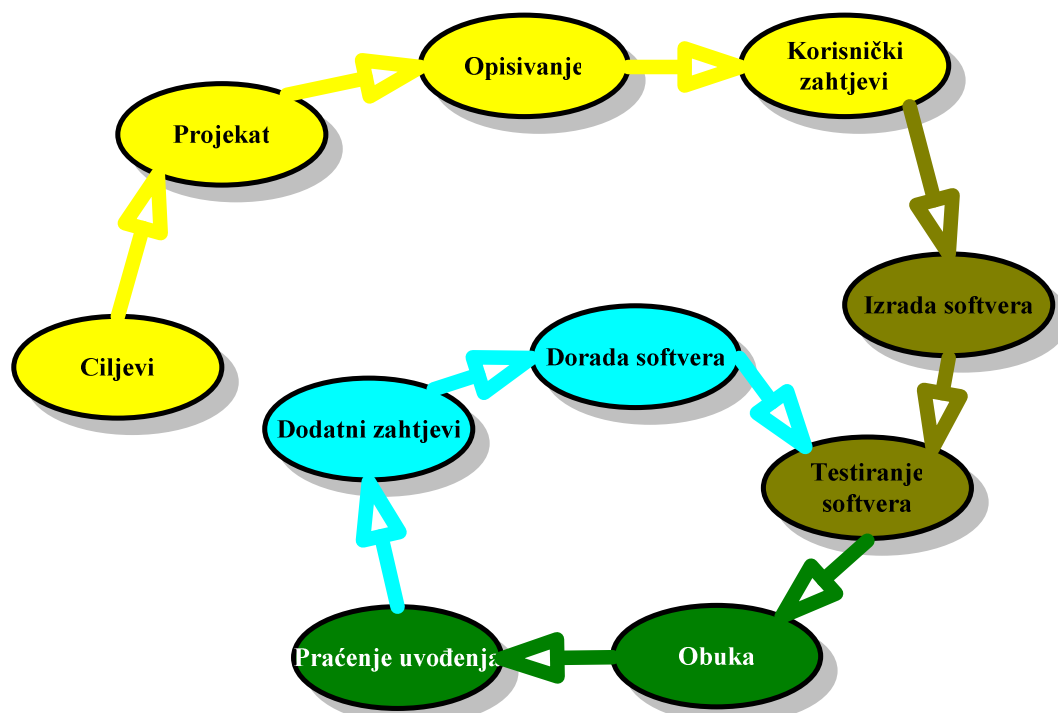
Sve treba shvatiti uslovno, ali 90% projekata, svih vrsta, ne uspije jer menadžmentu ove stvari nisu bile jasne. A 95% uzroka nezadovoljstva rezultatima projekata IS je u nejasnoći.

Ne treba se zavaravati: ako prije ulaska u projekat nije jasno šta će se njime dobiti, to se neće razjasniti u toku projekta. U svim projektima ovakve vrste potrebna je pomoć sa strane da bi se razumjelo šta se u stvari dešava. Ne treba se previše uzdati samo u svoje ljude - treba čuti i drugu, objektivnu stranu. Zato će, gotovo uvijek, trebati i konsultant.

### **2.3.2 Faktor resursi**

Razvoj IS ima sljedeće aktivnosti:

- ◆ Definisanje ciljeva: šta želimo da postignemo
- ◆ Definisanje projekta: određujemo globalnu dinamiku
- ◆ Detaljni projekti: opisujemo procese
- ◆ Definisanje korisničkih zahtjeva: šta krajnji korisnici žele
- ◆ Izrada softvera: programiranje
- ◆ Testiranje softvera: kako se ponaša u praksi
- ◆ Obuka za softver: budućih korisnika
- ◆ Praćenje uvođenja: probni period
- ◆ Prikupljanje dodatnih zahtjeva: povremeni dodatni zahtjevi usljed promijenjene prakse
- ◆ Dorada softvera.



