

*2023/24*

*PREDMET:*

*SE325 - Upravljanje projektima razvoja softvera*

**Razvoj online prodavnice računarske opreme**

Projektni zadatak 1

Asistent: Student:

**Nebojša Gavrilović David Bondžić 4887**

**Uvod**

U ovom dokumentu biće predstavljene tehnike i načini upravljanja projektima razvoja demonstrirane na online prodavnici računarske opreme. Veoma je važno pripremiti ceo proces kreiranja projekta od početka do kraja i u tome veoma pomažu standardizovane tehnike od kojih će neke biti primenjene ovde:

* Lista uspešnosti projekta (Project Success Chart)
* Struktura poslova WBS (Work Breakdown Structure)
* Lista zaduženja (Task Assignment List)
* Lista rokova (Task Duration List)
* Mrežni plan projekta (Project Network Diagram)
* Gantov dijagram raspored poslova (Grantt Chart Schedule)
* Lista budžeta projekta (Project Budget Chart)

**Apstrakt**

Ovaj projekat ima za cilj razvoj i implementaciju online prodavnice računarske opreme. U digitalnoj eri, potreba za brzim, pouzdanim i jednostavnim načinima nabavke računalne opreme je u stalnom porastu. Online prodavnica će omogućiti korisnicima da pretražuju, porede i kupuju širok spektar računarskih proizvoda, uključujući računare, komponente, periferne uređaje, dodatnu opremu i neke od najkorišćenijih softvera poput Microsoft Office softvera.

Platforma će biti razvijena koristeći najnovije tehnologije i najbolje prakse u e-trgovini, osiguravajući visok nivo sigurnosti, funkcionalnosti i korisničkog iskustva. Funkcionalnosti će uključivati detaljne opise proizvoda, recenzije korisnika, različite opcije plaćanja, kao i brzu i pouzdanu dostavu.

Projekat će se odvijati u nekoliko faza: analiza i planiranje zahteva, dizajn interfejsa i arhitekture sistema, razvoj softverskih komponenti, integracija sa platnim sistemima i logističkim servisima, testiranje i osiguranje kvaliteta, te lansiranje i kontinuirano održavanje sistema. Uspešna realizacija ovog projekta će omogućiti efikasnu i pristupačnu nabavku računalne opreme za korisnike, istovremeno povećavajući dostupnost i konkurentnost na tržištu e-trgovine.

**Lista uspešnosti projekta (Project Success Chart)**

Lista uspešnosti projekta predstavlja jedan od sedam koraka na putu ka uspešno razvijenom projektu, pogotovo prilikom razvoja projekata malih projekata. Ovaj korak podrazumeva korišćenje grafikona, tabela, dijagrama i drugih načina vizualnih prikaza podataka, upravljanjem kojih na adekvatan način pratimo usepešnost razvoja jednog projekta. Ova metoda nam lista faktore prema kojima se meri uspeh ishoda jednog projekta.

Kako bismo adekvatno svaki od ovih faktora uzeli u obzir, prevashodno je od važnosti definisati cilj prijekta sa svim neophodnim funkcionalnostima koje proizvod koji razvijamo mora raspolagati, a prema kojima ćemo i voditi metriku uspešnosti samog projekta.

Konkretizacija rokova i ispunjenja zahteva u odnosu na vremesku odrednicu jeste od ključne važnosti prilikom planiranja uspešnog projekta – samim tim, rangiranje uspešnosti projekta direktno zavisi od očekivanja stejkholdera u odnosu na jedinicu vremena. Tada se u obzir uzimaju njihova očekivanja izrade funkcionalnosti i kvaliteta samog proizvoda, u odnosu na rokove. Zbog toga je važno da svi članovi tima u svakom trenutku budu informisani o očekivanjima stejkholdera koji su jedinstveni za svaki projekat, a koji neposredno utiču na merenje uspeha rukovođenja jednim projektom.

