

Sveučilište Jurja Dobrile u Puli  
Fakultet informatike u Puli

Tomislava Kolar

**POGREŠKE U UPRAVLJANJU RAZVOJEM  
INFORMACIJSKOG SUSTAVA**

Završni rad

Pula, rujan 2018. godine  
Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

Tomislava Kolar

**POGREŠKE U UPRAVLJANJU RAZVOJEM  
INFORMACIJSKOG SUSTAVA**

Završni rad

**JMBAG:0303061143, redovni student**

**Studijski smjer: Informatika**

**Predmet:** Ekonomika informacijskih sustava

**Znanstveno područje:** Društvene znanosti

**Znanstveno polje:** Informacijske i komunikacijske znanosti

**Znanstvena grana:** Informacijski sustavi i informatologija

**Mentor:** Doc. dr. sc. Ivan Pogarčić

Pula, rujan 2018. godine



## IZJAVA O AKADAMSKOJ ČESTITOSTI

Ja, dolje potpisana Tomislava Kolar, kandidat za prvostupnika Informatike, ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedopušten način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

Tomislava Kolar

U Puli, 2. listopada 2018. godine



## IZJAVA

o korištenju autorskog djela

Ja, Tomislava Kolar, dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Pogreške u upravljanju razvojem informacijskog sustava koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst, trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu sa Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 2. listopada 2018. godine

Potpis Tomislava Kolar

# SADRŽAJ

<b>1. UVOD</b> .....	1
<b>2. DEFINIRANJE INFORMACIJSKIH SUSTAVA</b> .....	3
2.1. Osnovni pojmovi .....	3
2.2. Klasifikacija informacijskih sustava .....	5
2.3. Modeli razvoja .....	8
<b>3. UPRAVLJANJE RAZVOJEM INFORMACIJSKIH SUSTAVA</b> .....	14
3.1. Faze razvoja životnog ciklusa .....	14
3.2. Upravljanje rizicima, kontrolom i metrikom .....	17
3.2.1. Upravljanje rizikom .....	18
3.2.2. Kontrole .....	21
3.2.3. Metrika .....	23
<b>4. POGREŠKE U RAZVOJU INFORMACIJSKIH SUSTAVA</b> .....	25
4.1. Problem: Model informacijskog sustava ne odgovara organizacijskom sustavu. ....	25
4.2. Problem: Projekti razvoja informacijskog sustava kasne. ....	26
4.3. Problem: Troškovi razvoja informacijskog sustava udvostručili su se s obzirom na planirane troškove. ....	26
4.4. Problem: Veliki dio poslovnog sustava nije obuhvaćen automatskom obradom podataka, te je potrebno dalje ulagati resurse da se kompjutoriziraju poslovni sustavi. ....	27
4.5. Problem: Nedostatak kadrova za razvoj softvera, te postojeći kadrovi ne prate brzinu razvoja informacijske tehnologije. ....	27
4.6. Problem: Nerazumijevanje korisnika ili rješavanje krivog problema. ....	28
4.7. Problem: Mijenjanje korisničkih zahtjeva. ....	28
4.8. Problem: Nepostojanje strategijskog plana informatizacije.....	28
4.9. Problem: Nisu se koristile metode za dizajn informacijskog sustava. ....	29
4.10. Problem: Slabo kontroliranje projekta. ....	29
4.11. Problem: Slaba kvaliteta projekta i programskog proizvoda. ....	29
4.12. Problemi s kojima se programeri susreću.....	30
4.12.1. Nerazumijevanje korisnika.....	30
4.12.2. Otklanjanje pogrešaka (Debugging) .....	30
4.12.3. Održavanje tehnologije .....	31
4.12.4. Komunikacija .....	31
4.12.5. Procjena vremena .....	32
4.12.6. Sigurnosne prijetnje .....	32
4.12.7. Rad s programskim kodom druge osobe .....	32

<b>5. ZAKLJUČAK .....</b>	<b>34</b>
<b>6.TABLICA SLIKA .....</b>	<b>35</b>
<b>7. LITERATURA .....</b>	<b>36</b>
<b>8. SAŽETAK.....</b>	<b>37</b>
<b>9. SUMMARY .....</b>	<b>38</b>

# 1. UVOD

Informacijska je tehnologija nezaobilazni dio današnjeg vremena i suvremenog poslovanja. Informacijski sustavi su u današnje vrijeme vrlo napredni i nalaze primjenu u svim segmentima modernog društva. Pavlič [1] definira informacijski sustav kao skup povezanih dijelova kojima je cilj pribaviti i prenijeti informaciju i podatke za funkcioniranje, planiranje, odlučivanje i upravljanje poslovnom organizacijom. Danas, u doba ovisnosti o internetu i sve većoj ovisnosti o tehnologiji poslovne organizacije, ne mogu opstati i razvijati se bez pomoćnih informacijskih sustava koji se temelje na informacijsko-komunikacijskoj tehnologiji. Informacijski sustavi moraju nam dati rezultate koji su točni, precizni, pravodobni i prilagođeni potrebama korisnika.

Kada je jednom informacijski sustav napravljen tu nije kraj; treba ga (kao i sve ostale sustave) pratiti, kontrolirati, popravljati i revidirati. Da bi to bilo jasnije, u tekstu ću prikazati definiranje informacijskih sustava kroz neke osnovne pojmove, klasifikaciju informacijskih sustava i modele njihova razvoja.

Razvoj informacijskih sustava nikako nije jednostavan proces, a da bi se informacijski sustav djelotvorno razvijao mora postojati upravljanje razvojem informacijskih sustava. Upravljanje razvojem informacijskih sustava objasniti ću kroz faze životnog ciklusa, te objasniti kako upravljati rizikom.

Kada govorimo o savršenstvu u životu, ono navodno ne postoji, pa tako ni kod razvoja informacijskih sustava. Uvijek postoji mogućnost da se prilikom faze razvoja i implementacije dogode nekakve pogreške u samom informacijskom sustavu; može doći do nekih neželjenih problema u organizaciji koja razvija informacijski sustav, može doći do problema s rokovima isporuke ili u ugovorima između naručitelja i korisnika. U pravilu ne postoji neki obrazac po kojemu se pogreške ili problemi u razvoju informacijskih sustava događaju, pa u tom slučaju nitko ne može sa sigurnošću reći onome tko razvija informacijski sustav gdje su najčešće pogreške ili problemi.

Cilj ovog rada je ukazati na postojanje mogućnosti da dođe do pogrešaka koje se mogu riješiti tijekom faza razvoja, implementacije ili održavanja informacijskih sustava. Neke pogreške, koje nisu povezane s razvojem informacijskih sustava već su rezultat neočekivanih problema prilikom razvoja informacijskih sustava, kao što su nemogućnost poštivanja rokova isporuke ili povećani troškovi razvoja, mogu se riješiti samo u izravnom kontaktu s korisnikom. Imajući u vidu sve gore navedeno, ovim radom nastoji se ukazati na pogreške, probleme i neočekivane situacije u razvoju informacijskih sustava.



## 2. DEFINIRANJE INFORMACIJSKIH SUSTAVA

### 2.1. Osnovni pojmovi

Dio svakog poslovnog sustava je informacijski sustav, a razvoj tog informacijskog sustava kontinuiran je i vrlo zahtjevan posao. Pavlić [1] informacijski sustav definira kao skup povezanih dijelova (softver, hardver, ljudi, informacije, procedure i komunikacijske mreže), kojima je glavni cilj pribaviti i prenijeti informacije i podatke za planiranje, funkcioniranje, odlučivanje i upravljanje poslovnom organizacijom.

Povezanost s informacijskim sustavima je sve veća. Svijet u kojem se danas živi ovisan je sve više o internetizaciji i novim tehnologijama, poslovne organizacije ne bi opstale bez informacijskog sustava koji se temelji na informacijsko-komunikacijskoj tehnologiji. Treba svakako napomenuti da se informacijski sustav ne mora uvijek koristiti informacijskom tehnologijom i da informacijsku tehnologiju zato možemo gledati kao sredstvo, a ne kao svrhu postojanja informacijskog sustava. Informacijski sustav ima zadatak prikupiti, pohraniti, obraditi, čuvati i isporučiti informacije koje su bitan činbenik za uspješno odvijanje poslovnog procesa.

Informacijski sustav sastoji se od dva pojma - „sustav“ i „informacija“. Sustav je skup dijelova (elemenata), veza između dijelova te osobina dijelova svrsishodno organiziranih za neki proces, po prema Pavliću [2]. Uz informaciju vežemo pojmove kao što su: podatak, signal, kanal i znanje.

Podatak se definira kao skup znakova zapisanih na određenom mediju. Podaci mogu biti različitih vrsta:

- alfanumerički: slova, brojevi, specijalni znakovi...
- grafički: umjetnička slika, nacrt, fotografija, ljudsko sjećanje na oblik
- zvukovi: životinjski glasovi, ljudski glasovi, šumovi, buka...
- film: u slikama i zvukovima koje čovjek može percipirati kao odvijanje događaja

Informacija je podatak, odnosno činjenica s određenim značenjem. Informacija donosi novost, otklanja neizvjesnost, obavještava o nečemu i služi kao podloga pri odlučivanju. Primatelj koji informaciju dobije određuje njenu vrijednost i može je protumačiti iz svojeg konteksta znanja o stvarnosti. Osoba koja ne zna protumačiti rečenicu na tom jeziku, ta činjenica je podatak. Znači, ovisno o kontekstu, za nekoga informacija može biti podatak, i obrnuto.

Znanje se može definirati kao skup uređenih informacija o nečemu što je ljudski razumljivo i korisno za bilo koje ljudsko djelovanje. Znanje se stvara u ljudskom umu kao inovacija i zapisuje se, odnosno prenosi na daljnje korištenje. Također, znanje se sastoji od informacija s naputcima kako djelovati.

Informacijski sustav je skup povezanih i organiziranih informacija koje predstavljaju sustav i dio su toga sustava. Informacijski sustav je skup dobro definiranih pravila, postupaka i običaja pomoću kojih ljudi, oprema, ili jedno i drugo, rade na tome sa svrhom dobivanja informacija koje će zadovoljiti potrebe nekih pojedinaca u nekoj poslovnoj situaciji.

Cilj informacijskog sustava je dostaviti u organizaciji pravu informaciju, na pravo mjesto, u pravo vrijeme, te uz minimalne troškove. Funkcije informacijskog sustava dijele se na: prikupljanje podataka, obradu podataka, pohranjivanje informacija i podataka, kao i dostavu tih informacija i podataka korisnicima.