*Slika 1 Utjecaj resursa na razvoj IS*

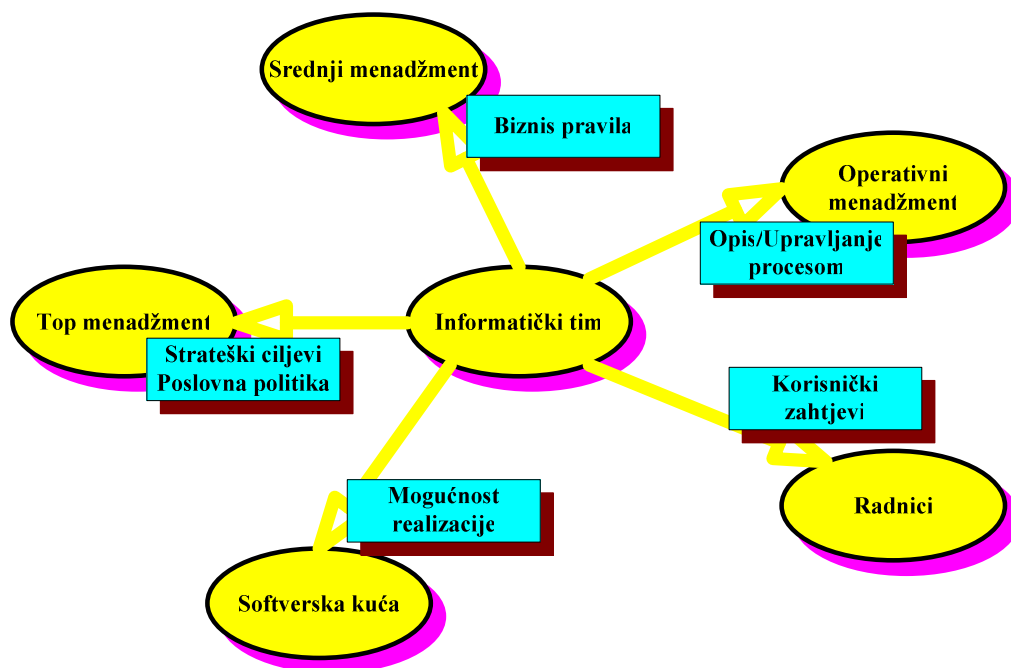
Treba zapaziti da projekat IS počinje nizom koraka, ali da kraja projekta nema - vrti se u petlji. Pravilo važi i za druge investicione projekte, npr. nabavka opreme: počinje se nabavkom i instalacijom opreme. Zatim slijedi stalno održavanje te opreme i unapređenje tehnologije. Specifičnost softvera je jedino u tome što je ciklus stalnih dorada mnogo izraženiji i zahtjevniji. Važno pitanje je da li i koliko angažovati spoljnu pomoć. A to u mnogome zavisi od odluke šta kupiti, a šta napraviti.

Angažovanje resursa je izbor da li za pojedine faze u uvođenju IS angažovati softverske kuće ili svoje ljude. Jedna od preporuka bi bila da se razdvoji 5 (pet) učesnika u tom: menadžment, budući krajnji korisnici softvera, informatički tim koji će razvijati softver, softverska kuća i stručni konsultant za menadžment.

Posebno treba obratiti pažnju na angažovanje resursa poslije implementacije. Često se barata ciframa da jedna četvrtina programera koja je učestvovala u razvoju ostaje trajno da radi na njegovom održavanju i doradi. Što je veća općina, veće je i održavanje. Bez održavanja svaki softver će brzo stati.

### **2.3.3 Faktor saradnja**

Slika prikazuje problem.

