

IMPLEMENTACIJA INFORMACIJSKIH SUSTAVA

prof. dr.sc. Mile Pavlič, mr.sc. Patrizia Pošćić, mr.sc. Sanja Čandrlić, Perino Krneta

SAŽETAK

Rad definira pedesetak aktivnosti razvoja informacijskog sustava i detaljnije opisuje aktivnosti faze implementacije s naglaskom na uvođenje i održavanje sustava. Kod uvođenja detaljnije se opisuju aktivnosti: instalacija gotovog softvera na produkcijsko okruženje, izrada uputa, prezentacija gotovog softvera, obuka, završne konverzije, završno testiranje, početak primjene nove aplikacije, uspostava novog sustava i potpisivanje primopredajnog zapisnika.

Ključne riječi: implementacija, informacijski sustav, softver, programski modul, baza podataka, korisnik, testiranje.

ABSTRACT

This paper defines some fifty activities regarding Information System Development and thoroughly describes the activities in the Implementation Phase, with the emphasis on Information System implementation and maintenance. The following implementation activities are described in more detail:

Installation of new applications in a working environment, help guide creation, final application presentation, user training, final conversions, final testing, application establishment and, finally, signing the Bill of Sale.

1. UVOD

Prema većini metodika (Pavlič, 1999), «implementacija informacijskog sustava» je faza života ciklusa razvoja sustava koja slijedi nakon projektiranja IS. Projektiranje se sastoji od analize poslovanja sustava te od dizajna budućeg sustava. Nakon dizajniranja IS slijedi njegova implementacija.

Faze i aktivnosti životnog ciklusa razvoja IS su prikazane u slijedećim tablicama.

Definicija: Implementacija je rad na provođenju čega; provedba, izvršenje, primjena, ostvarenje (Hrvatski enciklopedijski rječnik, 2002).

Implementacija IS je skup aktivnosti koji modele iz projekta IS pretvara u dijelove strukture budućeg IS te novonastale dijelove uvodi u postojeći sustav mijenjajući postojeći informacijski sustav.

Definicija: Aplikacija je računalni program stvoren da obavlja neku zadaću (Hrvatski enciklopedijski rječnik, 2002).

Aplikacija je programski proizvod i baza podataka koji obavljaju određenu poslovnu funkciju ili njen kohezivni dio.

Izrada aplikacija je faza u kojoj se specifikacije logičkog dizajna iz faze projektiranja IS prevode u prave programske module.

Faza 1: STRATEŠKO PLANIRANJE IS (SP)

| |
|--|
| 1.1 Analiza: Definiranje i obuka tima, dekompozicija procesa, popis dokumentacije i kretanje kroz sustav |
| 1.2 Podsustavi: Određivanje podsustava i veza |
| 1.3 Prioriteti: Određivanje prioriteta |
| 1.4 Resursi :Definiranje cjelovite infrastrukture |
| 1.5 Plan: Planiranje glavnih projekata i aktivnosti |

Faza 2: GLAVNI PROJEKT (GP)

| |
|---|
| 2.1 PZ : Izrada projektnog zadatka |
| 2.2 DTP: Intervjuiranje, raščlanjivanje i modeliranje procesa (DTP) |
| 2.3 Procesi GP: Analiza procesa, problema i prijedloga poboljšanja |
| 2.4 Podaci GP: Opisivanje podataka |
| 2.5 Plan GP: Planiranje izvedbenih projekata |
| 2.6 Resursi GP: Definiranje modela resursa glavnog projekta |

Faza 3: IZVEDBENI PROJEKT (IP)

| |
|---|
| 3.1 DEV: Intervjuiranje, apstrakcija i modeliranje podataka (EV) |
| 3.2 Prevođenje: Prevođenje modela podataka u shemu BP (RM) |
| 3.3 Arhitektura IP: Definiranje arhitekture programskog proizvoda (APP) |
| 3.4 Operacije IP: Projektiranje operacija nad shemom BP |