**Primena**

**Cilj projekta:**

Kreiranje online prodavnice računarske opreme koja korisnicima omogućava kupovinu opreme za računare od kuće. Rok za izradu projekta je 3 meseca, preciznije do početka oktobra 2024. godine.

**Isporuka delova softvera:**

1. Na početku projekta potrebno je nakon prikupljanja informacija i zahteva, koje dobijamo iz razgovora sa stejkholderima, da definišemo, analiziramo i validiramo te zahteve. Kao izlaz dobija se SRS dokument. Za ovaj početni deo određuje se da je potrebno 10 dana.
2. Sledeći korak, nakon adekvatno definisanih zahteva jeste projektovanje arhitekture sistema za šta neće biti potrebno više od 5 dana.
3. Nakon što je projektovanje arhitekture završeno i odobreno, počinje faza implementacije sistema, za koju je potrebno 25 radnih nedelja (okvirno 6 meseci). U ovoj fazi se vrši kompletna implementacija svih podsistema uz njihovo kontinualno testiranje i integraciju.
4. Završetkom faze inicijalizacije, sistem se nalazi u alfa fazi. Vrši se alfa testiranje sa predodređenom grupom test korisnika. Testiranje i analiza se vrši iterativno. Nakon faze testiranja koja traje nedelju dana narednih nedelju dana posle toga se vrši popravka potencijalnih defekata sistema. Ovaj proces se ponavlja najviše 3 puta ili dok se ne dostigne zadovoljavajući nivo rada sistema.

**Stejkholderi:**

1. Naručilac softvera
2. Korisnici sistema

**Kriterijumi uspešnosti:**

1. Projekat ispunjava sve zahteve sjektholdera. Svi fukncionalni zahtevi definisani u SRS dokumentu su ispunjeni u fleksibilnost neispunjavanja određenih nefunkcionalnih zahteva.
2. Projekat nije probio definisani budžet.
3. Većina korisnika sistema je zadovoljna korisničkim iskustvom.
4. Sistem je stabilan i ima brz odziv.
5. Naručilac softvera brzo pridobija nove korisnike na sistem.

**Struktura poslova WBS (Work Breakdown Structure)**

Svakom projektu razvijanom na nivou tima bitna je organizacija zadataka koji se tiču razvoja samog proizvoda. Ovo podrazumeva da se rad na jednoj funkcionalnosti pojednostavljuje na veliki broj manjih zadataka od kojih se sačinjava lista zadataka. Svi jednostavniji zadaci jesu delovi veće funkcionalnosti za čiju je izradu predodređen rok ka stejkholderima, i koji treba estimirati i komunicirati transparentno, sa uračunatim rizikom i uz dodatno pridavanje pažnje izvornim zahtevima stejkholdera.

**Primena**

Da bi isporučili naš sistem, potrebno je veće celine sistema podeliti na više manjih delova.

1. **Planiranje**
   1. Definisanje cilja
   2. Prikupljanje stejkholdera
   3. Prikupljanje zahteva
   4. Specifikacija zahteva
   5. Definisanje budžeta
   6. Formiranje tima
   7. Definisanje rokova
2. **Implementacija**
   1. Razvoj prototipa (opciono)
   2. Razvoj arhitekture sistema
   3. Razvoj baze podataka
   4. Razvoj korisničkog interfejsa
   5. Razvoj funckionalnosti
3. **Testiranje**
   1. Testiranje funckionalnosti
   2. Testiranje korisničkog interfejsa
   3. Alfa testiranje
   4. Beta testiranje
4. **Validacija**
   1. Ispravljanje grešaka
5. **Produkcija**
   1. Isporuka sistema na tržište
6. **Održavanje**
   1. Anketiranje krajnjih korisnika
   2. Dodavanje novih zadataka
   3. Ispravljanje grešaka

**Lista zaduženja (Task Assignment List)**

Raspodela odgovornosti ima veliki značaj u procesu razvoja softvera. Ovo podrazumeva konkretnu i transparentnu podelu poslova među članovima tima. Konkretnost i transparentnost kao dva stuba raspodele zadataka postoje radi preciznosti obavljanja posla, kao i međusobnoj informisanosti članova tima čija važnost dolazi do izražaja u trenucima spajanja jednostavnih zadataka u jednu opštu funkcionalnost samog proizvoda.