Elemente informacijskog sustava čine:

- Hardware
- Software
- Lifewar
- Orgware
- Netware
- Dataware

## 2.2 Klasifikacija informacijskih sustava

Informacijski sustavi mogu se klasificirati kao opći (genetički) ili specijalizirani, i osiguravaju informacije korisne svojim klijentima i članovima za pomoć u funkcioniranju i efikasnom upravljanju. Informacijski sustav na primjeru jednog skladišta trebao bi osigurati informacije o stanju neke robe na policama, informacijski sustav koji se koristi u maloprodaji trebao bi izdavati račune kupcu za robu kupljenu u trgovini, a informacijski sustav liječnika opće prakse morao bi sadržavati povijest bolesti pacijenta, pružene medicinske usluge, izdane uputnice specijalistima, bivše terapije i sl.

Informacijski sustav po načinu prikupljene informacije može biti, po Pavliču [1]:

- Formalni informacijski sustav sadrži informacije prikupljene na pravilan i unaprijed definiran način koji je nužan za funkcioniranje organizacije.
- Neformalni informacijski sustav sadrži informacije koje su prikupljene u neformalnim razgovorima, idejama i osobnim percepcijama pojedinaca, od kojih su neke lažne, poluistinite ili istinite. Informacije prikupljene neformalno mogu ponekad biti važne za sustav, međutim, organizacije bi trebale razvijati i prikupljati formalne informacijske sustave čiji je nositelj računalo.
- Izvršni informacijski sustav je sustav o temeljnim poslovnim procesima neke organizacije. Prema tome, informacijski sustav brine se za podatke o, na primjer, nabavi, održavanju, proizvodnji, kreditiranju i sl.
- Upravljački informacijski sustav je sustav koji osigurava informacije za upravljanje. Taj sustav podržava funkcije uvida u informacije potrebne menadžmentu za upravljanje organizacijom. Upravljanje organizacijama odvija se na strateškim, taktičkim i operativnim razinama.

Informacijske sustave, ovisno o vrsti softvera, može se podijeliti na, po Pavliču [2]:

- Generički ili opći informacijski sustav je sustav koji se razvija za široko tržište, ne nužno od istog proizvođača, a koristi se u organizacijama različitih vrsta (trgovačka društva, ustanove, banke, udruge i sl.) za opće funkcije potrebne različitim organizacijama. To uključuje: obračun plaća, izdavanje faktura, plaćanje računa, elektroničku poštu, sustave održavanja i sl. Organizacije mogu biti potpuno različite, ali svaka od njih ima veći broj istih funkcija pa mogu koristiti isti generički softver. Generički softver je nezavisni programski proizvod, a prodaje se na širokom tržištu, razvija ga proizvođač softvera, a ne korisnik, pa je u odnosu na naručeni informacijski sustav povoljniji na tržištu. Generički softver ima niz prednosti: nižu cijenu, funkcionalnosti koje su definirali i ugradili stručnjaci iz nekog područja, nije potrebna analiza korisničkih zahtjeva, a rezultat je unaprijed gotov proizvod (softver). Nedostatak je što u tim proizvodima nema ugrađenih funkcionalnosti prema specijalnim zahtjevima pojedinih krajnjih korisnika, već se gleda potreba cjelokupnog tržišta.
- Specijalizirani ili naručeni informacijski sustav je informacijski sustav koji softverski proizvod prilagođava organizaciji. Taj softver se proizvodi po narudžbi, a koristi se kao, na primjer: bankarski informacijski sustav, sustav za veleprodaju, sustav za rezervaciju u turizmu, sustav za naplatu parkinga i sl. S obzirom na to da granica između općega i specijaliziranog sustava granica nije stroga, vidljiva je razlika u cijeni, vremenu uvođenja, održavanju i mogućnostima. Kao prednost naručenih softvera može se istaknuti postojanje baza podataka i programskih modula koji krajnjim korisnicima najbolje odgovaraju, prate razvoj organizacije i prilagođavaju alate novim potrebama. Omogućavaju integraciju i povezivanje različitih informacijskih podsustava u jednu cjelinu, a imaju i ljude na raspolaganju samo za potrebe određenih korisnika. Nedostaci toga sustava su: potreban duži vremenski rok za izgradnju jer ne postoji gotovo rješenje, cijena je puno veća te su moguće pogreške, jer je gotovo nemoguće razviti softver bez pogreške.

Informacijski sustavi u literaturi se uobičajeno razvrstavaju na:

- **Sustav za automatizaciju uredskog poslovanja:** je informacijski sustav koji uključuje primjenu različite informacijske tehnologije u uredskom poslovanju. Sustav uredskog poslovanja također je element informacijskog sustava koji služi kao podrška u uredskom poslovanju, te kao manipulator dokumentima, podacima i pomoć u odlučivanju, komuniciranju i arhiviranju [10].
- **Transakcijski sustav:** je vrsta računalnih informacijskih sustava (takozvane transakcijski orijentirane aplikacije) koji pruža potporu tekućem odvijanju poslovnog procesa, tj. obrada transakcija. Prati rutinske poslovne događaje kao npr. praćenje i obrada narudžbi, fakturiranje, praćenje i obračun zaliha... Neke od glavnih funkcija transakcijskih sustava su vođenje evidencije, izdavanje, izvještavanje, odnosno praćenje poslovnog procesa [5].
- **MIS** (Management Information System): sustav za upravljanje informacijama koji pruža informacije organizacijama koje zahtijevaju učinkovito i djelotvorno upravljanje. MIS sustavi razlikuju se od drugih informacijskih sustava jer se koriste za analizu, te pomoć u strateškim i operativnim aktivnostima. Ti sustavi daju informacije potrebne za donošenje važnih poslovnih odluka i rješavanje problema, a prikupljaju i uspoređuju podatke iz različitih aspekata projekta [11].
- **DSS** (Decision Support System): sustav potpore odlučivanja je vrsta računalnih informacijskih sustava koji se koristi kao pomoć pri odlučivanju. Na svim razinama upravljanja, s naglaskom na odlučivanje kod strukturiranih i nestrukturiranih zadataka. DSS je također računalni sustav za organizaciju, identifikaciju i dohvat informacija, te transformaciju i analizu informacija. Možemo ga koristiti i za analizu dobivenih rezultata ili izbor modela odlučivanja [6].
- **Sustavi za upravljanje znanjem:** to su informacijski sustavi koji kreiraju, konstruiraju, identificiraju, prikupljaju, obrađuju, organiziraju, strukturiraju,

distribuiraju, pretražuju i primjenjuju znanje. Cilj sustava temeljenih na znanju je potpora u organizacijskom učenju te organizacijskoj efektivnosti [12].

- **Ekspertni sustavi:** sustavi iz područja umjetne inteligencije koji su namijenjeni rješavanju složenih problema uskog područja za koje su specijalizirani, odnosno izrađeni. Kroz te sustave nastoji se znanje stručnjaka temeljeno na iskustvu i intuiciji prikupiti i staviti u službu javnosti [13].
- **ERP** je integrirani sklop softverskih aplikacija dizajniran za upravljanje i korištenje širokih potreba poduzeća koje uključuju razne aspekte: planiranje proizvoda, financijski menadžment, logistiku i ljudske potencijale. ERP sustavi su bili alati za izgradnju integriranog informacijskog sustava. To su softverski paketi koji pokrivaju važne poslovne procese organizacija kao što su: obrada upita, obrada narudžbi, obrada zahtjeva, fakturiranje, planiranje proizvodnje, praćenje proizvoda, kalkulacija materijala, naručivanje i sl. [1].

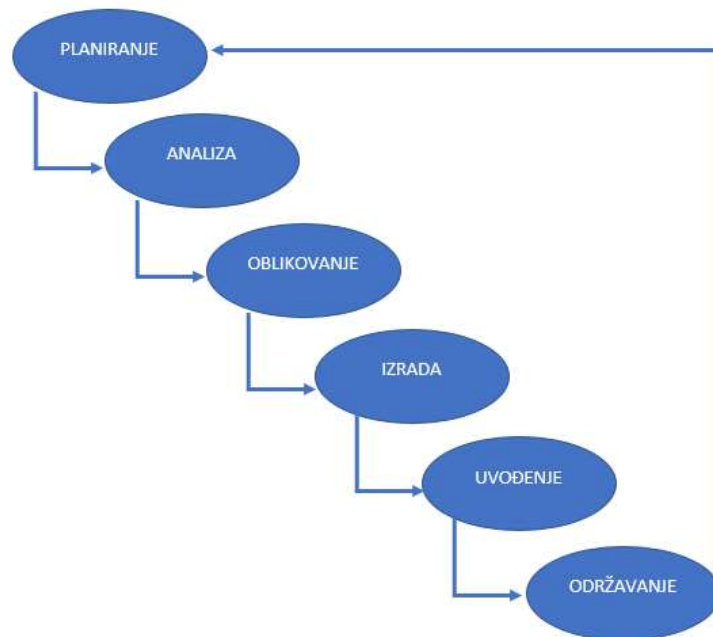
### 2.3. Modeli razvoja

Razvoj informacijskih sustava teče tako da se sukcesivno izvode aktivnosti i primjenjuju metode za izradu modela. U svakoj fazi razvoja koriste se ulazni skupovi informacija i pomoću metoda kao rezultat dobijemo model. Ne postoji jedna jedinstvena metoda za projektiranje svih vrsta informacijskih sustava, nego više odgovarajućih skupina metoda, gdje su neke metode definirane samo za određene faze.

Prema Pavliću [1], postoje sljedeći modeli razvoja:

- **Vodopadni model:** Vodopadni ili klasični model razvoja je aktivnost koja se izvodi slijedom kroz napredovanje iz faze u fazu. Kod tog modela nisu dopuštene naknadne promjene rezultata prethodnih faza. Dobar je za veće projekte i za dobro definirano okruženje kada je propisana procedura ručne obrade informacija ili računalni sustav, a rezultat je poznat iz prijašnjih projekata.