| Faza 4: PROIZVODNJA SOFTVERA (PS) |
|--|
| 4.1: PLANIRANJE PROIZVODNJE |
| <ul style="list-style-type: none"> • Planiranje aktivnosti proizvodnje SW • Određivanje izvršitelja za pojedine zadatke i određivanje rokova • Određivanje i kreiranje produkcijske, testne i razvojne okoline |
| 4.2: OBLIKOVANJE BAZE PODATAKA |
| <ul style="list-style-type: none"> • Prevođenje logičkog modela podataka u fizički model sheme baze podataka • Kreiranje razvojne okoline za svakog pojedinog programera • Punjenje sheme baze podataka u razvojnoj okolini iz postojeće produkcijske BP • Dodavanje novih koncepata iz modela podataka u razvojnu shemu BP (tip entiteta, ...) • Kreacija razvojne baze podataka • Inicijalno punjenja testne baze podataka |
| 4.3: RAZVOJ PROGRAMSKOG PROIZVODA |
| <ul style="list-style-type: none"> • Izrada glavnog izbornika (aplikacijskog stabla) ili dorada već postojećeg • Izrada ekrana za pregled redaka svake tablice po jednom ili više ključeva • Izrada ekrana za operacije nad jednim retkom tablice (unos, izmjena, brisanje i pregled) • Izrada programskih modula različitih vrsta i namjena kao: obračuna, procedura, funkcija kontrola, look-up-ova nad tablicama (s prvim testiranjem modula) • Izrada izvještaja (s prvim testiranjem modula) |
| 4.4: TESTIRANJE U TESTNOJ OKOLINI |
| <ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u testno okruženje • Spajanje novih modula s postojećim • Back-up verzija softvera • Testiranje prototipa softvera nad testnom bazom podataka • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih grupa aktivnosti i ponovno testiranje |
| 4.5: TESTIRANJE I ISPRAVLJANJE U RADNOJ OKOLINI |
| <ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u radno okruženje • Spajanje novih modula s postojećim • Back-up verzija softvera • Punjenje baze podataka • Testiranje prototipa softvera nad produkcijskom bazom podataka od strane programera • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih grupa aktivnosti i ponovno testiranje |
| 4.6: TESTIRANJE OD STRANE KORISNIKA |
| <ul style="list-style-type: none"> • Presentacija softvera korisniku • Testiranje od strane korisnika • Izrada popisa primjedbi korisnika • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih grupa aktivnosti i ponovno testiranje • Izrada zapisnika o testiranju i prihvaćanju faze uvođenja |

| Faza 5: UVOĐENJE (UVO) |
|--|
| 5.1 Instalacija gotovog softvera na produkcijsko okruženje (kod korisnika) |
| 5.2 Izrada uputa |
| 5.3 Presentacija gotovog softvera |
| 5.4 Obuka |
| 5.5 Završne konverzije |
| 5.6 Završno testiranje |
| 5.7 Početak primjene nove aplikacije |
| 5.8 Uspostava novog sustava i potpisivanje primopredajnog zapisnika |

| Faza 6: PRIMJENA I ODRŽAVANJE (ODR) |
|--|
| 6.1 Podešavanje novog aplikacijskog sustava |
| 6.2 Izvješće o procjeni novog projekta |
| 6.3 Raspodjela odgovornosti korisnika i programera |
| 6.4 Korištenje aplikacijskog sustava |
| 6.5 Postavljanja zahtjeva za izmjenama od strane korisnika |

Implementacija IS sastoji se od slijedećih faza:

- 9 proizvodnja softvera s oblikovanjem baze podataka,
- 10 uvođenje novog IS,
- 11 primjena i održavanje IS.

Faza «Proizvodnja softvera» započinje nakon izrade izvedbenog projekta IS uzimajući u obzir postojeće programske proizvode i bazu podataka, a ima za cilj reorganizirati bazu podataka IS i kreirati potreban novi programski proizvod odnosno izmijenjen postojeći programski proizvod.

Proizvodnja se naslanja na postojeće aplikacije i model dizajniran od strane projektanta. Planiranje proizvodnje je procjena aktivnosti i njihovog trajanja.

Timaska i profesionalna proizvodnja softvera odvija se na tri razine i to: mjesto gdje radi programer tzv. razvojna okolina, mjesto gdje se testira rezultat rada svih programera tzv. testna okolina i mjesto gdje se nalazi radna verzija operativne baze podataka i programa tzv. produkcijska okolina. Potreba za ove tri razine dokazana je na velikim razvojnim projektima.

