

SAŽETAK TUTORIJALA - AI SUMMARY

OSNOVE INFORMACIONIH SISTEMA

Sadržaj

1	Tutorijal 1: Uvod u Informacione Sisteme	4
1.1	Definicija sistema	4
1.2	Podaci i informacije	4
1.3	Definicije informacionog sistema	4
1.4	Karakteristike informacionog sistema	4
1.5	Cilj izgradnje IS-a	5
1.6	Tim za razvoj IS-a	5
1.7	Strategije konkurentske prednosti	6
1.8	Životni ciklus općih sistema (GSLC)	6
1.9	Životni ciklus informacionih sistema (ISLC)	6
1.10	Dimenzije zastarjevanja IS-a	7
1.11	Software Development Life Cycle (SDLC)	7
2	Tutorijal 2: SDLC - Faza Planiranja	8
2.1	Opis faze planiranja	8
2.2	Šta se radi u fazi planiranja SDLC-a	8
2.2.1	1. Identifikacija poslovnog problema ili prilike	8
2.2.2	2. Definisanje ciljeva projekta	8
2.2.3	3. Analiza zahtjeva zainteresiranih strana (Stakeholder analysis)	8
2.2.4	4. Izrada poslovnog slučaja i studije izvodljivosti	9
2.2.5	5. Definisanje opsega projekta (Scope definition)	9
2.2.6	6. Planiranje resursa i budžeta	9
2.2.7	7. Izrada vremenskog rasporeda (Project Schedule)	9
2.2.8	8. Analiza rizika (Risk analysis)	9
2.2.9	9. Izrada projektne povelje (Project Charter)	9
2.3	Zahtjev za sistemom / Analiza izvodljivosti	9
2.3.1	Identifikacija zahtjeva za sistemom	9
2.3.2	Analiza izvodljivosti	9
2.3.3	Dokumentovanje i validacija	10
2.4	Software Requirements Specification (SRS)	10
2.4.1	Šta je SRS dokument?	10
2.4.2	Sadržaj dokumenta	10
2.5	Upoznavanje sa organizacijom / kompanijom	11
2.6	Misija organizacije / kompanije	11
2.7	Vizija organizacije / kompanije	11

2.8	Ciljevi organizacije / kompanije	11
2.9	Stakeholderi	11
2.10	Zahtjev za sistemom	12
2.11	Ciljevi projekta	12
2.12	Ekonomska izvodljivost - Finansijski pokazatelji	12
2.12.1	Return on Investment (ROI)	12
2.12.2	Break-Even Point (BEP)	12
2.12.3	Payback Period (PB)	12
2.12.4	Present Value (PV) i Net Present Value (NPV)	13
3	Tutorijal 3: Upravljanje Projektom	13
3.1	SDLC faza - Planiranje (detaljno)	13
3.2	Aktivnost i projekat	14
3.3	Često korišteni projektni termini	14
3.4	Upravljanje projektom	14
3.5	Projekt menadžer i projektni tim	15
3.6	Plan projekta	15
3.7	Work Breakdown Structure (WBS)	15
3.8	Gantt Chart (Gantogram)	16
3.9	Upravljanje osobljem	16
3.10	Mrežno planiranje	16
3.10.1	Proračun naprijed	17
3.10.2	Proračun nazad	17
3.11	Critical Path Method (CPM)	17
3.12	DiSC Profili	17
3.13	Iron Triangle	18
3.14	Upravljanje kvalitetom	18
3.14.1	Koncepti upravljanja kvalitetom	18
3.14.2	PDCA / Deming Cycle	18
4	Tutorijal 4: skip	19
5	Tutorijal 5: skip	19
6	Tutorijal 6: Analiza zahtjeva	19
6.1	Definicija analize zahtjeva	19
6.2	4 koraka analize zahtjeva	19
6.3	Metode prikupljanja zahtjeva	19
6.3.1	Tradicionalne metode	19
6.3.2	Savremene metode	19
6.4	Analysis paralysis	20
6.5	Intervju	20
6.6	Analiza dokumenata	21
6.7	Upitnici	21
6.8	Funkcionalni vs nefunkcionalni zahtjevi	21
6.8.1	Funkcionalni zahtjevi	21
6.8.2	Nefunkcionalni zahtjevi	21
6.9	Use Case - Model slučaja upotrebe	22
6.9.1	Elementi slučaja upotrebe	22