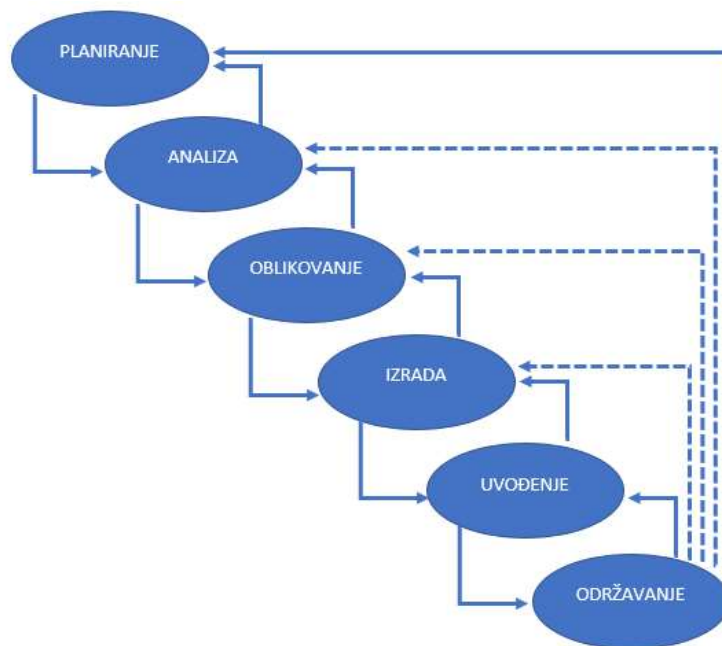
Slika 1 Model vodopada



Izvor: vlastita izrada prema: Pavlič, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka

- **Pseudostrukturni model:** je kaskadni model s ugrađenim mehanizmom povratne veze na ranije faze. Ovakav model omogućuje promjenu rezultata prethodnih faza i nastavak s napravljenim izmjenama.

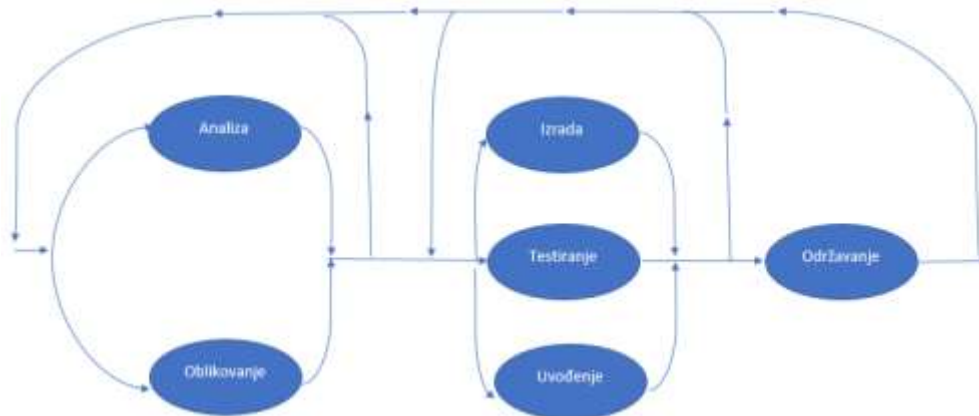
Slika 2 Pseudostrukturni model



Izvor: vlastita izrada prema: Pavlič, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka

- **Strukturni model:** definira faze razvoja s mehanizmom povratne veze, s tim da neke faze mogu teći istodobno, jer su relativno nezavisne. Osnovna ideja je da se faze izvode paralelno u skladu s logičkim oblikovanjem i fizičkom izvedbom sustava. Primjena ovog modela je najčešća u praksi, jer ima najveću slobodu, a zahtijeva projekt prije izrade bilo kakva proizvoda.

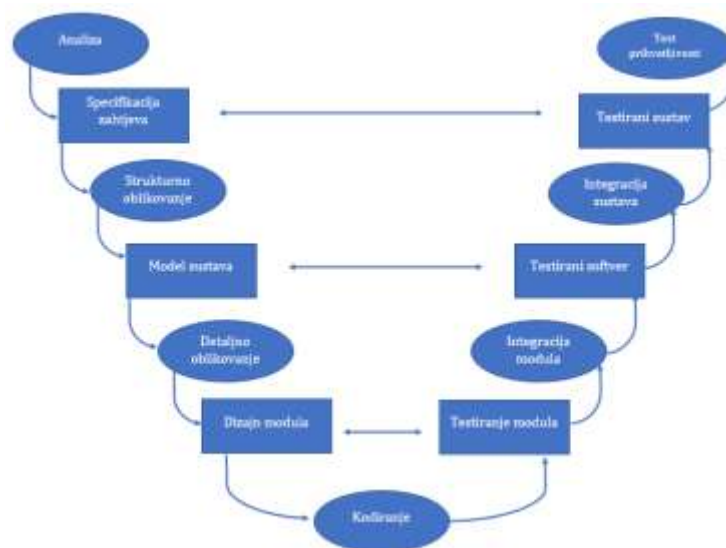
Slika 3 Strukturni model



Izvor: vlastita izrada prema: Pavlič, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka

- **V-model:** se izrađuje na osnovi činjenice da se sustav i proizvod sastoje od dijelova koji se opet sastoje od dijelova, i tako u više razina hijerarhije. Svrha modula je da se najprije razrađuje logički od cjeline prema detaljima, sve do logičkog modula koji će kodiranjem postati programski modul, a zatim da se od najmanjih programskih modula grade veće cjeline, i tako do kompletnog sustava. Taj je model podoban za manje softverske proizvode.

Slika 4 V-model



Izvor: vlastita izrada prema: Pavlič, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka

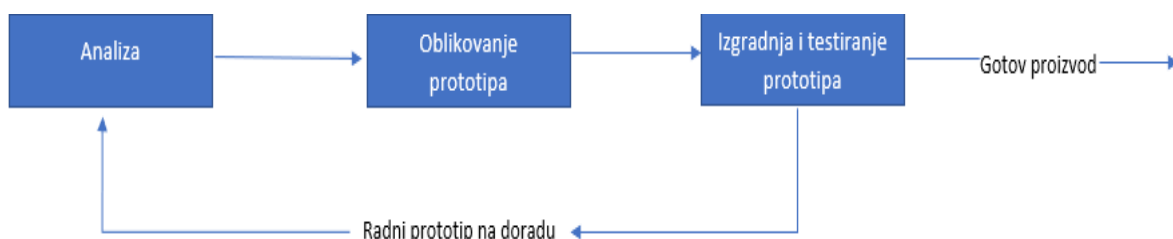


- **Prototipski model:** zamišljen je da izgradi prototip proizvoda koji nije gotov proizvod te se on provjerava i poboljšava dok ne zadovolji potrebe. Pogodan je za softverske proizvode ili dijelove informacijskog sustava. Kod prototipskog modela podržava se pristup u kojem se konačnom cilju približavamo mehanizmom povratne veze, a taj je pristup popraćen korištenjem naprednih softverskih alata za brzi razvoj programskog proizvoda, u kojem se brzo izrađuje ciljni softverski proizvod, takozvani radni model. Radni model daje se na korištenje pa korisnik daje svoje primjedbe i prijedloge za poboljšanje i izradu novog radnog modela. Prije izrade prvog prototipa potrebno je analizirati korisničke zahtjeve i dizajnirati radni model. Kada proizvod zadovoljava i postane dio sustava, takav pristup zovemo „brzo prototipiranje“.

Suprotno tome, u „ograničenom prototipskom modelu“ konačni se prototip ne mora koristiti kao rješenje u sustavu, nego se kao konačno rješenje gradi novi proizvod alatima, kod kojeg nisu moguće brze izmjene, ali se dobije kvalitetan proizvod na način i oblikom kakav je potvrđen prototipom.

- **Brzo prototipiranje** je brza izgradnja funkcionalnog proizvoda (prototipa), koji se onda ugrađuje kao rješenje u radni sustav koji je uobičajen za male projekte.

*Slika 5 Brzo prototipiranje*

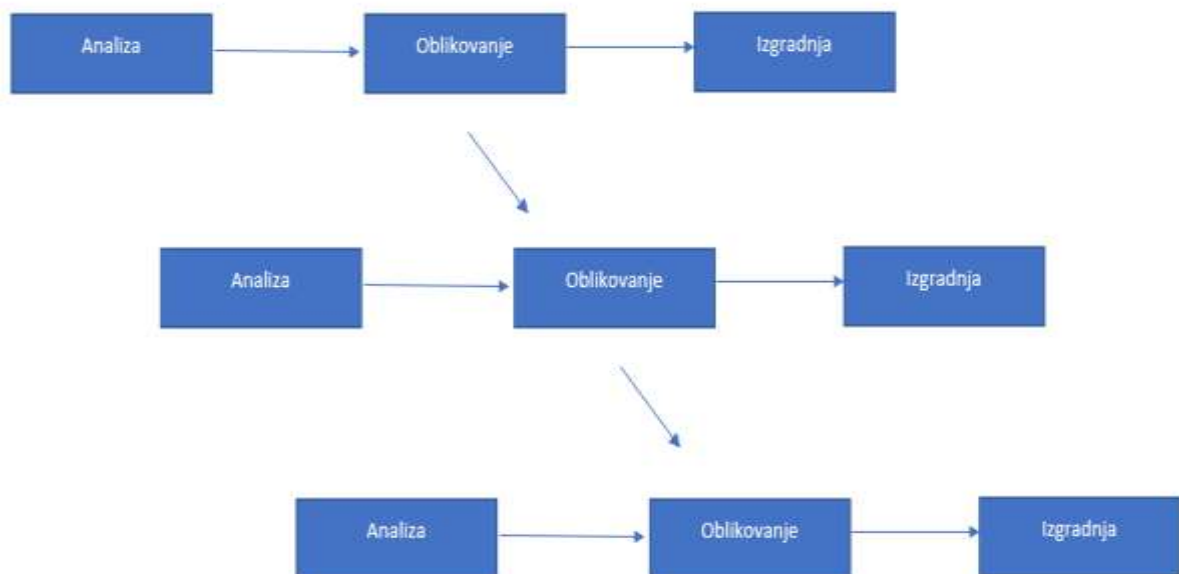


*Izvor: vlastita izrada prema: Pavlić, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka*

- **Ograničeno prototipiranje** je izrada nefunkcionalnog programskog proizvoda kao sredstva za utvrđivanje korisničkih zahtjeva i prikaz konačnog izgleda proizvoda. U određenom trenutku prototipiranje se zaustavlja i izrađuje se novi konačni programski proizvod.