Testiranje gotovog programskog proizvoda je važno i smješteno je na sve razine od strane više aktera.

Gotov programski proizvod se uvodi u sustav odnosno primjenjuje u svakodnevnom radu.

2. FAZA: UVOĐENJE

Izrađen novi programski proizvod treba zauzeti mjesto u poslovnom sustavu i zamijeniti postojeću «staru» aplikaciju. Proces zamjene stare novom aplikacijom nazivamo uvođenje. Kad se aplikacija uvede, počinje njeno dnevno operativno korištenje i tada se sustav nalazi u fazi primjene aplikacije. Uvođenje je razdoblje od nekoliko dana do nekoliko mjeseci. Primjena programskog proizvoda traje više godina. Uvođenje može biti i neuspješno, ako se pokažu nepremostivi problemi. Tada se planiraju korektivne radnje vezane uz probleme: izmjena softvera, dorada baze podataka, dodatna obuka, i dr. Po uklanjanju uzroka odgađanja uvođenja ponovo se pristupa uvođenju. Ovi se postupci mogu ponoviti više puta.

Aktivnosti faze uvođenja jesu:

- Instalacija gotovog softvera na produkcijsko okruženje (kod korisnika)
- Izrada uputa
- Prezentacija gotovog softvera
- Obuka
- Završne konverzije
- Završno testiranje
- Početak primjene nove aplikacije
- Uspostava novog sustava i potpisivanje pripremljenog zapisnika

Opišimo pojedine aktivnosti detaljnije.

2.1 Instalacija gotovog softvera kod korisnika

Ako je novi kupac prihvatio aplikacijski sustav od dobavljača analizirajući ga na privremenoj testnoj lokaciji, svojoj ili kod dobavljača, koja neće postati dio produkcijskog okruženja korisnika, onda treba pristupiti nizu radnji koje nazivamo «priprema produkcijske IT infrastrukture».

Instalacija gotovog softvera na produkcijsko okruženje se planira u suradnji s voditeljem projekta od strane korisnika. On treba dati svoju suglasnost i pripremiti sustav na promjenu. Ponekad je testiranje od strane korisnika izvedeno na produkcijskom okruženju tijekom

faze proizvodnje i tada nema posebne instalacije gotovog softvera.

Priprema informatičke infrastrukture se sastoji od priprema i uvođenja potrebnih IT resursa (prostor, mreža, LAN, sigurnost, operacijski sustav, potreban pomoćni softver, ljudi, ...) kao preduvjeta instalacije aplikacije. Ponekad ove radnje traju i nekoliko mjeseci.

2.2 Priprema poslovnog sustava

Priprema poslovnog sustava se sastoji od: izrada uputa za korisnike aplikacije, prezentacija gotovog softvera korisnicima i menadžmentu, detaljne obuke korisnika, završnih konverzija i završnog testiranja. Ove aktivnosti se mogu paralelno izvoditi s aktivnostima faze proizvodnje aplikacije. Dakle, aktivnosti uvođenja započinju prije gotovosti programskog proizvoda. Opišimo ukratko pojedine pripreme radnje.

Upute

Upute sadrže dio koji je namijenjen korisnicima (korisničke upute) i dio koji je namijenjen informatičarima korisnika (tehničke upute), a opisuju: instalacijske procedure, postupke spašavanja podataka, tehničke preduvjete, ograničenja i sl.

Upute za korisnika je moguće napraviti na dvije razine i to: detaljne i skraćene. Detaljne potpuno opisuju programski proizvod i namijenjene su početnicima. Skraćene upute, koje se češće koriste u svakodnevnom radu, pomažu korisnicima s iskustvom da brzo nađu odgovore na najvažnija i najčešća pitanja, a oni su zapisani na skraćeni način na nekoliko stranica.

Pored uputa u obliku knjige, upute trebaju biti dostupne i kroz programski proizvod pritiskom na tipku F1.