Vođa tima ima obavezu raspodele konkretnih zadataka pojedincima, i to u skladu sa njihovim znanjem, iskustvom i nivoom senioriteta. Važan uticaj na raspodelu poslova i listu zaduženja imaju i takozvani doušnici, odnosno posrednici u transferu znanja koje se tiče obavljanje samih projektnih zadataka. Ovi pojedinci zaduženi su za prenos korisnih informacija od strane razvoja drugog dela softvera koji može biti od pomoći onima koji razvijaju neki drugi segment proizvoda, kao i pružati neophodna pojašnjenja kojim se dobija na vremenu i efikasnosti projekta.

Veliki deo odgovornosti nose pojedinci koji listom zaduženja dobiju zadatak da odobre isporučeni deo proizvoda. Ponekad se ovaj zadatak ostavlja stejkholderima, medjutim, potrebno je uveriti se u kvalitet isporučenog segmenta proizvoda pre samog predstavljanja funkcionalnosti stejkholderima odnosno krajnjim korisnicima.

**Primena**

Prilikom formiranja tima treba uzeti u obzir obim posla, budžet i rok projekta kako bi se formirao idealan tim. Na ovom projektu radi 5 ljudi, svako sa svojom ulogom.

David Bondžić – Project Leader

Aleksa Aleksić – Backend Developer

Marko Marković – Frontend Developer

Petar Petrović – Test Inženjer

Luka Lukić – Analitičar Zahteva

U sledećoj tabeli možemo videti sve zadatke rapoređene po članovima tima:

| **Zadatak** | **Vlasnik Zadataka** | **Podrška** | **Pružalac Informacija** | **Odobrenje** |
| --- | --- | --- | --- | --- |
| Definisanje cilja projekta | David Bondžić | Business Analyst |  |  |
| Prikupljanje zahteva | Luka Lukić | Projekt Menadžer |  | David Bondžić |
| Specifikacija zahteva (SRS) | Luka Lukić | Petar Petrović |  | David Bondžić |
| Definisanje budžeta | David Bondžić | Luka Lukić |  |  |
| Formiranje tima | David Bondžić |  |  |  |
| Definisanje rokova | David Bondžić | Luka Lukić |  |  |
| Dizajn arhitekture sistema | Aleksa Aleksić | Marko Marković | Luka Lukić | David Bondžić |
| Razvoj baze podataka | Aleksa Aleksić |  | Luka Lukić | Petar Petrović |
| Razvoj korisničkog interfejsa | Marko Marković | Aleksa Aleksić | Luka Lukić | Petar Petrović |
| Razvoj funkcionalnosti | Aleksa Aleksić | Marko Marković | Luka Lukić | Petar Petrović |
| Implementacija API integracija | Aleksa Aleksić | Marko Marković | Luka Lukić | Petar Petrović |
| Razvoj prototipa (opciono) | Marko Marković | Aleksa Aleksić | Luka Lukić | David Bondžić |
| Unitarno testiranje | Petar Petrović | Aleksa Aleksić |  |  |
| Integraciono testiranje | Petar Petrović | Aleksa Aleksić |  |  |
| Sistematsko testiranje | Petar Petrović | Marko Marković |  |  |
| Alfa testiranje | Petar Petrović | Aleksa Aleksić | Testeri | David Bondžić |
| Beta testiranje | Petar Petrović | Aleksa Aleksić | Testeri | David Bondžić |
| Ispravljanje grešaka | Petar Petrović | Developeri | Luka Lukić | David Bondžić |
| Isporuka sistema na tržište | David Bondžić | Developeri | Luka Lukić |  |
| Anketiranje krajnjih korisnika | Luka Lukić |  | Korisnici | David Bondžić |
| Dodavanje novih funkcionalnosti | Aleksa Aleksić | Marko Marković | Luka Lukić | David Bondžić |
| Održavanje sistema | Aleksa Aleksić |  | Korisnici | Petar Petrović |