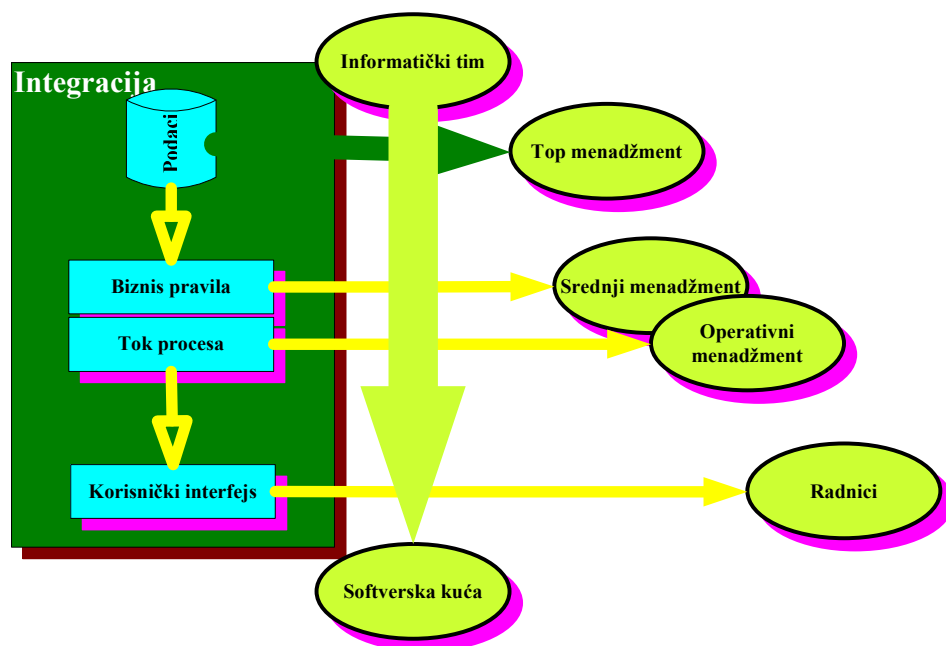


*Slika 2 Faktor saradnje*

Učesnici u razvoju IS postavljaju svoje zahtjeve i vuku na svoju stranu. Top menadžment bi kroz IS želio da riješi svoje organizacione i funkcionalne probleme i ostvari svoje ciljeve. Na drugoj strani radnici najviše žele da im rad bude što lakši, a da se pri tome ništa ne mijenja. Srednji menadžment, rukovodioci sektora i službi žele da se pravila rada konačno razjasne. Operativni menadžment želi da proces rada bude jasan i upravljiv. A sve ovo je ograničeno sposobnošću softverske kuće da to realizuje kroz aplikacije u ograničenom roku (što umnogome zavisi od softverske infrastrukture i sistemskih aplikacija). U sredini je informatički tim koji sve to mora da usaglasiti. Dakle, potrebni su saradnja i kompromisi.

Da se ne radi o preteškom bremenu za informatički tim? U stvari i nije, ako se zna o čemu se radi. Radi se o različitim nivoima softvera. Savremena teorija softvera (praktikuje se zadnjih 4-5 godina), govori o višenivooskom softveru, tako nešto treba i primijeniti.

Radnici u procesu, krajnji korisnici softvera, postavljaju zahtjeve za korisnički interfejs (kako izgleda aplikacija, koje opcije ima i sl). Operativni (tok procesa) i srednji (biznis pravila) menadžment daju potrebne informacije za biznis nivo. Top menadžment usmjerava tok razvoja softvera i time olakšava definisanje potrebnih podataka (baza podataka) i integraciju. Sve ove zahtjeve prikuplja informatički tim i proslijeđuje softverskoj kući. Prikupljanje tih zahtjeva je dugotrajni i inkrementalni proces - ne možete prikupiti sve zahtjeve odjednom. Informatički tim je bliži zaposlenima i procesima od softverske kuće, ima više vremena da se posveti tom poslu od nje, te je njegova uloga u prikupljanju zahtjeva ključna za kvalitet softvera.



*Slika 3 Faktor saradnje*

Prikupljanje korisničkih zahtjeva i opisivanje poslovnih procesa je tek početak saradnje između tvrtke i softverske kuće, ali i između ljudi u pojedinim timovima i timova sa informatičkim timom.

