

Poslovni proces upravljanja zahtjevima u informatičkoj industriji

Šteković Savić, Sanja

Professional thesis / Završni specijalistički

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:443855>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-23**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Sanja Šteković Savić

**POSLOVNI PROCES UPRAVLJANJA
ZAHTJEVIMA U INFORMATIČKOJ
INDUSTRIJI**

SPECIJALISTIČKI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Sanja Šteković Savić

Matični broj: IRODID-64/2006

Studij: Informacijski sustavi

POSLOVNI PROCES UPRAVLJANJA ZAHTJEVIMA U
INFORMATIČKOJ INDUSTRIJI

SPECIJALISTIČKI RAD

Mentor:

Prof. dr. sc. Vjeran Strahonja

Varaždin, prosinac 2019.

Sanja Šteković Savić

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom radu opisane su teorijske i praktične mogućnosti upravljanja zahtjevima prilikom razvoja programskog rješenja. Rad opisuje metode i tehnike koje je moguće koristiti prilikom razvoja programskog rješenja, te inženjerski pristup upravljanju zahtjevima. Nadalje, opisana je norma te je ukratko opisan ITIL koji se koristi kao skup najboljih praksi prilikom razvoja programskog rješenja, a samim time i upravljanja zahtjevima. Praktični dio rada prikazuje upravljanje zahtjevima u IT poduzeću koje ne koristi nikakvo programsko rješenja koje se koristi kao podrška procesu. Kao poboljšanje procesa uvedena je jedinstvena baza podataka koja se uvodi u proces te se s time dobiva na transparentnosti i vidljivosti u kojem stanju su zahtjevi. U sklopu rada provedena je preliminarna anketa kojoj je cilj utvrditi na koji način današnje IT tvrtke upravljaju zahtjevima te koje su prednosti i nedostaci upravljanja zahtjevima.

Ključne riječi: poslovni proces, modeliranje poslovnih procesa, upravljanje zahtjevima, razvoj programskog rješenja, alati za upravljanje zahtjevima

PODACI O SPECIJALISTIČKOM ZAVRŠNOM RADU

I. AUTOR

Ime i prezime	Sanja Šteković Savić
Datum i mjesto rođenja	24.1.1971., Zagreb
Naziv fakulteta i datum diplomiranja	Prirodoslovno-matematički fakultet, 29.4.1997.
Sadašnje zaposlene	Voditelj projektnog ureda

II. ZAVRŠNI RAD

Naslov	Poslovni proces upravljanja zahtjevima u informatičkoj industriji
Broj stranica, slika, tabela, priloga, bibliografskih podataka	Broj stranica 95, broj slika 43, broj bibliografskih podataka 24.
Znanstveno područje, smjer i disciplina iz koje je postignut akademski stupanj	Informacijske i komunikacijske znanosti
Mentor i voditelj rada	Prof.dr.sc. Vjeran Strahonja
Fakultet na kojem je rad obranjen	FOI
Oznaka i redni broj rada	IROID

III. OCJENA I OBRANA

Datum prihvaćanja teme od Fakultetskog vijeća	18.11.2008.
Datum predaje rada	20.12.2019.
Datum sjednice FV-a na kojoj je prihvaćena pozitivna ocjena rada	02.04.2020.
Sastav Povjerenstva koje je rad ocijenilo	Izv. prof. dr. sc. Katarina Tomčić-Pupek, Prof. dr. sc. Vjeran Strahonja, Prof. dr. sc. Melita Kozina
Datum obrane	20.4.2020.
Sastav Povjerenstva pred kojim je rad obranjen	Izv. prof. dr. sc. Katarina Tomčić-Pupek, Prof. dr. sc. Vjeran Strahonja, Prof. dr. sc. Melita Kozina
Datum promocije	

Sadržaj

Sadržaj	iii
1. Uvod	5
2. Metode i tehnike rada.....	6
3. Poslovni proces	7
3.1. Ciljevi modeliranja.....	7
3.2. Opis procesa modeliranja	10
3.3. Postupci i norme za modeliranje	20
4. Upravljanje zahtjevima	21
4.1. Uspostava projekta i početak razvoja	22
4.2. Faze u procesu upravljanja zahtjevima.....	22
4.3. Zahtjevi u iterativnom modelu	26
4.4. Zahtjevi za promjenama.....	31
5. Tehnike upravljanja zahtjevima	33
5.1. Alati za upravljanje zahtjevima.....	33
5.2. Sposobnost opisivanja i praćenja zahtjeva	34
5.3. Povezivanje zahtjeva i UML metoda.....	34
5.4. Alati za upravljanje zahtjevima.....	35
5.4.1. Git	36
5.4.2. Jira Atlassian	40
5.4.3. Trello.....	42
6. Inženjerstvo zahtjeva.....	44
6.1. Osnove zahtjeva programskog proizvoda.....	44
6.2. Programski proizvod i proces.....	44
6.3. Funkcionalni i nefunkcionalni zahtjevi.....	45
6.4. Neposredni zahtjevi	45
6.5. Kvantitativni zahtjevi	45
6.6. Sistemski i programski zahtjevi.....	45
6.7. Proces inženjerstva zahtjeva	46
6.7.1. Modeli procesa	46
6.7.2. Uloge u procesu	46
6.7.3. Kvaliteta i unaprjeđenje procesa	47
6.8. Prikupljanje zahtjeva	47
6.8.1. Izvori zahtjeva.....	47

6.8.2. Tehnike prikupljanja zahtjeva	48
6.9. Analiza zahtjeva	49
6.10. Konceptualno modeliranje	50
6.10.1. Arhitekturno dizajniranje te alokacije zahtjeva	50
6.10.2. Pregovaranje o zahtjevima	50
6.11. Specifikacija zahtjeva	51
6.11.1. Definicija sustava	51
6.11.2. Specifikacija zahtjeva sustava	51
6.11.3. Specifikacija programskog proizvoda	52
6.12. Validacija zahtjeva	52
6.12.1. Recenzije specifikacije	52
6.12.2. Prototipiranje	52
6.12.3. Test prihvatanja	53
7. Standardi za upravljanje zahtjevima	54
7.1. ISO 20000	54
7.2. FitSM	56
8. ITIL	57
8.1. Komponente ITIL-a	57
8.1.1. Strategija usluge	58
8.1.2. Dizajn usluge	59
8.1.3. Tranzicija usluge	59
8.1.4. Pružanje usluge	60
8.1.5. Kontinuirano poboljšanje usluge	61
9. Proces „Upravljanje zahtjevima korisnika“	62
9.1. Opis procesa „Upravljanje zahtjevima korisnika AS IS“	62
9.2. Tablica za proces „Upravljanje zahtjevima AS IS“	64
9.3. Model procesa „Upravljanje zahtjevima korisnika AS IS“	65
9.4. Opis detaljnog procesa „Zaprimiti zahtjeve korisnika“	66
9.5. Tablični prikaz detaljnog procesa „Zaprimiti zahtjeve korisnika“	67
9.6. Model detaljnog procesa „Zaprimiti zahtjeve korisnika“	68
9.7. Poslovni i organizacijski razlozi za poboljšanjem procesa	69
9.8. Model procesa „Upravljanje zahtjevima To BE“	70
10. Anketa i analiza rezultata	71
11. Zaključak	82
Popis literature	83
Popis slika	85

1. Uvod

Upravljanje zahtjevima, u poduzećima koja razvijaju aplikativna rješenja, je specifičan proces u kojem treba zadovoljiti korisnika i isporučiti mu traženo aplikativno rješenje. Praksa pokazuje da se zahtjevi mijenjaju usred projekta, ukoliko se upravljanje zahtjevima ne odvija kvalitetno i kontrolirano dovodi se u pitanje uspješnost projekta. U svakodnevnom radu sa zahtjevima korisnika primjećujem poteškoće u komunikaciji osoba u raznim rolama koje rade na obradi zahtjeva te čestim izmjenama zahtjeva u podmakloj fazi projekta. Zahtjevi koji dolaze od korisnika moraju biti obrađeni s više aspekata kako bi se kvalitetno izvršili. Upravo ta potreba, uključivanja više aktivnosti i osoba raznih vještina, ukazuje na potrebu dobrog upravljanja tim procesom. Ovim radom želim opisati i modelirati poslovni proces upravljanja zahtjevima, te ukazati na potrebu praćenja zahtjeva u raznim fazama obrade. Postavljanjem poslovnog modela moći će se bolje pratiti zahtjev, te će se lakše predvidjeti i planirati svaka promjena. Na taj način uštedjet će se vrijeme i resursi, te će se projekti završavati unutar zadanih rokova. Promjene zahtjeva su česta i prirodna pojava, pravilno upravljanje zahtjevima nameće se kao ključno za uspjeh projekta.

Poduzeća kojima je primarna djelatnost razvoj aplikativnih rješenja imaju naglasak na razvoju poslovnih aplikacija. Uočena je ovisnost većine poslovnih procesa od strane informacijske i komunikacijske tehnološke infrastrukture, kao i kvantitete, kvalitete i dostupnosti informacija koje takva infrastruktura osigurava i podržava. Iskustva današnjih IT menadžera ukazuju da je potrebna suradnja s poslovnim svijetom kako bi se kreirale nove poslovne mogućnosti. Najčešći poticaji za novi zahtjev je potreba za unapređenjem postojećeg procesa te potreba za informatiziranjem poslovnih procesa.

Ideja je ovim radom opisati i modelirati upravljanje zahtjevima kao poslovni proces. Za uspješno upravljanje zahtjevima mora se omogućiti da se u svakom trenutku može pregledati zahtjev te saznati tko i kada ga je zatražio, u kojem je statusu, zatim koji je prioritet zahtjeva i rok za izvršenjem. Modeliranje nam je potrebno kao bi sagledali životni ciklus svakog zahtjeva. Upravljanje zahtjevima sastoji se od više podprocesa: analiza zahtjeva, specifikacija, implementacija i testiranje samog rješenja. U svakom trenutku realno je očekivati promjenu zahtjeva. Kvalitetno upravljanje promjenama moguće je samo u dobro postavljenom poslovnom procesu. Ovim radom opisat će se poslovni proces upravljanja zahtjevima, što će omogućiti bolji pregled nad fazama i aktivnostima pri realizaciji zahtjeva od korisnika, što će rezultirati uspješnijim izvršenjem projekta i zadovoljstvom korisnika.

2. Metode i tehnike rada

U rješavanju opisanog problema koristit ću se kombinacijom slijedećih metoda:

- Metoda modeliranja – koristiti će se za pojašnjenje i bolji pregled poslovnog procesa upravljanja zahtjevima.
- Metoda analize – omogućuje prepoznavanje mjesta u kojim se proces može poboljšati. Obzirom da analiziranih procesa može biti više potrebno je na početku napraviti procjenu važnosti određenih procesa. Važniji procesi su oni koji u izvođenju troše više resursa i vremena (promatra se broj ponavljanja u jedinici vremena i utrošak resursa pri svakom ponavljanju).
- Statističke metode (usporedba srednjih vrijednosti, usporedba varijanci) – u ovom radu će pomoći pri donošenju zaključaka. Preciznost matematičkih operacija kod dobivanja rezultata ovim metodama osigurava objektivnost primjene.
- Metoda simulacije - omogućuje konkretniju usporedbu različitih prijedloga u dinamičkim uvjetima i provođenjima „što ako?“ analize. U ovom radu prikazat ću rezultate dobivene simulacijom postojećeg i predloženog procesa, te na osnovu tih rezultata zaključiti o uspješnosti predloženog procesa.
- Heuristički pristup - predstavlja korištenje iskustva, intuicije i vlastite procjene prilikom rješavanja nekog problema.
- Metoda studiji slučaja - ispituje međuovisnosti svih varijabli pojedinog procesa kako bi postigli ukupno razumijevanje tog procesa.
- Induktivna metoda i deduktivna metoda – omogućuju da na temelju promatrane pojave možemo zaključiti da će slične pojave koje se nisu ispitivale imati ista svojstva kao i pojave koje su se desile. Deduktivna metoda polazi od postavke da ono što vrijedi uopće, vrijedi i za jedan poseban slučaj .

U radu će se koristiti slijedeća informatička podrška:

- Bizagi Modeler- BPM alat za izradu modela procesa. Ovim alatom dokumentirat će se poslovni proces, pojednostaviti analiza te lakše pronaći mogućnost za unaprjeđenjem.
- MS Excel 2013 - tablični kalkulator: procjena troškova,
- MS Visio 2013 - alat za crtanje dijagrama, koristi će se za crtanje modela podataka

3. Poslovni proces

Poslovni proces je povezani skup aktivnosti i odluka , koji se izvodi na vanjski poticaj radi ostvarenja nekog mjerljivog cilja organizacije, troši vrijeme i pretvara ulazne resurse u specifične proizvode ili usluge od značaja za kupca ili korisnika. (2)

Organizacijski poslovni proces jasno opisuje rad koji izvode svi resursi uključeni u kreiranje izlaza, mjerljivih vrijednosti za njegove klijente i druge sudionike. Poslovni proces postoji u kontekstu nekog organizacijskog posla. Započinje s nekim događajem koji ga inicira i završava s namjeravanim rezultatom(ciljem) u vezi s kojim je donesena odluka. Mora se nedvosmisleno imenovati i koristiti konzistentno - biti u cijelosti razumljiv za njegove sudionike. (3)

Modeliranje poslovnih procesa je izrada dijagrama koji pokazuju aktivnosti slijedom kako se događaju kod postojećih (as-is) i budućih (to-be) poslovnih procesa organizacije radi analize i poboljšanja izvođenja postojećih procesa. Za izradu modela najčešće su zaduženi poslovni analitičari i vlasnici procesa koji žele unaprijediti efikasnost i kvalitetu procesa te kvalitetu rezultata procesa. U općenitom smislu unaprjeđenje ne mora uključivati podršku IT-a, ali je u praksi njeno korištenje najčešći pokretač aktivnosti modeliranja. U ovom poglavlju želim opisati:

- razloge odnosno ciljeve modeliranja,
- uloge koje sudjeluju u modeliranju poslovnih procesa,
- artefakte koji se koriste pri modeliranju bilo kao izvor informacija ili su rezultat aktivnosti modeliranja,
- tehnike koje tim može koristiti za prikupljanje i sistematizaciju informacija,
- alate i njihove uloge i na kraju
- korake složenog proces poslovnog modeliranja kao dio razvojnog procesa, ali i životnog ciklusa procesa koji je predmet interesa tima.

3.1. Ciljevi modeliranja

Modelirajući postizemo jasnu sliku radnih koraka (aktivnosti) i tokova informacija kojima su procesi povezani.(4)

Ovisno o poslovnoj situaciji razlozi za modeliranje poslovnih procesa, a time i ciljevi mogu biti različiti. Kao tipične ciljevi poslovnog modeliranja moguće je izdvojiti izrada dijagrama postojećeg poslovnog procesa, jednostavnija analiza i pronalaženje mogućnosti za

unaprjeđenje, automatiziranje procesa radi povećanja efikasnosti, analiza različitih prijedloga za poboljšanjem procesa.

Također, ovisno o opsegu projekta na kojem se modeliranje provodi, potrebno je postići jedan ili više navedenih ciljeva.

Dokumentiranje postojećih poslovnih procesa osnovni je korak prema njihovom kvalitetnom upravljanju. Najčešće su procesi organizacije dokumentirani i sistematizirani, ali ne na način pogodan za daljnju analizu, optimizaciju i automatizaciju odnosno kvalitetno upravljanje. Također često postoji odmak između dokumentiranog stanja i načina na koji se procesi odvijaju u praksi. Postoje situacije u kojima s vremenom u više okolina u kojima se proces izvodi nastanu različite varijante istog procesa. Osim standardizacije i sistematizacije dokumentiranjem, modeliranje će u tim slučajevima olakšati njihovo prepoznavanje i usklađivanje.

Izradom modela procesa dolazi se do slijedećih novih vrijednosti:

- Poslovni procesi dokumentiraju se na standardan i sistematski način;
- Organizacija i njeni članovi se „ponovno upoznaju“ s vlastitim poslovnim procesima;
- Prepoznaju se različite varijante istog procesa (postojanje kojih ne mora značiti loše stanje);
- Stvara se podloga za usporedbu, analizu i unaprjeđenje poslovnih procesa.
- Procjenjuje se učinak novog načina izvođenja procesa (simulacijom na modelu)

Samo modeliranje potiče sve uključene na sistematsko promišljanje te se često već tijekom modeliranja, ili kasnije jednostavnim uvidom u gotove modele, mogu prepoznati prilike za unaprjeđenja koja je u mnogim situacijama moguće implementirati u kratkom vremenu i uz mali trošak. Daljnjom analizom, moguće je prepoznati kompleksnija unaprjeđenja čija implementacija će možda zahtijevati i pokretanje projekata. Vrlo korisnim se može pokazati postojanje više varijanti istog procesa jer postojanje svake razlike ima svoje razloge, a njihovo pronalaženja dovodi do boljeg poznavanja procesa i okoline u kojoj se izvodi. Na temelju takvih informacija može se donijeti odluka o implementaciji nekoliko varijanti zavisno o okolinama ili vremenu odvijanja (npr. razlika između većih i manjih poslovnica ili odvijanja procesa u radno i neradno vrijeme). Jednako tako, u nekim će se slučajevima moći prepoznati i ukloniti razlike u okolinama u kojima se proces odvija i tako eliminirati nepotrebne varijante.

Koristi od ovakve analize su novi prijedlozi za jednostavnijim i kompleksnijim unaprjeđenjima, te stjecanje novih znanja o procesima i okolini.

Potpuna automatizacija cilj je svakog vlasnika procesa. Iako većinu trenutno nije moguće u potpunosti automatizirati, razvojem tehnologija i njihovim širenjem moguće je automatizirati sve više podprocesa. Preduvjet za automatizaciju je dobro poznavanje odvijanja procesa i svih vezanih pravila od strane više osoba različitih specijalnosti i nivoa predznanja. Rad na modelu uvelike olakšava komunikaciju i analizu odnosno pronalaženje prilika i načina za automatizaciju. Osim toga, kvalitetni i detaljni modeli kakvi se mogu raditi s alatom kao što je Bizagi Modeler mogu se iskoristiti kao osnova za implementaciju servisa (npr. razvoj servisa) odnosno poslovnih procesa kao i njihovu doradu. U tom smislu modeliranje poslovnih procesa je jedan od koraka u razvoju vođenom modelima.

Prednosti modeliranja radi automatizacije u odnosu na klasične komunikacijske tehnike su:

Preciznije i standardizirano specificiranje olakšava komunikaciju;

Podaci iz modela automatski se prebacuju u razvojne alate pa se smanjuje ručni rad i mogućnost pogreške;

Ubrzava se razvojni proces kao i ciklus unapređenja.

Često se u organizaciji pojavi više prijedloga za poboljšanjem postojećeg procesa ili prijedloga novih procesa. Izrada modela prijedloga korak je kojim se može olakšati njihova usporedba. Skup informacija dobivenih korištenjem modela bitno je opsežniji i detaljniji od onog dobivenog klasičnim opisom. Također, moguće je doći do tzv. referentnog modela procesa npr. modela u koji su uključena najbolja iskustva pa je za usporedbu procese (vlastiti i referentni) dobro svesti na zajednički nazivnik odnosno modelirati na jednak način. Provođenje simulacija, pomaže uspoređivanju procesa u dinamičkim uvjetima.

Često se unutar organizacije ili od strane vanjskih tijela (partnera, regulatornih tijela, radi dobivanja određenih certifikata i sl.) pojavljuju zahtjevi za standardnim dokumentiranjem poslovnih procesa. Pod tim standardima, između ostalih zahtjeva, traži se postojanje modela te i to može biti razlog za modeliranje.

Model je, u općenitom smislu, pojednostavljeni prikaz kompleksnog sustava načinjen s ciljem jednostavnije analize.⁽²⁴⁾ S obzirom na atribute poslovnih procesa, kad je riječ o njihovim modelima, govorimo o sljedećim dimenzijama:

Proces – slikoviti prikaz modela procesa

Slijedne veze - ukazuju na redoslijed izvođenja aktivnosti

Resursi – definicija tipova resursa i instanci koje su pridružene modelu

Informacije – prikaz podataka koji se koriste unutar poslovnog procesa

Organizacije – definicija i struktura organizacijskog ustrojstva i pridruženih resursa

Poslovna mjerila – definicija ključnih pokazatelja uspješnosti (KPI) i metrike u svrhu mjerenja sustava i okidača karakteristika procesa.

Dakle, model uključuje slijed aktivnosti s relevantnim informacijama koje su dodane; informacije koje su zaprimljene, koje se koriste i koje se stvaraju, informacije o tome tko što radi i gdje (resursi, role, rasporedi), informacije koje označavaju organiziranost tvrtke (organizacijske jedinice, lokacije, strukture), te informacije za grupiranje povezanih atributa za analizu.

3.2. Opis procesa modeliranja

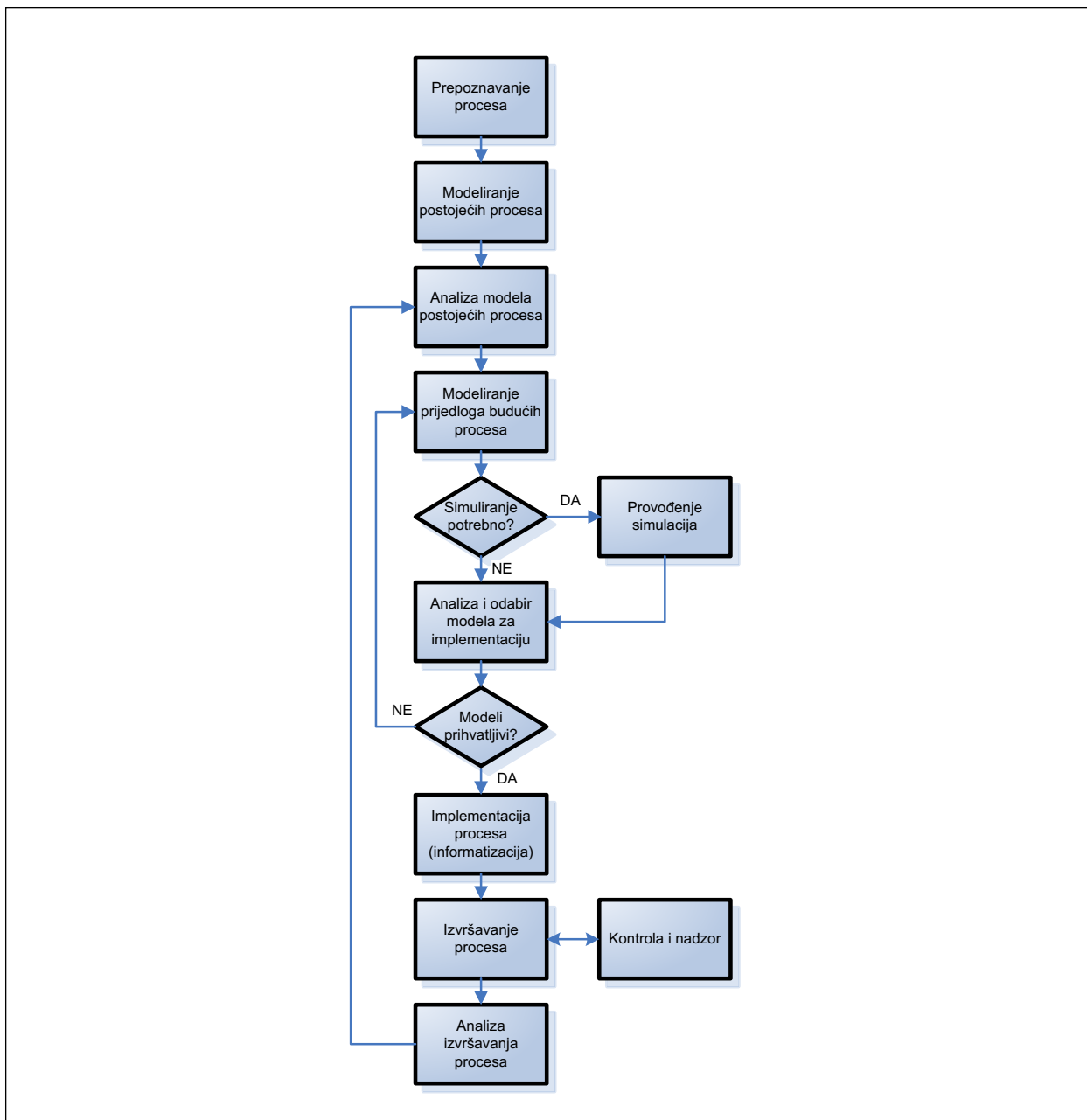
Aktivnosti iz procesa modeliranja poslovnih procesa dio su skupa aktivnosti kroz koje poslovni proces prolazi kroz svoj životni ciklus. Da bi slika bila potpuna u ovom je poglavlju opisan kompletan životni ciklus procesa modeliranja, naravno s naglaskom na aktivnosti modeliranja.

U praksi se, ovisno o situaciji, s modeliranjem kreće iz dvije polazišne točke.

Potrebno je poboljšati postojeći poslovni proces – kada poslovni proces postoji ali nije adekvatno dokumentiran kreće se od njegovog modeliranja, a model budućeg procesa nastaje nakon analize postojećeg.

Potrebno je osmisлити potpuno novi poslovni proces – polazi se od modeliranja prijedloga poslovnog procesa, eventualno na temelju dostupnog referentnog modela.

U oba je slučaja cilj na temelju poslovnih znanja i dostupnih informacija doći do kvalitetnog modela procesa za implementaciju na temelju kojeg se definiraju detaljne procedure, pravilnici, organizacijsko ustrojstvo i sl. te se nabavlja oprema i alati za njegovu podršku. Kada je sve spremno poslovni se proces pokreće, a zajedno s njim i odgovarajući procesi nadzora i kontrole. Rezultat tih procesa, važan za upravljanje poslovnim procesom, su nova poslovna znanja i iskustva na temelju kojih se može krenuti s poboljšanjem procesa kroz izradu novih modela. Tako započeti ciklusi ponavljaju se sve dok postoji potreba za odvijanjem poslovnog procesa. Model na slici 1. Životni ciklus procesa prikazuje tipični život procesa, a koraci su detaljnije objašnjeni u nastavku.



Slika 1 Životni ciklus procesa (Izvor: Vlastiti primjer)

Na aktivnostima modeliranja i, pogotovo, analize uključen je velik broj osoba. Svakoj se od njih, njenim uključivanjem u rad, dodjeljuje jedna ili više od uloga, trebaju sudjelovati svi kompetentni da pomognu u osmišljavanju najboljeg rješenja.

Poslovni analitičar – najvažnija uloga sa stanovišta poslovnog modeliranja. Zadužen je na temelju sakupljenih informacija održavati modele.

Vlasnik poslovnog procesa – najveći izvor poslovnih informacija. Ne treba zaboraviti i druge poznavatelje kao što su izvršitelji i korisnici odnosno njihovi predstavnici.

Softver arhitekt – predstavnik tima zaduženog za razvoj odnosno unaprjeđenje informatičkog sustava za podršku procesu. Osim te, nama najzanimljivije uloge, pri implementaciji se u rad može uključiti cijeli niz stručnjaka raznih disciplina.

Izvršitelji – osobe koje sudjeluju u izvršavanju procesa svakodnevno rade i promišljaju o njemu iz perspektive drugačije od vlasnika. Izvor su informacija o stvarnom odvijanju procesa te mogu ukazati na nelogičnosti propisanih procedura, predložiti i dati primjedbe na prijedloge za poboljšanjima. S obzirom da ih može biti velik broj u rad se uključuju njihovi predstavnici.

Korisnici (procesa) – vide proces iz potpuno drugačije perspektive, korisnici su njegovih rezultata, a s obzirom da je usmjerenost na njih jedan od ciljeva svake moderne organizacije, mogu i trebaju dati primjedbe na postojeće procese kao i prijedloge budućih. Kao i kod izvršitelja, u praksi se koristi mišljenje nekih korisnika odnosno relevantnih predstavnika.