**Lista rokova (Task Duration List)**

Lista rokova jeste faktor koji u velikom delu utiče na vreme izrade funkcionalnosti na softveru, a u krajnjoj liniji, vreme izrade kompletnog proizvoda. Organizacijski gledano, ova lista mora do detalja ispratiti listu zahteva o kojoj smo govorili u prethodnom pasusu. Kako bi se izbeglo nerealno ubrzavanje ili usporavanje samog razvoja, važno je realno estimirati odnosno predvide ti potrebno vreme izrade svake od funkcionalnosti. U velikoj meri će posao olakšati lista zahteva i “razbijanje” tih zahteva na manje celine – na ovaj način, developer lakše pravi predikciju o vremenu potrebnom za izradu svakog zahteva zasebno.

Okvirno predviđanje vremena potrebnog za izradu softverskog proizvoda developeri i stejkholderi definišu uzevši u obzir:

1. Optimistično vreme
2. Verovatno vreme
3. Pesimistično vreme
4. Ispostavljeno vreme

Ovo znači da vreme izrade predstavlja varijablu koju nije uvek moguće precizno definisati, te se u obzir uzima da optimični vremenski okviri mogu biti ostvareni u 10-20% slučajeva, dok se verovatno vreme ostvaruje u 50% slučajeva, a pesimistične estimacije vode od 80% do 90% uspeha što dodatno potrđuje činjenicu da veliki broj projekata “probija rokove” zbog nedovoljno realne esitmacije vremenskog okvira.

Sledeća tabela prikazuje okvirno vreme izvršavanja svakog zadatka, pokazano u danima:

| **Zadatak** | **Optimisticno vreme** | **Najverovatnije vreme** | **Pesimistično vreme** | **Ispostavljeno vreme** |
| --- | --- | --- | --- | --- |
| Definisanje cilja projekta | 1 | 2 | 3 |  |
| Prikupljanje zahteva | 2 | 4 | 6 |  |
| Specifikacija zahteva (SRS) | 3 | 5 | 7 |  |
| Definisanje budžeta | 1 | 2 | 3 |  |
| Formiranje tima | 1 | 2 | 3 |  |
| Definisanje rokova | 1 | 2 | 3 |  |
| Dizajn arhitekture sistema | 2 | 4 | 6 |  |
| Razvoj baze podataka | 4 | 6 | 8 |  |
| Razvoj korisničkog interfejsa | 5 | 7 | 10 |  |
| Razvoj funkcionalnosti | 15 | 20 | 25 |  |
| Implementacija API integracija | 5 | 7 | 10 |  |
| Razvoj prototipa (opciono) | 5 | 7 | 10 |  |
| Unitarno testiranje | 2 | 3 | 5 |  |
| Integraciono testiranje | 2 | 3 | 5 |  |
| Sistematsko testiranje | 3 | 5 | 7 |  |
| Alfa testiranje | 2 | 3 | 5 |  |
| Beta testiranje | 2 | 3 | 5 |  |
| Ispravljanje grešaka | 5 | 7 | 10 |  |
| Isporuka sistema na tržište | 1 | 2 | 3 |  |
| Anketiranje krajnjih korisnika | 1 | 2 | 3 |  |
| Dodavanje novih funkcionalnosti | 5 | 7 | 10 |  |
| Održavanje sistema | 2 | 3 | 5 |  |

**Mrežni plan projekta (Project Network Diagram)**

Nakon definisanja konkretnih zadataka i vremena potrebnog za njihovu implementaciju, važno je isplanirati redosled faza razvoja softvera. Ovo uključuje uzimanje u obzir međusobnih odnosa faza i njihove međuzavisnosti. Mrežni plan projekta definiše prioritete u izradi funkcionalnosti, posebno kada se faze razvoja nadovezuju jedna na drugu.