- **Evolucijski model** sadrži ideju kontinuiranog rasta i razvoja. Informacijski sustav sam po sebi nikada nije gotov i on se stalno mijenja. Korisnici kod izgradnje informacijskog sustava prilikom rada uočavaju moguća poboljšanja sustava i daju ideje i prijedloge za izmjenu. Primjena aplikacije mijenja pogled korisnika i potrebe se mijenjaju, i to tako da najčešće rastu. Prvih nekoliko mjeseci korištenja broj zahtjeva je maksimalan, a zatim ostaje konstantan tijekom korištenja, sve do velikih reorganizacija sustava. Informacijski sustav raste s rastom organizacijskog sustava.

*Slika 6 Evolucijski model*

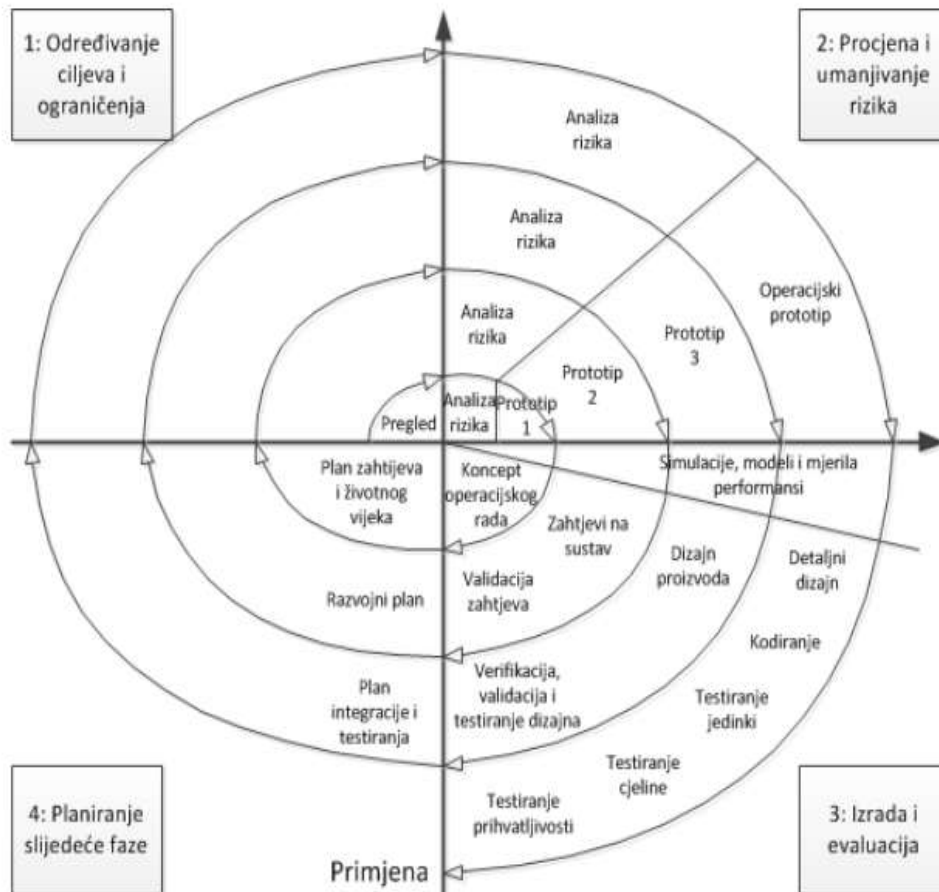


*Izvor: vlastita izrada prema: Pavlič, M.(2009):Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci, Rijeka*

- **Spiralni model** se odvija u fazama jedna za drugom u krug. Svaki pojedini krug predstavlja jednu fazu i krug je izrezan u isječke, od kojih svaki krug predstavlja jednu apstraktnu vrstu posla koja se izvodi u svakoj fazi. Svaki krug, odnosno faza, ima sve apstraktne aktivnosti za jednu vrstu cilja koji želimo postići u projektu tom fazom. Tako jedan od krugova može predstavljati izradu studije izvodljivosti, drugi može biti definicija korisničkih zahtjeva, treći krug može biti za dizajn sustava, četvrti krug implementacija, peti uvođenje i

integracija, a šesti krug održavanje. Krugovi u tom modelu predstavljaju faze ranijih modela.

Slika 7 Spiralni model



Izvor: preuzeto sa: <http://bib.irb.hr/datoteka/593801.NDraskovic-Diplomski.pdf>

- **Model integracije komponenti** fokusira se na proces integracije gotovih komponenti u integrirani sustav. Kroz određeno vremensko razdoblje nastao je veliki broj različitih rješenja u obliku gotovih komponenti, koje se mogu ponovo koristiti. Mnogo je gotovih projekata koji su rezultirali dobrim rješenjima, pa je poželjno da se prije razvoja pogleda stanje u sličnim projektima i sustavima. Nakon toga se ocjenjuje koliko je potrebno izmijeniti gotove komponente i ugraditi ih u sustav. Takav postupak ubrzava razvoj i često se koristi u softverskom inženjeringu.

## 3. UPRAVLJANJE RAZVOJEM INFORMACIJSKIH SUSTAVA

### 3.1. Faze razvoja životnog ciklusa

Razvoj informacijskih sustava počeo je od 1950 godine. Tradicionalna metodologija SDLC (eng. System Development Life Cycle), po Pavliču [2], metodologija je koja je dizajnirana u kasnim 1960-ima i to je najstarija metodologija iz koje su kasnije nastale mnoge druge, koje ću ukratko opisati. Temeljno načelo u toj metodologiji je da se svi zahtjevi za funkcioniranje poslovne organizacije mogu prikazati kao:

1. Ulazni podaci (informacije)
2. Procesi (radnje, akcije)
3. Izlazni podaci (informacije)

Dalje se svaki od tih može detaljno razraditi u sastavne dijelove. Također su vrlo rano uočene različite vrste poslova: kompjutorskog operatera, programera, sustavnog analitičara poslovnih procesa, sustavnog analitičara IT-a.

SDLC metodologija ima ove faze:

1. Prethodno istraživanje (engl. Preliminary Investigation)
2. Analiza sustava (engl. System Analysis)
3. Oblikovanje sustava (engl. Systems Design)
4. Razvoj sustava (engl. System Development)
5. Implementacija sustava (engl. Systems implementation)
6. Održavanje sustava (engl. Systems Maintenance).

Faze su zajedno još nazvane „konvencionalna sustavna analiza“, „tradicionalna sustavna analiza“, „životni krug razvoja sustava“ ili „vodopadni model“.

Oblikovanje sustava je jasno definiranje budućeg sustava tako da se istakne kako će taj sustav raditi da bi postigao ono što je u fazi analize definirano kao potreba.

Oblikovanje uključuje detaljan način rada, kako ručnog, tako i automatiziranog dijela budućeg sustava. Dokumentacija koja nastaje ima sljedeći sadržaj:

- Planirane i realizirane aktivnosti na projektu
- Projektna dokumentacija: popis procesa rada sustava, definiranje programskih proizvoda za unos podataka u bazu podataka, definiranje programa i izlaza iz sustava, definiranje strukture datoteka
- Dokumentacija o testiranju i planovi uvođenja
- Tehnička dokumentacija (redoslijed izvođenja programa, zaštitne mjere, back-up verzije...)
- Korisnička dokumentacija

Razvoj zahtijeva veliki napor da se izgradi programski proizvod i podrazumijeva testiranja programskog proizvoda što je više moguće.

Implementacija bi bila uvođenje u sustav novog hardvera i novog softvera uz kontrolu. Sastavni dio faze je i obrazovanje korisnika koji će poslove izvoditi na novi način. Kako bi se smanjio rizik, često se neko vrijeme paralelno radi na starom i novom sustavu. Testiranja se također izvode i u toj fazi, da se vidi uspješnost primjene. Stari sustav prestaje s radom tek kada se pokaže puna funkcionalnost novog sustava.

Održavanje informacijskih sustava značilo bi da se mijenjaju dijelovi informacijskog sustava koji je sam po sebi promjenjiv sustav, i on se prilagođava promjenama u organizacijskom sustavu. Postoje centri za informatiku koji se uglavnom bave održavanjem postojećih informacijskih sustava i rijetko razvijaju nove sustave. Problem održavanja su troškovi, koji su sve veći.

Nedostaci SDLC metodologije su:

- Opterećenost održavanja: Opterećenost održavanja javlja se zbog bržeg razvoja i prljavih rješenja koja dovode do lošeg dizajna, a samim time i do problema održavanja sustava.

- **Nezadovoljstvo korisnika:** Nezadovoljstvo korisnika javlja se jer oni vide gotov sustav i rade na provođenju toga sustava kada su korekcije u zahtjevima nemoguće i neovisno o njihovom intenzitetu i vremenu participacije tijekom dizajna i analize. Korisnici kada vide taj sustav, ocjenjuju ga i mogu biti nezadovoljni njime, pa je potrebna svakako i obuka korisnika metodama promatranja sustava na više razina apstrakcije.
- **Neuspjeh u zadovoljavanju potreba menadžmenta:** Budući da informacijski sustav uspješno rješava temeljne poslovne procese, menadžment ostaje bez njemu potrebnih informacija. Uglavnom svi informacijski sustavi zasnivaju se na obradi podataka za neke rutinske poslove koji se stalno ponavljaju na niskoj razini operativnih zadataka.
- **Neambiciozno oblikovanje novog sustava:** Neambiciozno oblikovanje novog sustava. Informacijski sustavi zasnovani na ručnoj obradi informacija neodgovarajući su u slučaju čestih promjena u sustavu. Taj pristup uglavnom uvodi IT, a novi sustav sličan je starom sustavu.
- **Procesi su nestabilni elementi u sustavu:** Konvencionalna SDCL metodologija zasnovana je na procesima. Ona pokušava poboljšati način na koji procesi u poslovanju izvode. Kako se mijenja poslovanje, naravno da postoji potreba i za promjenama procesa, odnosno programskih proizvoda. Postoje metodologije koje grade informacijski sustav i njegov model na konceptima koji su stabilni i nemaju taj nedostatak.
- **Konačan softverski proizvod je teško promjenjiv:** Tijekom razvoja na razini baze podataka i na razini programa nad bazom dobijemo oblik konačnog informacijskog sustava. Ako ima potrebe za promjenama sustava onda one zahtijevaju i promjenu oblika informacijskog sustava i njegovih komponenti, počevši od izmjene oblikovanja novog modela informacijskog sustava i samih programskih proizvoda. Ako dođe do takvih promjena, često dolazi do odgađanja predviđenog roka implementacije ili neodgovarajućeg informacijskog sustava.