Ostale zainteresirane strane (dionici, engl. stakeholders) – svi oni koji imaju interes u procesu koji se opisuje (5) . Najvažniji dionici su već izdvojeni kroz posebne uloge, međutim moguće je da postoje neke osobe koje ne spadaju ni u jednu skupinu (npr. oni koji su radili na informatizaciji postojećeg procesa), a imaju i mogu dati korisne informacije o procesu.

Vezano uz ključne uloge bitno je obratiti pažnju na njihove odnose ovisno o perspektivi iz koje se promatra.

Iz tog razloga je važno pri komunikaciji znati iz koje se perspektive nešto opisuje.

Cilj prvog koraka, životnog ciklusa procesa slika 1, je upoznati se s organizacijom i prepoznati poslovne procese koji će kasnije biti predmetom daljnje analize.

Upoznati se s poslovanjem organizacije – iako je poznavanje poslovnih procesa kao i pripadnost organizaciji prednost, specijalnost poslovnog analitičar je poznavanje procesa poslovne analize te je vrlo važno već na početku dobiti sliku o tome što organizacija radi na koji način posluje, kakav je njen položaj na tržištu, gdje posluje, tko su joj dobavljači, za koga radi i sl.

Upoznati se s organizacijskom strukturom – iz same strukture može se prepoznati koje su osnovne funkcije prisutne u organizaciji, koji se projekti vode ili, čak za procesno orijentirane tvrtke, koji su osnovni poslovni procesi. Također se može saznati čime se pojedine organizacijske jedinice bave kao i kakvi su njihovi međusobni odnosi.

Upoznati se s proizvodima i uslugama koje organizacija nudi bilo interno ili na tržištu – poznavanje proizvoda i usluga još je jedan važan dio slike o organizaciji, a poslovni procesi su u pravilu organizirani u smislu proizvodnje odnosno pružanja usluga.

Prepoznati procese i grupe procesa – od informacija prikupljenih kroz prethodne zadatke poslovni analitičar može prepoznati procese i njihove grupe. Top-down metodom radi

se dekompozicija procesa. Prvo se prepoznaju i popisuju krovni procesi organizacije. Nakon toga za svaki se prepoznati proces prepoznaju njihovi potprocesa i tako dok se ne dobije kompletna lista procesa. Broj nivoa nije čvrsto određen već ovisi o organizaciji i ciljevima modeliranja. Bottom-up pristup kreće od procesa niže razine koji se onda povezuju, na određeni način grupiraju, u procese više razine.

U poslovnom smislu procese se rano može podijeliti na tri vrste procesa: procese upravljanja, operativne procese i procese podrške. Proces upravljanja, kako i sam naziv govori, služe upravljanju organizacijom odnosno upravljanju ostalim procesima.

Operativni procesi su ključni za organizaciju, čine njen „core business“ te su u skladu misijom organizacije usmjereni prema klijentima. Proces podrške za zadatak imaju osiguravanje preduvjeta za nesmetano odvijanje operativnih procesa. Vrsta određenog procesa nije jednaka u različitim organizacijama već ovisi o samoj tvrtki. Za svaki proces potrebno je prepoznati vlasnika – vlasnik je osoba ili organizacijska jedinica koja se brine da se proces odvija na optimalan način. U funkcijski organiziranim organizacijama to je najčešće voditelj funkcijske jedinice u kojoj se proces odvija dok je u procesno organiziranim uloga vlasnika odvojena od izvršitelja. U svakom slučaju, vlasnik je taj koji može dati najviše informacija o procesu i koji ima najviše interesa da se proces poboljša.

U nastavku su popisane i opisane metode te potrebni zapisi kako bi se krenulo u snimku postojećeg stanja i prijedlog unapređenja.

Poslovna znanja i informacije o poslovanju – najveća vrijednost za kvalitetno upravljanje procesima. Više ili manje dokumentirana u različitoj formi i na različitim medijima.

Popis poslovnih procesa – jednostavna lista, hijerarhijska struktura ili dijagram koji popisuje poslovne procese organizacije grupirane prema poslovnim cjelinama.

Modeli poslovnih procesa – predstavljaju jedan od pogleda na poslovni proces, postojeći ili prijedlog budućeg. Uvelike olakšavaju komunikaciju, analiziranje i uspoređivanje procesa te provođenje simulacija.

Prijedlozi za unaprjeđenjem – jedan ili više dokumenata, komentara na samim modelima koji sadrže konstruktivne primjedbe relevantnih osoba na proces, njegov model ili dio procesa odnosno modela.

Poslovni ciljevi – jasni poslovni ciljevi kao i njihovi prioriteti uvelike pomažu u osmišljavanju pravih procesa.

Ključni pokazatelji uspješnosti – egzaktna mjerila na temelju kojih će se uspoređivati i odabirati verzije modela istog procesa. Koriste se i u fazi izvođenja implementiranog procesa za potrebe mjerenja njegove uspješnosti.

Referentni modeli – modeli istog ili sličnog procesa obično pribavljeni od neke druge organizacije, a koji dokumentiraju najbolje prakse. Najčešće nisu direktno primjenljivi već se moraju prilagoditi konkretnoj okolini.

Izveštaj o rezultatima simuliranja – dokumentira rezultate simulacija te pomaže kod izbora između više varijanti jednog procesa kao i pribavljanja novih prijedloga za unaprjeđenjem.

Popis modela procesa za implementaciju – na temelju niza prijedloga budućeg procesa odabire se jedan ili više njih koji će biti implementirani, a u ovom dokumentu se te odluke dokumentiraju.

Softver – rezultat implementacije (informatizacije) poslovnog procesa.

Pravilnici i procedure, alati, organizacijsko ustrojstvo i sl. – rezultati implementacije poslovnog procesa.

Proučavanje dokumentacije – najčešće unutar organizacije postoji određena količina dokumentacije vezana uz konkretno područje interesa. Iako je često pisana nesistematski, iz različitih perspektiva i za različite potrebe te sadrži više ili manje zastarjele podatke, takva dokumentacija može biti vrlo koristan izvor poslovnih informacija.

Intervju – unutar organizacije uvijek postoje ljudi koji su veliki poznavatelji poslovanja, procesa i poslovnih pravila te ih je korisno prepoznati i s njima razgovarati pojedinačno ili u manjim skupinama. Od njih se, u odnosu na dokumentaciju, mogu dobiti informacije o stvarnom odvijanju poslovnih procesa, ali isto tako mogu dati iskrivljenu sliku s obzirom da poslovanje opisuju prvenstveno iz vlastite perspektive.

Brainstorming – grupna metoda koja se upotrebljava za generiranje novih ideja unutar grupe. Nakon iznošenja problema od članova se očekuju prijedlozi rješenja. Kritiziranje i komentiranje danih prijedloga nije dozvoljeno dok se svi prijedlozi ne popišu. Nakon toga se prelazi na neorganiziranu raspravu te se analizom i ocjenjivanjem prijedloga donosi konsenzusno rješenje problema. Svakako treba voditi računa o tome da ne dođe do dominacije nekog pojedinca nad ostalima.

Delphi tehnika – iterativna tehnika kod koje se traži mišljenje grupe stručnjaka na način da svaki od njih nezavisno (u idealnom slučaju jedni ne znaju za druge) popunjavaju upitnike. Njihovi odgovori se mogu koristiti u pripremi slijedeće iteracije ili kao rezultat istraživanja. Primjenjuje se kada treba ispitati više osoba nego što ih može komunicirati licem u lice te kada se želi izbjeći dominantan utjecaj jedne osobe (6)

Benchmarking – je tehnika upravljanja kojom organizacija kroz različite poglede uspoređuje svoje procese sa sličnim procesima najčešće, ali ne i nužno, unutar iste djelatnosti da bi prepoznala i prihvatila bolja rješenja. (7) Tehnika omogućava borbu protiv stavova u smislu „Način na koji radimo je najbolji, jer oduvijek tako radimo“ (tzv. Paradigm blindness).

Prezentacije – tehnika kojom se veću grupu upućuje u neku temu, problem ili rješenje. Obično jedna osoba publici izloži temu nakon čega započinje rasprava publike.

Radionica – grupna tehnika prikupljanja i sistematiziranja informacija koja uključuje istovremeni praktični rad kao što je primjerice izrada ili detaljiziranje modela.

Revidiranje – tehnika se koristi za potvrđivanje kvalitete dokumenta ili modela. Prije objave artefakt se dostavlja stručnoj osobi, poznavatelju problematike (revizoru), da bi ga ona proučila i dala svoje mišljenje o artefaktu, primjedbe i savjete za unapređenjem. Moguće je organizirati više nivoa revidiranja ili istovremeno revidiranje od strane više revizora.

Modeliranje postojećih procesa je neophodno ukoliko procesi nisu dokumentirani, kako bi se svima uključenima predstavilo trenutno stanje. Pod postojećim procesom riječ je o procesu kakav se odvija u praksi.

Prilikom snimke stanje procesa potrebno je izraditi modele organizacijske strukture –. U pravilu je riječ o hijerarhijskom modelu koji prikazuje organizacijske jedinice, njihove međusobne odnose (nadređene i podređene), odgovarajuća radna mjesta i uloge. Modeliranje organizacijske strukture do nivoa osoba radi se rijetko, pogotovo u organizacijama s većim brojem zaposlenika. Važno je napomenuti da organizacijska struktura osim funkcionalne dimenzije često ima i druge, npr. projektna ili procesna.

Prepoznati i opisati podprocese i aktivnosti pa napraviti kostur modela procesa – u skladu s prije napravljenim popisom procesa kreće se u njihovo modeliranje. Za proces koji se modelira potrebno je odrediti podprocese i aktivnosti koji se u njemu odvijaju. Nakon toga određuju se i modeliraju njihovi međusobni odnosi, što čemu prethodi ili slijedi, što se može odvijati paralelno, postoje li kakva grananja i sl. Dobiveni model predstavlja tok procesa.

Prepoznati i opisati resurse/informacije povezane s podprocesima i aktivnostima – za odvijanje određene aktivnosti ili podprocesa potrebni su određeni resursi (npr. IT sustavi) koje treba uključiti na model. Osim toga svaka aktivnost ili podproces na temelju ulaznih informacija daje izlazne te je potrebno opisati te informacijske sadržaje koji povezuju aktivnosti i procese te i njih uključiti na model procesa.

Prepoznati organizacijske elemente povezane s podprocesima i aktivnostima i njihove odgovornosti (radi, provjerava, odobrava, propisuje) – Organizacijski elementi (organizacijske jedinice, uloge, radna mjesta, zaposlenici) koji se pojavljuju u modelu organizacijske strukture

unutar svojih odgovornosti sudjeluju u odvijanju procesa. Najčešće se veze u ovom smislu rade u skladu s nivoima, procesi višeg nivoa vežu se uz više organizacijske elemente i obrnuto. Primjerice procesi „Nabave“ u nadležnosti su sektora nabave odnosno njegovog voditelja dok se aktivnost unutar jednog od podprocesa npr. „Objava natječaja“ može vezati uz radno mjesto ili ulogu.

Prepoznati međusobne veze različitih procesa – Neke veze procesa vidljive su već kod prepoznavanja samih procesa, međutim pojavljivanje određenog podprocesa, aktivnosti ili informacije u više procesa lako može ostati nedokumentirana. Razlozi su najčešće komunikacijske prirode, timovi paralelno rade na svojim modelima i nisu dovoljno upoznati s radom i procesima drugih timova. Precizni standardi i njihovo pridržavanje, podjela posla i povremeno upoznavanje s procesima drugih timova uvelike mogu olakšati ovaj zadatak.

Nakon što su modeli postojećih procesa potvrđeni kreće se na njihovu analizu s ciljem prepoznavanja mjesta u kojima se proces može poboljšati i prikupljanja prijedloga za poboljšanjem. Iskustva izvršitelja na procesu kao i korisnika mogu biti vrlo vrijedna te ih treba, ako je ikako moguće, uključiti u analizu. Kako procesa koji se analiziraju može biti puno može se na početku napraviti procjena važnosti određenih podprocesa. Važniji su podproces u ovome smislu oni koji u izvođenju troše više resursa i vremena (promatra se broj ponavljanja u jedinici vremena i utrošak resursa pri svakom ponavljanju). Poboljšanjem takvih procesa može se dobiti sveukupno bolji rezultat nego poboljšanjem manje važnih procesa. U analizu modela korisno je uključiti predstavnike svih zainteresiranih strana u koje, osim vlasnika procesa, čine sudionici u izvođenju, nadzoru ili kontroli procesa, članovi projektnog tima koji radi na automatizaciji, drugi stručnjacima unutar ili izvan organizacije, a u nekim slučajevima i korisnici procesa. Svatko od njih treba samostalno ili u manjim grupama proučiti modele i dati svoje komentare koji će dati korisne informacije o procesu i njegovu modelu iz različitih perspektiva. Zanimljive informacije pokazuju predstavlja li model dovoljno dobro stvaran proces i koliko se poklapa s propisanim procedurama, postoje li važnije iznimke ili područja u kojima se češće pojavljuju problemi. Također je prilika s obzirom na raznolikost publiku prikupiti prijedloge za unapređenjem. Tim koji proučava modele, samim time i procese, i prikupljene primjedbe i komentare treba dobro shvatiti poslovna pravila i razloge koji iza njih stoje. Često proces nije kroz dugo vremena bio unapređivan te su postojeća pravila zastarjela i/ili neprimjenjiva. Razumijevanje situacije u kojoj su nastala i usporedbom na postojeće stanje može se zaključiti koja pravila trebaju ostati, koja treba mijenjati, a koja ukinuti.

Prepoznati i opisati razlike u odnosu na propisane procedure – propisane procedure trebale bi točno opisivati procese što u praksi često nije slučaj. Razlika može biti na bolje, odnosno stvarni proces je kroz praksu prilagođen novim okolnostima odnosno poboljšán, ali i na gore jer se zbog nepoštivanja procedure u jednom dijelu stvaraju problemi u nekom drugom

dijelu procesa. Proces koji se kroz duži period odvija u odvojenim okolinama (npr. podružnicama) s vremenom se može promijeniti tako da nastane nekoliko različitih varijanti istog procesa. Međusobne razlike također treba uzeti u obzir jer one mogu pokazati da su uvjeti u tim okolinama različiti, ali i pokazati koja su mjesta pogodna za unapređenje uz istovremeno nuđenje nekoliko rješenja razvijenih kroz praksu.

Prepoznavanjem sličnih procesa i njihovom usporedbom može se doći do zanimljivih zaključaka. Nekad se oni mogu objediniti, a češće se dobra (ili loša) iskustva s jednog mogu iskoristiti za unapređenje sličnih procesa.

Na temelju informacija dobivenih analizom ili poslovnih zahtjeva, ako je riječ o osmišljavanju potpuno novih procesa, kreće se u modeliranje prijedloga budućih procesa. Vrlo važna informacija su jasni poslovni ciljevi prema kojima će se tražiti optimalno rješenje. Jasni poslovni ciljevi su važan preduvjet za osmišljavanje kvalitetnih prijedloga poslovnih procesa. Poslovni ciljevi zaposlenika trebaju biti u skladu sa ciljevima organizacijskih jedinica, a oni opet sa ciljevima cijele organizacije. Sukladno tome prije modeliranja trebaju biti određeni jasni poslovni ciljevi procesa na temelju kojih se osmišljavaju budućí procesi. Razumijevanje tih poslovnih ciljeva od strane analitičara, ali i svih uključenih u osmišljavanje procesa važan je preduvjet za kvalitetan rad jer se u ovisnosti o ciljevima rezultati mogu uvelike razlikovati. Primjerice, razlika između procesa čiji je cilj maksimalna jednostavnost za klijenta i procesa koji treba biti što jeftiniji za organizaciju može biti jako velika.

Utjecaj okoline na proces, kao i utjecaj procesa na okolinu ni u kom se slučaju ne smije zanemariti. Potrebno je shvatiti organizacijske, vremenske, kulturne i slične okolnosti u kojima će se proces odvijati. Proces prikladan za malu poslovnicu (manji broj zaposlenika, jednostavnija oprema, ali i manji broj klijenata) ne mora biti prikladan za veliku. Dnevna smjena u odnosu na noćnu i ljetna u odnosu na zimsku ne moraju isti posao raditi na isti način itd.

Potrebno je odrediti ključne pokazatelje uspješnosti, ključni pokazatelji uspješnosti određuju se na temelju poslovnih ciljeva, a u praksi omogućuju objektivno mjerenje razine njihova postizanja. Za svaki cilj treba odrediti barem jedan pokazatelj uspješnosti i njegove referentne vrijednosti. Tijekom izvođenja procesa mjerit će se stvarne vrijednosti čija usporedba s referentnima će pokazati do koje je razine cilj postignut. Jednako tako, ovi pokazatelji i referentne vrijednosti mogu se koristiti i pri dinamičkim analizama odnosno kod simulacija.

Slično kao kod modeliranja postojećih procesa i ovdje treba odrediti podprocese i aktivnosti koji se odvijaju unutar procesa, odrediti i modelirati njihove međusobne odnose (prethodnici/sljedbenici, paralelni rad, grananja i sl.) čime se dolazi do modela toka procesa. Često se, umjesto izrade modela od početka, kao predložak mogu iskoristiti modeli postojećih

procesa. Na temelju rezultata analize postojećih procesa takvi se modeli prilagođavaju te postaju modeli prijedloga procesa.

Prepoznati i opisati resurse/informacije/dokumente povezane s podprocesima i aktivnostima, ovi dijelovi procesa mogu se, ali i ne moraju, mijenjati s promjenom toka procesa, ali je u svakom slučaju to nužno provjeriti. Jednako tako, moguće je unaprjeđenje kod koje se tijek procesa ne mijenja već se samo mijenjaju resursi ili informacije koji se koriste.

Prepoznati organizacijske elemente povezane s podprocesima i aktivnostima i njihove odgovornosti bez obzira je li novi model rađen od početka ili je nastao izmjenama postojećeg potrebno je dodati ili ažurirati organizacijske elemente. Slično kao kod resursa, moguća je situacija, istina rjeđe, da je razlika između postojećeg i novog modela samo u nekom organizacijskom elementu.

Ukoliko prijedlog za unaprjeđenjem utječe na organizacijsku strukturu primjerice kroz dodavanje nove, ukidanje postojeće organizacijske jedinice ili nekog drugog organizacijskog elementa potrebno je tu promjenu dokumentirati u modelu.

U ovom se slučaju, s obzirom da je riječ o prijedlozima budućih procesa i njihovim varijantama, rade samo osnovne veze nužne za analizu prijedloga i provođenje simulacija. Svakako treba razlikovati veze modela prijedloga budućih procesa od onih na modele postojećih procesa. Kroz implementaciju prihvaćenog prijedloga, odnosno kad on postane model postojećeg procesa napraviti će se odnosno ažurirati veze prema ostalim modelima postojećih procesa.

Često su razlike između više prijedloga istog procesa velike. Ako je uz to procijenjeno da će cijena implementacije biti velika moguće je odlučiti se za provođenje simulacija. Simulacije omogućuju konkretniju usporedbu različitih prijedloga u dinamičkim uvjetima i provođenje „što ako?“ (what if?) analize. U takvom je slučaju modele potrebno detaljnije razraditi, ali je taj trošak bitno manji od mogućeg troška implementacije lošijeg rješenja.

Trajanje i cijena procesa direktno ovise o parametrima pojedinih aktivnosti. Potrebno je težiti točnijoj procjeni na nivou aktivnosti što će dovesti do boljeg modela za simulaciju. S druge strane treba imati na umu da se radi na modelu koji nikad neće savršeno prikazivati sustav pa u procjenama ne treba pretjerivati. Pogotovo ne na aktivnostima koje se rijetko pojavljuju i malo koštaju. Podaci o stvarnom izvršenju aktivnosti su najbolji te ovaj zadatak bitno jednostavniji ako postoje zapisi u tom smislu.

Potrebno je procijeniti vjerojatnosti pojedinih grananja, jer o grananjima ovisi koji će se niz aktivnosti i koliko puta izvršiti. Kao i kod prethodnog zadatka i ovdje treba težiti što realnijim procjenama, naravno u razumnim granicama i po mogućnosti pribaviti stvarne podatke.

Nadalje treba procijeniti učestalost i vremensku razdiobu događaja koji pokreću poslovne procese. Nije isto pokreće li se proces jednoliko kroz dulji period ili koncentrirano u određenim trenutcima. Posljedica mogu biti različiti optimalni procesi pa čak i drugačija organizacijska struktura. Stoga je i ove parametre potrebno procijeniti prije provođenja simulacija.

Modele procesa na kojima će se vršiti simuliranje potrebno je dopuniti procjenama iz prethodnih zadataka. Provođenje simulacija je u biti jednostavan posao ako se koristi dobar alat. Rezultat su usporedivi podaci o ponašanju procesa u dinamičkim uvjetima.

Analizom rezultata simulacije dolazi se do novih informacija koje mogu upućivati na nedovoljno dobre procjene, ali i na mogućnosti za poboljšanjem modela procesa. U skladu s tim zaključcima potrebno je doraditi modele te pokrenuti novi ciklus simulacija. Kada se rezultati simulacija pokažu zadovoljavajućima potrebno ih je dokumentirati da bi se na temelju njih mogle donositi daljnje poslovne odluke, ali i da bi ostali zapisani za usporedbu sa stvarnim podacima kada se proces počne izvršavati. Bez obzira na to jesu li ili nisu provedene simulacije moguća je situacija kod koje postoji nekoliko prijedloga jednog procesa koji ispunjavaju definirane zahtjeve. U ovom ih se koraku procjenjuje te se odabiru oni koji će biti implementirani. Važno je napomenuti da je mogući rezultat nekoliko varijanti istog procesa, a koje će se odvijati u različitim okolinama odnosno okolnostima. Da bi se donijela konačna odluka koje od predloženih varijanti procesa će biti implementirane potrebno je detaljno proučiti sve prijedloge. Ako je odrađeno simuliranje ono je rezultiralo dodatnim informacijama o ponašanju modela u dinamičkim uvjetima. Te informacije su još jedna pomoć kod donošenja u ovom koraku nužnih odluka. Obzirom da će proces funkcionirati kroz duže vrijeme i da će uz tom periodu proći kroz nekoliko ciklusa optimizacije dobro je odluke donesene u ovom trenutku dokumentirati. Takvi će dokumenti pomoći timu, bez obzira hoće li u njemu sudjelovati isti ili drugi članovi, koji će na procesu odnosno njegovu modelu raditi u budućnosti. Za svaku važniju odluku zašto će neko rješenje biti ili ne implementirano potrebno je opisati prednosti i mane u odnosu na konkurentna rješenja.

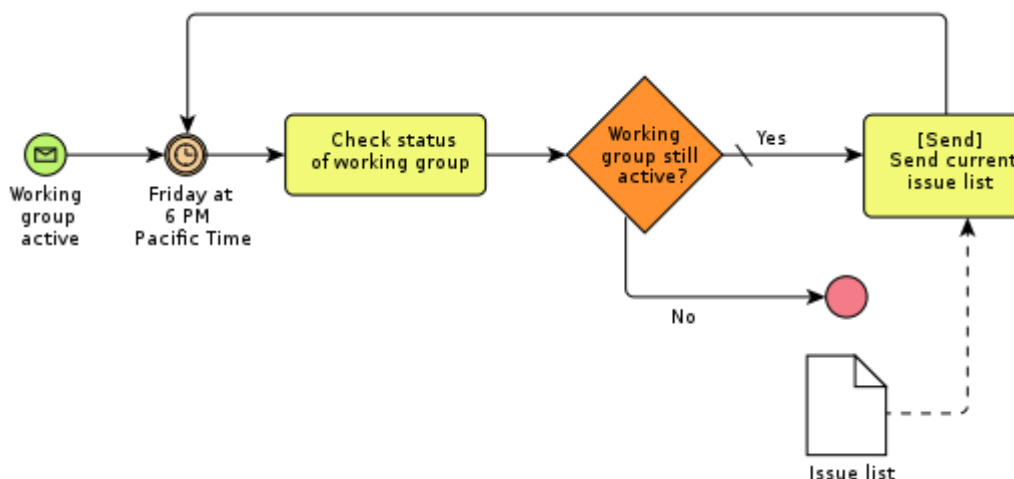
Na temelju prepoznatih prednosti i mana različitih varijanti procesa odabiru se one koje će biti implementirane. Nije nužno da jedan proces ima jednu implementaciju pogotovo ako će se odvijati u bitno drugačijim okolnostima (drugačiji prostor, raspoloživi resursi, vrijeme kroz koje se proces odvija i sl.)

Tijekom izvršavanja prikupljaju se podaci o samom procesu. Ti podaci mogu se više ili manje podudarati s očekivanjima, a njihovim se korištenjem i analizom dolazi do informacija koje će pomoći u dodatnim poboljšanjima procesa. Alati koji sudjeluju u izvođenju procesa, izvještaji ljudi koji sudjeluju u nadzoru i kontroli kao i iskustva svih uključenih u izvođenje

procesa glavni su izvori informacija o procesu. Njihovo prikupljanje i analiza omogućava prikupljanje novih prijedloga za unapređenjem odnosno neprestano poboljšanje procesa što je u interesu kako vlasnika procesa tako i njegovih korisnika. Obzirom na to da nakon implementacije proces prolazi kroz fazu uhodavanja korisnije je prikupljati i koristiti podatke nakon njena završetka. Koliko god se vremena i truda utrošilo kod modeliranja prijedloga i provođenja simulacija podaci neće biti toliko precizni koliko mogu biti podaci o realnom izvođenju procesa. Razlike stoga mogu biti u procjenama vrijednosti atributa elemenata ili nepreciznosti samog modela. Prepoznate razlike treba dokumentirati najbolje u samim modelima. Razlozi odstupanja mogu biti i u očekivanoj i stvarnoj okolini odvijanja procesa. U nekim je situacijama moguće unapređenje stvarne okoline (npr. dopuna resursa), a u nekim će kroz slijedeći ciklus trebati prilagoditi modele. S obzirom da ciklus, barem korak implementacije, najčešće ne kreće odmah, sve je prepoznate razlike potrebno dokumentirati.

3.3. Postupci i norme za modeliranje

U ovom poglavlju ukratko će se opisati metode i tehnike modeliranja. Poslovni proces modeliranja (BPM) je aktivnost kojom se predstavlja poslovni procese, na taj se način trenutni proces može analizirati i poboljšati. (8)



Slika 2 Modeliranje poslovnih procesa u razvoju aplikacija (Izvor: Magistarska teza - Modeliranje poslovnih procesa u razvoju aplikacija elektronskog poslovanja)

Pojam Business Process Modeling odnosi se na dizajniranje ,upravljanje i izvršavanje poslovnog procesa. BPM znači sagledavanje organizacije kao skupa procesa koji se mogu definirati, kojim se može upravljati, i koji se mogu optimizirati.

Razlozi za korištenje BPM su slijedeći (9)

Formalizacija postojećeg procesa

Bolje razumijevanje procesa

Efikasnije izvršavanje poslovnih procesa zbog korištenja BPM

Povećanje produktivnosti i smanjenje učešća ljudi u izvršavanje procesa

Omogućavanje ljudima da riješe komplicirane probleme

Temeljni ciljevi BPMN-a su:

Osigurati standardnu notaciju razumljivu svim poslovnim sudionicima, od izvršnog osoblja, preko projektanata i softverskih inženjera do uprave (management). Na taj način BPMN omogućuje premošćivanje jaza između modeliranja poslovnih procesa i izrade informacijske potpore za njih.

Omogućiti generiranje izvršnog jezika, zvanog BPEL (Business Process Execution Language), za obavljanje poslovnih procesa izravno iz grafičkog modela. (10)

4. Upravljanje zahtjevima

Upravljanje zahtjevima vrlo je specifična aktivnost, zbog potrebe za različitim tehnikama prikupljanja zahtjeva, a najviše zbog čestih promjena samih zahtjeva usred projekta. Kod izgradnje novog sustava potrebno je definirati i analizirati problem. Analizu problema, prikupljanje i dokumentiranje zahtjeva potrebno je sistematizirati. Upravo tu sistematizaciju nazivamo upravljanje zahtjevima, uz što je neizbježno upravljanje promjenama zahtjeva.