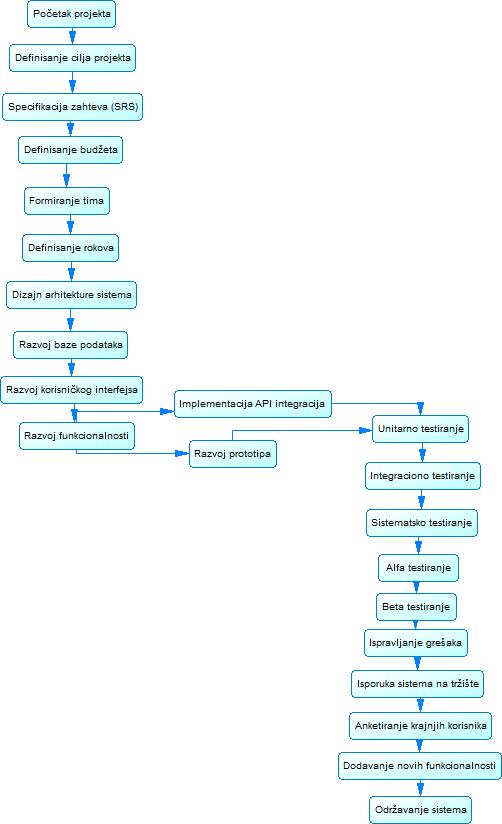
Važno je grupisati srodne i paralelne zadatke kako bi zadaci koji zavise od njih bili obavljeni u narednoj fazi razvoja. Nezavisni zadaci mogu se obavljati paralelno, dok zadaci koji zavise od drugih moraju čekati na njihovo prethodno izvršenje.

Mrežni plan projekta pruža uvid u srodne zahteve koji formiraju „linije života“, koje razvijaju različite funkcionalnosti zasebno, a na kraju se povezuju u celovit softverski proizvod. Najvažnija je „kritična linija života“, koja je najduža i najkompleksnija putanja na dijagramu projekta, i esencijalno zavisi od uspeha softverskog proizvoda. Ostale putanje, ili „linije snabdevanja“, podržavaju celokupni proizvod, ali bez kritične linije, softver ne može biti isporučen.

Mrežni plan takođe zavisi od liste rokova i zadataka. Važno je pronaći i grupisati sve neophodne funkcionalnosti kako bi kritična linija bila što konkretnije definisana, s posebnim fokusom na kolonu verovatnog vremena, kako bi najduža linija na projektu bila optimalno predviđena.

**Primena**

Primenom mrežnog plana projekta daje nam na uvid koja komponenta zavisi od razvoja drugih, koje komponente se mogu paralelno razvijati kao i uvid u vremenski period realizacije komponenata.