Prezentacija gotovog softvera i obuka

Prezentacija gotovog softvera i obuka su slične aktivnosti. Prezentacija predstavlja prvo upoznavanje korisnika s novim softverom, a obuka je detaljno prenošenje znanja na korisnike kako bi korisnik mogao dnevno operativno brzo i kvalitetno obaviti svoj posao. Dokazano je da (Carey, 1988) što duži period korisnik participira u proizvodnji softvera na testiranju ili drugim aktivnostima, to manji otpor pruža prema tim promjenama.

Obuku treba planirati. Ponekad se treba obučiti stotine korisnika geografski dislociranih, te je planiranje mjesta, vremena, polaznika, predavača i nastavnog sadržaja složeno. Plan obuke treba imati slijedeće elemente: tko izvodi obuku, tko od korisnika prolazi obuku, gdje se obuka izvodi, kada započinje, kada završava, što je sadržaj obuke, kako se provjerava je li postignuta potrebna razina znanja.

Nakon što je obuka izvršena, korisniku se ostavlja vremena za samostalno uvježbavanje rada na programu. U ovoj fazi korisnik uči i testira aplikaciju. Moguće je da otkrije poneku pogrešku.

Konverzija podataka

Prema planu dogovorenog dana, kada je i obuka gotova, pristupa se završnim konverzijama podataka. Konverzije su posebna grupa aktivnosti na projektu i podrazumijevaju prijenos podataka iz stare baze podataka u novu bazu podataka kako bi se korisniku olakšao posao inicijalnog unosa velike količine potrebnih podataka.

Novi aplikacijski sustav podrazumijeva i novu organizaciju baze podataka s uglavnom strogim pravilima referencijalnog integriteta i dodatnim ograničenjima. Stara aplikacija, bilo ručna ili implementirana na starom hardveru i softveru ima

uglavnom manji stupanj ograničenja u bazi i manji fond podataka te zahtijeva prepisivanje podataka iz starog u novi sustav s obogaćivanjem.

Dio tih podataka su tablice šifarskog sustava, a dio prometni podaci o proteklom razdoblju. Ponekad povijesni prometni podaci ne trebaju u novoj bazi, već samo početna stanja te je posao konverzije podataka olakšan. Šifarnici se uglavnom lakše konvertiraju.

Za konverziju podataka potrebno je poznavati strukture podataka obje aplikacije. Na osnovi toga napravi se plan koje će se tablice ručno upisati (obično manji šifarnici do 100 redaka) a koje prepisati u nove tablice. Programi za konverzije nastaju i testiraju se u fazi proizvodnje po pravilima tog procesa i spremaju se za upotrebu sve do trenutka prelaska na novi sustav. Stari podaci se čuvaju i arhiviraju. Do njih korisnik može doći u slučaju potrebe

Konverzija se sastoji iz tri osnovne aktivnosti i to:

- pisanje programa za konverziju (započinje paralelno s izradom programskog proizvoda),
- testiranje konverzije i
- izvođenje završne konverzije na produkcijskom okruženju.

Programi napisani za konverziju koriste se jednokratno, odnosno nakon konverzije arhiviraju se za slučaj potrebe slične konverzije u budućnosti. Oni se uglavnom više ne koriste.

Završno testiranje

Nakon konverzije potrebno je testirati sadržaj produkcijske baze podataka te spoj gotovog programskog proizvoda i pune baze relevantnih podataka. Ovo testiranje je kratko i traje od nekoliko sati do nekoliko dana (vikendom). To završno testiranje izvode programeri i tester. Ako je testiranje uspješno, može započeti korištenje gotovog programskog proizvoda.

Početak primjene nove aplikacije

Prvi slijedeći radni dan započinje primjena nove aplikacije, novog programskog proizvoda na novoj bazi podataka. Poželjno je da sustav funkcionira uredno. Moguće su manje greške i ispravci tijekom korištenja novog sustava, bilo u bazi podataka bilo u programskom proizvodu. Sustav se zaustavi na kratko razdoblje, ako je to moguće, i zamijeni potreban programski modul ili ponovi ispravna konverzija.

Ako se ipak pojavi velika greška u radu sustava, koja se ne može brzo otkloniti, i bude ugrožena funkcionalnost poslovnog sustava, potrebno je zaustaviti novi sustav i vratiti se na stari sustav.