Upravljanje zahtjevima na sistematizirani način omogućuje izradu transparentne specifikacije sustava koja omogućuje postizanje i kontinuirano održavanje dogovora s korisnicima oko toga što točno budući sustav treba raditi. (12)

Svrha upravljanja zahtjevima:

Uspostavljanje i održavanje suglasnosti s korisnicima i drugim interesnim stranama što sustav treba raditi

Omogućavanje razvojnom timu bolje razumijevanje zahtjeva na sustavu

Definiranje jasnih granica sustava

Omogućavanje osnove za planiranje tehničkog sadržaja pojedinih iteracija razvoja sustava

Omogućavanje osnove za procjenu troškova i vremena za razvoj sustava

4.1. Uspostava projekta i početak razvoja

Prilikom upravljanja projektima razvoja softvera želimo istovremeno postići cilj, imati rizike pod kontrolom, prevladati sve prepreke i isporučiti produkt koji zadovoljava zahtjevu naručitelja (onog koji plaća) i potrebama krajnjeg korisnika. To je iznimno težak zadatak. U početnoj fazi projekt želimo postići, razumijevanje između stakeholdera o ciljevima i opsegu projekta, definirati okvir projekta, popisati zahtjeve korisnika, prepoznati poslovne slučajeve (use case), opisati značajne poslovne slučajeve.

Na početku je potrebno potrošiti puno vremena na proučavanje poslovnog procesa prije razvoja same aplikacije. Potrebno je prepoznati koji poslovni procesi su obavezni da bi podržali poslovanje svih zainteresiranih strana (engl. stakeholdera). Ti procesi se prikazuju sa poslovnim slučajevima, koji opisuju servise koje poslovanje pruža svojim korisnicima. Najvažniji analitičarev zadatak je jasno definiranje što sustav treba raditi i prikupiti zahtjeve od stakeholdera. Zahtjev je ono što stakeholder treba od projekta ili produkta.^{3}

4.2. Faze u procesu upravljanja zahtjevima

U procesu upravljanja zahtjevima slijedimo određeni redoslijed aktivnosti. Prva faza u procesu upravljanja zahtjevima je razumijevanje potreba i zahtjeva korisnika. U komunikaciji sa korisnikom moramo biti strpljivi jer je potrebno više iteracija razgovora, intervjuja ili nekih drugih tehnika prikupljanja zahtjeva. Ova je vrlo zahtjevna faza, sa kojom se nikako ne smije

brzati i korisničke specifikacije zahtjeva pokušati oblikovati nakon prve iteracije razgovora. Aktivnost identificiranja i razumijevanja zahtjeva provodi se u pet koraka (13)

Prepoznati izvor informacija o zahtjevima

Postavljati odgovarajuća pitanja kako bi postigli razumijevanje korisničkih potreba

Otkriti eventualne nekonzistentnosti zahtjeva, te odrediti opseg prikupljenih informacija

Potvrditi razumijevanje zahtjeva

Dokumentirati prikupljene zahtjeve

Prikupljene zahtjeve modeliramo s ciljem uspostave boljeg razumijevanja između poslovne i informatičke domene.

Slijedi aktivnost analize probleme, što treba rezultirati razlaganjem korisničkih zahtjeva na komponente. Potrebno je potrošiti puno vremena na identifikaciju i proučavanje poslovnih potreba obaveznih da bi podržali poslovanje svih zainteresiranih strana (stakeholdera). Cilj ove aktivnosti je postizanje razumijevanja svih uključenih strana, stvaranje dobre podloge za pronalaženja dizajna rješenja. Završetkom ove aktivnosti pojašnjene su granice zahtjeva, te se postavlja rješenje koje određuje opseg projekta.

Specifikacija zahtjeva sadrži ciljeve koje sustav treba zadovoljiti, opseg zahtjeva, popis funkcionalnosti, kratak opis sustava i korisnika, interakcije s postojećim sustavima, moguće rizike, zakonske zahtjeve i standarde.

Kao što je ranije opisano u analizi koriste se razne metode modeliranja koji ja moguće opisati Referentni modeli pomažu usklađivanju i pojednostavljenju raznih procesa. Razlikujemo poslovne modele, modele informacijskog sustava, razne zahtjeve prema informacijskom sustavu itd. Obzirom da referentni modeli proizlaze iz najbolje prakse smatramo da Rational Unified Process, po kojem se radi prilikom razvijanja aplikativnih rješenja, možemo promatrati kao referentni model. Referentni model RUP pomaže u minimiziranju pogrešaka i povećanju produktivnosti u razvoju aplikativnih rješenja.

RUP je skup znanja, metodologija koja okuplja i klasificira uloge (role), aktivnosti (discipline) i isporuke (artefakte) u projektima razvoja software-a. Cilj RUP-a je osigurati produkciju visoko kvalitetnih aplikativnih rješenja koja zadovoljavaju zahtjeve korisnika i to unutar zadanih rokova i raspoloživog proračuna. Rational Unified Process je procesni okvir kojeg je moguće prilagoditi potrebama pojedine organizacije. To je proces koji opisuje TKO radi ŠTO, KAKO i KADA.

Koraci koji opisuju izradu Specifikaciju zahtjeva:

Prepoznati potrebe korisnika i ostalih zainteresiranih strana (stakeholderi)

Upoznati zakonsku regulativu

Dogovarati sa korisnikom definiciju zahtjeva

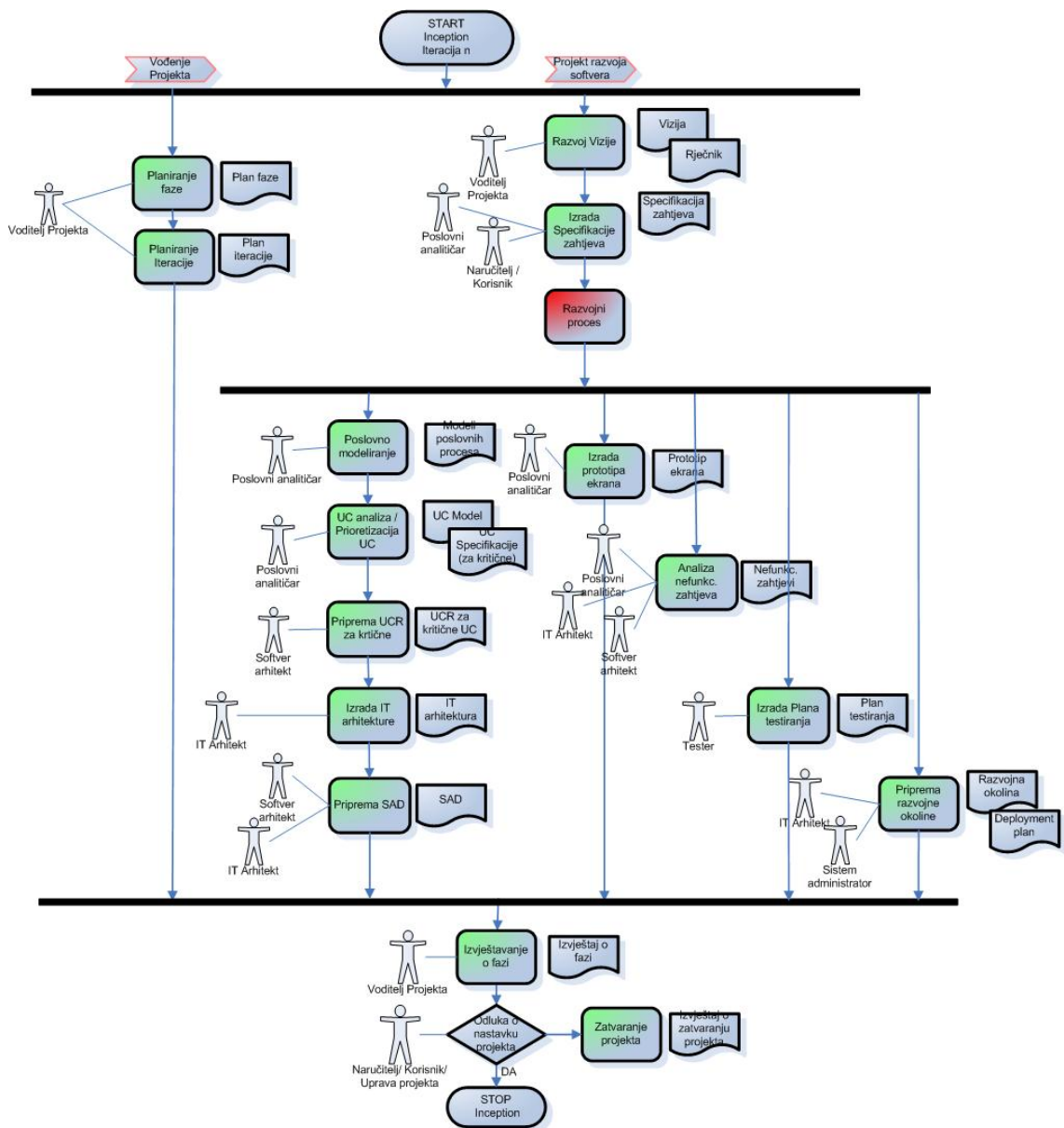
Opisati funkcionalnosti sustava

Prepoznati i opisati interakciju sa postojećim sustavima (ako postoje)

Prepoznati moguće rizike

Odrediti prioritet izrade sustava

Nakon početnog zahtjeva, slijedi detaljnija specifikacija. To je zahtjevna faza jer, ukoliko je specifikacija nekompletna i pogrešna nastaju veliki problemi u daljnjem razvoju rješenja. Specifikacija mora biti jasna, a ne dvosmisljena, kako ne bi došlo do krivih interpretacija onoga što je napisano. Use Case analizom postavlja se UC model koji opisuje funkcionalne zahtjeve kroz aktere i use caseove. Akteri su fizički korisnici sustava ili neki vanjski sustavi koji komuniciraju sa sustavom koji gradimo. Akteri su uvijek izvan sustava i nisu odgovornost sustava. Use Case (slučaj uporabe) opisuje način kako pojedini akter komunicira sa sustavom, serija interakcija između vanjskog entiteta i sustava. Mora davati mjerljivu vrijednost određenom akteru (izvještaj, neka događaj ili uvjet kao ishod akcije). Use case treba pisati jezikom koji korisnik razumije, ne smije biti vezan niti uz jednu tehnologiju. Slika prikazuje TKO, ŠTO u inicijalnoj fazi.

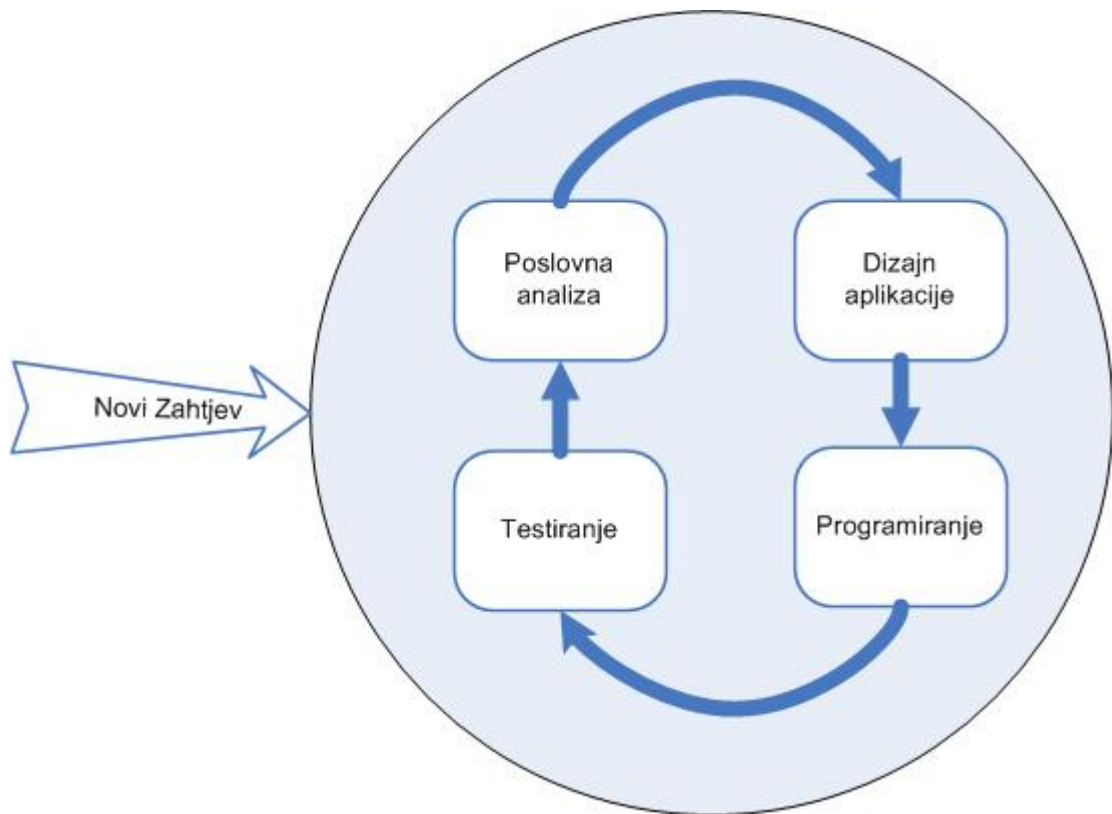


Slika 3 Role i aktivnosti u inicijalnoj fazi (Izvor: izrada autora)

Nadalje, zahtjevi moraju biti validirani u suradnji sa korisnikom. Time se postiže zajednička suglasnost zainteresiranih strana da je to upravo ono što korisnik želi. Tek nakon validacije, po definiranim zahtjevi započinje proces razvoja i testiranja rješenja.

Nakon ovih faza zahtjevi trebaju biti pohranjeni u jedinstveni repozitorij kako bi se mogli pratiti, verzionirati, povezivati high level zahtjevi sa komponentama zahtjeva i obrnuto.

Navedene faze dovode do sistematizacije i opisa proces, što glavne aktivnosti pri izradi aplikativnog rješenja, Takav način rada primjenjiv je u tvrtkama kojima je djelatnost razvoj software-a. Izvođenjem aktivnosti kako su opisane u fazama i koracima, isporukom opisanih artefakta olakšava se proces razvoja i osigurava isporuka kvalitetnog aplikativnog rješenja.

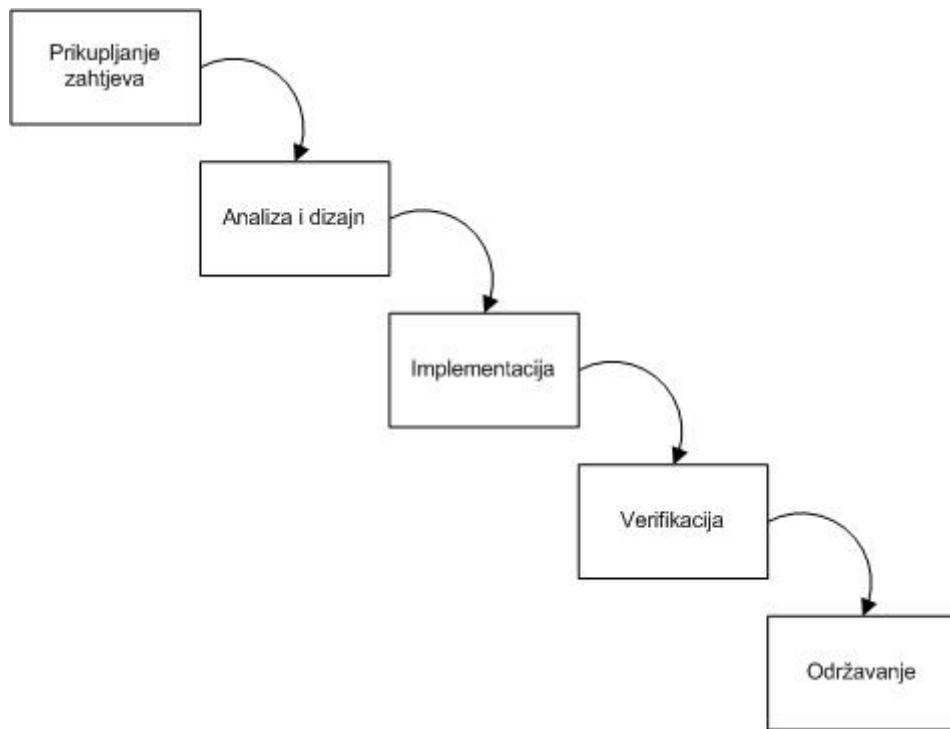


Slika 4 Proces razvoja zahtjeva, globalni pogled (Izvor: izrada autora)

4.3. Zahtjevi u iterativnom modelu

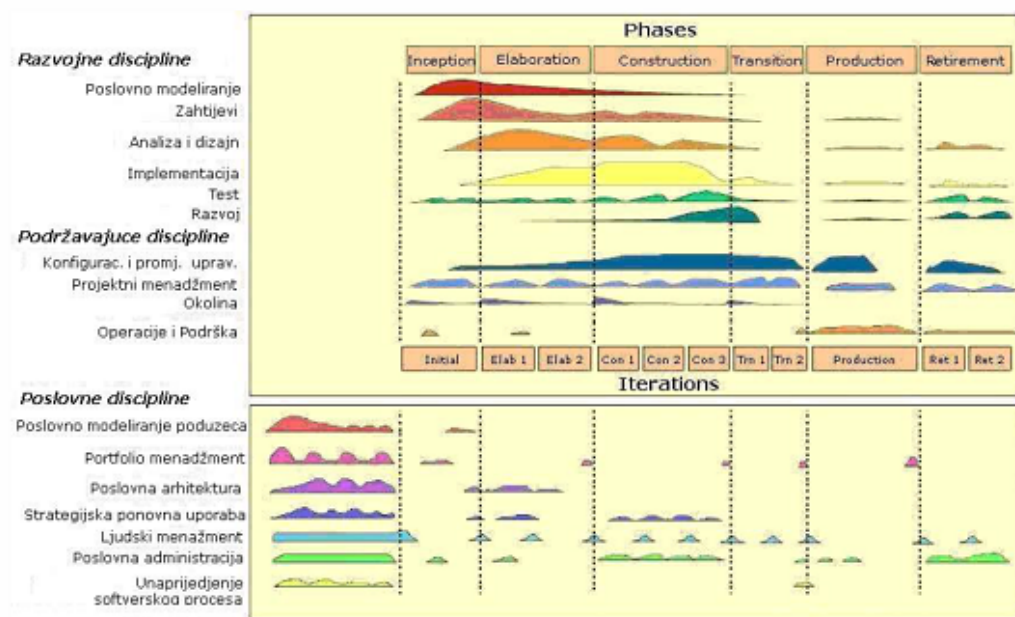
Tradicionalni model procesa razvoja rješenja prikupljenim zahtjevima je pravocrtni (slika Pravocrtni model procesa razvoja). Prikupljanje zahtjeva – Analiza i dizajn – Implementacija sustava – Integracija i testiranje sustava – Održavanje sustava.

U pravocrtnom modelu aktivnosti se dešavaju jedna za drugom, tek nakon potpunog završetka jedne aktivnosti slijedi druga aktivnost i tako do kraja izgradnje sustava. Problem u takvom načinu rada je što je potrebno čekati završetak jedne aktivnosti da bi mogla započeti sa slijedećom, tako da nema napretka u razvoju i eventualne pogreška u definiciji zahtjeva otkrivaju se prekasno.



Slika 5 Vodopadni model procesa razvoja programskog rješenja (Izvor: http://en.wikipedia.org/wiki/Waterfall_model)

Prilikom razvoja informacijskih sustava najbolja praksa je iterativni pristup (slika Iterativni proces)



Slika 6 Iterativni model procesa razvoja (Izvor: <http://www.agiledata.org/essays/enterpriseArchitectureTechniques.html#UML>)

Iterativnim modelom procesa razvoja moguće je da se aktivnosti odvijaju u iteracijama, tijekom svake iteracije rano se otkivaju i uklanjaju rizici. Iteracije se ponavljaju unutar nekoliko faza.

Iterativni pristup podržan je u RUP (Rational Unified Process) metodologiji. U RUP-u prepoznajemo dvije procesne strukture. Prva (horizontalna) prezentira vrijeme i pokazuje životni ciklus procesa kako je prikazano „hupserima“ slika 7. Ova dimenzija prezentira dinamiku aktivnosti procesa i izražava ih u terminima ciklusa, faza, iteracija i ključnih točaka (milestones). Druga procesna struktura (vertikalna) prezentira osnovu procesa radnog tijeka, koji logično i prirodno grupira aktivnosti. Ova nam prezentira statički aspekt procesa, njegove opise u terminima procesnih komponenti, aktivnosti, isporuka i uloga (rola)

Prilikom razvoja informacijskog sustava upravljanjem zahtjevima želimo istovremeno postići cilj, imati rizike pod kontrolom, prevladati sve prepreke i isporučiti produkt koji zadovoljava zahtjevu naručitelja (onog koji plaća) i potrebama krajnjeg korisnika. To je iznimno težak zadatak. Cilj upravljanja zahtjevima razvoja informacijskog rješenja u RUP-u je olakšati taj zadatak osiguravajući vođenje (usmjeravanje) na tom području. To nije recept za uspjeh, ali on povećava šanse za uspjeh uspješne isporuke rješenja.

Svaka od četiri RUP faze (incepcija, elaboracija, konstrukcija i tranzicija) završava dobro definiranim kontrolnim točkama (milestone). Incepcija i elaboracija su dvije najkreativnije faze. Funkcionalnosti i dizajn su uspostavljeni, svi zahtjevi su prepoznati.

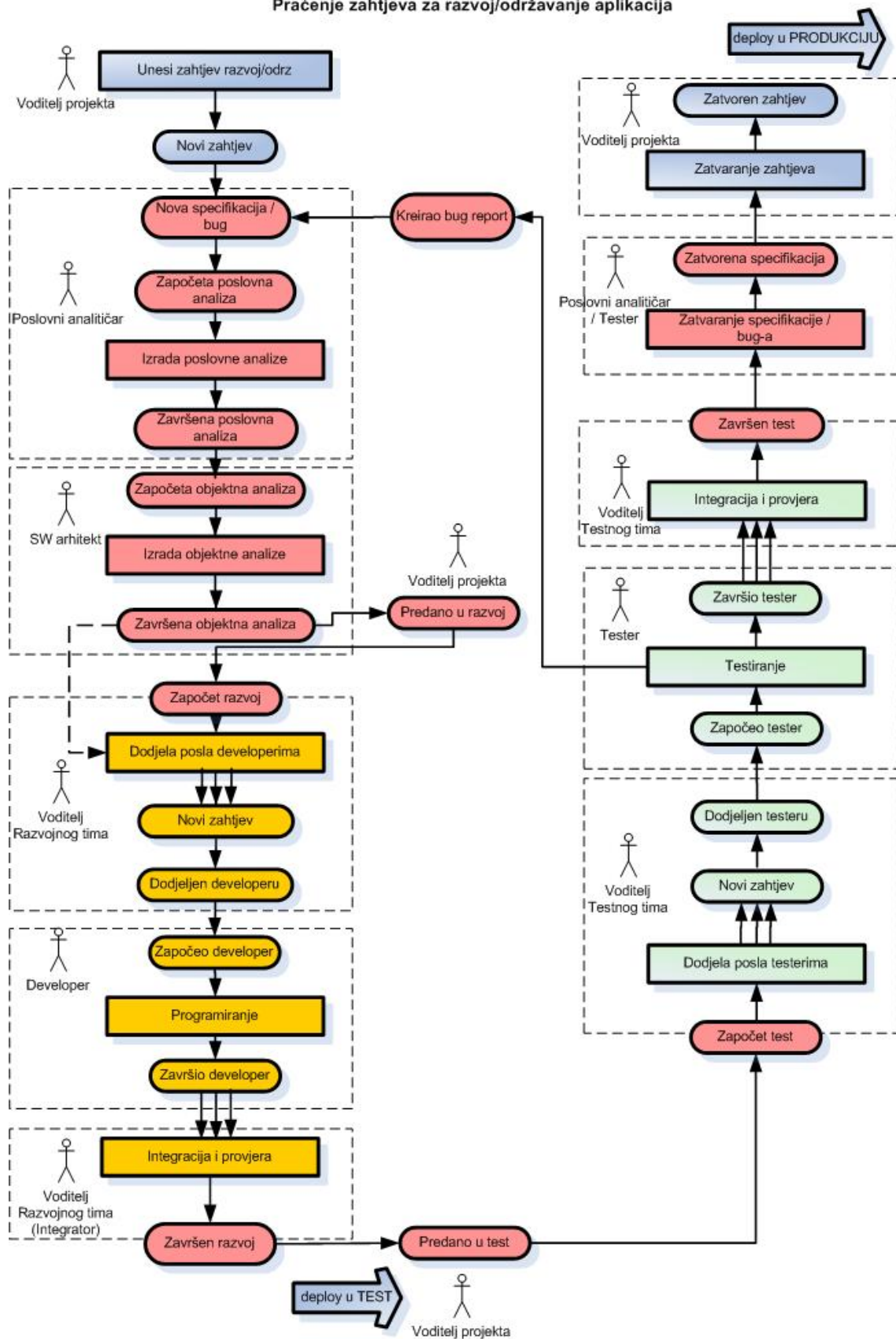
Konstrukcija i tranzicija su faze izrade (izgradnje), programiranje, testiranje i isporuka

Svaka faza može se podijeliti na iteracije. Iteracija je kompletan razvojni ciklus koji rezultira isporukom (internom ili eksternom) izvršnog produkta, podskupa konačnog produkta koji raste inkrementalno iz iteracije u iteraciju.

Prednosti iterativnog pristupa su što u svakoj fazi i iteraciji ponavlja se disciplina prikupljanja i sistematizacije korisničkih zahtjeva. Tajna pri upravljanju zahtjevima je prihvatiti da se zahtjevi mijenjaju. Identificiranje zahtjeva je kontinuirani proces koji je potrebno raditi zajedno sa krajnjim korisnikom.

Prilikom upravljanja zahtjeva važno je omogućiti praćenje statusa zahtjeva u svakom trenutku slika 7. Praćenje razvoja zahtjeva, dakle potrebno je implementirati alata kako bi sve zainteresirane strane bile obaviještene u kojoj fazi je izvršenje zahtjeva.

Praćenje zahtjeva za razvoj/održavanje aplikacija



Slika 7 Praćenje razvoja zahtjeva (Izvor: Vlastiti primjer)

4.4. Zahtjevi za promjenama

„Panta Rei“ jedan od koncepata Heraklitove filozofije, potpuna je istina primjenjiva na današnja dešavanja oko korisničkih zahtjeva. Sve se mijenja i sve je podložno promjenama. Promjene nije moguće zaustaviti, ali ukoliko prihvatimo činjenicu da su promjene neizbježne možemo se uhvatiti u koštac sa njima. Dobrom pripremom promjene nas neće iznenaditi i moći ćemo upravljati njima kad se pojave. Zahtjevi za promjenom odražavaju se kroz proces upravljanja promjenama. Važno je imati jedinstveno mjesto gdje se prijavljuju zahtjevi. Svaki zahtjev za promjenom mora biti odobren od strane tima koji se sastoji od ljudi koji procjenjuju izvodivost i stvarnu potrebu zahtijevane promjene. Ukoliko je zahtjev za promjenom odobren slijede faze kao i kod novog zahtjeva, analiza, detaljnija specifikacija, validacija. Naglasak je na jedinstvenom repozitoriju zahtjeva i zahtjeva za promjenom. Nakon pokretanja novog sustava u produkciji i svakodnevne upotrebe od strane krajnjeg korisnika pojavljuju se želje za proširenjem i izmjenama funkcionalnosti. Kako bi se olakšalo prikupljanje zahtjeva potrebno je koristiti neki alat koji će se na jednostavan način kategorizirati i odrediti prioritet prijavljenog zahtjeva. Zahtjev je obavezno pratiti po njegovim statusima od početka do stavljanja u produkciju. Projekti se grade s više timova, smještenih na različitim mjestima, na različitim platformama. Važno je osigurati konstantnu sinkronizaciju dijelova sustava i verifikaciju izmjena. Proces izmjene zahtjeva mora biti definiran i ponovljiv. Postojanje formalnog zahtjeva za izmjenom olakšava komunikaciju. Promjene mogu biti održavane u jednostavnom (čvrstom, otpornom, robusnom) i prilagodljivom sustavu.