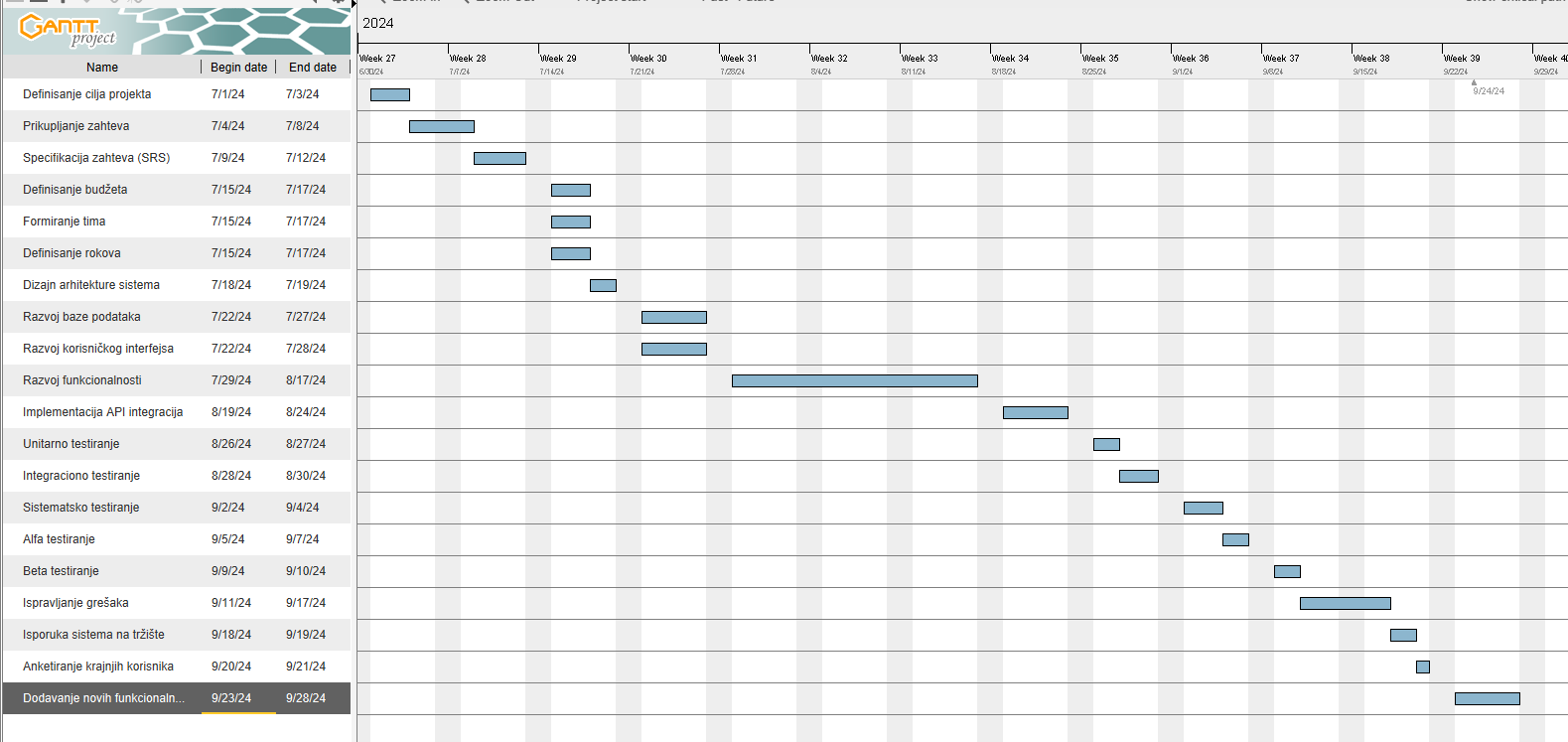


**Gantov dijagram rasporeda poslova (Grantt Chart Schedule)**

Ova tehnika se striktno fokusira na menadžment vremena kao važnog ograničavajućeg faktora i resursa. Osim predstavljanja podele poslova, važno je estimacijom odrediti dužinu trajanja faza projekata, u okviru kojih će se odvijati razvoj zactanih tokova razvoja funkcionalnosti jednog softverskog proizvoda.

Gantov dijagram, odnosno Gantogram, predstavlja grafički prikaz rokova sa analizom i podelama zaduženja u okviru jednog tima. Ovo znači da Gantov dijagram mora biti u koheziji sa svim ostalim tehnikama planiranja – listom zahteva, listom rokova, podelom zadataka itd. Ova tehnika nam pomaže da sve resurse rasporedimo optimalno u odnosu na konkretne vremenske rokove koji su nam omogućeni. Na taj način se sama estimacija potrebnog vremena precizira, te menadžerima ostaje dodatnog prostora i za vreme baferovanja, odnosno „kupovina dodatnog vremena“ koje se čuva za krajnje kvalitetnije testiranje proizvoda.