- Problemi s dokumentacijom: SDLC je dobro dokumentirana metodologija, međutim, ta dokumentacija namijenjena je dizajnerima, a ne korisniku, i tu nastaje problem, jer je on može potpisati kao prihvatljivu i ne može biti odgovoran ako sustav koji dobije nije taj koji mu zaista odgovara.
- Nedostatak točnih procjena resursa: Procjene su se pokazale nerealnima iako metodologija omogućuje mrežno planiranje pristupu i procjenjuje potrebne resurse. Također složenost informacijskog sustava i neiskustvo nekih procjenitelja uzrok su nerealnih procjena. Posljedica toga je da se IT stručnjaci smatraju nepouzdanima, a to je onda i uzrok razočaranja u realizaciji informacijskog sustava.
- Spor razvoj novih aplikacija: Budući da postoje ručni informacijski sustavi koje treba razviti, korisnici čekaju godinama razvoj programskog proizvoda. Korisnici mogu odustati od opravdanih zahtjeva za poboljšanjima sustava jer su svjesni problema sporog razvoja.
- Problemi s „idealnim“ pristupom: SDLC metodologija zasniva se na ideji korak po korak i gore-dolje, priroda toga posla je ponavljajuća, pa se novi zahtjevi mogu pokazati nepotrebnima, niz koraka može se preskočiti, a i ljudi ne mogu robovati apstraktnim propisima metodologije.

### 3.2. Upravljanje rizicima, kontrolom i metrikom

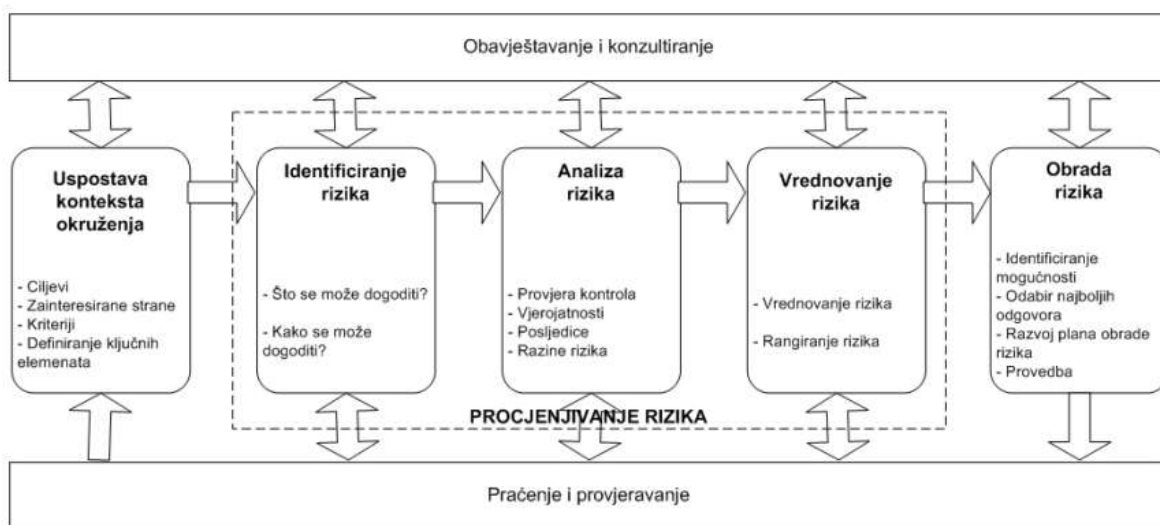
Usporedbu koncepata i metoda u području upravljanja informacijskim sustavima razmotrit ćemo kroz analizu triju tipičnih područja u sklopu upravljanja u kojem postoje sličnosti. Jedno od područja je područje upravljanja rizikom, zatim područje sigurnosnih, odnosno IT kontrola, te područje metrike, odnosno općenito mjerenje u upravljanju procesima.

### 3.2.1. Upravljanje rizikom

Koncept metoda procjenjivanja i upravljanja rizicima informacijskih sustava u osnovi se sastoji od tri faze [4]:

1. Procjenjivanje rizika (identificiranje, analiza i vrednovanje),
2. Obrada rizika (izbjegavanje, smanjenje, prihvaćanje i prijenos),
3. Uspostava cjeloživotnog procesa upravljanja rizikom

Slika 8 Faze procesa upravljanja rizikom



izvor: preuzeto sa: [https://bib.irb.hr/datoteka/521789.UIS\\_seminar\\_AK\\_28062010.pdf](https://bib.irb.hr/datoteka/521789.UIS_seminar_AK_28062010.pdf)

Da bi upravljanje rizikom sigurnosti informacijskog sustava bilo što učinkovitije i detaljnije u sklopu određene organizacije, nužno ga je integrirati u životni ciklus razvoja informacijskog sustava. Tablica 1 prikazuje podjelu životnog ciklusa razvoja sustava, ali i osnovne karakteristike faza razvoja, a u desnoj koloni prikazana je mogućnost integracije mjera upravljanja rizikom u svakoj od faza životnog ciklusa.

Tablica 1 Upravljanje rizicima kroz faze životnog ciklusa

Faze životnog ciklusa Razvoja sustava (SDLC)	Karakteristika faze	Potporna aktivnosti upravljanja rizikom
1. Početni zahtjevi	Izražava se potreba te dokumentira namjena i obujam IT Sustava	Identificirani rizici se koriste u razvoju zahtjeva sustava, uključujući sigurnosne zahtjeve, kao i sigurnosni operativni koncept (strategija)
2. Razvoj ili nabava	IT sustav je projektiran,	Rizici koji su identificirani



	nabavljen, programiran, razvijen ili na drugi način konstruiran	tijekom ove faze mogu biti korišteni za podršku sigurnosne analize IT sustava koja može voditi u potrebu izmjena arhitekture ili projekta tijekom razvoja sustava
<b>3. Izvedba</b>	Sigurnosna obilježja sustava trebaju biti konfigurirana, omogućena, ispitana i potvrđena	Proces upravljanja rizicima podržava procjenjivanje izvedbe sustava sukladno početnim zahtjevima i u sklopu modeliranog operativnog okruženja. Odluke koje se tiču identificiranih rizika moraju biti donesene prije puštanja sustava u rad i odgovarajuće sigurnosne kontrole trebaju biti implementirane
<b>4. Primjena i održavanje</b>	Sustav izvodi predviđene funkcionalnosti. Tipično, sustav se prilagođava na stalnoj osnovi kroz dodavanje sklopovske ili programske podrške, kao i promjenama organizacijskih procesa, politika i procedura.	Aktivnosti upravljanja rizikom provode se kroz periodičnu re-akreditaciju sustava ili kada se događaju velike promjene u IT sustavu u njegovoj operativnoj-produkcijskoj okolini (npr. nova sučelja na sustavu – ispitivanje kontrola i zaštitnih mjera).
<b>5. Odlaganje</b>	Ova faza može uključivati odlaganje sklopovske i programske podrške te dokumentacije sustava. Aktivnosti mogu uključivati prenošenje, arhiviranje, uništavanje podataka, kao i odgovarajuće zbrinjavanje sklopovske i programske podrške.	Aktivnosti upravljanja rizikom provode se za komponente sustava koje će se odlagati ili zamijeniti, kako bi se osiguralo odgovarajuće zbrinjavanje sklopovske i programske podrške te preostalih podataka u komponentama sustava, odnosno provođenje migracije sustava na siguran i sustavan način.

izvor: preuzeto sa: [https://bib.irb.hr/datoteka/521789.UIS\\_seminar\\_AK\\_28062010.pdf](https://bib.irb.hr/datoteka/521789.UIS_seminar_AK_28062010.pdf)

Pojam operativni rizik u novije se vrijeme povezuje s upravljanjem rizikom na području informacijske sigurnosti. Definicija operativnog rizika [4] opisuje operativni rizik kao rizik gubitka nastao zbog neodgovarajućih unutarnjih poslovnih procesa ili procesa u kojima postoje slabosti ili pogreške, odnosno zbog ljudskog faktora i tehničkih sustava ili zbog vanjskog događaja. Kod suvremenog pristupa informacijskog upravljanja rizicima postoji metoda koja se temelji na identificiranoj imovini unutar obujma sustava upravljanja, zatim na identificiranim prijetnjama za tu imovinu, identificiranim ranjivostima koje bi te prijetnje mogle iskoristiti, kao i na procjeni utjecaja koje gubitak povjerljivosti, cjelovitosti ili raspoloživosti može imati na imovinu. Zaštitne mjere štite imovinu, tj. vrijednosti koje je organizacija identificirala u sklopu obujma ISMS-a. Tako da se i tu može reći da definicija operativnog rizika odgovara metodi razvoja ISMS-a.

Ovako definirani operativni rizici uključuju sve ili gotovo sve oblike prijetnji, bilo da se radi u sklopu unutarnjih slabosti organizacije, prirodnih nepogoda, ili zbog drugih vanjskih prijetnji, te zbog namjernog ili nenamjernog napada. Stoga možemo reći da upravljanje operativnim rizikom znači balansiranje između rizika koji povezuje neku aktivnost i rizika koji povezuje neprovođenje te aktivnosti. Također, pojedinačni rizici međusobno utječu na način da ublažavanje jednog rizika gotovo sigurno povećava neki drugi. Prema tome, bitno je poznavati cjelokupno poslovanje u kojem se upravlja operativnim rizikom.

Nekoliko primjera karakterističnih rizika operativnog rizika [4]:

1. Rizik informacijske sigurnosti, povezan s neovlaštenim otkrivanjem, modificiranjem ili gubitkom raspoloživosti podataka koji imaju svojstvo povjerljivosti,
2. Rizik održavanja i operativnog okruženja poslovnog objekta, povezan s upravljanjem kontinuitetom poslovanja i održavanja IT-a,
3. Rizik sukladnosti s legislativom i regulativom, povezan s propisanim obavezama iz područja informacijske sigurnosti,
4. Rizik klimatoloških uvjeta i vremenskih nepogoda, povezan s upravljanjem kontinuitetom poslovanja.

Operativni rizici u svojoj osnovnoj podjeli na nezgode, otkaze i napade, tj. u općenitoj definiciji rizika od gubitka u tijeku poslovnih redovitih aktivnosti, temelj su za različite metode koje onda uvode detaljniju razradu rizika, a u sklopu definicije operativnih rizika. Kod upravljanja projektima isto je slična situacija koja, prepoznavanjem rizika, usmjerava prema osnovnim obilježjima projektnog okruženja kao što su: planovi, rokovi, razvojno okruženje, korisnički zahtjevi i dr. To su operativni rizici kod kojih može doći do gubitka zbog kašnjenja projekta, gubitaka zbog neispunjavanja korisničkih zahtjeva ili propasti samog projekta.