Saradnja podrazumijeva postojanje kulture timskog rada. Možda je to i osnova koju treba staviti na početak spiska zadataka prije ulaska u IS, u suprotnom desit će se ono što se dešava svaki dan i u općinama i preduzećima. Umjesto timova, preovlađaće interesne grupe koje će se brzo iskristalizovati u dva tabora: oni koji su za uvođenje IS u nekom obliku (top menadžment, informatički tim i još poneko) i ostala većina koja je protiv. U jednom trenutku će top menadžment narediti da se implementira softver do nekog datuma, a time otpada sva priča oko konsenzusa i njegovom prilagođenju učesnicima u procesu i samim procesima.

Znači: (1) obuka za timski rad za širok krug menadžera prije ulaska u projekat IS, (2) definisanje zajedničkih ciljeva IS konsenzusom, (3) usaglašavanje oko prioriteta i dinamike, (4) uređivanje odnosa između timova i informatičkog tima, (5) uređivanje odnosa između informatičkog tima i softverske kuće i (6) rad na projektu IS.

### **2.3.4 Faktor novac**

Iskustveno, iznos koji se namjerava dati za razvoj IS treba pomnožiti sa dva i tako doći do sume koliko će koštati projekat.

To se mora uzeti iz razloga što se polazi sa pretpostavkom da će se riješiti svi problemi. Kupljeni softver gotovo uvijek mora da se dorađuje i ponovo se plaća, ponekad se mora krenuti iz početka.

IS košta puno, barem za naše uslove. Kada se nema dovoljno para za ove namjene, treba početi sa automatizacijom samo dijela procesa i malim brojem računara (kako je do sada u većini slučajeva i urađeno). Sada kad je jasno koliko treba da se plati da bi se nešto dobilo, treba početi sa kalkulacijom.

I treba imati na umu: najskuplja investicija je promašena investicija. Može se kupiti i najskuplji softver i da opet nema koristi od njega.

### **2.3.5 Faktor vrijeme**

Za automatizaciju jednog procesa (npr. plaće, materijal i sl) treba najmanje 6 mjeseci. Jednostavnije aplikacije (procesi) mogu da se pokrenu za 3 mjeseca, složeniji traže i do 9 mjeseci.

Ako uzmemo i ograničenje da se ne može raditi na više od 3 procesa istovremeno, jer su i resursi ograničeni (informatički tim, softverska kuća, raspoloživost menadžera i novac). Praktično, svaka ozbiljnija implementacija IS, znači pokrivanje 5-7 ključnih poslovnih procesa traži barem 18 mjeseci.

Požurivanje ne uspeva. Još šezdesetih je utvrđeno pravilo da dodavanje novih ljudi u red projekta ne ubrzava projekat, već ga usporava.

Neophodno je napomenuti još neke od nedostataka u metodologiji MIRIS ako je primjenjujemo u razvoju IS tvrtke, ti nedostaci bi bili:

- ◆ Definicija zadatka nije dobra i rješava se drugi problem.
- ◆ Zahtjevi korisnika u kasnim fazama projekta su u raskoraku s njihovim utvrđenim zahtjevima na početku projekta ili su u raskoraku s drugim korisničkim zahtjevima.
- ◆ Zahtjevi mogu uključivati neprimijećeni problem u implementaciji koji se otkrije tek kasnije u procesu razvoja.
- ◆ Zahtjevi se stalno mijenjaju zbog vanjskih događaja kao šta su promjene zakona ili političke okolnosti.
- ◆ Prihvatljivost i lakoća korištenja nisu zadovoljavajući. Ne analiziraju se potrebne informacije za odabrani posao.
- ◆ Pitanje je treba li se usmjeriti na formulaciju problema i njegovo rješavanje ili na rješenje projekta cjeline u kojoj će problemi nestati.
- ◆ Plan aktivnosti treba biti baziran na strategiji poslovanja a suprotnosti pojedinaca i njihovi pojedinačni ciljevi treba rješavati konsenzusom.