Po ispravljanju greške i ponovnom testiranju pristupa se uvođenju. Uvođenje takvog sustava može biti odgođeno i više mjeseci. Pogotovo ako se stari sustav održava i ako se na njega postavi velik broj hitnih zahtjeva potrebnih menadžmentu za vođenje poslovanja. Tada se sve izmjene paralelno izvode na starom i novom sustavu.

2.3 Uspostava novog sustava

Ako je uvođenje proteklo ispravno, novi sustav je uveden. Primjenjuju se novi programski proizvodi, dnevno se puni baza podataka novim podacima, i može se pristupiti potpisivanju primopredajnog zapisnika. Obično se u ugovorima dogovori rok kada će se potpisati primopredajni zapisnik. To može biti nakon mjesec dana (ili manje) korištenja novog sustava.

Ponekad, ako je plaćanje vezano uz potpis primopredajnog zapisnika, i ako je iskustvo s korisnikom na ranijim projektima negativno, takvo da on ne želi

potpisati zapisnik više mjeseci nakon uvođenja, može se zahtijevati potpisivanje zapisnika o primopredaji nakon završnog testiranja od strane korisnika u fazi proizvodnje. U tom slučaju ne pristupamo uvođenju dok korisnik (kupac) ne prihvati potpisati da je suglasan s novim programskim proizvodom i njegovim funkcionalnostima jer se u to sam uvjerio testiranjem.

U tom slučaju je povećana njegova odgovornost za moguću grešku kod proizvodnje i on pažljivije testira programe. Tada imamo manje problema s uvođenjem.

Uspostava novog sustava i uklanjanje starog je promjena odnosno preobrazba sustava. Moguća su tri načina promjene starog u novi sustav (Merle, 2003) i to:

- direktna,
- paralelna i
- fazna (postupna).

Direktna promjena je slučaj kada se dio starog sustava zaustavlja i prepušta funkcioniranju novom sustavu.

Karakteristike ovog pristupa jesu:

- 12 najjeftinija je od tri alternative jer nema dva odvojena sustava koja rade u isto vrijeme
- 13 najmanje zbunjuje korisnike jer ima jedan skup ulaza i izlaza
- 14 traži više obveza od korisnika jer nema starog sustava na koji se može vratiti ako novi sustav padne
- 15 potpuno je dizajniran i testiran jer su se analitičari i programeri potrudili osigurati da novi sustav radi dobro jer nema starog sustava na koji se može vratiti
- 16 riskantan je jer ako novi sustav padne, ne postoji više stari sustav.

Paralelna promjena znači korištenje starih i novih aplikacija sustava kroz neko razdoblje dok se «svi» ne priviknu na novi sustav i uvjere se da on dobro radi.

Karakteristike ovog pristupa jesu:

- a) najsigurnija je od tri alternative, osigurava povratak na staro ako novi sustav ne radi dobro
- b) najskuplja je jer dva aplikacijska sustava moraju raditi u isto vrijeme
- c) najmanje navodi korisnike da se posvete novom sustavu jer stari sustav i dalje postoji
- d) najlakše ga se pregleda jer se rezultati novog sustava mogu usporediti transakciju po transakciju sa starim sustavom.

Postupna promjena je mješavina dvije prethodne alternative. Stari sustav se isključuje, a novi uključuje korak po korak. Postoje dvije osnovne varijacije ove alternative i njihova mješavina:

- **Pilot projekt na jednoj lokaciji.** Aplikacijski sustav se u cijelosti uvodi na jednoj izdvojenoj lokaciji (npr. jedno manje skladište). Iskustva s te lokacije služe za doradu aplikacijskog sustava i uvođenje svih ostalih lokacija. Rizik je umjeren, a uvođenje olakšano.
- **Fazno uvođenje po funkcijama.** Moduli novog aplikacijskog sustava pokrivaju funkcionalnosti poslovnog sustava te se planira uvođenje jedne po jedne grupe funkcija odnosno grupe programskih modula. Npr., u trgovačkom društvu se može uvesti ekonomat s narudžbom dobavljaču, a kad taj modul profunkcionira onda se uvodi komercijala koja ima proširenje za npr. narudžbu kupcu i dr. Ponekad se neke funkcije koriste tek na koncu obračunskog razdoblja te je fazno uvođenje funkcija prirodno.