6.9.2	Dijagram slučaja upotrebe	22
6.9.3	Definiranje sistemskih zahtjeva	23
6.9.4	Scenario	23
6.9.5	Veze između elemenata modela	23
7	Tutorijal 7: Modeliranje podataka i procesa	24
7.1	Uvod u modeliranje	24
7.2	Modeliranje podataka	24
7.2.1	Nivoi modeliranja podataka	24
7.2.2	Vrste modela podataka	25
7.3	ER model	25
7.3.1	Entitet	25
7.3.2	Atributi	25
7.3.3	Veze i kardinalnost	26
7.4	Normalizacija	26
7.4.1	Prva normalna forma (1NF)	26
7.4.2	Druga normalna forma (2NF)	27
7.4.3	Treća normalna forma (3NF)	27
7.4.4	Kako kreirati ERD?	27
7.4.5	Primjer: ERD za kompaniju za osiguranje automobila	28
7.5	Modeliranje procesa	28
7.6	Dijagrami aktivnosti	28
7.6.1	Elementi dijagrama aktivnosti	28
7.6.2	Primjeri dijagrama aktivnosti	29

1 Tutorijal 1: Uvod u Informacione Sisteme

1.1 Definicija sistema

Sistem je skup objekata (entiteta) i njihovih međusobnih veza usmjerenih ka ostvarivanju zajedničkog cilja, prihvatanjem ulaza i proizvodnjom izlaza u organizacijskom transformacionom procesu.

1.2 Podaci i informacije

Podaci (engl. data) su prosti zapisi koji imaju smisao samo u kontekstu u kome su zapisani. Primjer: "11. oktobar 2022.", "Tutorijal", "17:00".

Informacije su podaci koji su obrađeni i zapisani u složenom kontekstu u kome imaju određeni smisao i značenje. Primjer: "Tutorijal iz Osnova informacionih sistema je 11. oktobra 2022. u 17:00."

Hijerarhija: **Znanje** → **Informacije** → **Podaci**

1.3 Definicije informacionog sistema

Koncept informacionog sistema javlja se oko 1960. godine. Iako se smatra dobro uspostavljenim konceptom, i dalje se teško precizno definira.

Ključne definicije:

- Sistem koji uzima podatke kao ulaz i pretvara ih u informacije na izlazu.
- **Thierauf R.:** "Informacioni sistem je određeni skup metoda, postupaka i resursa, oblikovanih tako da se potpomogne postizanje nekih ciljeva."
- **IFIP:** "Informacioni sistem je sistem koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju i društvo, tako da budu dostupne i upotrebljive za svakog ko se želi njima koristiti."
- Informacioni sistem je uređeni sistem koji čine ljudi, podaci, procesi i interfejsi koji međusobno saraduju na podršci i unapređivanju svakodnevnih operacija u poslovanju.

1.4 Karakteristike informacionog sistema

Dobri informacioni sistemi:

- Učinkovitost
- Pouzdanost
- Prilagodljivost
- Sigurnost
- Kvalitetni izlazi
- Integracija

- Podrška korisnicima

Loši informacijski sistemi:

- Neučinkovitost
- Nestabilnost / Nepouzdanost
- Nedostatak prilagodljivosti
- Nesigurnost
- Niska kvaliteta izlaza
- Nedostatak integracije
- Nedostatak podrške korisnicima

1.5 Cilj izgradnje IS-a

Osnovni cilj izgradnje informacijskog sistema jeste postizanje boljih rezultata poslovanja. Svi drugi ciljevi proizilaze iz ciljeva poslovnog sistema i sa njima moraju biti usklađeni.

Kakav treba biti IS:

- Razumljiv svim korisnicima
- Jednostavan u prezentiranju informacija
- Pouzdan
- Omogućava iskazivanje obrađenih informacija u vrlo kratkim vremenskim intervalima

1.6 Tim za razvoj IS-a

Ključni članovi tima:

- Projekt manager (voditelj projekta)
- Analitičar poslovnih procesa
- Programeri i razvojni inženjeri
- Dizajner interfejsa
- Administrator baze podataka
- Inženjer za sigurnost podataka
- Testeri i QA analitičari
- Korisnici i naručitelji

Ključne karakteristike tima:

- Raznolikost vještina

- Podjela odgovornosti
- Stručno znanje
- Komunikacija i suradnja
- Kontinuirani nadzor i podrška
- Rizik i krizni menadžment
- Kvaliteta i kontrola

1.7 Strategije konkurentske prednosti

- Proizvesti najjeftiniji proizvod ili ponuditi najjeftiniju poslovnu uslugu
- Kreirati unikatan proizvod za kojeg ne postoji konkurentski proizvod
- Pronaći određeno područje tržišta (niša) u kojem proizvod dominira svojim privlačnim kvalitetom
- Uspostaviti takve barijere tako da njihovo prevazilaženje bi obeshrabrilo potencijalne konkurente

Ključna misao: IS je dobar” ako su korisnici zadovoljni!