## **2.4 Mijenjanje zahtjeva i postupni razvoj u metodologiji MIRIS**

Ako je definicija zahtjeva pogrešna, onda takav mora biti i razvijeni sistem. Često se zahtjevi mijenjaju iz slijedećih razloga:

- ◆ Korisnici ne znaju svoje zahtjeve;
- ◆ Korisnici se ne slažu oko svojih zahtjeva;
- ◆ Problemi u komunikaciji između analitičara i korisnika potiču nerazumijevanje onoga šta će sistem donijeti u vezi sa zahtjevima;
- ◆ Organizacijsko okruženje se promijenilo u vremenu dok je sistem bio u razvoju

U slučaju velikih sistema poželjan je postupni razvoj i postupna isporuka pojedinih izvedbenih projekata. Ovo uključuje podjelu sistema da bude razvijen/isporučen u samozaostavljive, funkcionalne i korisne dijelove koji mogu biti isporučeni u relativno kratkom vremenskom periodu. Skraćeno vrijeme isporuke prati manje promjene zahtjeva i lakše je održavanje manjih izmjena.

Psihološki posljedica je povećana motivacija razvojnog tima kroz pozitivnu korisničku povratnu vezu i pojačano povjerenje korisnika u one koji razvijaju. Poteškoće kod postupnog razvoja svode se na pitanje kako podijeliti sistem i odrediti prioritete pojedinih izvedbenih projekata. Na ova pitanja odgovara analiza u glavnom projektu.

Pitanje je kako postići konsenzus zahtjeva za novi sistem pomoću sudjelovanja, rasprave i spajanja korisničkih pogleda sa rješavanjem razlika među korisnicima za iste funkcije na različitim lokacijama, ako izmjena funkcija zadire u prošlosti traži izmjenu prošlih ugovora. Postupni razvoj u uvođenje po lokacijama korisnika jedino je moguće rješenje. Neuočeni problemi najbolje se rješavaju postupnim razvojem.

MIRIS je neovisan o relacijskoj bazi podataka. Relacijska shema se može crtati raznim alatima. Semantički bogata shema s konceptima koji nose znanje o sistemu, može se crtati bilo kojim grafičkim alatom ako skup simbola CASE alata nije otvoren za širenje i izmjenu.

U MIRIS-u je poboljšana proces modeliranja, načinjena bolja povezanost među metodama, poboljšanja priprema za intervjuiranje, proširene metode za modeliranje podataka, pojednostavljen proces prevođenja u relacijsku shemu baze podataka. Skraćeno je vrijeme rada projektanta ali je povećan rad korisnika i članova projektnog tima koji poznaju sistem.

Odlika MIRIS-a je bila i ostala, naći odgovor na pitanje koje metode izabrati i kako izabranim metodama modelirati sistem.

## **2.5 Projekcija upotrebe metodologije**

S obzirom na namjenu Metodologije razvoja informacionih sistema, koji je na samom početku prijedloga za razvoj metodologije za razvoj IS tvrtke, može se napraviti i projekcija njene upotrebe.

Ako se uzme u obzir da je jedan od korisnika vođa projekta razvoja informacijskog sistema, koji će MIRIS upotrijebiti kao pomoć za definisanje odgovarajućih projekatanskih i aktivnosti, odabrati nadzorne tačke kao i odrediti alate, koji će nastati u procesu razvoja. Vođa projekta će izabrati odgovarajući pristup (strukturni, objekatski,...), po kojem će teći razvoj projekta. U djelu metodologije, gdje je predstavljen postupak razvoja IS kao rezultat će se pojaviti izlaz koji će se kasnije koristiti i koji će određivati tok narednih faza, iz kojih je i sastavljen projekat. Naravno, vođa projekta će moći zadržati stvarno stanje i na osnovu njegova iskustva zadržati preporučene faze ili ih čak razbiti na više faza. Predstavljene podjele na faze u metodologiji ne smijemo uzimati kao apsolutne, nego kao najobičniji primjer, koji odgovara većini stvarnih projekata razvoja, ali ne baš svima. Da predlagana podjela na faze nije apsolutna, rečeno je već u strukturnom djelu prijedloga metodologije, gdje bi se na osnovi prvog predstavljenog cjelovitog pristupa predstavio još i skraćeni pristup. U skraćenom pristupu neke faze moraju biti združene, s tim da su ustvari okrnjene i aktivnosti unutar tih faza. Nadalje će vođa projekta u dijelu, gdje je opisan postupak razvoja, naći prijedlog aktivnosti i njihov redoslijed unutar fazi. Aktivnosti i njihov redoslijed je određen na osnovu iskustava u radu na više projekata, zato je preporučljivo da vođa projekta predloženo koristi koliko je to moguće, a naravno može pojedinačne aktivnosti po vlastitoj presudi združiti ili mjenjati.