Ponekad se funkcije ne mogu uvoditi pojedinačno, pa se uvodi niz funkcionalnosti odjednom.

- **Pilot projekt na jednoj lokaciji jedne funkcije.** Ponekad je moguće na jednoj lokaciji uvesti jednu funkciju, a nakon njenog uvođenja na istoj toj lokaciji uvoditi drugu funkciju. Nakon testiranja mogu se dalje širiti u sustav u paketu ili pojedinačno.

Za vrijeme procesa promjene starog novim sustavom mogu se dogoditi mnogi problemi. Ti problemi mogu negativno utjecati na velik broj korisnika i kupaca. Stoga je uvođenje važan i kritičan dio projekta.

3. FAZA: PRIMJENA I ODRŽAVANJE SUSTAVA

Jednom uspostavljen novi sustav prelazi u fazu primjene i održavanja. Primjena programskih proizvoda je dnevno korištenje modula u obavljanju poslovanja.

Održavanje znači izvođenje aktivnosti bilo koje ranije faze radi prilagodbe programskih proizvoda potrebama korisnika. Uzroci održavanja jesu: uvođenje novih poslovnih procesa, izmjene postojećih poslovnih procesa i otklanjanje grešaka

U toj fazi su definirane procedure postavljanja zahtjeva za izmjenama od strane korisnika, odobravanje zahtjeva od nadležnog menadžera, projektiranje i proizvodnja novih programskih modula te testiranje i uvođenje novih funkcionalnosti.

3.1 Podešavanje novog aplikacijskog sustava

Novi aplikacijski sustav je instaliran i obavlja dnevne operacije za korisnika. Prvi mjeseci rada sustava su naporni za razvojni tim iz niza razloga: greške, neznanje i neiskustvo u korištenju od strane korisnika, participiranje u pomoći korisnicima i obuci djelatnika u Help desk-u. U tom razdoblju je potrebno pomoći korisnicima u prvim koracima korištenja i istraživanju uzroka krivih izvještaja i pogrešnih unosa u bazu podataka.

Procjenjuje se efikasnost novog sustava i poboljšava njegova funkcionalnost. Odgovornost za funkcioniranje novog sustava se ne prebacuje na organizacijske dijelove i korisnike, već i dalje o tome aktivno brine razvojni tim.

Korisnici očekuju savršeno dizajniran i izgrađen aplikacijski sustav kao što je to slučaj sa proizvodima koje kupuju u trgovinama. Projektiranje i izgradnja aplikacijskog sustava je složena, a nije serijska proizvodnja te ima nedostataka. Projektanti i programeri trebaju upozoriti korisnike da očekuju manje greške. Treba reći da ih ima, ali da sada nitko ne zna gdje su. Greške se ne mogu naći testiranjem i brzo će se ukloniti kad ih netko otkrije. Ako korisnik očekuje greške, neće izgubiti povjerenje u sustav kad se te greške i dogode.

U slučaju da dobavljač softvera prodaje gotov proizvod koji je implementiran kod više korisnika tada se ne očekuje pojavljivanje grešaka već postavljanje zahtjeva koje do sada nisu tražili drugi korisnici.

3.2 Izvješće o procjeni novog projekta

Nakon razdoblja od oko 6 mjeseci pristupa se procjeni funkcioniranja novog aplikacijskog sustava u odnosu na plan projekta i izrada izvješća o uspjehu projekta. Izvješće sadrži: usporedbu planiranih i ostvarenih funkcionalnosti, nepostignute ciljeve, neimplementirane funkcionalnosti, stupanj organizacijske pokrivenosti, plan aktivnosti potreban da se dostigne puna funkcionalnost

sustava s rokovima i datumom ponovnog ocjenjivanja stanja projekta.