### 3.2.2. Kontrole

Prema [4] područje upravljanja informacijskim sustavima i područje sigurnosti informacijskih sustava usko su povezani i koriste sličan pristup upravljanju rizikom, a koji se primjenjuje u nizu suvremenih, međunarodnih i nacionalnih normi informacijske sigurnosti. Velik broj sigurnosnih prijetnji i ranjivosti povezan je sa suvremenom informacijskom i komunikacijskom tehnologijom. Tako da upravljanje rizikom ima za cilj svesti vjerojatnost rizika primjene informacijske tehnologije u okvire prihvatljive za upravu određene organizacije. A to znači da će vjerojatnost da neka prijetnja iskoristi ranjivost informacijskog sustava biti umanjena na odgovarajući način i u skladu s ciljevima i načelima pristupa organizacije riziku.

Prema [4] postoje dva ključna smjera upravljanja informacijskom tehnologijom kod primjene sigurnosnih kontrola u području sigurnosti informacijskih sustava. Prvi smjer bi bio životni ciklus informacijskih sustava; u sklopu ovog procesa moraju se prepoznati i na odgovarajući način sigurnosno usmjeravati sve specifičnosti informacijske tehnologije, koje dolaze u različitim fazama životnog ciklusa informacijskog sustava. Drugi smjer predstavlja specifičnosti IT kontrola i općenito se može prikazati kao na slici:

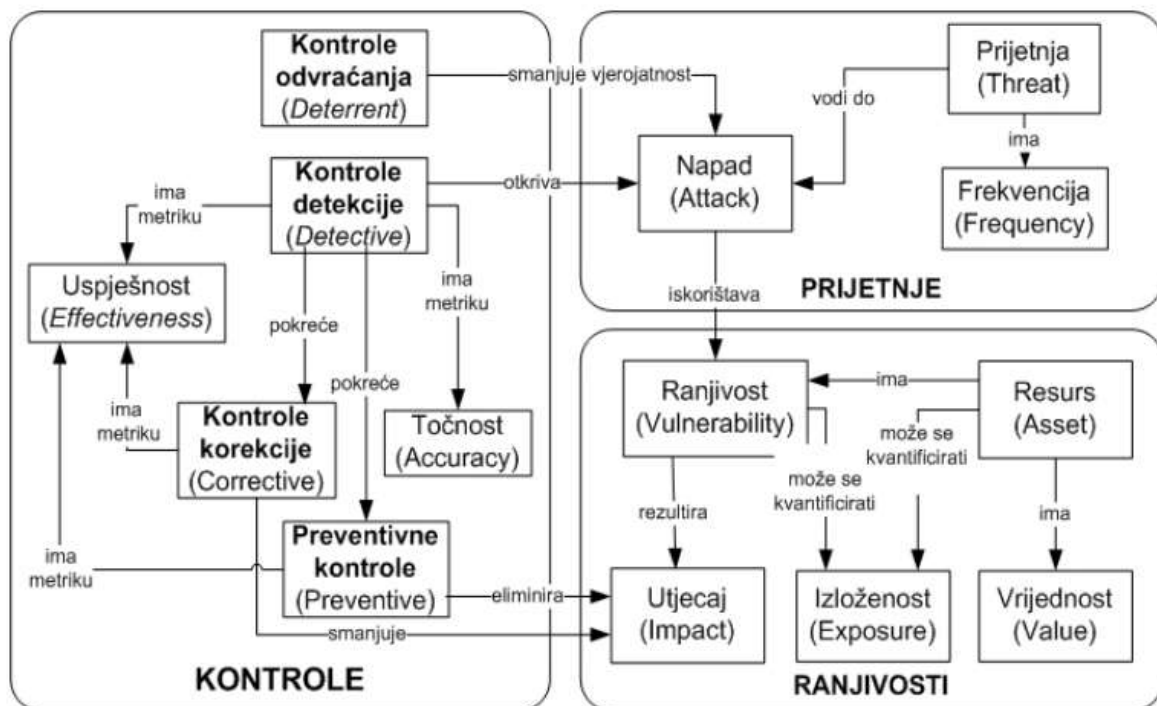
*Slika 9 Slojevi u okviru kojih se realiziraju IT*



izvor: preuzeto sa: [https://bib.irb.hr/datoteka/521789.UIS\\_seminar\\_AK\\_28062010.pdf](https://bib.irb.hr/datoteka/521789.UIS_seminar_AK_28062010.pdf)

Kao što na slici možemo vidjeti, ovaj model IT kontrola u stvari obuhvaća segment korporativnih upravljačkih politika (obavlja dio kojemu pripadaju zakonodavne obaveze, kao i politika informacijske sigurnosti), segment upravljačkih IT kontrola (srednji dio, koji obuhvaća regulativne obaveze i općenito upravljanje tehnologijom u okruženju), i na kraju skup tehničkih kontrola koje pokrivaju problematiku uvođenja informacijskih sustava kroz razvoj, kao i problematiku sustavske i aplikacijske programske podrške. Odabir kontrola u uskoj je vezi s obilježjima informacijskog sustava, a i s okolinom, tj. prijetnjama. Logični model sigurnosnih kontrola i način djelovanja, kao i njihova obilježja, utemeljeni su na analizi međudjelovanja kontrola, prijetnji i ranjivosti. Kod logičnog modela moguće je uvođenje metrika za pojedine veličine koje obilježavaju prijetnje, ranjivosti i kontrole. To je prikazano na slici:

Slika 10 Logički model sigurnosnih kontrola



izvor: preuzeto sa: [https://bib.irb.hr/datoteka/521789.UIS\\_seminar\\_AK\\_28062010.pdf](https://bib.irb.hr/datoteka/521789.UIS_seminar_AK_28062010.pdf)

Četiri vrste kontrola koje su opisane primjenjuju se kod različitih kategorija kontrola, a najčešće se koriste podjele na sljedeće kategorije sigurnosnih kontrola [4]:

1. Tehničke kontrole (engl. Technica),
2. Upravljačke kontrole (engl. Management),
3. Operativne kontrole (engl. Operational).

### 3.2.3. Metrika

Metrika je općenito sustav [4] ili postupak mjerenja, pri čemu su očekivana svojstva takvog mjerenja konzistentnost, lakoća prikupljanja podataka, kvantitativno izražavanje i korištenje jedinica mjere. Pod sigurnosnom metrikom podrazumijevamo analizu i interpretaciju podataka dobivenih mjerenjima, iz kojih je moguće zaključiti koje je akcije potrebno poduzeti radi uklanjanja sigurnosnog rizika i poboljšanja cjelokupnog stanja sigurnosti.

Danas dolaze zahtjevi za mjerenje iz različitih izvora, najčešće zakonodavno-regulatornih, te se odnosi na poslove revizije. Budući da je potreba za upravljanjem različitim procesima sve veća, tako se postavljaju zahtjevi za odgovarajuće mjerenje veličina na temelju kojih se upravlja procesima.

Kod informacijskih sustava tradicionalni pristupi upravljanju informacijskom sigurnošću, jednako kao i suvremeni, ne obraćaju dovoljno pažnju na mjerenje. Jedini načini prikupljanja podataka o stanju sigurnosti u većini organizacija su kroz proces upravljanja rizikom ili kroz neku vrstu revizije. Usmjerenost upravljanja rizikom nije na performansama ili strateškom povezivanju sigurnosti s poslovnim ciljevima, već na identificiranju rizika informacijske imovine i na razvoju sigurnosnih kontrola. Najčešće se ovdje radi o operativnoj razini podataka, ponekad taktičkoj, a rijetko strateškoj razini u smislu poslovnog upravljanja. Jedan od problema je ako je procjena subjektivna u metodama upravljanja rizikom. S druge, pak, strane, revizija osigurava najvećim dijelom "povijesne" podatke o usklađenosti sa zahtijevanom normom, pa teško može služiti za strateško upravljanje, odnosno procjenu trendova koji su značajni za upravljanje određenim procesom. Procesima kojima se upravlja, prati (engl. monitoring) i mjeri mogu biti upotrebljivi za upravljanje informacijskom sigurnošću. Cilj je dobiti potrebne mjerne veličine o stanju sigurnosti, radi donošenja odluka o usmjeravanju sigurnosnog programa (strateška razina) ili njegovu upravljanju (taktička-operativna razina).

Metrike možemo podijeliti na više načina [4], po tome što mjere (proces, performanse, rezultate, kvalitetu, trendove, uskladivost...), ili po tome kako mjere

(metode zrelosti, uravnotežene kartice postignuća, vrednovanje, statistička analiza,...), odnosno na temelju tih izmjerenih vrijednosti kao kvantitativne, kvalitativne i hibridne metode. Jedna od tipičnih metrika performansi su ključni indikatori ciljeva i ključni indikatori procesa, koji se općenito koriste za praćenje ostvarenja ciljeva, odnosno za praćenje djelotvornosti različitih procesnih aktivnosti u poslovnom ili sigurnosnom segmentu. Teško je ove indikatore nekad svrstati u upravljanje organizacijom ili projektom, odnosno upravljanjem sigurnošću, kao što je to slučaj za poslovni cilj održavanja tržišnog ugleda, pri čemu je ključni indikator cilja broj incidenata koji narušavaju tržišni ugled. Indikator cilja kao takav može se tretirati i kao poslovni, strateški indikator, ali i kao sigurnosni indikator, a predstavlja primjer poveznice strateške i taktičke razine.

Slika 11 Obilježja sustavskih metrika



izvor: preuzeto sa: [https://bib.irb.hr/datoteka/521789.UIS\\_seminar\\_AK\\_28062010.pdf](https://bib.irb.hr/datoteka/521789.UIS_seminar_AK_28062010.pdf)

Obilježja metrike moguće je opisati [8] odgovorima na pitanja koja se vide na prethodnoj slici, čime se može provesti i kategorizacija različitih metrika. Uobičajena podjela metrika je tehnička (tehnički objekti – algoritmi, uređaji, projekti...), organizacijska (procesi i programi) i operativna kategorija metrika (opisuje kakvi su sustavi, operativne prakse i specifična okruženja). S obzirom na to da je područje metrike u sustavskom inženjerstvu, kao i u sigurnosnom inženjerstvu, dosta dugo prisutno, ova grana razvoja još je dosta nerazvijena i postoji dosta nejasnoća i nerazumijevanja u korištenju pojmova metrike, mjerenja ili indikatora.