Promjene se uvijek događaju. Efikasno upravljanje promjenama je jedna od najvažnijih aktivnosti tijekom provedbe zahtjeva. Izvori promjena obično se nalaze u sljedećim kategorijama: Potrebe/zahtjevi klijenata – dodavanje zahtjeva na proizvod ili usluge su česti izvor promjena. Dodavanje ili izmjena zahtjeva proizvoda ili usluga usred razvoja ili testiranja proizvoda može imati značajan utjecaj zbog potrebnih dorada i ispravaka.

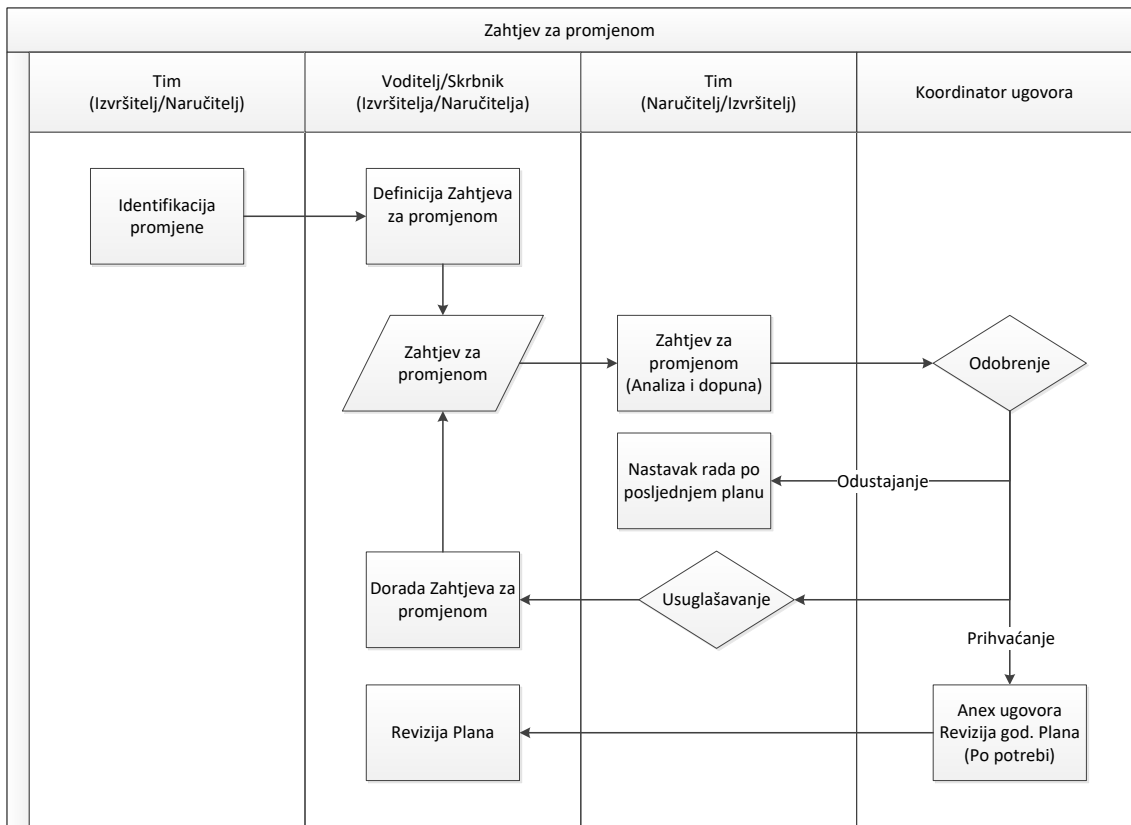
Svrha upravljanja promjenama je racionalizirati sustav promjena i pružiti kronološki pregled akcija nad zahtjevima. Životni ciklus zahtjeva je opis i slijed koraka potrebnih za praćenje promjena nad zahtjevima u sustavu. Životni ciklus je proces koji se prezentira dijagramom stanja. Osnovni elementi dijagrama su:

Stanje – trenutni status zahtjeva

Akcija – aktivnosti koje mijenjaju status zahtjeva

Pravila – propisana pravila za izvršavanje pojedinih akcija

Uloge – izvršitelji akcija



Slika 8 Praćenje razvoja zahtjeva (Izvor: Izrada autora)

5. Tehnike upravljanja zahtjevima

Jedna od potreba kvalitetnog upravljanja zahtjevima je sposobnost praćenja zahtjeva, kako bi se nadgledao status svakog pojedinačnog zahtjeva, ali i sustava u cjelini. Na taj se način omogućava pogled u definirane zahtjeve i da li oni odgovaraju svim elementima sustava, te nadalje dobiva se uvid u povezanost zahtjeva sa elementima modela sustava, sadržavajući sve potrebne informacije za razumijevanje zahtjeva. Opisivanjem i praćenjem zahtjeva nadgledamo i mjerimo kvalitetu sustava i zrelost razvojnog procesa.

5.1. Alati za upravljanje zahtjevima

Već dugi niz godina na tržištu postoje razni alati za praćenje i opisivanje zahtjeva, pomoću kojih je moguće mjerenje kvalitete i spremnosti sustava koji razvijamo. To su alati koji imaju mogućnost podrške za nadgledanje zahtjeva, verzioniranje dokumentacije, te povezivanje detaljnije specifikacije sa krovnim zahtjevom. Alati pomažu u povezivanju raznih artefakata između raznih rola u razvojnom procesu, te onemogućavaju istovremeno ažuriranje istih. Uz prednosti koje donose postoje i potencijalni problemi unutar jedne organizacije, odrediti platformu na kojoj će se pokretati alati kako bi bili dostupni sudionicima projekta sa raznim ulogama. Iz iskustva rada u razvojnom procesu vidim prednosti u implementaciji vlastitog rješenja sustava upravljanja zahtjevima. Prije svega potrebno je razviti svijest o potrebi centraliziranog zaprimanja zahtjeva, određivanja prioriteta, rokova i postavljanja statusa zahtjeva. Kada se takav proces ugradi u svijest sudionika u procesu, daleko je jednostavnije posložiti vlastito informatičko rješenje, nego kupovati alat i prilagođavati se prema tom kupljenom rješenju. Kao što je navedeno u poglavlju 4.3.1. postoji više elemenata dijagrama stanja, stvarajući vlastito rješenje moguće je utjecati na stanja, akcije, uloge.

Postavljanjem svojih pravila postizemo bolju komunikaciju i koordinaciju između raznih timova i raznih uloga.

5.2. Sposobnost opisivanja i praćenja zahtjeva

Perspektive na zahtjeve, od strane sudionika razvojnog procesa, su različite. Voditelja projekta zanimaju statusi izvođenja, te da li su svi zahtjevi pokriveni odgovarajućom dokumentacijom i poslovnom analizom. Voditelja razvojnog tima mora provjeriti da li su zavedene sve potrebne informacije za razumijevanje zahtjeva, te koji su rokovi i povezanost na zadatke koji proizlaze iz krovnog zahtjeva. Sposobnost opisivanja i praćenja zahtjeva važna je zbog dopunjavanja zahtjeva u stvarnom vremenu uz zadržavanje konzistentnosti raznih artefakata vezanih u pojedine zahtjeve. U razvojnog procesu uključeno je više rola koje proizvode različite dokumente koji su vezani uz zahtjev: model poslovnog procesa, UC specifikacija, UC realizacije, model podataka, UML dijagrami, testne skripte. U izradu dokumentacije uključeni su svi sudionici razvoja, neke role izrađuju i dopunjavaju dokumente, a neke role su zainteresirane za napredak i status projekta.

5.3. Povezivanje zahtjeva i UML metoda

Tijek poslova u poslovnog procesu može se opisati tekstualno u opisu ali i modelirati. Postoji više tehnika modeliranja poslovnih procesa. Jedan od standardnih jezika za modeliranje je UML (Unified Model Language). UML jedan je od alata za vizualno modeliranje. Vizualna apstrakcija pomaže:

- boljem sagledavanju software-a koji razvijamo sa različitih aspekata;

- u sagledavanju kako su elementi sustava povezani;

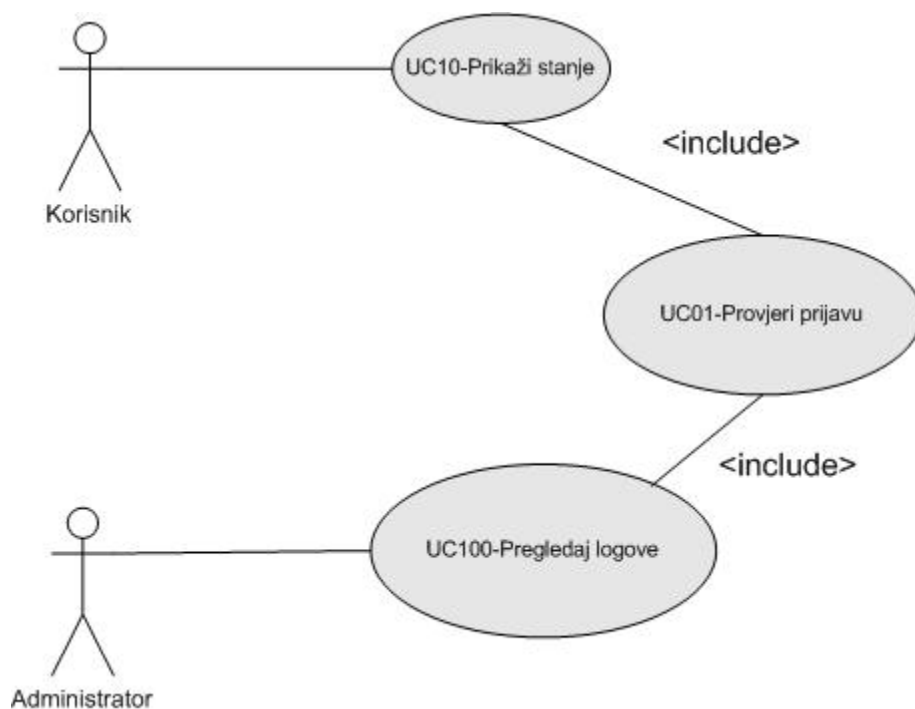
- sačuvati dosljednost između dizajna i njegove implementacije;

- boljem razumijevanju sustava;

- nedvosmislenoj komunikaciji.

Putem modeliranja korisnicima se lakše predoče njihovi procesi i veze prema ostalim sustavima. „Slika govori više nego tisuću riječi.“ Nakon prvih razgovora i prepoznavanja zahtjeva moguće je nacrtati dijagram toka procesa, prototip korisničkog sučelja, scenariji uporabe, Use Case dijagram slika 2. Pomoću raznih dijagrama analiziramo zahtjeve, lakše

razgovaramo sa korisnicima, pokušavamo definirati funkcionalnosti, te ih povezati sa raznim rolama krajnjih korisnika. Modeliranje pomaže analitičaru zahtjeva da pripremi programere i lakše objasni korisnikove potrebe. Time se postiže bolja i brža komunikacija između poslovnog i tehničkog svijeta te se osim samih procesa ubrzavaju i njihova unapređenja. Korištenjem raznih alata za upravljanjem zahtjeva mogu se povezati elementi modela u zahtjevima prikupljenim od strane korisnika navedenim, u svrhu osiguranja dvosmjerne povezanosti između zahtjeva i poslovnih procesa. Moguća je i jedinstvena povezanost prikupljenih zahtjeva i elemenata, naravno, jednom uspostavljena veza se ažurira svakom promjenom.



Slika 9 Primjer Use case dijagrama (Izvor: Izrada autora)

5.4. Alati za upravljanje zahtjevima

U ovom poglavlju osvrnut ću se na aplikacijska rješenja koja se koriste u procesu upravljanja zahtjevima prilikom razvoja aplikacijskog rješenja. Prije samog opisa već gotovih aplikacijskih rješenja koja se koriste u procesu upravljanja zahtjevima prilikom razvoja

aplikacijskog rješenja smatram kako u kratkim crtama treba navesti neke osnovne značajke koje takva rješenja trebaju podržavati.

Ukoliko se pogledaju trendovi u razvoju softverskog rješenja, uz standardne norme primjene „čistog koda“ nastoji se uvesti čim veća efektivnost upravljanja zahtjevima te vezanje istih s programskim kodom. Tako aplikacije koje se koriste u procesu upravljanja zahtjevima trebaju dati mogućnost korisniku da:

Upiše sve funkcionalne zahtjeve u aplikaciju

Prati i dodaje resurse za svaki zahtjev (resurs u ovom kontekstu može biti: vremenski, novčani ili se odnosi na zaposlenika koji radi na razvoju zahtjeva)

Prati status svakog pojedinog zahtjeva, kao i cjeline

Integrira funkcionalne zahtjeve s programskim kodom

Prilagođene tijekove rada (koji su vezani uz način razvoja programskog koda i zahtjeva)

Kreira potrebne izvještaje (npr. O utrošenim radnim satima, o procjenama razvoja pojedinih zahtjeva i slično)

Dijele znanje unutar same aplikacije (npr. „wiki sustav“)

Prethodno nabrojane mogućnosti navedene su temeljem dugogodišnjeg iskustva u području vođenja projekata.

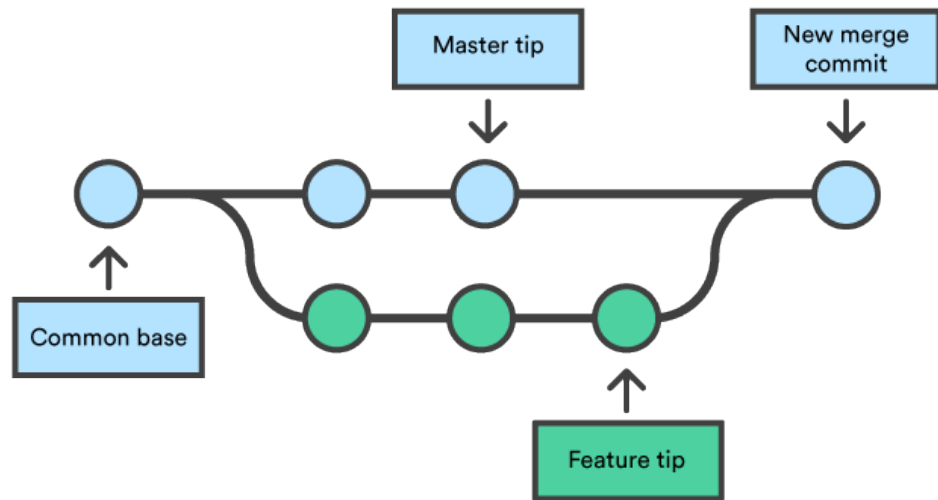
U nastavku rada ukratko ću se osvrnuti na najčešće korištene komercijalne i nekomercijalne aplikacije koje se koriste kao potpora u procesu upravljanja zahtjevima prilikom razvoja aplikacijskog rješenja.

5.4.1. Git

Git je distribuirani sustav za kontrolu verzije programskog koda koja se koristi za praćenje promjena u istom prilikom razvoja samog programskog rješenja. Samo programsko rješenje koristi se za koordinaciju posla između programskih inženjera, ali se koristi i za praćenje promjena kod bilo kakvog seta datoteka. Glavni cilj Git-a odnosi se na brzinu, integritet podataka i podršku prilikom razvoja aplikacijskog rješenja. (21)

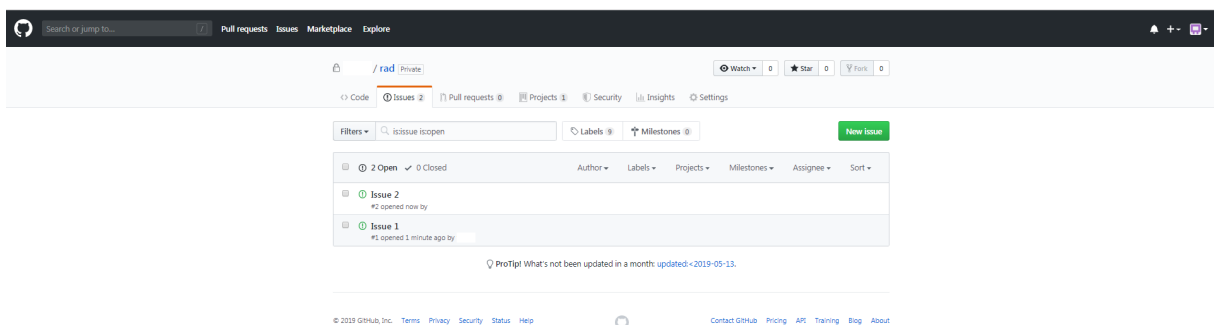
Git je besplatan i open-source softver, a u nastavku će biti nabrojane i prikazane neke od karakteristika samog sustava.

Glavne karakteristike Git-a vidljive su u podršci razvoja tzv. „nelinearnog“ razvoja programskog rješenja. Nelinearan razvoj programskog rješenja odnosi se na mogućnost grananja, spajanja i navigaciju kroz takav razvoj programskog rješenja.



Slika 10 Grananje tijekom razvoja softvera (Izvor: izrada autora)

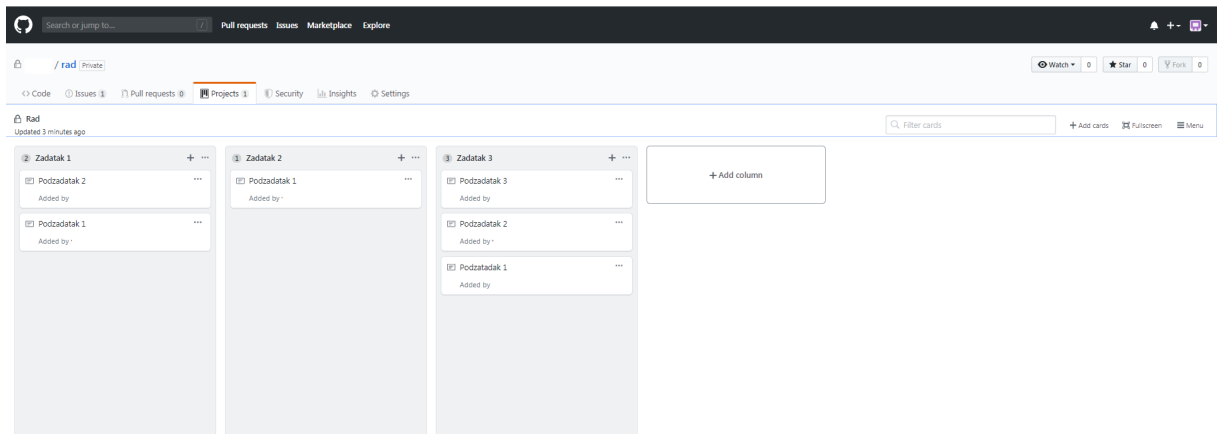
Nadalje, kao glavnu karakteristiku mogu navesti da svaki od programskih inženjer a posjeduje lokalnu kopiju cijelog razvojnog procesa. Sve promjene koje se naprave na „grani“ vidljive su u svim repozitorijima. Ujedno, Git je kompatibilan sa eksternim sustavima i protokolima (poput HTTP) te je ujedno skalabilan te brz. Ono što bi navela kao nedostatak jest nemogućnost praćenja statusa zahtjeva.



Slika 11 Pregled zadataka u Gitu (Izvor: izrada autora)

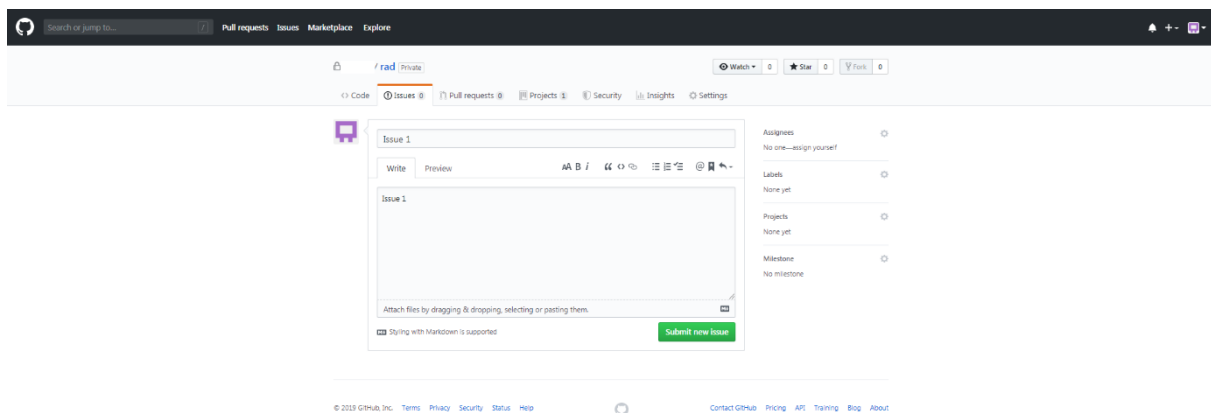
Na gornjoj slici prikazano je sučelje u kojem se nalaze zadatci koji su uneseni u Git sustav. Kao što je vidljivo, oba zadatka su u „Open“ statusu. Ovdje bih željela napomenuti da

zadatci u Gitu mogu biti u nekoliko statusa, npr. greška, duplikat, poboljšanje, dokumentacija i slično.



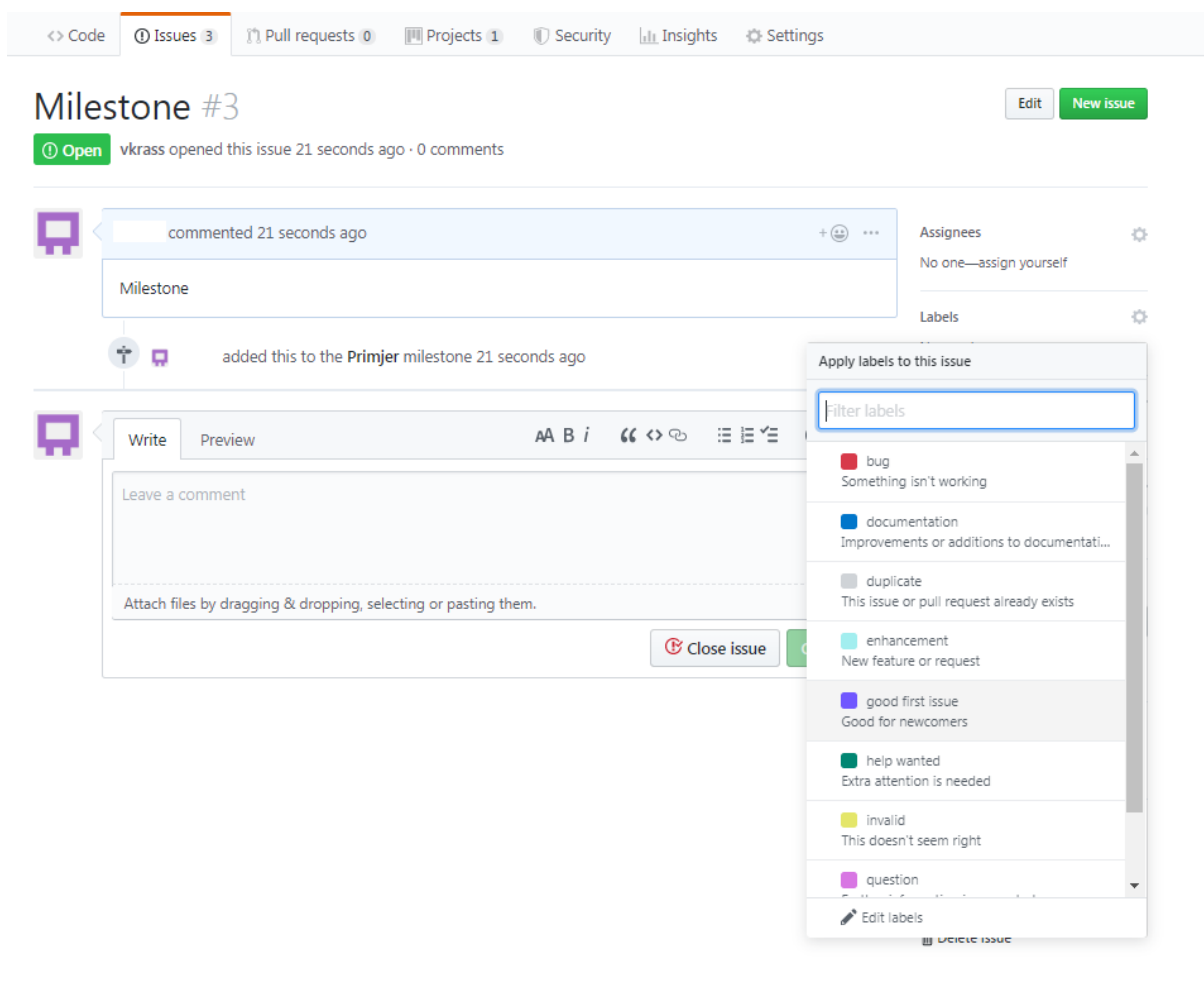
Slika 12 Definiranje zadataka i podzadataka (Izvor: izrada autora)

Gornja slika prikazuje jedan testni primjer projekta te glavne zadatke (1,2,3) i njihove podzadatke. Istraživanjem i radom u programu nedostatak je vidljiv u tome što nisu prikazane osnovne informacije koje sadrže podzadatci nego samo ime podzadatka. Ukoliko se klikne na podzadatak, nisu vidljive nikakve daljnje informacije za što smatram da je nedostatak aplikacije.



Slika 13 Kreiranje zadataka (Izvor: izrada autora)

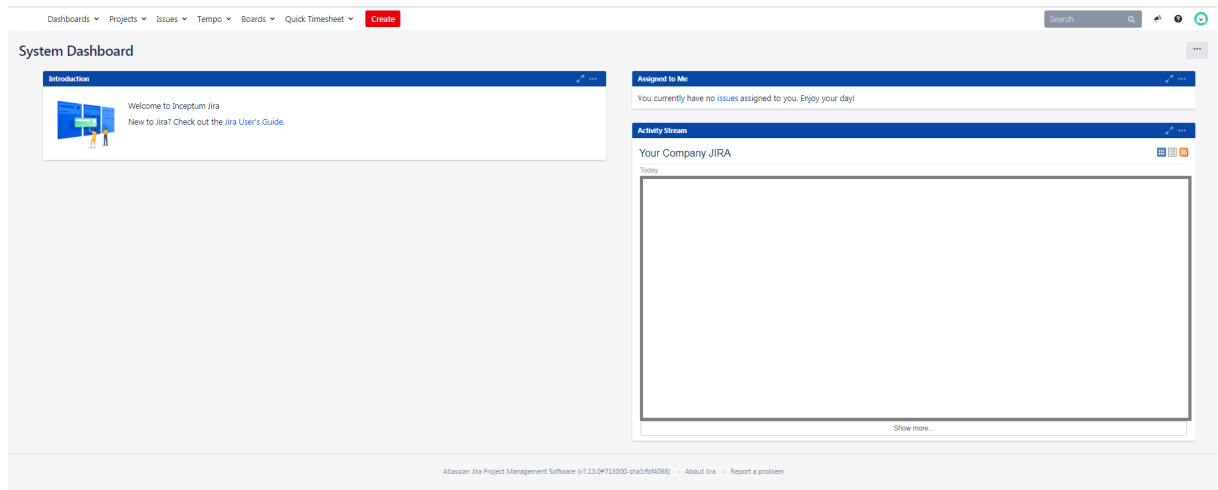
Na prethodnoj slici prikazano je korisničko sučelje prilikom izrade zadataka. Vidljivo je kako se zadatku može pridružiti osoba koja će odraditi zadatak, projekt kojem zadatak pripada, tip zadatka i slično.