3.3 raspodjela odgovornosti korisnika i programera

Po gotovosti projekta raspušta se projektni tim. Definiraju se programeri koji će aplikacijski sustav održavati i korisnici koji će biti ovlašteni i odgovorni za upravljanje aplikacijskim sustavom. To je svojevrsan ugovor, odnosno dogovor o raspodjeli posla i odgovornosti između korisnika i programera na održavanju sustava. U tom ugovoru definiraju se kanali komunikacije, brzina ispravljanja pogrešaka, koja se poboljšanja mogu očekivati, dokumentacija za komunikaciju, oblik zahtjeva za izmjenama pa sve do plana kada će se inicirati novi projekt i kreirati nova aplikacija koja će zamijeniti i nadograditi upravo uvedenu.

3.4 Korištenje aplikacijskog sustava

Novi aplikacijski sustav svakodnevno radi. Programske alate pokreću korisnici za obavljanje poslovanja.

Moguće je da se pojave greške u radu aplikacije zbog: skrivene greške u programu, pogreške kod konverzije podataka, pogrešnog unosa od strane korisnika koji se ne može kontrolirati, pogreške kod unosa koje se mogu dodatno spriječiti i to: unošenjem ograničenja u bazu podataka ili dodavanjem kontrola na ekrane za unos podataka.

Ponekad će korisnici uočiti moguća poboljšanja rada i predložiti doradu programa.

Moguće je da se promijene interna pravila ili eksterni uvjeti koji traže izmjenu programa.

Neovisno o uzroku potrebno je izraditi zahtjev za izmjenu programskog proizvoda.

3.5 postavljanja zahtjeva za izmjenama od strane korisnika

Svi zahtjevi za promjenama u aplikaciji idu programerskom timu za održavanje programskog proizvoda. Zahtjev za izmjenu sadrži niz podataka: tko traži izmjenu, kome se šalje zahtjev, tko ga odobrava, kada je načinjen, do kada posao treba biti gotov, opis onoga što treba mijenjati.

Zahtjev ide na odobrenje osobi koja je odgovorna za angažiranje tima za održavanje. U slučaju internih resursa na održavanju treba planirati raspored rada programera, a ako održavanje izvodi vanjska tvrtka treba tražiti ponudu s rokom i cijenom za izvođenje posla. Odobren zahtjev isporučuje se timu za održavanje.

Analitičari i projektanti koji su kreirali i uspostavili IS su obično prebačeni na novi projekt. Oni su zbog toga zauzeti, nisu raspoloživi za održavanje. Ako programeri dobiju složene zahtjeve koje ne mogu izvesti bez projektanata onda se i projektanti uključuju dok se ne definira model promjena.

4. ZAKLJUČAK

Implementacija se u osnovi sastoji od tri faze i to: proizvodnje programskog proizvoda, uvođenja novog IS i održavanja IS. Ponekad se pod implementacijom podrazumijeva samo faza uvođenja i održavanja

U ovom radu je detaljnije opisana faza uvođenja i održavanja.

Sve aktivnosti bilo koje faze, projektiranja, proizvodnje, uvođenja i održavanja izvode se tako da se u bilo kojem trenutku po potrebi može vratiti na ranije aktivnost. Isto

tako se na svim projektima i ne moraju izvoditi sve aktivnosti, ovisno o veličini projekta i gotovosti modula integralnog informacijskog sustava.

Plan rada na projektu može se napraviti tako da se tablice s popisom aktivnosti koriste kao metaaktivnosti te se one dorađuju i razvrstavaju u vremenu. Svaki pojedini projekt ima svoj zaseban plan.

Izmjene tijekom projekta razvoja su uobičajene i to ih je više što projekt duže traje. Uzroci izmjena su: zakoni, potrebe korisnika, ubrzanje procedura i dr. Izmjene utječu na izmjene plana, a potom i aktivnosti razvojnog tima.

Uvođenje je napor zamjene stare novom aplikacijom. Važan dio uvođenja su završne konverzije s ispravljanjem pogrešnih podataka u BP.

Svaki novi softver će u budućnosti postati «stari». Sigurno će doći dan kada će sada novi aplikacijski sustav postati nedovoljan i zastario. Razlozi za to mogu biti različiti. Tada će menadžment (pokrovitelj projekta, investitor, vlasnik poslovnog procesa) inicirati novi projekt za izgradnju zamjenskog aplikacijskog sustava. Pokrenut će se novi razvojni ciklus, formirati tim, otvoriti projekt, projektirati, proizvoditi i uvoditi novi aplikacijski sustav. Čini se da ovoj priči nema kraja.