## 4. POGREŠKE U RAZVOJU INFORMACIJSKIH SUSTAVA

U svakodnevnom životu događaju se neke pogreške, pa tako i kod samog razvoja informacijskog sustava. Cilj svake metode razvoja informacijskog sustava je stvoriti kvalitetan i produktivan sustav, koji treba zadovoljavati utvrđene zahtjeve, te mora biti razvijen na vrijeme i unutar planiranih troškova. U nastavku će biti nabrojane neke od pogrešaka do kojih može doći.

### 4.1. Problem: Model informacijskog sustava ne odgovara organizacijskom sustavu.

Zašto dolazi do ovakvih pogrešaka? Osnovni je problem naći dovoljno dobar te dovoljno precizan sustav zapisivanja znanja. Grafički prikaz znanja danas je sadržan u različitim vrstama sematičkih mreža, u kojima je znanje smješteno u čvorovima, koji predstavljaju složene koncepte, te binarnim vezama između čvorova. Kod crtanja dijagrama koriste se: klasifikacija, agregacija, generalizacija i hijerarhija, te takve sheme sadrže znanje o interakcijama među činjenicama i nije potrebno neko dodatno traganje za znanjem. Baze podataka sadrže činjenice koje su korisne samo za pretraživanje tipa pitanje-odgovor, dok programi sadrže činjenice korisne samo za vrijeme izvedbe procedure. Modeli ne mogu samostalno zaključivati na osnovi ugrađenih činjenica bez neke interakcije s vanjskim svijetom. Nije dopuštena nepotpunost opisa sustava, odnosno postupan rast i razvoj opisa i aplikacije tijekom korištenja. Traže se potpunost i sveobuhvatnost odmah od početka rada. Projektant stvori viziju i dizajnira model jednom zauvijek.

Korištenje već postojećih programa vrlo je ekonomično, te su troškovi za kupnju takvih programa minimalni, a većina zadovoljava osnovne potrebe korisnika.

## 5.2. Problem: Projekti razvoja informacijskog sustava kasne.

Tko je kriv za kašnjenje? Krivi su dijelom korisnik i njegovi promjenjivi poslovni sustavi, te dijelom informatičar sa svojim softverskim proizvodima. Razlozi za ta kašnjenja su najčešće:

- u 34 % slučajeva podcijenjena je složenost projekta
- u 34 % slučajeva ljudi su zauzeti održavanjem i drugim poslovima
- u 25 % slučajeva kasni oprema
- u 6 % slučajeva ljudska je pogreška
- u 1 % slučajeva krivi su alati za razvoj.

Informacijski sustavi su sustavi koji se često moraju mijenjati, te koji se prilagođavaju promjenama u organizaciji kako bi i dalje služili svojoj namjeni. Te promjene su opsežne, tako da projektanti tijekom projektiranja novih sustava često imaju potrebu za vraćanje na promjene gotovih informacijskih sustava, što dovodi do kašnjenja projektiranja novih informacijskih sustava. Kako bi se to kašnjenje spriječilo trebali bi nakon planiranja potrebnog vremena za razvoj sustava povećati planirano vrijeme, pa ga čak i udvostručiti u nekim slučajevima kad je sustav kompliciraniji.

Sigurno je da samo kašnjenje projektiranja povisuje troškove razmjerno s vremenom trajanja izrade projekta, a mogu narasti i do maksimuma, međutim, sustavi koji se na ovakav način razvijaju su u pravilu bolji i dugovječniji, što je na kraju isplativije za krajnjeg korisnika.

## 4.3. Problem: Troškovi razvoja informacijskog sustava udvostručili su se s obzirom na planirane troškove.

Stvarna cijena softvera informacijskog sustava bit će ona koja se dobije pažljivim računanjem te ako je udvostručimo. Ta metoda procjene je nepreporučljiva, jer uvijek postoje neke aktivnosti za koje ne možemo predvidjeti koliko će stajati. Softverske kuće koje razvijaju informacijske



sustave i uvode ih u organizacije povećavaju planiranu cijenu i više od očekivanog, neki i četiri puta više, kako bi se osigurali od gubitaka.

4.4. Problem: Veliki dio poslovnog sustava nije obuhvaćen automatskom obradom podataka, te je potrebno dalje ulagati resurse da se kompjutoriziraju poslovni sustavi.

Dan-danas se nađe pokoji sustav u kojem se poslovi obavljaju ručno, bez korištenja računala. Uvođenje novih informacijskih sustava već se godinama odgađa. Postojeći informacijski sustav je nedostatan, preostalo je malo kvalitetnih kadrova zbog negativne selekcije. Rješenje ovakvog problema je kupnja odgovarajućeg softvera ili izbor novog menadžera.

Najekonomičnije za korisnika je da uvede informacijski sustav i mijenja ga stalno, u skladu s napretkom tehnologije i sustava, pa samim time neće doći do ogromnih troškova za uvođenje novih sustava ili modernizaciju starih. Informacijski sustavi omogućuju puno brži i efikasniji rad te samim time smanjuju troškove krajnjeg korisnika.

4.5. Problem: Nedostatak kadrova za razvoj softvera, te postojeći kadrovi ne prate brzinu razvoja informacijske tehnologije.

Mreža korisnika informacijskih sustava danas raste takvom brzinom da ni jedna država ne može proizvesti dovoljan broj stručnjaka koji bi mogli pratiti taj brzi rast i razvoj.

Dodatnom edukacijom ili prekvalifikacijom mogao bi se djelomično riješiti taj problem, što bi uvelike povećalo brzinu razvoja informacijske tehnologije, ali i povećalo troškove onima koji razvijaju softvere, pa tako i onima koji ih koriste. Naravno da je veći broj educiranih djelatnika dobrobit za sve u lancu razvoja i korištenja informacijskih sustava, bez obzira na troškove.

#### 4.6. Problem: Nerazumijevanje korisnika ili rješavanje krivog problema.

Dizajner može krivo shvatiti korisničke zahtjeve te ih krivo interpretirati. Novi sustav obavlja funkciju koja nije ni od kakve koristi samom korisniku.

Troškovi ponovnog razvoja softvera su ogromni, promjena implementiranog softvera dugotrajan je proces, a krivo shvaćanje korisnika najčešće je dodatni trošak onome tko razvija softver.

#### 4.7. Problem: Mijenjanje korisničkih zahtjeva.

Početna točka kod razvoja informacijskog sustava je utvrđivanje korisničkih zahtjeva. Ako je korisnička specifikacija pogrešna, bit će takav i informacijski sustav. Također, tijekom razvoja informacijskih sustava može doći do promjene korisničkih zahtjeva i ako se sustav nastavi razvijati po staroj specifikaciji neće odgovarati korisniku te će biti neupotrebljiv. Do promjene korisničkih zahtjeva najčešće dolazi zbog:

- toga što korisnici ne znaju što žele
- ne slažu se sa svojim zahtjevima
- problema u komunikaciji, nerazumijevanja onoga što sustav treba raditi
- promjene organizacijskog okruženja.

Svaka promjena korisničkih zahtjeva odgađa rok isporuke i povećava broj radnih sati, što naposljetku donosi nove troškove za korisnika. Kako ne bi došlo do takve pogreške važno je da se promjene zahtjeva neprekidno prate cijelo vrijeme razvoja sustava, što će na kraju rezultirati manjim troškovima i bolje usmjerenim sustavom.

#### 4.8. Problem: Nepostojanje strategijskog plana informatizacije.

Ako se informacijski sustav razvija tako da se grade aplikacije jedna po jedna za određenu poslovnu funkciju, bez ikakvog strateškog plana temeljenog na

poslovnoj strategiji, dolazi do problema. Takav način razvoja ne donosi nikakvu korist korisniku te je potrebno ponovo razviti sustav.

Razvoj novog informacijskog sustava uvijek je skupa opcija pa prilikom razvoja treba imati dobar strategijski plan koji uključuje nadogradnju informacijskih sustava, što je uvijek povoljnija opcija za korisnika. Ako se na vrijeme planiraju buduće nadogradnje to će smanjiti troškove i na taj način korisniku uštedjeti vrijeme i novac.

#### 4.9. Problem: Nisu se koristile metode za dizajn informacijskog sustava.

Ako se ne koriste metode za razvoj informacijskog sustava svaka aplikacija predstavlja građevinu bez nacrtu. Tijekom razvoja informacijskog sustava potrebo je odabrati jednu metodu razvoja programskog proizvoda te nju prilagođavati za održavanje, jer se novije metode razvoja razlikuju od postojećih.

Odabirom dobre metode za razvoj sustava te temeljitim planiranjem možemo smanjiti troškove i ubrzati razvoj sustava. Ako je metoda koju smo odabrali dobra, onda imamo i dobre temelje na kojima se može raditi i u budućnosti, što donosi manje troškove i zadovoljnog korisnika.

#### 4.10. Problem: Slabo kontroliranje projekta.

Članovi razvojnog tima dosta često ne uoče pogreške na vrijeme ili vjeruju kako sve dobro radi te ne zatraže pomoć u kontroli proizvoda, pa dolazi do kašnjenja projekta.

Uvođenje dodatnih kontrola možda povećava troškove i vrijeme razvoja, ali izlazni proizvod sadrži manje pogrešaka, odnosno kvalitetniji je.

#### 4.11. Problem: Slaba kvaliteta projekta i programskog proizvoda.

Kvaliteta programskog proizvoda opisana je nizom standarada, a jedan od najpoznatijih je ISO 9001 za slučaj kada je proizvodnja prilagođena zahtjevima

korisnika. Jedna od najpoznatijih metoda podizanja kvalitete je primjena CMM ili Six Sigma metodologije.

## 4.12. Problemi s kojima se programeri susreću

### 4.12.1. Nerazumijevanje korisnika

#### **Problem**

Kod izrade softvera, najvažniji prioritet su korisnici. Naravno, mora se znati što točno korisnici žele. Korisnici mogu imati različita mišljenja o načinu na koji proizvod treba raditi, a i članovi razvojnog tima mogu imati drukčija mišljenja. Novi programeri teško shvaćaju što korisnici žele, jer su se rijetko našli interakciji s njima. Naravno, tehnike upravljanja projektima, kao što su agilne metode, olakšavaju razvojnim timovima ažuriranje softvera, jer se korisnički zahtjevi stalno mijenjaju tijekom cijelog razvoja softvera.