Slika 14 Kategorija zadatka (Izvor: izrada autora)

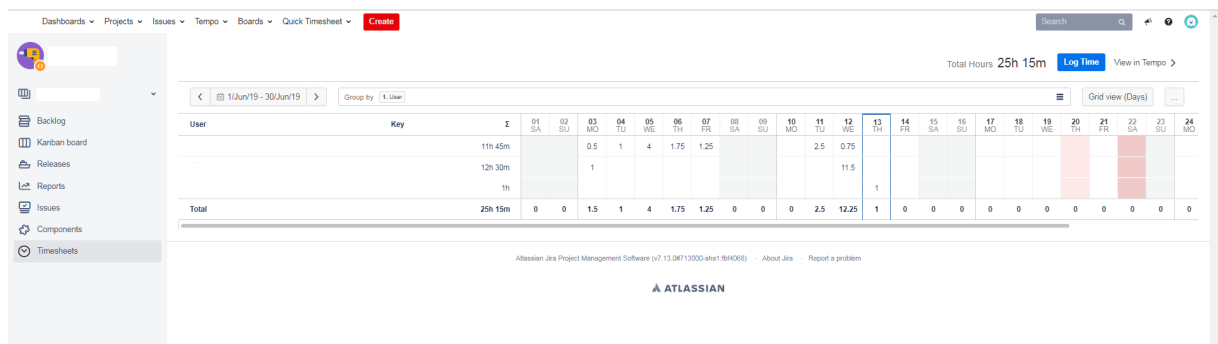
5.4.2. Jira Atlassian

Jira Atlassian je programsko rješenje koje se koristi najčešće u kombinaciji sa Git sustavom. Naime Jira je specijalizirani alat koji služi za praćenje zahtjeva. Samo programsko rješenje unutar sebe sadrži i tzv. „Service Desk“ koji se koristi za podršku korisnicima. Prednost Jira programskog rješenja u odnosu na Git jest vidljiv status zadataka te mogućnost razlikovanja koji zadatak spada u cjelinu. (22)



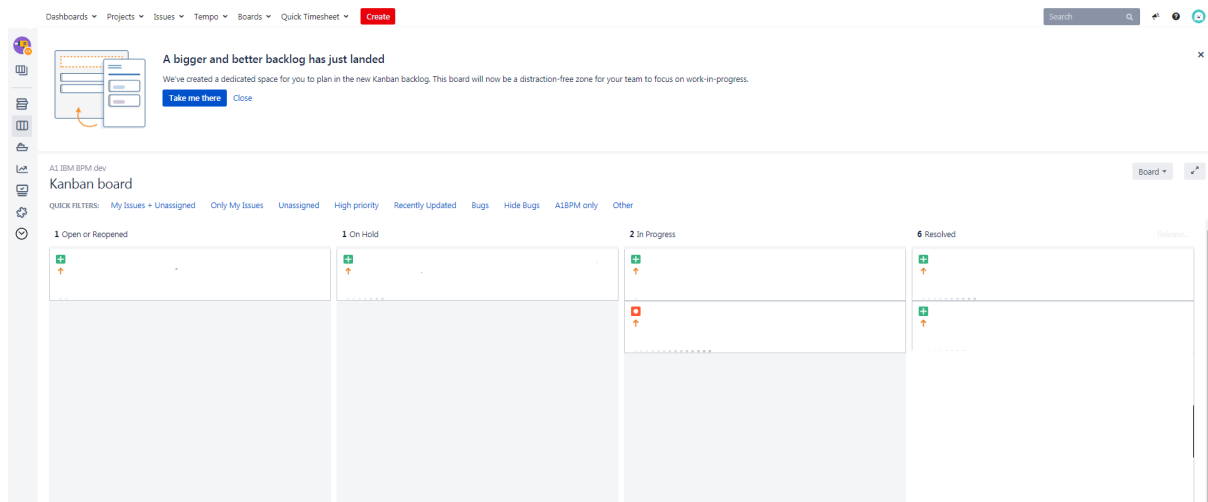
Slika 15 Jira dashboard (Izvor: izrada autora)

Na prethodnoj slici prikazano je početno korisničko sučelje Jira Atlassian programskog rješenja. Kada se korisnik prijavi svojim korisničkim podacima može vidjeti koji zadatci su mu dodijeljeni što korisnicima (developerima i sl.) omogućava lakši i jednostavniji rad s obzirom da se unaprijed zna koje zadatke trebaju isti odraditi.



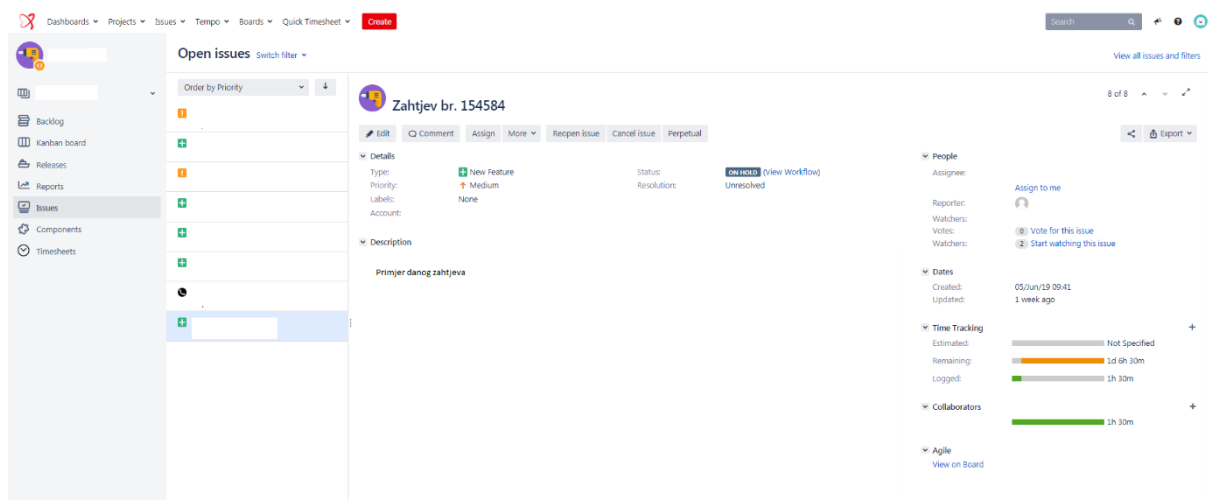
Slika 16 Praćenje utrošenog vremena (Izvor: izrada autora)

Gornja slika prikazuje i utrošak sati na projektu za svakog poslenika posebno, ali u gornjem desnom kutu prikazuje i ukupan broj odradenih sati. Time je na jednostavniji način moguće utvrditi stvarni utrošak sati na nekom od projekta.



Slika 17 Status zahtjeva (Izvor: izrada autora)

Gornja slika prikazuje prikaz u kojem statusu su određeni zadatci. Jira automatski stvara 4 statusa – open, on hold, in progress i resolved. Tj. Na hrvatskom, primljeni zahtjevi, na čekanju, u postupku rješavanja te riješeni zahtjevi.

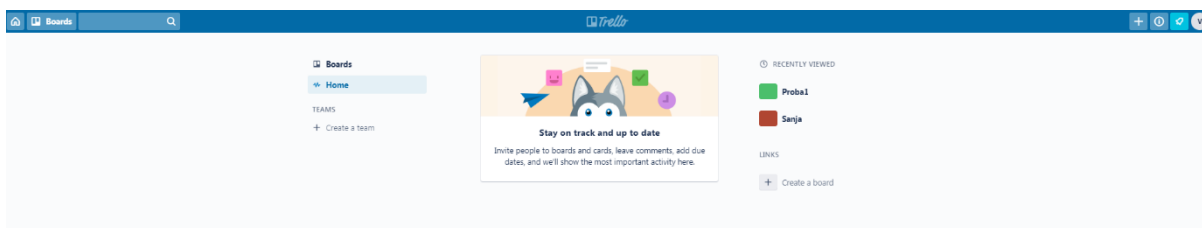


Slika 18 Opis zahtjeva (Izvor: izrada autora)

Gornja slika prikazuje prikaz detalja za svaki zadatak. Može se uočiti kako svaki zadatak bilježi tko isti odrađuje, koliko je procijenjeno vrijeme rješavanja, koliko je vremena utrošeno na rješavanje zadatka i slično.

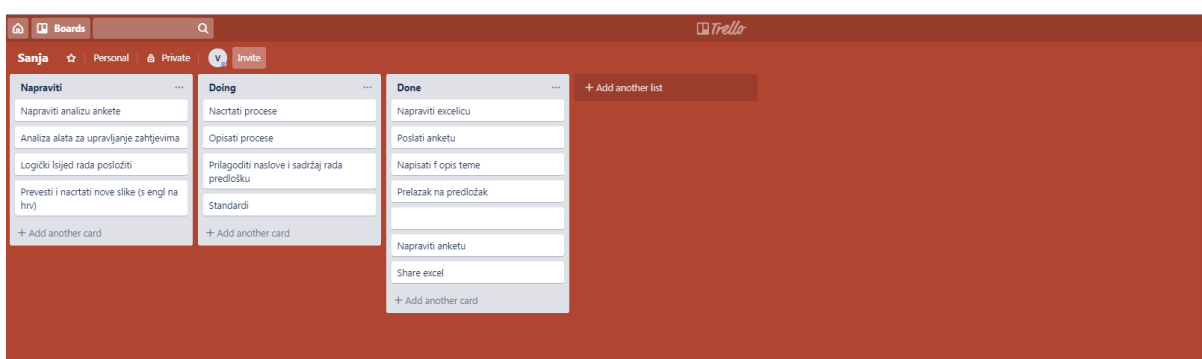
5.4.3.Trello

Trello je programsko rješenje koje je više prilagođeno za upravljanje projektima. Trello ima user-friendly korisničko sučelje te funkcioniра drag-drop principom. Trello je web based programsko rješenje koje je besplatno. U nastavku ću prikazati nekoliko korisničkih sučelja. (23)



Slika 19 Početni zaslon (Izvor: izrada autora)

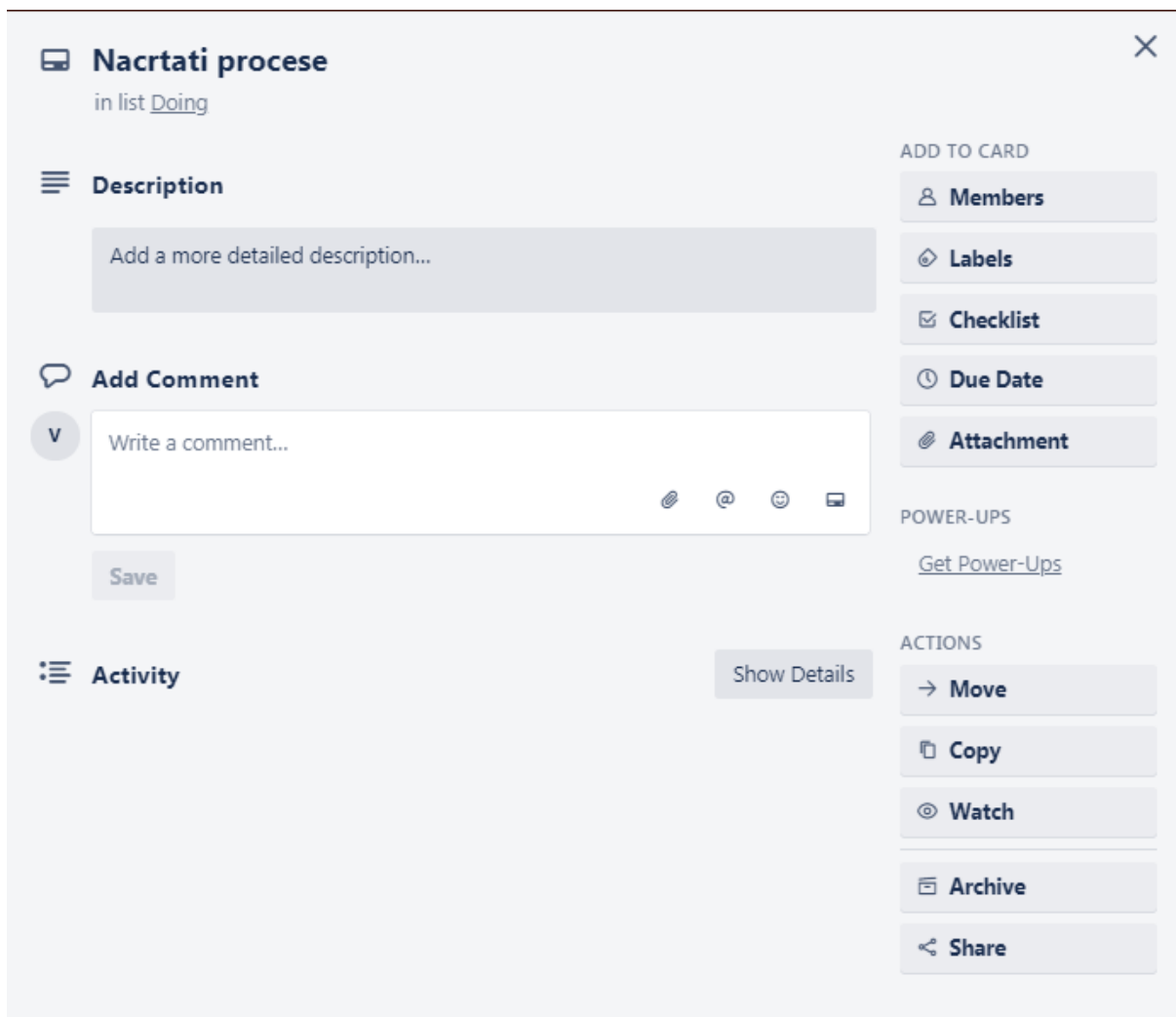
Gornja slika prikazuje početno korisničko sučelje. S desne strane se nalazi popis svih projekata. Ukoliko se klikne na jedan od projekata otvara nam se sučelje koje se nalazi u nastavku.



Slika 20 Status zadataka (Izvor: izrada autora)

Kao što je vidljivo iz slike 20, Trello prati status zahtjeva slično kao i programsko rješenje Jira Atlassian. Svaki zahtjev, odnosno zadatak nalazi se u jednom od statusa. Sam korisnik može kreirati broj statusa prema osobnim potrebama. Za potrebe ovog rada, u gornjem primjeru je vidljivo kako se zadatci nalaze u jednom od tri stanja – napraviti, u radu, završeno.

Ukoliko se klikne na jedan od zadataka otvara nam se nova forma koja je prikazana u nastavku.



Slika 21 Detalji zadatka (Izvor: izrada autora)

Iz same forme vidljivo jest da se mogu dodati neke osnovne stvari poput opisa zadatka, komentari, članovi koji su zaduženi za rješavanje zadatka, prilozi te rok do kada isti treba biti izvršen.

6. Inženjerstvo zahtjeva

Ovo poglavlje objasniti će osnovne pojmove vezane uz upravljanje zahtjeva gledajući ih s inženjerske strane. Ukratko će se dati definicije i objašnjenja osnovnih pojmova poput zahtjeva, uloga koje sudjeluju u procesu upravljanja zahtjevima, analize zahtjeva, specifikacije i validacije.

6.1. Osnove zahtjeva programskog proizvoda

Bourque i Fairley definiraju zahtjev kao svojstvo koje mora biti izloženo od strane nekog entiteta s ciljem rješavanja problema u stvarnom svijetu (15). Najčešće se zahtjevi odnose na automatizaciju (djelomičnu ili cjelovitu) zadatka neke osobe u poslovnom procesu, korekciju postojećeg programskog rješenja ili kontrolu određenog uređaja. Zahtjevi prema programskom proizvodu najčešće su kompleksna kombinacija „želja i potreba“ različitih ljudi koji se nalaze na „različitim“ mjestima u organizaciji, a koji su uključeni u samo okruženje i korištenje programskog proizvoda.

Prema programskom inženjerstvu najvažnijim svojstvom programskog proizvoda smatra se verifikacija zahtjeva bez obzira radi li se o funkcionalnom ili nefunkcionalnom svojstvu programskog proizvoda. Zahtjevi sadrže mnogo atributa po kojima se mogu rangirati, testirati te na kraju se može i ocijeniti sama kvaliteta odrađenog zahtjeva. Prema prethodno spomenutim atributima najčešće se rade određeni kompromisi prilikom definiranja, razvoja i kontrole samog programskog svojstva.

6.2. Programski proizvod i proces

Prema Bourgue i Fairley-u zahtjev u (programskom) proizvodu definira se kao potreba ili ograničenje prema programskom proizvodu. Zahtjev prema procesu razvoja programskog proizvoda u suštini se odnosi na ograničenje u samom procesu razvoja programskog proizvoda.

Neki od zahtjeva programskog proizvoda generiraju implicitna ograničenja/zahtjeve u samom procesu razvoja programskog proizvoda. Primjer tome može biti poseban način verifikacije samog programskog rješenja.

6.3. Funkcionalni i nefunkcionalni zahtjevi

Funkcionalni zahtjevi opisuju funkcije koje programsko rješenje treba izvoditi(16). Tekst mora biti oblikovan prema posebnim uputama (font, veličina i slično) – primjer funkcionalnog zahtjeva.

Nefunkcionalni zahtjevi opisuju ograničenja programskog proizvoda. Najčešće se definiraju kao ograničenja, no mogu se koristiti i kao zahtjevi prilikom provjere kvalitete programskog proizvoda, zahtjevi performansi, sigurnosni zahtjevi i slično.

6.4. Neposredni zahtjevi

Nakon definiranja zahtjeva programskog proizvoda implicitno se postavljaju neposredni zahtjevi. Neposredni zahtjevi su zahtjevi koji nisu navedeni pomoću jedne komponente zahtjeva nego ovise o integraciji svih komponenti programskog proizvoda. Neposredni zahtjevi ovise o arhitekturi programskog proizvoda.

6.5. Kvantitativni zahtjevi

Zahtjevi programskog proizvoda moraju biti jasno definirani, te gdje je moguće kvantitativni. Kvantitativnost zahtjeva posebno se odnosi na nefunkcionalne zahtjeve programskog proizvoda.

6.6. Sistemski i programski zahtjevi

Sistemski zahtjevi definiraju se kao zahtjevi cijelog sustava. Ako sustav sadrži programske komponente , onda se programski zahtjevi izvode iz sistemskih zahtjeva. Sistemski zahtjevi sadrže zahtjeve krajnjeg korisnika, zahtjeve dionika te zahtjeve koji nisu definirane od strane ljudi.

6.7. Proces inženjerstva zahtjeva

U ovom poglavlju ukratko ću objasniti modele procesa, uloge, podršku procesu te kvalitetu procesa programskog inženjerstva.

6.7.1. Modeli procesa

Srž modela procesa pri inženjerskom pristupu upravljanja zahtjevima u razvoju programskog proizvoda jest da prikaže da sam razvoj programskog proizvoda nije jednosmjerna aktivnost u ciklusu razvoja programskog proizvoda (17). Modelom procesa želi se prikazati kako se inicijalni model procesa mijenja i redefinira kroz sam period razvoja programskog proizvoda.

Model procesa upravljanja zahtjevima definira programske zahtjeve kao konfiguracijske značajke i upravlja njima pomoću programskog proizvoda za upravljanje konfiguracijom, kao i drugim programskim proizvodima u ciklusu razvoja programskog proizvoda.

Svaki model procesa upravljanja zahtjevima mora biti prilagođen organizaciji te projektu razvoja programskog proizvoda.

6.7.2. Uloge u procesu

Prema Bourque i Fairleyu proces upravljanja zahtjevima jest fundamentalno interdisciplinarni te zahtjeva da u istom sudjeluje specijalist za prikupljanje i analizu zahtjeva. Specijalist se u procesu koristi kao posrednik između dionika i programskog inženjera.

Najčešći uloge koje sudjeluju u procesu upravljanja zahtjevima su:

Korisnici – grupa ljudi koja će koristiti programski proizvod

Krajnji korisnici – grupa ljudi za koje je namijenjen programski proizvod na tržištu

Analitičar – grupa ljudi koja se bavi istraživanjem tržišta

Regulatori – grupa ljudi koji moraju osigurati da su zahtjevi programskog proizvoda u skladu sa zakonima u danom području

Programski inženjeri – individualne osobe ili grupa ljudi koja razvija programski proizvod

Važno je naglasiti kako je nemoguće perfektno zadovoljiti zahtjeve svakog sudionika u procesu, te je proces programskog inženjera ili specijalista da pregovorom osigura i dogovori

kompromis koji je prihvatljiv svim sudionicima u procesu. U samim pregovorima pregovara se o budžetu, tehničkim, regulatornim i drugim uvjetima. Kako bi se sam proces pregovaranja što bolje odvio potrebno je identificirati te analizirati sve sudionike u procesu razvoja programskog proizvoda.

6.7.3. Kvaliteta i unaprjeđenje procesa

Glavni cilj kvalitete i unaprjeđenja procesa s inženjerskog pogleda na sam proces upravljanje zahtjevima odnosi se na troškove i trajanje ciklusa razvoja programskog proizvoda te zadovoljstva korisnika isporučenim.

6.8. Prikupljanje zahtjeva

Prikupljanje zahtjeva bavi se izvorima zahtjeva programskog proizvoda i načinom na koji ih programski inženjer prikuplja. Prva faza prilikom razvoja programskog proizvoda odnosi se na razumijevanje poslovne potrebe koju programski proizvod treba riješiti. Fundamentalno se radi o ljudskoj aktivnosti te to podrazumijeva definiranje svih sudionika i veza između sudionika koji sudjeluju u samom ciklusu razvoja programskog proizvoda. Što kvalitetnije prikupljanje zahtjeva podrazumijeva efektivnu komunikaciju između svih sudionika koja se proteže kroz cijeli životni ciklus razvoja programskog proizvoda.

Najkritičniji element prilikom prikupljanja zahtjeva odnosi se na definiranje projektnog opsega. Definiranje projektnog opsega uključuje opis i svrhu razvoja programskog proizvoda, isporuke i popis prioriteta kako bi se osiguralo zadovoljstvo korisnika. Time se smanjuje vrijeme i rizik nepotrebne potrošnje resursa. S druge strane, opis mora biti skalabilan i mora biti u mogućnosti prihvaćati dodatne zahtjeve koji nisu bili „postavljeni“ prilikom inicijalnog procesa prikupljanja zahtjeva.

6.8.1. Izvori zahtjeva

Kako bi se što bolje i detaljnije opisali zahtjevi programskog proizvoda, esencijalno je definirati sve potencijalne izvore. Najčešći izvori zahtjeva su:

Ciljevi - u pogledu programskog proizvoda odnosi se na grubo definirane ciljeve programskog proizvoda, tj. koje probleme treba riješiti korištenje programskog proizvoda

Domensko znanje – programski inženjer mora biti u mogućnosti steći ili već posjedovati specifično znanje vezano uz aplikacijsku domenu (primjena konkretnih i relevantnih koncepata prilikom razvoja programskog proizvoda za određenu aplikacijsku domenu).

Sudionici – sam programski inženjer ili medijator koji skuplja zahtjeve programskog proizvoda mora biti u mogućnosti definirati, analizirati i upravljati s različitim „pogledima“ sudionika prilikom razvoja i budućeg korištenja programskog proizvoda

Poslovna pravila – odnosi se na specifične činjenice koje definiraju ili ograničavaju određene aspekte strukture ili ponašanja programskog proizvoda

Operativno okruženje – odnosi se na dobivanje zahtjeva iz okoline u kojima će se programski proizvod koristiti

Organizacijsko okruženje – programski proizvod treba podržati i upravljati poslovnim procesima

6.8.2. Tehnike prikupljanja zahtjeva

Nakon što se definiraju svi izvori i sudionici koji sudjeluju u životnom ciklusu razvoja programskog proizvoda, programski inženjer ili poslovni analitičar (specijalista za sakupljanje zahtjeva) može započeti s prikupljanjem zahtjeva sudionika. U ovom poglavlju primarni fokus stavlja se na tehnike prikupljanja zahtjeva. Prikupljanje zahtjeva jedan je od težih zadataka te sam poslovni analitičar treba biti svjestan činjenice da ponekad korisnici imaju problem prilikom opisivanja traženih funkcionalnosti ili tijeka procesa, da mogu izostaviti bitne informacije ili ne žele ili ne mogu sudjelovati prilikom procesa prikupljanja zahtjeva.

Najčešće tehnike prikupljanja zahtjeva programskog proizvoda su:

Intervju – „tradicionalna“ tehnika u kojem se poslovni analitičar razgovara sa svim sudionicima te prikuplja potrebne informacije direktno od korisnika

Scenariji upotrebe– kao tehnika prikupljanja zahtjeva važnost se stavlja na kontekst samog scenarija. Pomoću scenarija dobiva se okvir u kojem poslovni analitičar postavlja pitanja o zadacima vezanih uz zadani scenarij. Najčešća pitanja u slučaju scenarija odnose se na „ako ovo“ (uvjeti izvođenja) i „kako se obavlja“ (tijek procesa). Scenariji se opisuju pomoću use case dijagrama.

Prototipiranje – kao tehnika prikupljanja zahtjeva je bitna jer se time definiraju jasni nedvosmisleni zahtjevi. Ponašaju se slično kao i scenariji, no glavnom prednosti smatra se prikaz konteksta samom korisniku te je u ovom slučaju sam korisnik upoznat s informacijom koju on treba „dati“

Radionice – cilj ovakvog načina prikupljanja zahtjeva jest dobivanje „zajedničke slike“ što programski proizvod treba raditi. Sudionici u grupama na radionicama doprinose boljem uvidu u zahtjeve programskog korisnika, za razliku od individualnog intervjua s korisnikom. Na radionicama se mogu koristiti dodatne tehnike poput brainstorminga te se ujedno mogu redefinirati ideje do kojih bi se teško došlo pomoću individualnih intervjua. Ujedno, prednost radionica vidljiva jest u tome što se konfliktni zahtjevi jako rano uočavaju te je lakše upozoriti sudionike na iste kada dođe do istih

Opservacija – kao tehnika u fokus stavlja važnost programskog proizvoda u organizacijskom kontekstu. Kako bi se izveli potrebni zahtjevi motre se korisnički zadatci (na koji način se obavljaju, pomoću kojih alata i slično).

Korisničke priče – kao tehnika koristi kratke, opisne priče potrebne funkcionalnosti te opisuje na koji način korisnik koristi zadanu funkcionalnost

Ostale tehnike – odnose se na analizu konkurenata, analiziranje podataka i slično.

6.9. Analiza zahtjeva

Ovo poglavlje bavi se procesom analiziranja dobivenih zahtjeva kako bi se identificirali i riješili potencijalni konflikti između dobivenih zahtjeva, kako bi se uspostavile granice programskog proizvoda te načina na koji isti komunicira s organizacijskom i operativnom okolinom.

Zahtjevi se mogu klasificirati po različitim osnovama, a neke od njih su podjela na:

- Funkcionalne i nefunkcionalne zahtjeve
- Hijerarhija zahtjeva
- Zahtjev temeljen na procesu izvođenja ili programskog proizvoda
- Prioritetne i ne prioritetne zahtjeve
- Opseg zahtjeva – odnosi se na opseg na koji način zahtjev ima utjecaj na sam programski proizvod i na programske komponente
- Stabilnost zahtjeva – s obzirom da se zahtjevi mogu mijenjati u bilo kojem ciklusu razvoja programskog proizvoda korisno je napraviti procjenu da li će se zahtjev promijeniti

6.10. Konceptualno modeliranje

Razvoj modela stvarnog problema smatra se glavnim „ključem“ prilikom izvršavanja analize zahtjeva programskog proizvoda. Glavni cilj je dobiti bolji uvid i razumijevanje u stvarnu situaciju i okolinu u kojoj problem nastaje. Primjena konceptualnih modela u procesu upravljanja zahtjevima programskog proizvoda vidljiva je u nizu dostupnih alata poput korištenja grafa slučaja korištenja, podatkovnih modela, objektnih modela, modela podataka i mnogih drugih. Notacija koja se koristi prilikom razvoja konceptualnih modela zove se UML (engl. *Unified Modeling Language*).