Literatura:

- 1 Carey, J., Understanding resistance to system change: An empirical study. In Human factors in management information systems. J.M.Carey, Ed., pp. 195-206. Norwood, NJ:Ablex, 1988.
- 2 Merle, P., Martin, Systems Implementation, Encyclopedia of Information Systems, Volume 4, H. Bidgoli, Ed., pp. 379 – 389, Academic Press, 2003.
- 3 Pavlič, M., Ivašić, M., Zamlić, I., Methodology MIRIS, Proceedings of the eight Electrotechnical and Computer Science Conference ERK'99", 23.09. - 25.09. 1999., Slovenian Section IEEE, Portorož, Slovenija, 1999. str. 309-312.
- 4 Sartipi K., Software Architecture Recovery, Dept. Of Computer Science, Waterloo, Ontario, 1999.
- 5 Pavlič, M., Razvoj informacijskih sustava - projektiranje, praktična iskustva, metodologija, Znak, Zagreb, 1996
- 6 Sommerville, I.: Software Engineering, Fifth Edition, Addison-Wesley, Harlow, 1995., p. 742

Podaci o autorima:

prof. dr.sc. Mile Pavlič

Ris d.o.o.

Pilepčić 10

51215 Kastav

tel: 051/687-500

fax: 051/687-501

e-mail: mile.pavlic@ris.hr

Prof. dr. sc. Mile Pavlič bavi se metodologijom projektiranja i razvojem informacijskih sustava. Direktor je tvrtke RIS osnovane 1993. za razvoj informacijskih sustava. Aktivno izučava metode za projektiranje informacijskih sustava. Predavač predmeta o projektiranju informacijskih sustava na Odsjeku za informatiku na Filozofskom fakultetu u Rijeci. Bio je pročelnik Odsjeka od 1995. do 1998. Aktivno sudjeluje na razvoju niza projekata u privredi (MOHV, HRT, CROATIA i dr.). Stalni predavač metoda za projektiranje IS od 1986. za projektante. Član programskog odbora za organiziranje CASE - alati i metode savjetovanja. Završio studij Fizika s matematikom na Pedagoškom fakultetu u Rijeci 1980. godine. Na poslovima analitičara/programera u ERC-u, radio od rujna 1982. godine u brodogradilištu 3. Maj u Rijeci. Prvi informacijski sustav o praćenju stanja skladišta završio 1983. godine. Radio na nizu projekata, bilo na projektiranju bilo na fizičkoj realizaciji, kao voditelj, projektant, analitičar ili programer. Održavao predavanja o: arhitekturi operativnih sustava računara, radu prevoditelja, bazi podataka, ekspertnim sustavima, DSS, modeliranju procesa i podataka, organizaciji informatike, informacijskim sustavima i dr. Radio na radnom mjestu projektanta i direktora INFO centra u RiAdria banci d.d. Rijeka. Biran za područje informacijskih znanosti u znanstvenoistraživačko zvanje znanstvenog asistenta 17. ožujka 1992. a u zvanje znanstvenog suradnika 9. studenog 1994. te u docenta 1997. godine. Objavio niz znanstvenih i stručnih radova. Magistrirao na usporedbi raznih metoda za modeliranje podataka, a doktorirao na temi izučavanja procesa praktične primjene informatičkog inženjeringa u poduzećima.

mr.sc. Patrizia Poščić

Sveučilište u Rijeci, Filozofski fakultet u Rijeci

Omladinska 14

51000 Rijeka

tel: 051 345 046

fax: 051 345 207

e-mail: patrizia@ffri.hr

mr.sc. Sanja Čandrić

Sveučilište u Rijeci, Filozofski fakultet u Rijeci

Omladinska 14

51000 Rijeka

tel: 051 345 050

fax: 051 345 207

e-mail: sanjac@ffri.hr

Perino Krneta

Ris d.o.o.

Pilepčić 10

51215 Kastav

tel: 051/687-500

fax: 051/687-501

e-mail: perino.krneta@ris.hr