#### **Kako riješiti problem?**

Ljudi koji će koristiti proizvod bit će krajnji korisnici. Korisnici znaju koji zadatak program treba obavljati, ali ne i značajke, a posao programera je da to shvati. Jedno od rješenja je razgovor s ljudima koji imaju direktan pristup korisnicima, kao na primjer stručnjaci za korisnike i dizajneri. Njihov je najčešći zadatak pristupanje svakom proizvodu te izravan pristup ljudima koji će zapravo koristiti taj proizvod.

### 4.12.2. Otklanjanje pogrešaka (Debugging)

#### **Problem**

Otklanjanje pogrešaka zvuči dosta nadmoćno, a pogotovo za nove programere. Neke pogreške lako se uklanjaju, ali dosta njih i ne, što dovodi do kašnjenja u razvoju te frustracija kod programera. Bagovi su uobičajen stvar u programiranju. Čak i najbolje programirani program može ih imati te se one mogu popraviti.

### **Kako riješiti problem?**

Provođenjem bezbroj sati pokušavajući riješiti problem, a najbitnije je shvatiti zašto je do problema, odnosno pogreške, došlo.

#### 4.12.3. Održavanje tehnologije

##### **Problem**

Tehnologija svaki dan sve brže raste te programeri trebaju pratiti taj razvoj. Okviri, alati i biblioteke sve brže postaju neupotrebljivi, odnosno zastarjeli. Tako npr. front-end okviri obično traju godinu ili dvije prije dolaska novih, ažuriranih verzija. Ažurirane verzije su dobre, te učinkovitije i olakšavaju rad, ali se također treba brzo naviknuti na njih.

##### **Kako riješiti problem?**

Odvajanje vremena za upoznavanje s novim sustavima svakako može pridonijeti rješavanju toga problema.

#### 4.12.4. Komunikacija

##### **Problem**

Loša komunikacija je problem s kojim se suočava većina programera, pogotovo novih, te ona može izazvati sukobe na radnom mjestu. Ako se ne izgradi dobra komunikacija u timu, dolazi do problema.

##### **Kako riješiti problem?**

Kada je u pitanju razvoj softvera komunikacijske vještine su jednako važne kao i tehničke vještine. Bitno je povezati se sa svojim kolegama, te im postavljati pitanja o problemima s kojima se suočavamo tijekom razvoja softvera.

#### 4.12.5. Procjena vremena

##### **Problem**

Ako se ne zna dobro procijeniti koliko je vremena potrebno za projekt, dolazi do prekoračenja roka isporuke. Procjene su važne u razvoju softvera. One mogu biti osnova za cijenu te raspored projekta. Ako razvoj kasni, to može uzrokovati veliki broj problema i ugroziti povjerenje.

##### **Kako riješiti problem?**

Najbolji način za upravljanje velikim brojem zadataka je da se zadaci razbiju na niz manjih zadataka. Ako si dobro rasporedimo vrijeme i dobro isplaniramo razvoj neće doći do kašnjenja projekta.

#### 4.12.6. Sigurnosne prijetnje

##### **Problem**

Podaci su vrijedna roba. Klijenti se oslanjaju na programere te se nadaju da će njihove informacije čuvati od različitih prijetnji. Početnici najčešće zanemaruju sigurnosne rupe u svom kodu, te ne postaju svjesni tih rupa sve do trenutka kada se dogodi neki od sigurnosnih prekršaja.

##### **Kako riješiti problem?**

Teško je zaustaviti nekoga od pokušaja hakiranja, ali jedna stvar koja se može učiniti je osigurati se od uobičajenih metoda hakiranja kako korisnici ne bi ostali zakinuti te da ne bi procurile neke informacije.

#### 4.12.7. Rad s programskim kodom druge osobe

##### **Problem**

Svi zaposlenici moraju raditi na projektima koje je stvorio netko drugi. U programiranju se radi na programskom kodu koji je napisao neki drugi razvojni programer te takva situacija može dovesti do problema.

### **Kako riješiti problem?**

Rad na nekom drugom programskom kodu može stvarati problem, ali je to jedan od problema koji se lako daje riješiti. Jedan od najboljih načina je da se tom kodu pristupi kao izazovu te da se posveti neko određeno vrijeme tome kodu, da bismo shvatili način razmišljanja i rada drugog programera te njegov pristup i stil.

## 5. ZAKLJUČAK

Kroz ovaj rad može se vidjeti koliko je razvoj informacijskih sustava zahtjevan i složen proces, koji u pojedinim slučajevima može završiti na neočekivan način, s pogreškama, nefunkcionalan, da ne zadovoljava zahtjeve naručitelja i sličnim problemima. Da ne dođe do takvih problema potrebno je pridržavati se faza razvoja životnog ciklusa informacijskih sustava te upravljanja razvojem informacijskih sustava.

Pogreške koje se događaju vrlo često su rezultat nerazumijevanja između onoga tko razvija informacijski sustav i onoga tko ga naručuje, jer su najčešće te dvije strane stručnjaci u različitim poljima i dolazi do nesporazuma. Da bi se takva nerazumijevanja svela na što je moguće manju mjeru, prije razvoja informacijskih sustava potrebno se upoznati s područjem za koje se određeni informacijski sustav razvija te na taj način bolje razumjeti onoga tko informacijski sustav naručuje.

Pogreške i problemi ne mogu se predvidjeti i nema sheme po kojima se oni događaju. Stoga ih treba rješavati odmah po njihovu nastajanju, da ne bi prouzročili veću štetu. Razvojem tehnologije i alata za pomoć pri razvoju informacijskih sustava pogreške postaju sve rjeđe i čine manju štetu, ali i dalje moramo biti oprezni i držati se spomenutih faza u razvoju životnog ciklusa informacijskih sustava.

Brzina razvoja u današnje vrijeme je takva da društvo teško odgovara razvojem sustava obrazovanja stručnih kadrova i brojem proizvedenih stručnjaka, te je stoga na samim stručnim kadrovima da se sami usavršavaju, prate trendove razvoja, a svoja stečena znanja dijele sa zajednicom i pojedincima i na taj način dovedu do toga da se problemi i pogreške svedu na što je moguće manju razinu.

Iako su problemi i pogreške prisutni, to ne umanjuje značaj koji informacijski sustavi imaju u modernom društvu, a daljnjim razvojem taj se značaj samo još može povećati te pogreške i problemi smanjiti.



## 6.TABLICA SLIKA

Slika 1 Model vodopada .....	9
Slika 2 Pseudostrukturni model .....	9
Slika 3 Strukturni model.....	10
Slika 4 V-model .....	10
Slika 5 Brzo prototipiranje .....	11
Slika 6 Evolucijski model .....	12
Slika 7 Spiralni model .....	13
Slika 8 Faze procesa upravljanja rizikom.....	18
Slika 9 Slojevi u okviru kojih se realiziraju IT .....	21
Slika 10 Logički model sigurnosnih kontrola.....	22
Slika 11 Obilježja sustavskih metrika .....	24

## 7. LITERATURA

Knjige:

- [1] Pavlič, M. (2009): *Informacijski sustavi*, Odjel za informatiku Sveučilišta u Rijeci, Rijeka
- [2] Pavlič, M. (1996): *Razvoj informacijskih sustava*, Znak, Zagreb
- [3] Manger, R.(2016): *Softversko inženjerstvo*, Element, Zagreb

Ostalo:

- [4] Klaić, A. (2010): *Usporedba koncepata i metoda koje se koriste u područjima upravljanja informacijskim sustavima i upravljanja informacijskom sigurnošću - seminar, FER, Zagreb*
- [5] Ćurko, K. i Varga, M. (2010): *Što je informacijski sustav?- materijali za predavanja, Ekonomski fakultet, Zagreb*
- [6] Zekić-Sušac, M. (2016): *Sustavi za potporu odlučivanja - materijali za predavanja, Ekonomski fakultet, Rijeka*

Internet:

- [7] The 9 Most Common Problems New Programmers Face: <<https://simpleprogrammer.com/9-common-problems-new-programmers-face/>> [9. 9. 2018.]
- [8] The 6 most common problems in software development: <<https://unifaceinfo.com/the-6-most-common-problems-in-software-development/>> [9. 9. 2018.]
- [9] Problems in information system development: <[http://users.jyu.fi/~jpt/doc/thesis/ime-1\\_1.html](http://users.jyu.fi/~jpt/doc/thesis/ime-1_1.html)> [9. 9.2018.]
- [10] Uredski informacijski sustav: <<http://www.seminarski-diplomski.co.rs/INFORMACIONI%20SISTEMI/UredskiInformacijskiSustav%20.html>> [5. 9. 2018.]
- [11] Što je MIS/ERP sustav? <<http://www.printera.hr/sto-je-mis-erp-sustav/>> [5. 9. 2018.]
- [12] KMS: Knowledge Managment System <<http://www.infodom.hr/default.aspx?id=34>> [5. 9. 2018.]
- [13] Ekspertni sustavi : <<http://www.enciklopedija.hr/natuknica.aspx?id=17426>> [5. 9. 2018.]

## 8. SAŽETAK

Informacijski sustav je sustav koji se sastoji od skupa međusobno povezanih dijelova koji prikuplja i razmjenjuje informacije za planiranje, odlučivanje te upravljanje nekom organizacijom. Informacijski sustavi danas su vrlo napredni te se primjenjuju u gotovo svim područjima. Razvoj informacijskog sustava složen je proces, a da bi se taj sustav razvio na odgovarajući način trebamo dobro isplanirati razvoj informacijskog sustava. U pojedinim slučajevima razvoj informacijskog sustava može završiti na potpuno neočekivan način, s puno pogrešaka, nefunkcionalan, da ne zadovoljava korisničke zahtjeve, a troškovi su udvostručeni, te sličnim problemima.

**Ključne riječi:** informacijski sustav, razvoj, pogreške, problemi

## 9. SUMMARY

An information system is a system consisting of a set of interconnected parts that collect and exchange information for planning, decision making and management of an organization. Information systems are very advanced today and are applied in almost all areas. The development of the information system is a complex process, and in order to develop this system properly we need to plan the development information system well. In some cases, the development of the information system can end in a completely unexpected way, with lots of errors, dysfunctional, does not meet user requirements, costs are doubled and similar problems.

**Keywords:** information system, development, errors, problems