U drugom dijelu ovog poglavlja je usput rečeno, da je metodologija namjenjena vođama i članovima timova za uspostavljanje IS. Vođa tima za postizanje zadovoljavajućeg kvaliteta je naročito interesantan dio, gdje je predstavljen postupak razvoja, jer između ostalih postiže prijedlog, kako bi se najbolje na projektu razvoja izvodile pojedine faze, koji izlazi bi se koristili kao ulazi u neku od faza. Proizvodi analize i planiranja su od ključnog značaja pri razvoju IS, tu se kriju najviše zamke i najčešći nedostaci koji se kasnije pojave u samoj fazi razvoja IS

## 2.6 Podrška mrežnim sistemima

Potrebno je ponekad riješiti ozbiljnije i složenije informatičke probleme na mrežnim informacionim sistemima. Intervencije se posebno odnose na hardwareske ili sistem-softwareske probleme na serverima ( bilo da su oni pod Novell, Linux ili nekim Microsoft-ovim proizvodom), mrežnim radnim stanicama (kao klijentima) ili nekim komponentama na infrastrukturi informacionog sistema (preklopnici i slično). Vrlo je popularan ugovorni odnos s klijentima i partnerima čime se ubrzava vrijeme izlaska na intervenciju, smanjuju troškovi korisnika jer je neophodno dogovoriti da je količina usluge neograničena te kvalitetno i kontinuirano vodi briga o informacionom sistemu klijenta.

Usluga instalacije mrežnog operativnog sistema na server :

- ◆ instalacija operativnog sistema
- ◆ instalacija file servisa
- ◆ instalacija print servisa
- ◆ instalacija dial-out servisa
- ◆ instalacija mail servera unutar sistema
- ◆ isporuka, testiranje, fino konfiguriranje i puštanje u pogon

Usluga instalacije dopunskih servisa na serveru[50]:

- ◆ firewall \*
- ◆ dial-in \*
- ◆ antivirusni software
- ◆ razno \*
  - Usluga hardware-ske instalacije ili intervencije na serverskom računaru, radnoj stanici ili infrastrukturi
  - Usluga software-ske instalacije ili intervencije na serverskom računaru, radnoj stanici ili infrastrukturi
  - Usluga izrade projekta i pripadajuće projektne dokumentacije strukturnog kabliranja
  - Usluga izrade projekta i pripadajuće projektne dokumentacije aktivne opreme informacionog sistema

Kod zaštita računarskih mreža i održavanja IS neophodno je uvijek imati u vidu potencijalne napadače i poteškoće u radu i funkcionisanju IS.

Kritični faktori uspjeha:

- ◆ Zaštitne aktivnosti moraju biti zasnovane na poslovnim zahtjevima i vođene od strane tima ili osobe zadužene za održavanje IS
- ◆ Vidljiva podrška i saglasnost od “top management”-a mora da postoji
- ◆ Mora da postoji dobro razumijevanje sigurnosnih rizika na potencijalne prijetnje i ranjivosti sistema
- ◆ Osnovni koncepti zaštite moraju biti izloženi svim menadžerima i zaposlenima kako bi svi bili upoznati sa važnošću zaštite
- ◆ Sažeta uputstva za primjenu zaštitne politike i standarda moraju biti distribuirana svim zaposlenima, kao i svim saradnicima koji nisu stalno zaposleni.