**Primena**



**Lista budžeta projekta (Project Budget Chart)**

Odgovornost planiranja budžeta jednog projekta pripada menadžerima koji u skladu sa načinom raspolaganja resursima – ljudima, vremenom, prihodom – dolaze do optimalnog budžetiranja jednog projekta. Ukoliko utvrde da je projekat neisplativ, da neće povratiti uloženo uz dodatu vrednost, odbaciće projekat. Prilikom sastavljanja liste budžeta projekta, potrebno je voditi računa o sinergiji svih tehnika planiranja iz prethodnih faza planiranja projekta, a pogotovo je važno uzeti u obzir Gantov dijagram sa listom rokova.

Kao i vreme, tako i troškovi mogu biti optimistični, verovatni, pesimistični i ispostavljeni. Zbog toga se određenom formulom mora doći do „budžetske kontigencije“ koja podrazumeva krajnji ishod budžeta sa uračunatim faktorima koji na njega utiču. To znači da od svih zadataka oduzmemo polovinu pesimističnih i verovatnih troškova, te ovu sumu saberemo sa sumom najverovatnijih troškova. Tada dobijamo najpribližniju moguću vrednost izračunatu prema Goldratt aproksimaciji troškova.

**Primena**

Sledeća tabela predstavlja troškove po zadatku, gde su cene predstavljene u evrima. Kao I kod sati izrade zadataka, cena je predstavljena optimistički, najverovatniji troškovi, pesimistički troškovi I ispostavljeni troškovi, gde će kolona za ispostavljene troškove ostati prazna kako smo još u fazi planiranja projekta.

| **Zadatak** | **Optimizovani troškovi (EUR)** | **Najverovatniji troškovi (EUR)** | **Pesimistični troškovi (EUR)** | **Ispostavljeni troškovi (EUR)** |
| --- | --- | --- | --- | --- |
| Definisanje cilja projekta | 400 | 500 | 600 |  |
| Prikupljanje zahteva | 800 | 1,000 | 1,200 |  |
| Specifikacija zahteva (SRS) | 1,200 | 1,500 | 1,800 |  |
| Definisanje budžeta | 200 | 300 | 400 |  |
| Formiranje tima | 200 | 300 | 400 |  |
| Definisanje rokova | 200 | 300 | 400 |  |
| Dizajn arhitekture sistema | 1,600 | 2,000 | 2,400 |  |
| Razvoj baze podataka | 2,400 | 3,000 | 3,600 |  |
| Razvoj korisničkog interfejsa | 2,800 | 3,500 | 4,200 |  |
| Razvoj funkcionalnosti | 16,000 | 20,000 | 24,000 |  |
| Implementacija API integracija | 2,800 | 3,500 | 4,200 |  |
| Unitarno testiranje | 1,200 | 1,500 | 1,800 |  |
| Integraciono testiranje | 1,200 | 1,500 | 1,800 |  |
| Sistematsko testiranje | 2,000 | 2,500 | 3,000 |  |
| Alfa testiranje | 1,200 | 1,500 | 1,800 |  |
| Beta testiranje | 1,200 | 1,500 | 1,800 |  |
| Ispravljanje grešaka | 3,200 | 4,000 | 4,800 |  |
| Isporuka sistema na tržište | 800 | 1,000 | 1,200 |  |
| Anketiranje krajnjih korisnika | 800 | 1,000 | 1,200 |  |
| Dodavanje novih funkcionalnosti | 2,800 | 3,500 | 4,200 |  |
| Održavanje sistema | 1,200 | 1,500 | 1,800 |  |
| **Ukupno** | **45,400** | **56,400** | **67,600** |  |

**Zaključak**

**Navedne tehnike su se pokazale kao veoma važne pri rukovođenju ovim projektom i njegovim inicijalnim planiranjem. Neke od tehnika kao što je na primer gantogram i Project Network Diagram su pružile dobar uvid u to koji zadaci mogu i kao su isplanirani za paralelizaciju. U svakom slučaju je iskutvo kreiranja ove dokumentacije doprinelo boljem razumevanju svrhe i posla projektnog menadžera.**

**Literatura**

* LAMS
* Dohn Kissinger, MBA, PhD, PMP, A Step-by-Step Approach for Planning Your Small Project