Izbor korištenja pojedinog koncepta te načina njegovog modeliranja ovisi o „prirodi“ problema koji se rješava, ekspertizom i znanjem programskog inženjera te procesnim zahtjevima korisnika. Najbitnije je krenuti s modeliranjem konteksta programskog proizvoda jer se na taj način modeliraju veze programskog proizvoda sa operativnom okolinom. Time se definiraju sva potreba korisnička sučelja prema okruženju u kojem radi, ili će raditi, programski proizvod.

6.10.1. Arhitekturalno dizajniranje te alokacije zahtjeva

U samom životnom ciklusu razvoja programskog proizvoda mora se javiti pojam arhitekture sustava te se isti treba definirati. Arhitekturalnim dizajnom smatra se točka u životnom ciklusu razvoja programskog proizvoda u kojem se procesni zahtjevi preklapaju s zahtjevima programskog ili sistemskog dizajna. Arhitekt programskog proizvoda uključuje se u sam razvoj programskog proizvoda kako bi izradio analizu zahtjeva za dizajnom komponenta programskog proizvoda koji bi zajedno kolaborirajući trebali zadovoljiti zahtjeve korisnika.

Alokacija zahtjeva u ovom kontekstu odnosi se na dodjelu arhitekturalnih komponenti programskog proizvoda zadužene za ispunjenje dobivenih zahtjeva korisnika. Arhitekturalno dizajniranje povezano je s konceptualnim modeliranjem.

6.10.2. Pregovaranje o zahtjevima

Kako bi se riješili konflikti između zahtjeva u ovom poglavlju ukratko ću se osvrnuti na rješavanje konflikata te pregovaranje o zahtjevima.

U kontekstu prikupljanja i upravljanja zahtjevima, konfliktom se smatraju dva ili više „poprečnih“ zahtjeva između dva ili više sudionika. Takvi zahtjevi najčešće sadrže dvije ili više nekompatibilnih značajki programskog proizvoda, zahtjeve za resursima ili funkcionalne i nefunkcionalne zahtjeve koji nisu povezani. Zbog toga je potrebno postići konsenzus sa svim sudionicima te napraviti prikladne kompromise između zahtjeva.

Ovo poglavlje stavljeno je u analizu zahtjeva zato jer se konflikti najčešće javljaju kao posljedica analize zahtjeva.

6.11. Specifikacija zahtjeva

U kontekstu programskog inženjerstva specifikacija zahtjeva odnosi se na dokument koji je sustavno izrađen, recenziran te odobren. Prilikom analize kompleksnih sustava izrađuju se tri dokumenta: definicija sustava, specifikacija zahtjeva sustava te zahtjevi programskog proizvoda. Prethodno spomenuti dokument najčešće se dopunjuju te se koriste „zajedno“. U nastavku ću se ukratko osvrnuti na njih.

6.11.1. Definicija sustava

Ovaj dokument sadrži sve zahtjeve sustava koji su definirani na „high-level“ način. Opisuje sve korisnike koji sudjeluju u korištenju sustava u danoj domeni. Osim korisnika sadrži sve ciljeve koje sustav kao cjelina treba ispuniti, okruženje u kojem radi, nefunkcionalne zahtjeve te ograničenja sustava. Kao dokument može sadržavati konceptualne modele koji ilustriraju kontekst samog sustava, primjenu i korištenje scenarija sustava te prikaz procesa te glavnih principa entiteta domene.

6.11.2. Specifikacija zahtjeva sustava

Ovaj dokument sadrži detaljno specificirane zahtjeve sustava te se iz tog dokumenta izvode zahtjevi programskog/ih proizvoda, a nakon toga se mogu specificirati zahtjevi komponenti programskog/ih proizvoda.

6.11.3. Specifikacija programskog proizvoda

Specifikacija programskog proizvoda koristi se kao sporazum između korisnika te „ugovorenog izvođača“ te sadrži sve smjernice i značajke što programski proizvod sadrži te što isti ne sadrži.

Specifikacija proizvoda sadrži „grubu“ procjenu zahtjeva prije izrade dizajna te smanjuje troškove naknadnog redizajniranja programskog rješenja u kasnijim stadijima razvoja programskog proizvoda. „Gruba“ procjena sadržava procjenu troškova, rizika te terminskog plana razvoja programskog proizvoda. Sama specifikacija programskog proizvoda nadalje se može koristiti od strane organizacije prilikom verifikacije i validacije da programski proizvod radi kako i što treba.

6.12. Validacija zahtjeva

Specifikacija programskog proizvoda mora biti verificirana od strane svih sudionika koji sudjeluju u samom procesu razvoja programskog proizvoda – od krajnjeg korisnika do programskog inženjera. Na taj način osigurava da se svi sudionici razumiju zahtjeve te glavne isporuke programskog proizvoda. Glavni cilj validacije zahtjeva jest da se riješe problemi prije nego se dodaju potrebni resursi za razvoj zahtjeva.

6.12.1. Recenzije specifikacije

Jedna od najčešćih metoda validacije specifikacije koristi se recenzija specifikacije programskog proizvoda. Grupa ljudi bavi se proučavanjem specifikacije te im je zadatak da pronađu moguće greške, krive pretpostavke i sve što se razlikuje od standardne prakse. Grupa recenzenata koristi smjernice koje govore na što obratiti pažnju te su takve smjernice najčešće dane u obliku check liste.

6.12.2. Prototipiranje

Prototipiranje kao način validacije najčešće se odnosi na verifikaciju interpretacije programskog inženjera na dane zahtjeve. Prednost korištenja prototipa kao verifikacijskog alata vidljiva je u tome što korisnici mogu lakše interpretirati pogled softverskog inženjera i dati potrebnu povratnu informaciju što je dobro te što je krivo. Nedostaci korištenja prototipa kao

validacijskog alata vidljivi su u tome što se korisnikova pažnja usmjerava s funkcionalnih zahtjeva na zahtjeve dizajna.

6.12.3. Test prihvatanja

Test prihvatanja jest zapravo provjera od strane korisnika da finalni proizvod radi ono što treba i što je dogovoreno. Svodi se na dizajniranje testa koji ocjenjuje kako krajnji korisnik koristi i pristupa sustavu uz pomoć razvijenog programskog proizvoda.

7. Standardi za upravljanje zahtjevima

U ovom poglavlju obradit ću jedan standard te jedan okvir koji se koriste za upravljanje zahtjevima. Postoje brojni standardi, koje su izdale različite organizacije. U svakom standardu navest ću tko ga je izdao i koja je namjena izdanog standarda.

7.1. ISO 20000

ISO 20000 je prvi internacionalni standard namijenjen upravljanju IT uslugama. Prva verzija razvijena je 2005. godine, te je trenutno na snazi treće izdanje iz 2018. godine. Temelj ove norme jest prijašnji standard BS 15000 razvijen od BSI grupe. (18)

Originalno je standard razvijen s ciljem da sadrži najbolje prakse i smjernice te da uključuje i ITIL (Information Technology Infrastructure Library). ITIL je skup najboljih praksi te smjernica kojima se nastoji pružiti što bolja kvaliteta i podražka poslovnim procesima koristeći IT.

Norma ISO 20000 koristi se u organizacijama kada:

- Organizacija traži usluge od pružatelja usluga i traže osiguranje da će njihovi zahtjevi biti ispunjeni
- Organizacija želi pokazati svoj kapacitet u dizajnu usluge, tranziciji, isporuci te kontinuiranom poboljšanju ispunjenja zahtjeva usluga

Kako bi se uspješno implementirala norma potrebno je opisati 3 glavna uvjeta:

- Odgovornost menadžmenta – top menadžment svojim akcijama mora prikazati predanost razvoju, implementaciji te kontinuiranom poboljšanju procesu upravljanja uslugama u kontekstu organizacijskih i korisničkih zahtjeva
- Važnost dokumentiranja – pružatelji usluga trebaju pružiti dokumente i zapisnike koji osiguravaju efektivno planiranje, izvođenje te kontrolu nad upravljanjem uslugama
- Razvoj vještina – sve odgovornosti i role u procesu upravljanja uslugama trebaju biti definirane. Nadalje, s ciljem što efikasnijeg i kvalitetnijeg pružanja i održavanja IT usluga, potrebno je razvijati vještine zaposlenika. Ujedno, menadžment mora objasniti svojim zaposlenicima važnost uloge i aktivnosti koji isti obavljaju i način na koji pridodaju uspješnosti upravljanja uslugom.

Metodologija koja se veže uz ISO 20000, poznatije pod nazivom „*Plan-Do-Check-Act*“ može biti implementirana na sve procese.

U „Plan“ fazi potrebno je odrediti ciljeve i procese neophodne za dobivanje rezultata koji su u skladu s korisničkim i organizacijskim zahtjevima. „Do“ faza odnosi se na implementaciju procesa, dok „Check“ faza podrazumijeva nadzor i mjerenje procesa (trajanje, troškovi i sl.) te prikazati rezultate u odnosu na politike, ciljeve i prethodno definirane zahtjeve. Faza „Act“ podrazumijeva kontinuirano poboljšanje procesa.

Postoje 5 glavnih „područja“ procesa, a to su:

- Isporuka usluge – procesi koji osiguravaju najbolju kvalitetu isporučene usluge. Glavni cilj ovih procesa odnosi se na definiranje, prihvatanje i održavanje razine usluga. U ovu kategoriju spadaju procesi:
 - Upravljanje razinom usluge – od definiranja razine usluge, prihvata od strane korisnika do isporuke
 - Izvještavanje o usluzi
 - Neprekinuti rad usluge – osigurava da je usluga operativna i dostupna u svim uvjetima
 - Upravljanje financijama – odnosi se na planiranje i praćenje troškova koji se koriste za isporuku usluge
 - Upravljanje kapacitetom – osigurava da je dovoljno resursa dostupno za uslugu
 - Upravljanje sigurnošću
- Proces upravljanja komunikacijom – cilj ovih procesa je uspostavljanje komunikacije između isporučitelja usluge i korisnika iste.
- Proces rješavanja konflikata – cilj ovog procesa jest rješavanje problema (bilo kakvog tipa – tehničkih, dopuna i slično) prilikom isporuke usluge korisnicima
- Kontrolni procesi – integrirani u sustav u kojem se ocjenjuje razina usluge. Uključuje dva procesa, a to su: upravljanje konfiguracijom te upravljanje promjenama.
- Upravljanje isporukom – odnosi se na procese isporuke i praćenja jedne ili više promjena u produkcijsko okruženje

Važno je napomenuti da se uz ovu normu još najčešće vežu i norme ISO 9000 (norma o sigurnosti informacijskih sustava) te norma 15504 (norma koja daje tehničke standarde o dokumentaciji koja se koristi prilikom razvoja programskog rješenja).

7.2. FitSM

FitSM je besplatan standard kojem je cilj omogućavanje efektivnog upravljanja uslugama u širokom području djelovanja organizacija. Glavni ciljevi FitSM-a su (19):

- Kreiranje jasnog i ostvarivog standarda koji omogućuje učinkovito upravljanje IT uslugama
- Verzija standarda upravljanja uslugama koji je pristupačniji za širi raspon okruženja
- Predložiti rješenje koje se može učinkovito implementirati u organizacijama bez obzira na veličinu istih
- Definiranje početnih vrijednosti učinkovitosti upravljanja uslugama koje se mogu postići, ali i povećati ako je potrebno

FitSM je vrlo slično strukturiran prema ISO standardima, tj. dokumentiran je u nekoliko dijelova poput:

- Glavni pregled – nudi općenit pregled standarda
- Zahtjevi – dokument sadrži sve zahtjeve koje je potrebno ispuniti za efektivno upravljanje uslugama.
- Ciljevi i aktivnosti – dokument koji sadrži opise svih rezultata procesa definiranih u standardu
- Uloge – dokument koji sadrži popis svih uloga te aktivnosti koje isti izvršavaju
- Obrasci i primjeri – dokument koji sadrži primjer i opis dokumenata koji se koristi u procesu upravljanja uslugama, poput SLA, politika, kataloga usluga i slično.
- Smjernice – dokument koji sadrži smjernice za mnoštvo tema vezanih uz upravljanje uslugama
- Procjena zrelosti – tehnika i alat koji se koristi za jednostavniju procjenu zrelosti implementiranih procesa upravljanja uslugama

8. ITIL

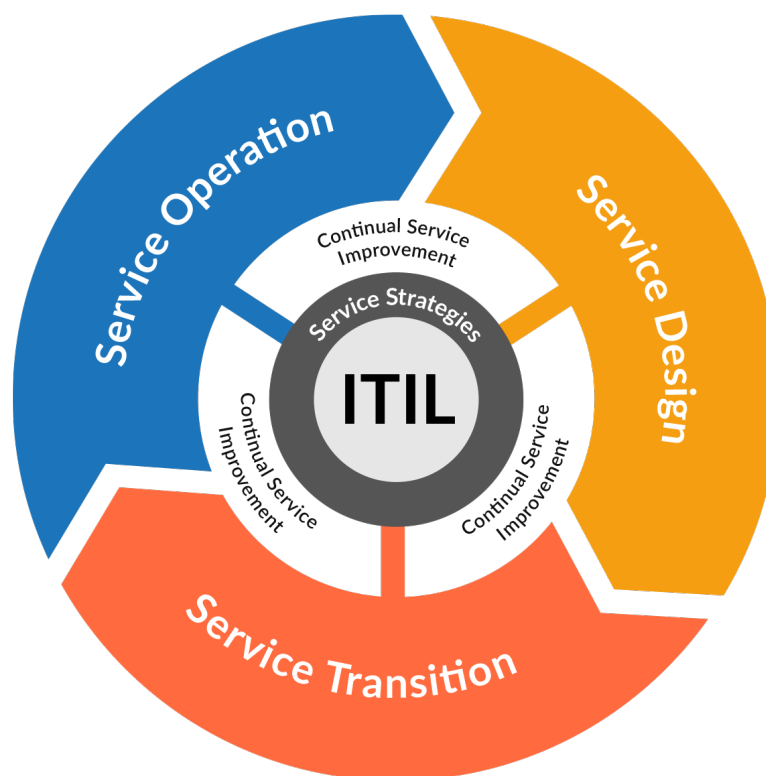
ITIL (engl., Information Technology Infrastructure Library, u nastavku: ITIL), je okvir dizajniran za odabir, planiranje, isporuku, te podršku IT usluge u poslovanju. Glavni cilj je poboljšanje efektivnosti te dostignuće željene razine usluge. ITIL daje smjernice koje upućuju na najbolje prakse koje usklađuju korištenje IT-a s poslovnim potrebama. ITIL-ove smjernice mogu se primijeniti u bilo kojoj organizaciji. (20)

8.1. Komponente ITIL-a

S obzirom da se norma ISO 20000 temelji na ITIL-u, važno je prikazati i objasniti što je ITIL. ITIL je okvir dizajniran za planiranje, isporuku te podršku IT usluga u poslovanju. Glavni cilj je poboljšanje efektivnosti i postavljanje predviđene razine IT usluga. ITIL je okvir koji sadrži smjernice najboljih praksi koje usklađuju IT s poslovnim sustavima i potrebama. ITIL pruža praktične smjernice za bilo koju organizaciju o strategiji i uslugama.

ITIL se sastoji od 5 komponenti, koje čine različite faze životnog ciklusa usluge, a te komponente su sljedeće:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement



Slika 22 ITIL, izvor: https://fiverr-res.cloudinary.com/images/t_main1,q_auto,f_auto/gigs/107521108/original/9a3df94822bd93ddc1fb6ff0ce8c660d32dd8466/design-an-ital-process-for-your-business.png

8.1.1.Strategija usluge

Glavna svrha ove komponente je pružanje strategije životnog ciklusa, koja bi trebala biti usklađena s poslovnim ciljevima kupca, kao i upravljanjem uslugama, uključujući time i njen opseg. Cilj je definirati strategiju i načine upravljanja.

Procesi koji su sadržani u procesu usluge su:

- Upravljanje portfeljem usluge – proces kojim se identificiraju, opisuju i vrednuju usluge.
- Upravljanje potražnjom – odnosi se na razumijevanje i utjecanje na potražnju usluge od strane kupca.
- Upravljanje financijama – proces upravljanja troškovima i prilikama povezanim s financijskim uvjetima.
- Stvaranje strategije – osigurava efektivnu i efikasnu izvedbu usluga, koje ispunjavaju zahtjeve korisnika, rješavaju probleme i obavljanje rutinskih zadataka.

8.1.2. Dizajn usluge

Odnosi se na dizajniranje usluge i potporu svih elemenata koji su važni za uvođenje usluge u okolinu. 4P dizajna usluge, predstavljaju područja na koja se treba obratiti pažnja prilikom dizajniranja usluge, a oni su:

- Ljudi (eng. People) – ljudski potencijali i organizacijske strukture potrebne za podršku uslugama,
- Procesi – procesi upravljanja uslugama potrebni za pružanje podrške uslugama,
- Proizvodi – tehnološka i ostala infrastruktura potrebna za pružanje podrške uslugama,
- Partneri – treće strane koje pružaju dodatnu podršku potrebnu za pružanje podrške uslugama.

Procesi dizajna usluge su:

- Upravljanje katalogom usluge – sadrži sve usluge koje su trenutno dostupne korisnicima
- Upravljanje razinom usluge – proces koji je zadužen za osiguravanje i upravljanje ugovorima između korisnika i pružatelja usluge. Rezultat procesa je dokument SLA (eng, Service Level Agreement), tj. ugovor o razini usluge
- Upravljanje raspoloživosti – proces koji se odnosi na upravljanje i postizanje razine usluge prema SLA
- Upravljanje kapacitetima – bavi se osiguravanjem postojanja troškovno učinkovitog kapaciteta koji zadovoljava ili premašuje SLA.
- Upravljanje kontinuitetom usluge – osigurava da davatelj usluga uvijek može pružiti minimalne dogovorene razine usluge.
- Upravljanje sigurnošću informacija
- Upravljanje odnosima s dobavljačima – proces zadužen za dobivanje vrijednosti za novac.

8.1.3. Tranzicija usluge

Cilj procesa tranzicije usluge je izgradnja i implementacija IT usluga. Nakon što se usluga testira, ista se stavlja u produkcijsku okolinu kako bi bila dostupna korisnicima. Cilj tranzicije usluge jest da se pobrine da se sve promjene u procesima izvođenja usluga i samim uslugama odvijaju na koordiniran način.

Procesi ove faze su:

- Upravljanje promjenama – proces je zadužen za kontrolom, analizom i provedbom svih promjena, a glavni cilj se odnosi na omogućavanje korisnih promjena s minimalnim ometanjem postojećih IT usluga.
- Evaluacija promjena – cilj evaluacije jest izrada analize zaprimljenog zahtjeva za promjenom koji se može odnositi na razvoj i uvođenje nove usluge ili većom izmjenom postojeće usluge
- Upravljanje projektima – proces je usmjeren na planiranje i koordinaciju korištenja resursa za implementaciju razvijene usluge (troškovi, vrijeme, zaposlenici te kvaliteta usluge)
- Razvoj aplikacija – proces koji uključuje razvoj i održavanje programskog rješenja koje je rađeno po narudžbi
- Upravljanje isporukom i implementacijom – procesa planira, raspoređuje i kontrolira kretanje izdanja za testiranje i produkcijsko okruženje. Primarna svrha ovog procesa je osigurati zaštitu cjelovitosti korištenih okolina i ispravak isporučenih komponenti.
- Validacija i testiranje usluge – proces kojim osiguravamo da isporučena usluga zadovoljava zahtjeve korisnika
- Upravljanje znanjem – cilj je prikupiti, analizirati, pohraniti i dijeliti znanja i informacije unutar organizacije, kako bi se poboljšala učinkovitost smanjivanjem potrebe za ponovnim otkrivanjem znanja.

8.1.4.Pružanje usluge

Pružanje usluge odnosi se na dio ITIL-a kojem je cilj ispunjenje korisničkih zahtjeva, rješavanje problema te obavljanje rutinski operativnih zadataka.

Uključuje slijedeće procese:

- Upravljanje događajima – cilj ovog procesa je osigurati da se zahtjevi prate, analiziraju i kategoriziraju, kako bi se odlučilo o daljnjim akcijama.
- Upravljanje incidentima – cilj je upravljati životnim ciklusom svih incidenata, kako bi se IT usluga što prije vratila korisnicima.
- Zadovoljavanje zahtjeva – cilj je ispuniti zahtjeve vezane za usluge, koji su u većini usluga manje promjene ili zahtjevi za informacijama.

- Upravljanje pristupom – cilj je dodjela prava ovlaštenim korisnicima te sprječavanje neovlaštenog pristupa

- Upravljanje problemima – cilj je upravljati životnim ciklusom svih problema, kojemu je svrha spriječiti nastanak incidenata ili smanjivanje utjecaja incidenata koji se ne mogu spriječiti.

- Upravljanje IT operacijama – cilj je nadzirati i kontrolirati IT usluge i njihovu osnovnu infrastrukturu. Postiže se svakodnevnim rutinskim zadacima koji su vezani uz rad infrastrukturnih komponenti i aplikacija.

Upravljanje programskim rješenjima – proces odgovoran za upravljanje programskim rješenjima tijekom njihovog životnog ciklusa.

- Upravljanje tehnikom – pruža tehničku podršku za upravljanje IT infrastrukturom.

8.1.5. Kontinuirano poboljšanje usluge

Cilj ovog procesa je korištenje metoda upravljanja kvalitetom, radi zadovoljenja zahtjeva korisnika te učenja iz prethodnih ciklusa razvoja programskog rješenja. Sam proces je u skladu s prethodno ukratko opisanom normom ISO 20000. Kontinuirano poboljšanje usluge odnosi se na metode i tehnike kojima je cilj poboljšanje i praćenje usluge te podizanje efektivnosti i efikasnosti procesa upravljanja IT usluga u organizaciji.

9. Proces „Upravljanje zahtjevima korisnika“

Razvojni proces aplikativnih rješenja može se sagledati kao poslovni proces neke tvrtke kojoj je primarna djelatnost razvoj informatičkih sustava po zahtjevu korisnika. Svaka tvrtka mora opisati svoje poslovne procese pa tako i proces razvoja informatičkih rješenja. Kako se sustavi ne razvijaju da služe sami sebi, već se razvijaju da ispune potrebe korisnika, važno je u tom procesu pratiti zahtjeve od strane korisnika. Potrebno je omogućiti da se u svakom trenutku može pregledati zahtjev te saznati tko i kada ga je zatražio, u kojem je statusu, zatim koji je prioritet zahtjeva i rok za izvršenjem. Obrada zahtjeva korisnika samo je jedan u nizu procesa u cjelokupnom razvoju aplikativnih rješenja, koji ćemo obraditi u ovom radu. U ovom radu prikazat će se i rezultati dobiveni praćenjem procesa upravljanja zahtjevima koji nije informatiziran (AS IS), i predloženog budućeg (TO BE). Poboľšanja se odnose na upotrebu tehnologije i automatizacije koje u sadašnjem izvođenju procesa nisu u potpunosti iskorištene.

9.1. Opis procesa „Upravljanje zahtjevima korisnika AS IS“

Razni su poticaji za novi zahtjev kao na primjer uvođenje novih ili izmjene postojećih zakona, zatim potreba za informatiziranjem i unapređenjem poslovnih procesa. Dokumenti koji prethode samom zahtjevu su narodne novine, novi zakon, nova uredba, novi pravilnik. Svi ti dokumenti ulaze u proces kreiranja zahtjeva što rezultira samim zahtjevom koji ulazi u proces obrade zahtjeva. Razvoj aplikativnih rješenja sastoji se od više procesa, to su analiza zahtjeva, specifikacija, implementacija i testiranje samog rješenja.

Proces počinje sastankom s korisnikom na kojoj poslovni analitičar prikuplja inicijalne zahtjeve. Nakon završenog sastanka, određuje se kategorija zahtjeva, tj. da li zahtjev spada u funkcionalnu ili nefunkcionalnu kategoriju. Nakon što se odredi kategorija, daljnji korak u procesu odnosi se na određivanje odgovorne osobe za izvršenje tog zahtjeva. Slijedeći koraci su pisanje funkcionalne i tehničke specifikacije zahtjeva. To se obavlja istodobno, pri čemu se pisanje tehničke specifikacije odnosi na izradu prijedloga arhitekture programskog rješenja te izrade podatkovnog modela (tj. baze podataka). Aktivnost pisanja funkcionalne specifikacije u sebi sadrži aktivnosti koje se odnose na opisivanje programskog rješenja, opis koraka izvršavanja funkcionalnosti, izradu prototipa, opisa podatkovnih entiteta te poslovnih pravila. Nakon što su funkcionalna i tehnička specifikacija napisane ista se prezentira korisniku te je

cilj da se korisnik složi sa predloženim specifikacijama. U slučaju da se korisnik ne složi s prezentiranim specifikacijama, cijeli proces ide na početak, tj. ponovno se organizira radionica s klijentom na kojoj se prepravljaju ili usvajaju novi zahtjevi. Ukoliko se korisnik slaže sa specifikacijom, daljnji tijek proces odnosi se na klasične aktivnosti upravljanja projektom – od izrade plana razvoja, određivanje rokova izvršenja, upravljaju dodatnim zahtjevima korisnika te nakon razvoja slijede aktivnosti implementacije razvijenog programskog rješenja te predavanje istog na održavanje.

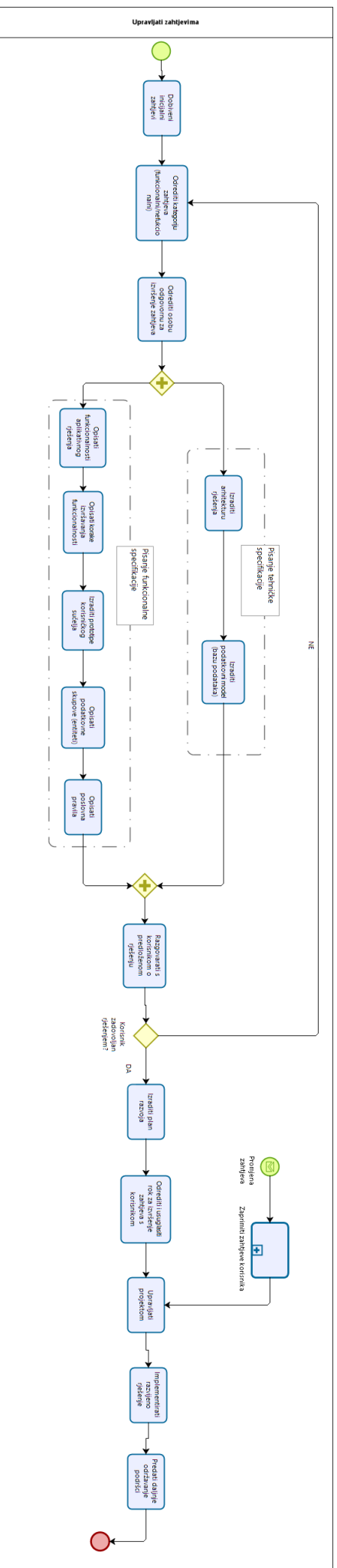
U nastavku se nalazi tablični i slikovni prikaz prethodno opisanog procesa.