## 2.7 ZAKLJUČAK

Iako je pitanje izbora metodologije koja bi pokrila cijeli životni ciklus razvoja IS logično, možda bi bilo ispravnije razmišljati u smjeru izbora metoda koje ćemo koristiti u pojedinim fazama razvoja IS. Naime, ukoliko se i pokazalo da je određena metodologija dobra za razvoj

IS u jednoj organizaciju, to ne znači da će ista metodologija dati dobre rezultate u razvoju IS za neku drugu organizaciju. Ovo proizlazi iz činjenice da su organizacije čiji se IS projektiraju, u svojoj prirodi različite, pa zahtijevaju drugačije metode za određene faze životnog ciklusa razvoja IS. Prema tome, umjesto odabira cijelih gotovih metodologija za razvoj IS jedne organizacije, pravilnije je birati metode za pojedine faze životnog ciklusa razvoja IS i tako stvoriti “virtualnu metodologiju”. Pri tome treba biti oprezan, jer metodologija, da bi bila primjenjiva, mora ispuniti određene uslove [Dobr 98].

Općenito, metodom možemo smatrati organiziran postupak za postizanje određenog jasnog cilja. Metoda može imati formalne teoretske temelje, iako to nije nužnost. Metodologija, kao skup odabranih metoda, nastaje obično “destilacijom” iskustava u provedbi najbolje prakse (best practices), u određenom području ili domeni.

Da bi se izvršio pravilan izbor metoda u “virtualnu” metodologiju, treba razmotriti tri karakteristike metoda:

- 1) *Definicija metode.* Definicija metode se uspostavlja karakterizirajući osnovnu motivaciju metode, koncepte i teoretske temelje. Nju razvijaju oni koji su metodu stvorili prema inženjerskim načelima (npr. ontologija metode, dizajn formalnog jezika metode i sl.).
- 2) *Način primjene metode.* Način primjene metode uključuje sintaksu metode i postupke kojima se metoda primjenjuje. Mnoge metode imaju višestruku sintaksu. Ove sintakse su se razvijale vremenom ili imaju naglasak na različite koncepte u djelokrugu primjene metode. Način primjene metode je, u stvari, sučelje metode prema korisniku.
- 3) *Upotreba metode.* Ova komponenta metode opisuje kako primijeniti metodu u različitim situacijama, kao npr. kad se metoda primjenjuje zajedno s nekim drugim metodama ili samostalno.

Upotreba metodologije, odnosno familije metoda, može se objasniti kroz arhitekturu informacijskog sistema. John Zachman, tvorac “arhitekture informacijskog sistema”, upućuje na vezu između metoda i arhitekture IS [Zach 87]:

*“Ne postoji arhitektura, već niz arhitekturnih prezentacija. Ne stoji da je jedna dobra, a druga loša. Arhitekture su različite. One su komplementarne. Postoje duboki razlozi za stvaranje svake arhitekturne prezentacije. I postoje rizici povezani s ne stvaranjem bilo koje od arhitekturnih prezentacija”*

Konzistentan i pouzdan dizajn korektne arhitekturne prezentacije povezan je s uporabom određenih metoda. Svaka metoda i njena arhitekturna prezentacija, fokusira se na ograničen skup karakteristika organizacije i ignoriše one karakteristike koje nisu predmet razmatranja te metode. Radi toga nije moguće razviti “super metodu” čija bi arhitekturna prezentacija opisala sve karakteristike organizacije bitne za razvoj IS. Umjesto potrage za ovakvom “super metodom”, vrši se puno realnija potraga za skupom metoda koje će, kad se primjene na određenu organizaciju, dati njenu kompletnu informacijsku arhitekturu. Prema tome, arhitektura informacijskog sistema određene organizacije predstavlja skup arhitekturnih prezentacija nastalih primjenom odabranih metoda za pojedine faze životnog ciklusa razvoja IS. Cjeloviti pristup razvoju IS podrazumijeva stvaranje jedne ovakve arhitekture [Sowa 92].