9.2. Tablica za proces „Upravljanje zahtjevima AS IS“

Naziv procesa: Upravljanje zahtjevima korisnika AS IS

Podobni događaji		Završni događaji										
Poruka razvojnog aplikacijskog inženjera	Dočuvani zahtjevi inženjera	Opisni zahtjeva (funkcionalni/verifikacijski)	Opisni odgovornu izvršenje zahtjeva	Priručje za specifikacije	Priručje tehničke funkcionalne specifikacije	Razgovor s korisnikom predložbenom inženjeru	Izraditi plan razvojnog projekta	Opisni rok za izvršenje zahtjeva s korisnikom	Upravljanje projektom	Implementirati razvojno inženjeru	Prodati održavanje podrške	Razvijeno implementirano aplikacijsko inženjeru
Izvršitelji	Posloveni analitičar Korisnik	Posloveni analitičar Korisnik	Posloveni analitičar	Arhitekt inženjer Administrativator baze podataka	Posloveni analitičar	Posloveni analitičar	Voditelj projekta Posloveni analitičar	Voditelj projekta Posloveni analitičar	Voditelj projekta	Programsku inženjeru Voditelj projekta	Voditelj projekta Podrška	Izvršitelji
Razlozi za promjenu												
<ul style="list-style-type: none"> Nepostojanje reprezentivna zahtjeva Nemoguća jednostavno predenje statusa zahtjeva Dugotrajni postupak komuniciranja s ostalim službenima vezano uz izvršavanje zahtjeva (telefonom, e-mailom, na hodniku, ...) Potrebna za održanjem procesa obrade zahtjeva korisnika 						Vrijedi novog poslovnog procesa						<ul style="list-style-type: none"> Koristiti reprezentiv (bazu podataka) svih zahtjeva sa statusima, odgovornim osobama, rokovima i svim ključnim podacima Omogućiti brz uvid u sve ključne podatke zahtjeva Automatizirati stanje obavijesti o primku zahtjeva (osobama zaduženim za pojedine aktivnosti prilikom izvršenja zahtjeva, te ostalim službenima po potrebi)
Učesnici		Mehanizmi		Metrika								
Podnosioci zahtjeva (korisnik) Osoba koja preuzima zahtjeva Posloveni analitičar Voditelj projekta Arhitekt inženjeru		Aplikacija za pregled podataka zahtjeva E-mail osobama zaduženim za pojedine aktivnosti prilikom izvršenja zahtjeva Zahtjevi od strane korisnika Zajednici sa sastanka		Trajanje postupka Trajanje postupka								

9.3. Model procesa „Upravljanje zahtjevima korisnika AS IS“



Slika 1 Model procesa (Izvor: izrada autora)

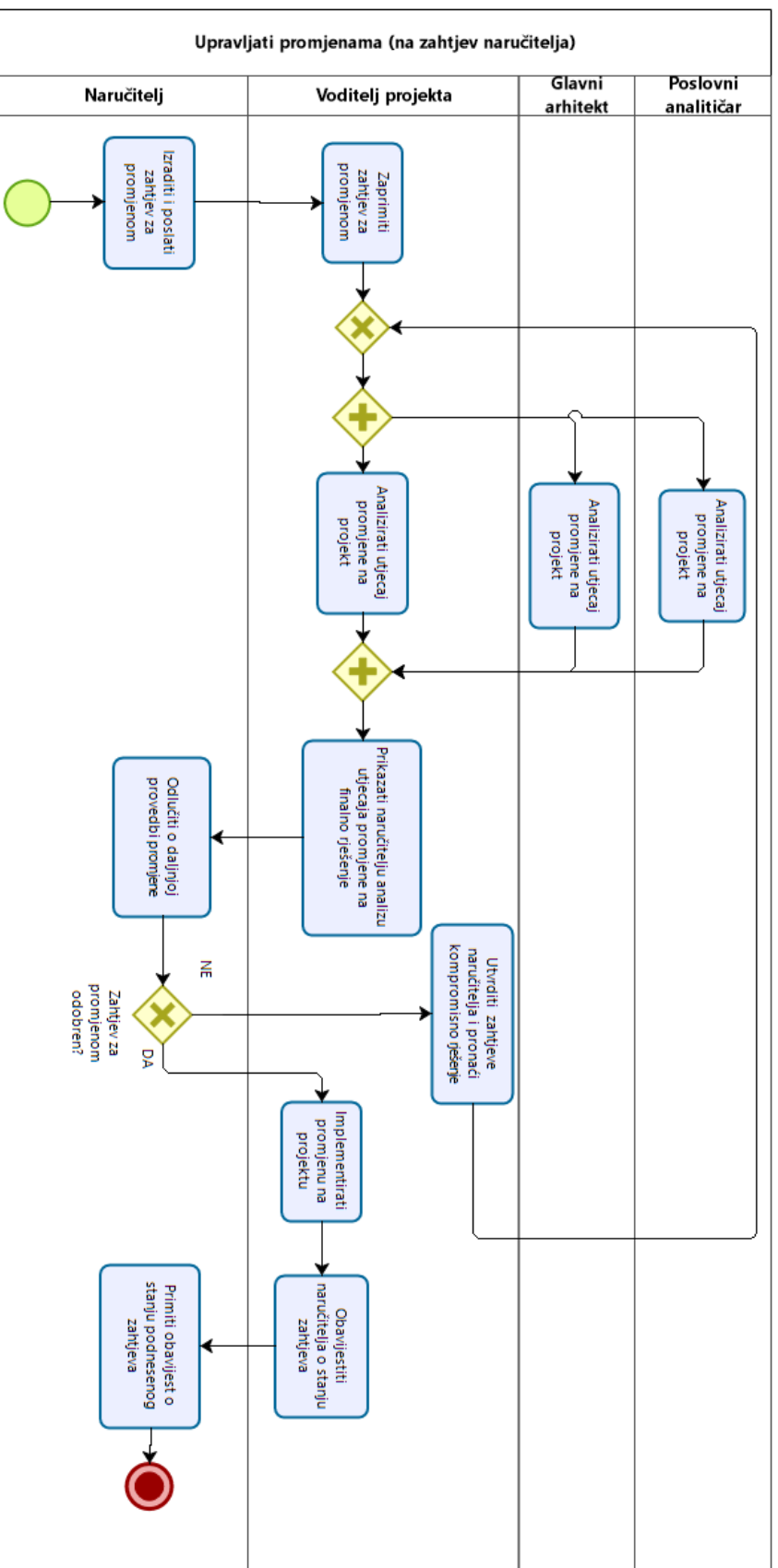
9.4. Opis detaljnog procesa „Zaprimiti zahtjeve korisnika“

U preglednom proces „Upravljanje zahtjevima korisnika“ zasebno je modeliran proces „Zaprimiti zahtjeve korisnika“ iz razloga jer se najčešće radi o zahtjevima korisnika koji se javljaju nakon prihvaćenih specifikacija i u samom tijeku razvoja programskog rješenja. Proces „Zaprimiti zahtjeve korisnika“ ima sveukupno 11 aktivnosti, a započinje kada korisnik zatraži dodatne značajke programskog rješenja koje bi želio da se razviju. U tom slučaju, nakon zaprimanja zahtjeva za novom značajkom ili promjenom već dogovorene, prvo se analizira sam zahtjev od strane poslovnog analitičara, arhitekta rješenja te voditelja projekta. Nakon izrađene analize i novog prijedloga rješenja isto se prikazuje korisniku s kojim se dogovara daljnja provedba tražene promjene/razvoja nove značajke programskog rješenja. S obzirom da je cilj da je korisnik zadovoljan isporučenim razvijenim programskim rješenjem, u slučaju da korisnik nije zadovoljan s analizom i predloženim rješenjem voditelj projekta nastoji pronaći kompromisno rješenje koje vraća proces natrag na korak analize. Ukoliko se korisnik slaže sa predloženom promjenom i analizom zahtjev se implementira u postojeći projekt te se korisnik obavještava o stanju zahtjeva.

9.5. Tablični prikaz detaljnog procesa „Zaprimiti zahtjeve korisnika“

Naziv procesa: Zaprimiti zahtjeve korisnika									
Početni događaj	AKTIVNOSTI						Završni događaj		
	Izraditi i poslati zahtjev za promjenom	Zaprimiti zahtjev za promjenom	Analizirati utjecaj promjene na projekt	Prikazati naručitelju analizu utjecaja promjene na finalno rješenje	Odlučiti o daljnjoj provedbi promjene	Zahtjev odobren? NE: Utvrditi zahtjeve naručitelja i pronaći kompromisno rješenje DA: Implementirati promjenu na projektu		Obavijestiti naručitelja o stanju zahtjeva	Primiti obavijest o stanju podnesenog zahtjeva
Klijent je utvrdio potrebnu novu funkcionalnost								Klijent je zaprimio obavijest	
Izvršitelj	Naručitelj	Voditelj projekta,	Voditelj projekta, Poslovni analitičar, Glavni arhitekt	Voditelj projekta	Voditelj projekta	Naručitelj	Voditelj projekta	Naručitelj	Izvršitelj
Učesnici									
Metrika									
Voditelj projekta, Naručitelj, Poslovni analitičar, Glavni arhitekt								Broj zaprimljenih zahtjeva za promjenom, broj izrađenih analiza, broj odbijenih i odobrenih zahtjeva za promjenom	

9.6. Model detaljnog procesa „Zaprimiti zahtjeve korisnika“



Slika 24 Model procesa (Izvor: izrada autora)

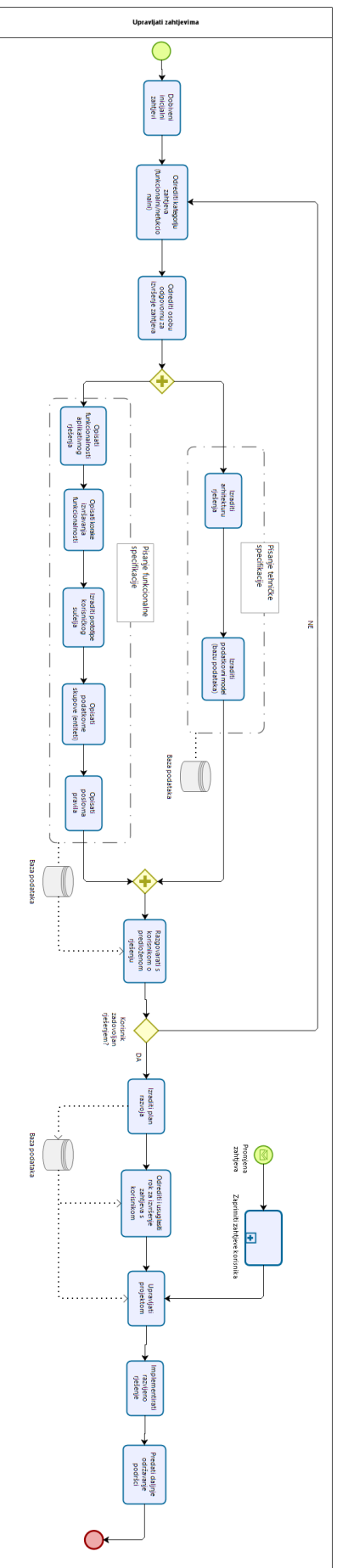
9.7. Poslovni i organizacijski razlozi za poboljšanjem procesa

Proces koji se opisuje iniciran je zahtjevom od korisnika. Nakon što je zahtjev zaprimljen potrebno je odrediti tip zahtjeva, pri tome mislimo na to da li je to zahtjev za izmjenom ili zahtjev za novom funkcionalnošću. Zaprimljeni zahtjev potrebno je registrirati, što trenutno nije podržano nekim repozitorijom, već se zahtjev registrira u tablicu. Osoba koja je zaprimila zahtjev je kontakt prema korisniku i sada je potrebno da se odredi osoba koja će biti zadužena i odgovorna za ostale potrebne aktivnosti nad registriranim zahtjevom. Po potrebi odgovorna osoba usuglašava zahtjev i prioritet zahtjeva sa korisnikom, zatim detaljnije analizira zahtjev ovisno o njegovim specifičnostima. Analiza zahtjeva može pokazati potrebu za aktivnostima neke druge službe u poduzeću, ovisno o detaljnijoj analizi i potrebi za uključivanjem jedne ili više službe određuje se težina samog zahtjeva, pa sukladno težini i rok potreban za izvršenjem. U svakom slučaju potrebno je usuglasiti rok sa korisnikom i na kraju svih aktivnosti obavijestiti podnositelja zahtjeva o prihvaćanju ili odbijanju zahtjeva.

Potrebno je naglasiti da niti jedna od ovih aktivnosti nije automatizirana, već je potrebno razmjenjivati informacije mailom ili usmeno, što usporava proces obrade zahtjeva. Nedostatak je što zahtjev sa svojim podacima nije spremljen u nekom repozitoriju, posebni nedostatak je što se dinamika izmjene statusa zahtjeva ne odvija automatizirano i nije evidentirana u repozitoriju, jer time je onemogućen jednostavan i brz uvid u stanje u koje se zahtjev nalazi. Otežana je i metrika praćenja koliko je zahtjeva zaprimljeno, prihvaćeno ili odbijeno u nekom periodu. To su sve nedostaci koje bi modelom to-be ispravili. Kako bi se uspjelo u implementaciji informacijskih sustava, organizacije ne smiju gledati na informacijsku tehnologiju samo kao automatizaciju starih procesa, nužno je uskladiti tehnologiju sa stvarnim poslovnim potrebama.(11) U raspisivanju modela to-be vodimo brigu oko izmjene i samog procesa ukoliko prepoznamo potrebu, a ne samo automatizacije postojećeg.

S obzirom da se uvodi aplikacija i jedinstvena baza podataka u sam proces, tijekom izvođenja procesa ostaje isti kao i u dosadašnjem stanju. Jedina razlika jest vidljiva u tome što se podaci o zahtjevima spremaju u bazu podataka. U nastavku slijedi prikaz modela procesa „Upravljanje zahtjevima korisnika TO BE“.

9.8. Model procesa „Upravljanje zahtjevima To BE“



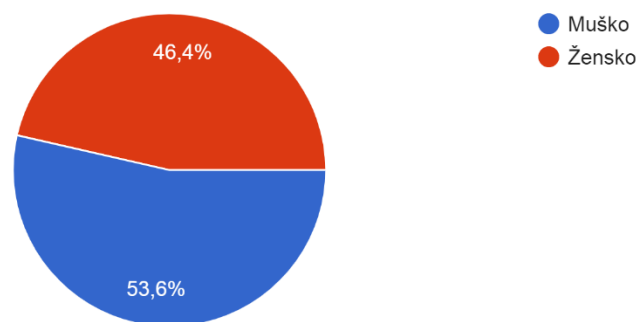
Slika 25 Model procesa TO BE (Izvor: izrada autora)

10. Anketa i analiza rezultata

U sklopu izrade ovog rada provela sam preliminarno istraživanje čija je svrha dobivanje detaljnije slike o načinu upravljanja zahtjevima u procesu razvoja aplikativnog rješenja u IT poduzećima. Anketa sadrži 20-tak pitanja, od kojih su prva 6 opća (ispitivanje o imenu poduzeća, dobnoj skupini i radnom mjestu ispitanika te slično) dok je ostatkom ankete želi ispitati osoban stav ispitanika u vezi upravljanja zahtjevima prilikom razvoja aplikativnog rješenja. Poduzeća koja su sudjelovala u anketi bave se isporukom informatičko-komunikacijskih rješenja. Anketa je prosljeđena na 15-tak e-mail adresa, a istu je ukupno ispunilo 28 ispitanika.

Spol ispitanika:

28 odgovora

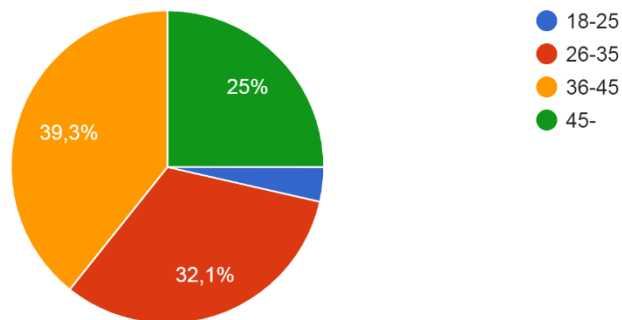


Slika 26 Spol ispitanika (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je da je anketu ispunilo 15 osoba muškog spola te 13 osoba ženskog spola.

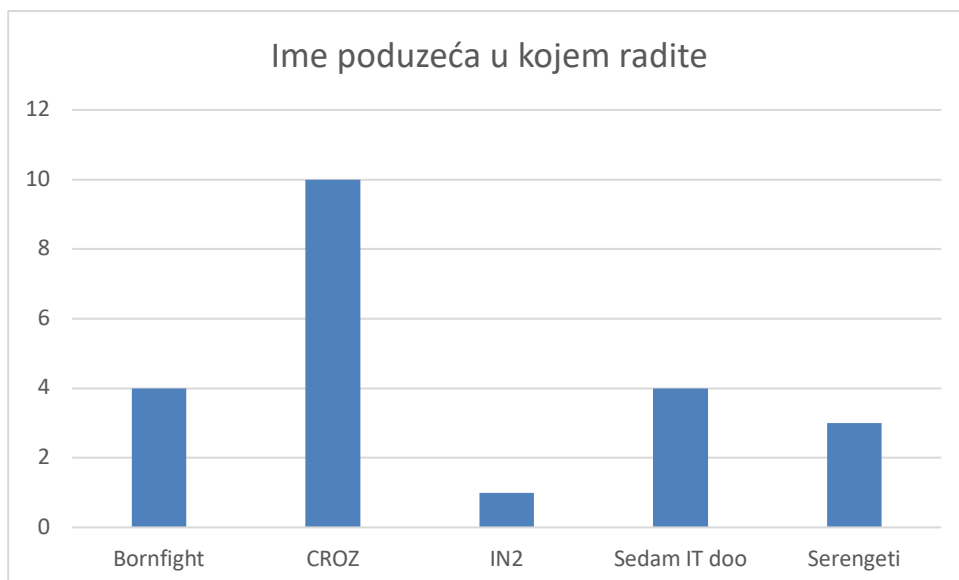
Koliko godina imate?

28 odgovora



Slika 27 Dob ispitanika (Izvor: izrada autora)

Gornji grafikon prikazuje dobne skupine ispitanika po kojem se najviše ispitanika, tj. njih 11 nalazi između 36 i 45 godina. Nakon toga, 9 ispitanika spada u dobnu skupinu od 26 do 35 godina. Nadalje, 7 ispitanika ima 45 i više godina te 1 ispitanik spada u dobnu skupinu od 18 do 25 godina.



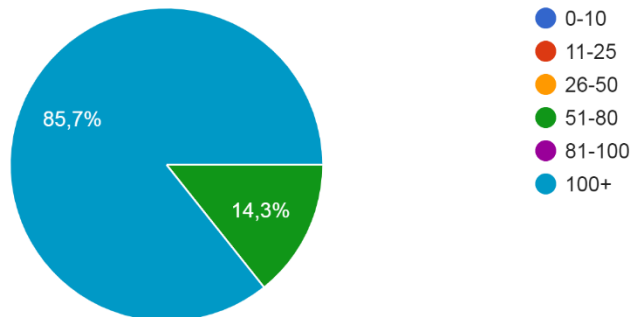
Slika 28 Ime poduzeća (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je kako 10 ispitanika radi u CROZ d.o.o., po 4-ero ispitanika radi u tvrtkama Bornflight i Sedam IT d.o.o.. 3 ispitanika radi u Serengeti te 1 ispitanik radi u

IN2. S obzirom da ovo pitanje nije bilo obavezno za ispuniti, vidljivo je kako pojedini je ovo pitanje ispunilo ukupno 22 od ukupnih 28 ispitanika.

Koliko zaposlenih ima Vaše poduzeće?

28 odgovora

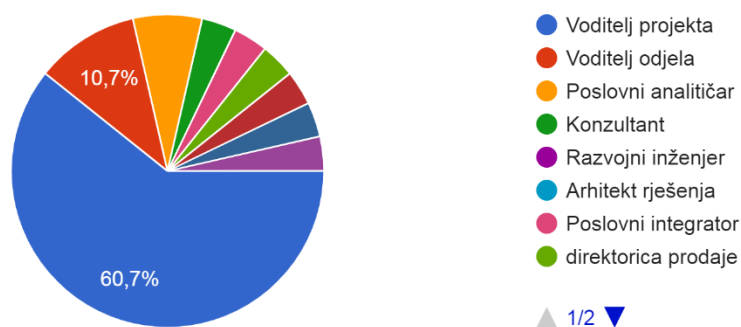


Slika 29 Broj zaposlenika (Izvor: izrada autora)

Gornji grafikon prikazuje da 24-ero ispitanika radi u poduzeću koje ima više od 100 zaposlenih, dok 4 ispitanika radi u poduzeću koje ima 50 do 80 zaposlenika.

Koje je ime Vašeg radnog mjesta?

28 odgovora

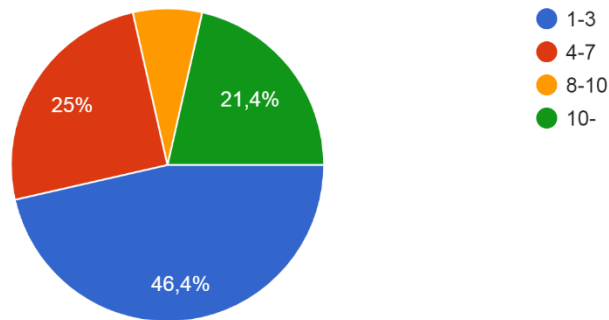


Slika 30 Ime radnog mjesta (Izvor: izrada autora)

Gornji grafikon prikazuje na kojem radnom mjestu ispitanici rade. Najviše ispitanika, tj. njih 17 radi kao voditelji projekata, 3 ispitanika rade kao voditelji odjela. Nadalje najveća slijedeća grupa, tj. 2 ispitanika rade kao poslovni analitičari. Nakon toga imamo ispitanike koji rade kao razvojni inženjeri, arhitekti rješenja poslovni integratori, direktori prodaje, članovi uprave te voditelji programa.

Koliko dugo radite na trenutnom radnom mjestu? (u godinama)

28 odgovora

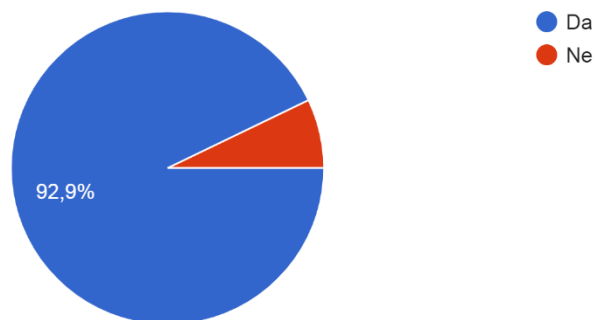


Slika 31 Godine rada ispitanika (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je kako 13 ispitanika na trenutnom radnom mjestu od 1 do 3 godine. Nadalje, slijedeća najveća grupa, tj. 7 ispitanika nalazi se na trenutnom radnom mjestu od 4 do 7 godina, dok je 6 ispitanika na trenutnom radnom mjestu 10 i više godina. Samo jedan ispitanik radi na trenutnom radnom mjestu 8 do 10 godina.

Imate li iskustvo u upravljanju projektima?

28 odgovora

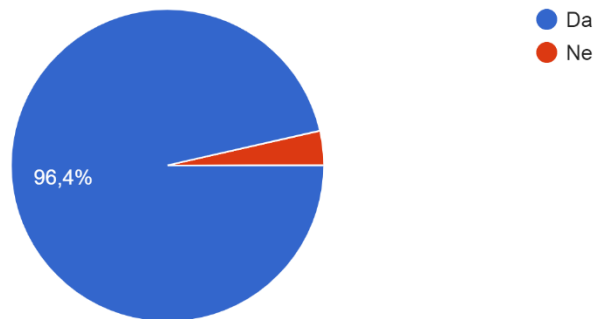


Slika 32 Iskustvo u upravljanju projektima (Izvor: izrada autora)

Iz gornjeg grafikona je vidljivo kako samo 2 od 28 ispitanika nema iskustva u upravljanju projektima.

Planirate li aktivnosti upravljanja zahtjevima kod određivanja projektnog opsega?

28 odgovora

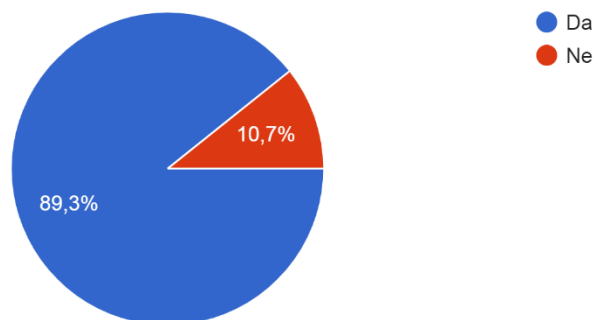


Slika 33 Planiranje aktivnosti upravljanja zahtjevima (Izvor: izrada autora)

Gornjim grafikonom utvrdili smo da prilikom utvrđivanja projektnog opsega projekta samo 1 ispitanik ne planira aktivnosti upravljanja zahtjevima.

Da li Vaše poduzeće razvija ideje/zahtjeve svojih zaposlenika?

28 odgovora

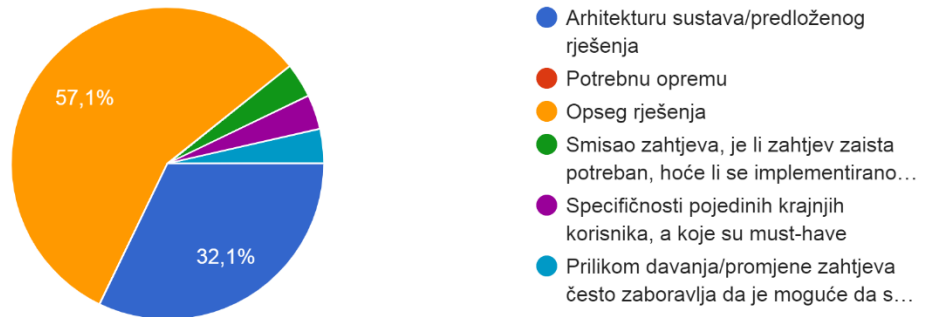


Slika 34 Razvoj ideja zaposlenika (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je kako 25 ispitanika prepoznalo da poduzeće u kojem rade razvija i njihove ideje.

Po Vašem iskustvu, koje uvjete/utjecaje korisnik najčešće zaboravlja prilikom davanja/promjenom zahtjeva?

28 odgovora

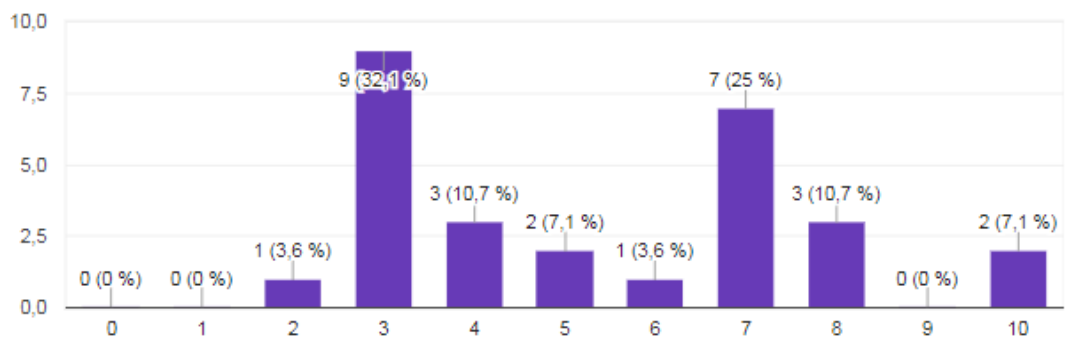


Slika 35 Utjecaji zahtjeva na predloženo rješenje (Izvor: izrada autora)

Detaljnijom analizom prethodnog grafikona utvrđeno je kako korisnici najčešće zaboravljaju na dogovoreni opseg projekta. Nakon toga najčešći uvjet o kojem korisnik ne razmišlja vezan je uz arhitekturu aplikativnog rješenja. U manjoj mjeri korisnik ne uzima u obzir specifičnosti i budžet rješenja koji su vezani uz razvoj aplikativnog rješenja za određeni industrijski sektor. Ujedno, jedan ispitanik je mišljenja kako korisnik ponekad nema smislene zahtjeve te hoće li se implementirano rješenje koristiti.

Prilikom izvođenja projekta, u kojem postotku dolazi do izmjena planiranih aktivnosti razvoja programskog rješenja zbog promjena zahtjeva? (u postocima)

28 odgovora

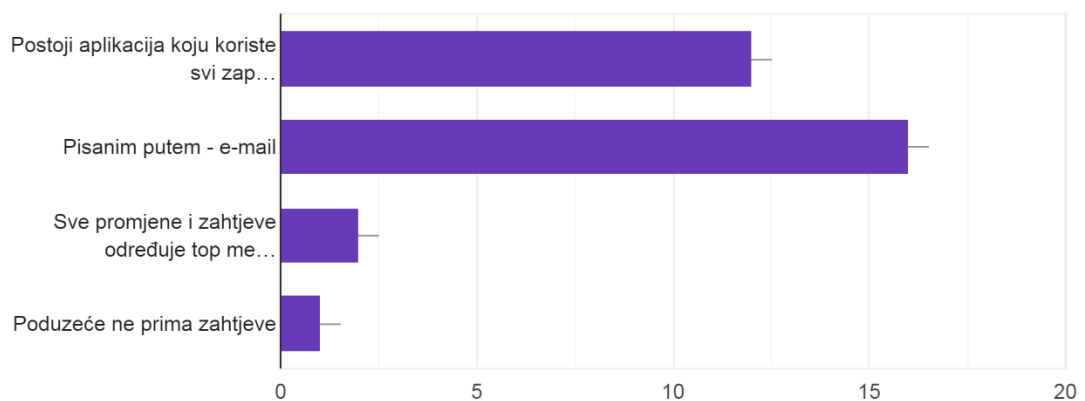


Slika 36 Postotak promjena (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je kako najviše 9 ispitanika smatra da dolazi do 30% promjena u zahtjevima prilikom razvoja aplikacijskog rješenja. 7 ispitanika tvrdi kako dolazi do 70% izmjene planiranih aktivnosti zbog promjene zahtjeva u razvoju programskog rješenja. Po 3-je ispitanika smatra kako dolazi do 40% i 80% izmjene planiranih aktivnosti zbog promjene zahtjeva aplikacijskog rješenja.

Na koji način Vaše poduzeće zaprima i obrađuje interne zahtjeve od strane svojih zaposlenika?

27 odgovora

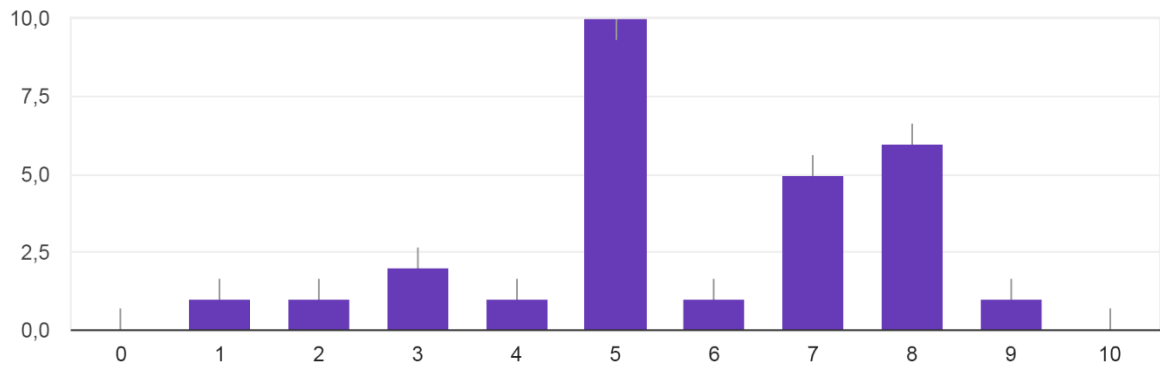


Slika 37 Način obrade zahtjeva (Izvor: izrada autora)

Najviše, tj 16 ispitanika smatra da poduzeće zaprima i obrađuje zahtjeve putem e-maila. Nakon toga 12 ispitanika smatra kako njihova tvrtka koristi aplikaciju koju koriste svi zaposlenici. Ujedno, 2 ispitanika smatra kako sve zahtjeve obrađuje top menadžment dok 1 ispitanik smatra kako poduzeće ne zaprima i obrađuje zahtjeve svojih zaposlenika.

U kojoj mjeri odobravate SVE zahtjeve korisnika?

28 odgovora

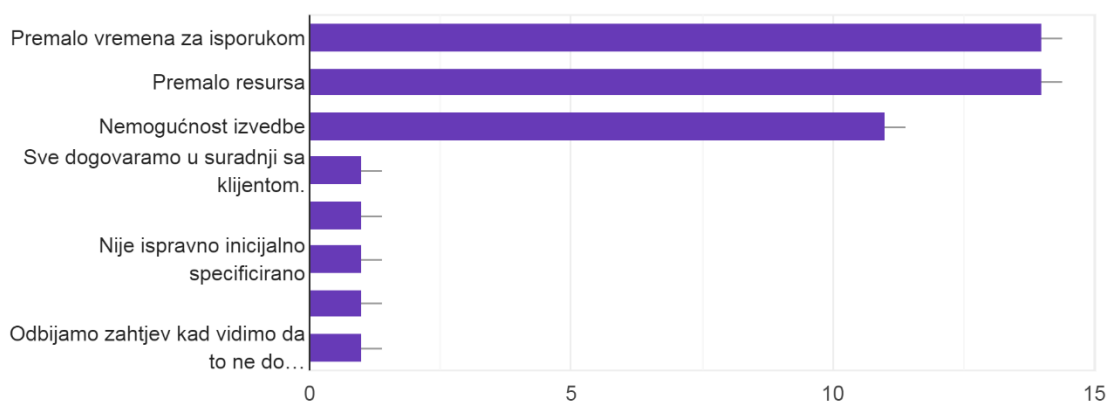


Slika 38 Odobravanje zahtjeva (Izvor: izrada autora)

Gornjim grafikonom vidljivo je kako ispitanici smatraju da u 50% slučajeva moraju odobriti sve zahtjeve korisnika. Nakon toga, 8 zaposlenika smatra da u 80% slučajeva odobre sve zahtjeve korisnika. Nakon toga, 5 ispitanika smatra da se u 70% slučajeva odobravaju svi zahtjevi korisnika. Ostali ispitanici smatraju kako u maloj mjeri dolazi do odobrenja zahtjeva korisnika. To je najčešće moguće ukoliko je na početku projekta jasno i detaljno isplaniran opseg projekta.

U kojim slučajevima najčešće morate odbiti zahtjev?

28 odgovora

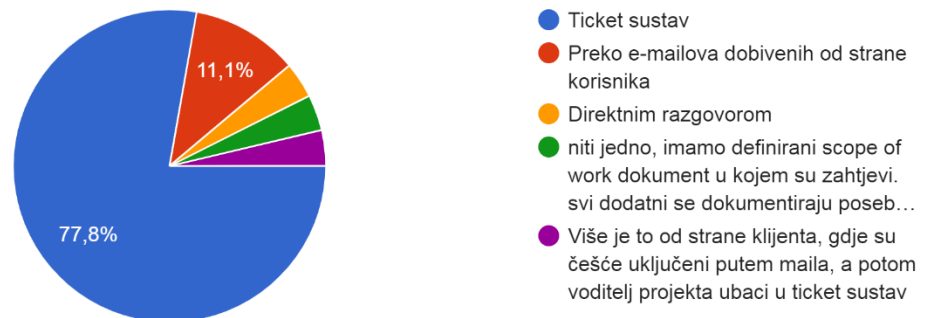


Slika 39 Odbijanje zahtjeva (Izvor: izrada autora)

Iz gornjeg grafa vidljivo je kako su najčešći razlozi za odbijanje zahtjeva premalo vremena za isporukom ili premalo resursa. Nakon toga najčešći razlog za odbijanje zahtjeva korisnika vezan je uz nemogućnost izvedbe. Neki ispitanici smatraju kako sve zahtjeve dogovaraju direktno s klijentom ili kada zahtjevi nisu inicijalno specificirani.

Na koji način pratite zahtjeve prilikom razvoja aplikativnog rješenja?

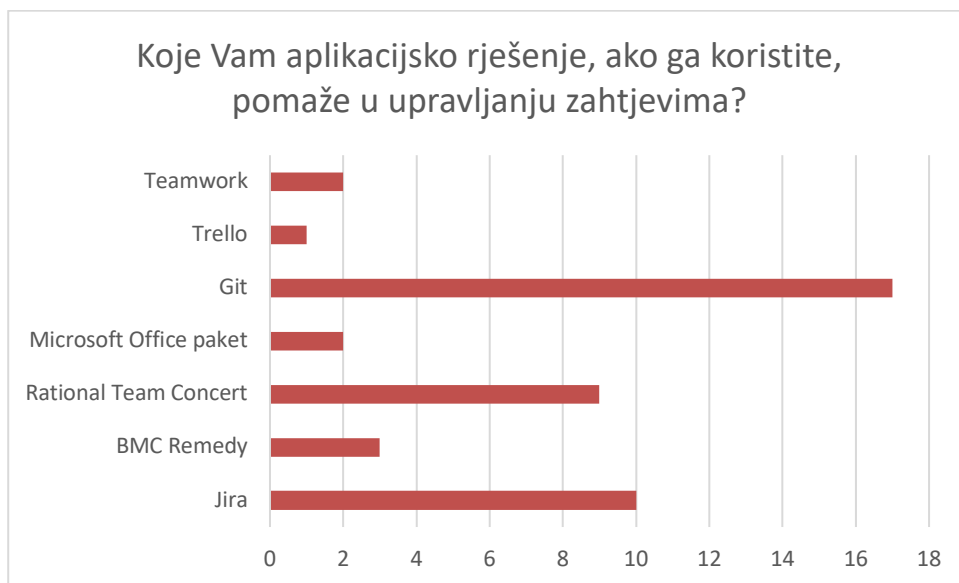
27 odgovora



Slika 40 Praćenje zahtjeva (Izvor: izrada autora)

Prethodni graf prikazuje kako IT poduzeća najčešće koriste ticketing sustav za praćenje zahtjeva prilikom razvoja aplikativnog rješenja. Drugi najčešći način praćenja zahtjeva jest e-mail sustav dok su ostali načini praćenja zahtjeva prilikom razvoja aplikativnog rješenja vezani uz direktan razgovor s korisnikom te dogovorenim opsegom projekta pri kojem se svi dodatni zahtjevi dokumentiraju zasebno.

Slijedeće pitanje u anketi bilo je: „Kako upravljanje zahtjevima utječe na kvalitetu konačnog proizvoda?“. S obzirom da se od ispitanika tražio deskriptivan odgovor, iz rezultata ankete, može se zaključiti kako ispitanici smatraju da se ispravnim upravljanjem zahtjevima povećava kvaliteta i šansa za uspješnim implementiranjem proizvoda. Ujedno, jedan od odgovora navodi kako se ispravnim upravljanjem zahtjevima korisnika utječe na zadovoljstvo korisnika pa je s toga potrebno uspostaviti način rada u kojem su promjene dobrodošle s obzirom da se kreće od pretpostavke da niti jedan korisnik u pravilu na početku ne zna artikulirati sve svoje zahtjeve. Ujedno, jedan ispitanik smatra kako upravljanje zahtjevima negativno utječe na održavanje zbog nelogičnosti i grešaka koji se javljaju u aplikativnom rješenju.

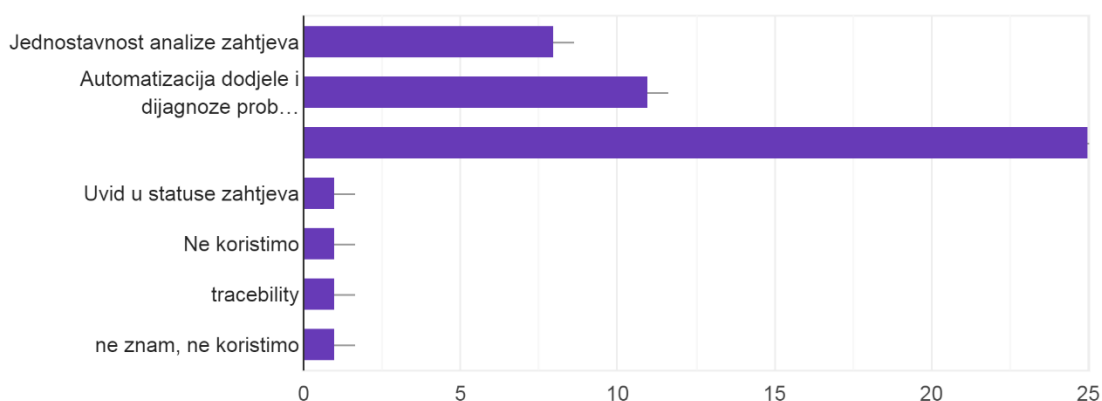


Slika 41 Aplikacijsko rješenje za praćenje zahtjeva (Izvor: izrada autora)

Iz gornjeg grafikona vidljivo je kako najviše ispitanika za upravljanje zahtjevima koristi Git sustav, nakon toga slijedi JIRA te slijedeći najzastupljeniji jest Rational Team Concert. Ujedno, potrebno je naglasiti da se detaljnom analizom odgovora ispitanika može zaključiti kako se najčešće ne koristi samo jedan od sustava nego kombinacija dvoje ili više alata.

Koje su po Vama ključne koristiti upravljanja zahtjevima uz pomoć aplikacije?

28 odgovora

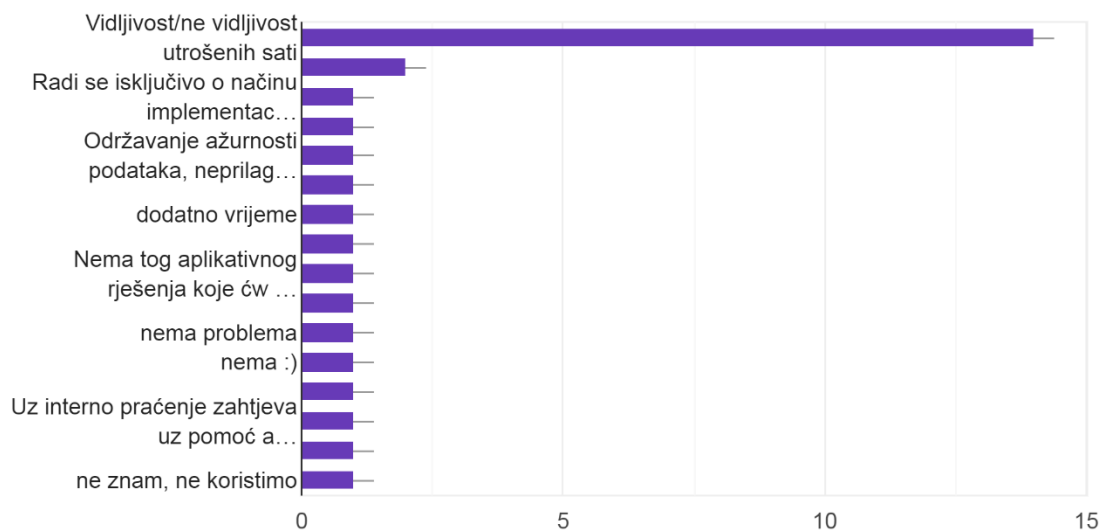


Slika 42 Prednosti upravljanja zahtjeva uz pomoć aplikacije (Izvor: izrada autora)

Gornjim grafom utvrđeno je kako ispitanici vide korist prilikom korištenja aplikacijskog rješenja za upravljanje zahtjevima u centraliziranom mjestu kojim se upravlja svim zahtjevima. Nakon toga slijede automatizacija dodjele i dijagnoze problema, jednostavnost analize zahtjeva. Dvoje ispitanika smatraju kako je glavna korist uvid u statuse zahtjeva te slijednost (tzv. traceability) u upravljanju zahtjeva. Dvoje ispitanika nije moglo odabrati glavne prednosti korištenja aplikacije u procesu upravljanja zahtjevima jer isti ne koriste.

Koji su po Vama problemi upravljanja zahtjevima uz pomoć aplikacije?

28 odgovora



Slika 43 Problemi upravljanja zahtjevima uz pomoć aplikacije (Izvor: izrada autora)

Gornjim grafikonom prikazano je kako ispitanici smatraju da su najveći problemi upravljanja zahtjevima uz pomoć aplikacije vezani uz vidljivost stvarno utrošenih sati razvoja po zadatku. Nakon toga slijedi nedovoljan razvoj aplikacije, tj. neki alati ne podržavaju određene potrebne funkcije poput priloga, ažurnosti podataka, neprilagođenosti aplikacije specifičnom poslovnom procesu upravljanja zahtjevima i slično. Ujedno, jedan ispitanik navodi kako je nemoguće utvrditi tko je odradio zahtjev.

11. Zaključak

U želji za poboljšanjem poslovanja potrebno je „pretresti“ sve poslovne procese koji se odvijaju unutar jedne organizacije. Svi poslovni procesi trebaju biti opisani, dokumentirani kako bi se mogli optimizirati. U organizaciji gdje je primarna djelatnost razvoj informatičkih rješenja potrebno je sagledati i opisati proces razvoja aplikativnih rješenja, kako bi se postiglo skraćivanje razvojnog ciklusa IT proizvoda, smanjili troškovi IT razvoja, smanjili troškovi održavanja aplikacija, što će rezultirati zadovoljnijim korisnicima i boljoj konkurentnosti na tržištu.

Cilj svake organizacije je smanjenje trajanja i pojednostavljenje poslovnih procesa. Važno je sagledati poslovni proces, i jednako je važno opisati poslovne aktivnosti koji se izvršavaju razmjennom raznih dokumenata u poslovnom procesu.

Popis literature

1. Ward J., Peppard J.: Strategic Planning for Information Systems, John Wiley, New York 2002.
2. Innovation and Leadership Business Academy, Zagreb, 13. lipnja 2012., Prof. dr. Josip Brumec, Dr. Slaven Brumec)
3. BPManifesto, 2012 http://www.bptrends.com/manifesto/BPManifesto_EN_A4.pdf.)
4. Dobrović Ž., Brumec J., Tomičić-Pupek K.: Designing the Document Management System: Metamodeling Approach, 16th International Conference on Information and Intelligent Systems – IIS 2005, Varaždin, p.211-220.
5. Business analysis for practitioners BA Standards, Project management Institute
6. A Guide to the Project Management Body of Knowledge (*PMBOK® Guide*)—Fifth Edition, Project management Institute
7. Requirements management : a practice guide.: , Project Management Institute
8. https://en.wikipedia.org/wiki/Business_process_modeling
9. Magistarska teza - Modeliranje poslovnih procesa u razvoju aplikacija elektronskog poslovanja
10. Slaven Brumec, Razvoj mobilnih informacijskih sustava, Magistarski rad
11. Brumec J., Vrček N.: Strategic Planning of Information Systems-A Survey of Methodology, Journal of Computing and Information Technology – CIT 10, 2002, 3 pp 225-231.
12. Leffingwel D., Widring D.: Managing Software Requirements: A Unified Approach, Hardcore, 1999.
13. Raghavan, Zelesnik, & Ford „Lecture Notes of Requirements Elicitation,“ 1994
14. Standard for lightweight IT service management, <https://fitsm.itemo.org/>, pristupano: 28.06.2019
15. Bourque, Fairley : „Guide to Software Engineering Body of Knowledge v 3.0“, 2014.
16. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.
17. K.E. Wiegers, Software requirements, 2nd ed., Microsoft press, 2003.
18. Adesh Chandra, ISO 2000 and ITIL & correlation between them
19. <https://www.fitsm.eu/>

20. https://en.wikipedia.org/wiki/IT_service_management
21. <https://git-scm.com/>
22. <https://www.atlassian.com/software/jira>
23. <https://trello.com/>
24. Prof. dr. Josip Brumec Modeliranje poslovnih procesa Prvi dio: Uvod u modeliranje
<https://koris.hr/preuzmi/koris-uvod-u-modeliranje-poslovnih-procesa.pdf>

Popis slika

Slika 1 Životni ciklus procesa (Izvor: Vlastiti primjer)	11
Slika 2 Modeliranje poslovnih procesa u razvoju aplikacija (Izvor: Magistarska teza - Modeliranje poslovnih procesa u razvoju aplikacija elektronskog poslovanja)	20
Slika 3 Role i aktivnosti u inicijalnoj fazi (Izvor: izrada autora)	25
Slika 4 Proces razvoja zahtjeva, globalni pogled (Izvor: izrada autora)	26
Slika 5 Vodopadni model procesa razvoja programskog rješenja (Izvor: http://en.wikipedia.org/wiki/Waterfall_model)	27
Slika 6 Iterativni model procesa razvoja (Izvor: http://www.agiledata.org/essays/enterpriseArchitectureTechniques.html#UML)	28
Slika 7 Praćenje razvoja zahtjeva (Izvor: Vlastiti primjer)	30
Slika 8 Praćenje razvoja zahtjeva (Izvor: Izrada autora)	32
Slika 9 Primjer Use case dijagrama (Izvor: Izrada autora)	35
Slika 10 Grananje tijekom razvoja softvera (Izvor: izrada autora)	37
Slika 11 Pregled zadataka u Gitu (Izvor: izrada autora)	37
Slika 12 Definiranje zadataka i podzadataka (Izvor: izrada autora)	38
Slika 13 Kreiranje zadataka (Izvor: izrada autora)	38
Slika 14 Kategorija zadatka (Izvor: izrada autora)	39
Slika 15 Jira dashboard (Izvor: izrada autora)	40
Slika 16 Praćenje utrošenog vremena (Izvor: izrada autora)	40
Slika 17 Status zahtjeva (Izvor: izrada autora)	41
Slika 18 Opis zahtjeva (Izvor: izrada autora)	41
Slika 19 Početni zaslon (Izvor: izrada autora)	42
Slika 20 Status zadataka (Izvor: izrada autora)	42
Slika 21 Detalji zadatka (Izvor: izrada autora)	43
Slika 22 ITIL, izvor: https://fiverr-res.cloudinary.com/images/t_main1,q_auto,f_auto/gigs/107521108/original/9a3df94822bd93ddc1fb6ff0ce8c660d32dd8466/design-an-itol-process-for-your-business.png	58
Slika 23 Model procesa (Izvor: izrada autora)	65
Slika 24 Model procesa (Izvor: izrada autora)	68
Slika 25 Model procesa TO BE (Izvor: izrada autora)	70
Slika 26 Spol ispitanika (Izvor: izrada autora)	71
Slika 27 Dob ispitanika (Izvor: izrada autora)	72
Slika 28 Ime poduzeća (Izvor: izrada autora)	72
Slika 29 Broj zaposlenika (Izvor: izrada autora)	73
Slika 30 Ime radnog mjesta (Izvor: izrada autora)	73
Slika 31 Godine rada ispitanika (Izvor: izrada autora)	74
Slika 32 Iskustvo u upravljanju projektima (Izvor: izrada autora)	74
Slika 33 Planiranje aktivnosti upravljanja zahtjevima (Izvor: izrada autora)	75
Slika 34 Razvoj ideja zaposlenika (Izvor: izrada autora)	75
Slika 35 Utjecaji zahtjeva na predloženo rješenje (Izvor: izrada autora)	76
Slika 36 Postotak promjena (Izvor: izrada autora)	76
Slika 37 Način obrade zahtjeva (Izvor: izrada autora)	77
Slika 38 Odobranje zahtjeva (Izvor: izrada autora)	78
Slika 39 Odbijanje zahtjeva (Izvor: izrada autora)	78
Slika 40 Praćenje zahtjeva (Izvor: izrada autora)	79
Slika 41 Aplikacijsko rješenje za praćenje zahtjeva (Izvor: izrada autora)	80
Slika 42 Prednosti upravljanja zahtjevima uz pomoć aplikacije (Izvor: izrada autora)	80

Slika 43 Problemi upravljanja zahtjevima uz pomoć aplikacije (Izvor: izrada autora)..... 81

Upravljanje zahtjevima

Ova anketa provodi se u svrhu izrade specijalističkog rada.

Cilj ove preliminarne ankete jest dobivanje detaljnije slike o načinu upravljanja zahtjevima (korisničkih, internih, implicitnih i eksplicitnih) prilikom razvoja aplikativnog rješenja u IT poduzećima. Prvi dio ankete odnosi se na opće podatke, dok se drugi dio ankete fokusira uključenost ispitanika u projektne aktivnosti. Zadnjim dijelom ankete želi se utvrditi u kojoj mjeri su korisnici upoznati implicitnim i eksplicitnim zahtjevima, kako utječu zahtjevi na IT projekt te način upravljanja zahtjevima.

Potrebno vrijeme da se anketa popuni jest 10 do 15 minuta.

Hvala Vam na ispunjenju!

Opći dio:

Ovim dijelom ankete žele se utvrditi osnovne informacije o ispitaniku.

Spol ispitanika: *

- Muško
- Žensko

Koliko godina imate? *

- 18-25
- 26-35
- 36-45
- 45-

Koliko zaposlenih ima Vaše poduzeće?

- 0-10
- 11-25
- 26-50

Koje je ime Vašeg radnog mjesta? *

- Voditelj projekta
- Voditelj odjela
- Poslovni analitičar
- Konzultant
- Razvojni inženjer
- Arhitekt rješenja
- Ostalo...

Koliko dugo radite na trenutnom radnom mjestu? (u godinama) *

- 1-3
- 4-7
- 8-10
- 10-

Proces upravljanja zahtjevima

Ovim dijelom ankete utvrđuje se utjecaj upravljanja zahtjevima prilikom izrade programskog rješenja.

Imate li iskustvo u upravljanju projektima? *

Da

Ne

Planirate li aktivnosti upravljanja zahtjevima kod određivanja projektnog opsega? *

Da

Ne

Da li Vaše poduzeće razvija ideje/zahtjeve svojih zaposlenika? *

Da

Ne



Po Vašem iskustvu, koje uvjete/utjecaje korisnik najčešće zaboravlja prilikom *
prilikom davanja/promjenom zahtjeva?

- Arhitekturu sustava/predloženog rješenja
- Potrebnu opremu
- Opseg rješenja
- Ostalo...

Po Vašem iskustvu, kako promjene zahtjeva najčešće utječu na izvođenje IT *
projekta?

- Produljuju trajanje i povećavaju trošak resursa
- Smanjuju trajanje i smanjuju trošak resursa
- Ostalo...

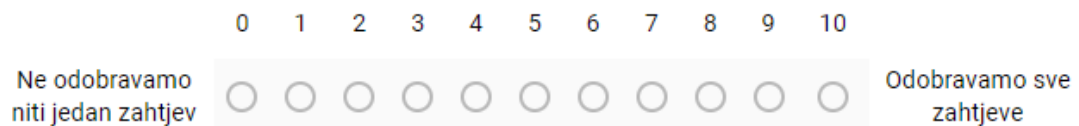
Prilikom izvođenja projekta, u kojem postotku dolazi do izmjena planiranih *
aktivnosti razvoja programskog rješenja zbog promjena zahtjeva? (u
postocima)

	0	1	2	3	4	5	6	7	8	9	10	
0 posto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	100 posto

Na koji način Vaše poduzeće zaprima i obrađuje interne zahtjeve od strane svojih zaposlenika?

- Postoji aplikacija koju koriste svi zaposlenici
- Pisanim putem - e-mail
- Sve promjene i zahtjeve određuje top menadžment
- Poduzeće ne prima zahtjeve

U kojoj mjeri odobravate SVE zahtjeve korisnika?



U kojim slučajevima najčešće morate odbiti zahtjev?

- Premalo vremena za isporukom
- Premalo resursa
- Nemogućnost izvedbe
- Ostalo...

...

Na koji način pratite zahtjeve prilikom razvoja aplikativnog rješenja?

- Ticket sustav
- Preko e-mailova dobivenih od strane korisnika
- Ostalo...

Kako upravljanje zahtjevima utječe na kvalitetu konačnog proizvoda?

Tekst dugog odgovora

Koje Vam aplikacijsko rješenje, ako ga koristite, pomaže u upravljanju zahtjevima?

Tekst kratkog odgovora

Koje su po Vama ključne koristiti upravljanja zahtjevima uz pomoć aplikacije?

- Jednostavnost analize zahtjeva
- Automatizacija dodjele i dijagnoze problema
- Centralizirano mjesto za upravljanje zahtjevima
- Ostalo...

Koji su po Vama problemi upravljanja zahtjevima uz pomoć aplikacije?

- Vidljivost/ne vidljivost utrošenih sati
- Nemogućnost stvarnog utvrđivanja tko je odradio zahtjev
- Ostalo...