1.8 Životni ciklus općih sistema (GSLC)

Svi sistemi koji postoje (biološki, fizički, socijalni) posjeduju velik broj zajedničkih karakteristika. GSLC predstavlja model sa četiri faze:

1. **Nastanak sistema** - osnivanje poslovne kompanije
2. **Razvoj i rast** - pojava poslovne organizacije na tržištu u borbi sa konkurencijom
3. **Zrelost sistema** - kompanija prestaje dalje rasti i dostiže stabilnu poziciju
4. **Deterioracija sistema** - kompanija gubi svoju tržišnu poziciju

1.9 Životni ciklus informacionih sistema (ISLC)

ISLC je sličan GSLC-u. Faze su:

- Razvoj sistema (dizajn)
- Implementacija sistema
- Rad i korištenje sistema (održavanje)
- Zastarjelost sistema (obsoletnost)

Karakteristika posljednje faze je ta da se ona rijetko očituje u fizičkoj formi, već sistemi postaju tehnološki i tehnički zastarjeli i nisu više u mogućnosti ispunjavati funkcionalne zahtjeve koje im postavljaju korisnici.

1.10 Dimenzije zastarjevanja IS-a

1. **Računovodstvena dimenzija** - U svakoj profitnoj organizaciji, knjigovodstveno se vodi amortizacija poslovne opreme. Period amortizacije varira, dok je negdje amortizacija za računarske sisteme potpuno ukinuta zbog intenzivnog razvoja i zastarjevanja ovih sistema.
2. **Tehnološka dimenzija** - Ukoliko konkurentne poslovne organizacije uvedu novije i modernije IS, tada i druge organizacije moraju, u najmanjoj mjeri, preispitati svoje sisteme.
3. **Deterioracija** - IS doživljavaju i fizičko trošenje. Stalno je prisutno pitanje nadogradnje, zamjene ili potpunog izbacivanja sistema iz prakse.
4. **Korisnička očekivanja** - Iako IS može funkcionirati i poslije njegove 'knjigovodstvene smrti', ne mora biti tehnološki zastario i ne pokazivati simptome fizičke deterioracije, on ipak može biti neprihvatljiv i nezadovoljavajući jer su se korisnička očekivanja vremenom promjenila.
5. **Eksterni utjecaji** - Nekada IS moraju biti zamjenjeni zbog vanjskih uticaja koji vrše pritisak na poslovnu organizaciju.

1.11 Software Development Life Cycle (SDLC)

SDLC je proces koji uključuje određivanje načina na koji će informacijski sistem podržati poslovne potrebe, dizajniranje sistema, izgradnju sistema te isporuku sistema korisnicima.

SDLC predstavlja konceptualni model koji se koristi u upravljanju projektima, kako bi se opisali faze i zadaci koji su uključeni u svaki korak razvoja i implementacije softvera.

Faze SDLC-a:

1. Planiranje

- Inicijacija projekta: kakvu poslovnu vrijednost sistem isporučuje organizaciji
- Upravljanje projektom: projekt menadžer kreira plan projekta, uspostavlja tim, dodjeljuje zadatke članovima tima

2. Analiza

- Strategija analize: istražuje se postojeći sistem, njegovi nedostaci i problemi
- Prikupljanje zahtjeva: zahtjevi se prikupljaju kroz intervju, grupne radionice, upitnike, proučavanje dokumentacije
- Kreiranje slučajeva upotrebe
- Modeliranje procesa i podataka

3. Dizajn

- Strategija dizajna: određuje način razvoja softvera
- Dizajn arhitekture: opisuje hardver, softver i mrežna infrastruktura
- Dizajn interfejsa i izvještaja: specificira izgled i navigaciju korisničkog interfejsa

- Dizajn baze podataka: definira koji će podaci biti pohranjivani i kako će se pohranjivati
- Dizajn programa: definira programe i šta koji program radi

4. Implementacija

- Razvoj sistema: Programiranje i testiranje
- Instalacija sistema: Stavljanje sistema u operativno okruženje

5. Održavanje

- Popravljanje i unapređivanje sistema

2 Tutorijal 2: SDLC - Faza Planiranja

2.1 Opis faze planiranja

SDLC faza planiranja je prva faza u životnom ciklusu razvoja softvera. Ova faza je ključna jer postavlja temelje za cijeli projekt i definira njegove ciljeve, opseg, potrebne resurse i vremenski okvir.

Osnovni zadatak faze planiranja je osigurati da svi članovi tima i zainteresirane strane (stakeholderi) razumiju što se želi postići i na koji način će se to postići.

2.2 Šta se radi u fazi planiranja SDLC-a

2.2.1 1. Identifikacija poslovnog problema ili prilike

U ovoj fazi se analizira i definiše poslovni problem koji softver treba riješiti ili prilika koju treba iskoristiti. To može uključivati izazove sa trenutnim sistemima ili prepoznavanje novih potreba tržišta.

2.2.2 2. Definisanje ciljeva projekta

Ciljevi moraju biti jasni, mjerljivi, ostvarivi, relevantni i vremenski ograničeni (**SMART ciljevi**):

- Specific (specifični)
- Measurable (mjerljivi)
- Achievable (ostvarivi)
- Relevant (relevantni)
- Time-bound (vremenski ograničeni)

2.2.3 3. Analiza zahtjeva zainteresiranih strana (Stakeholder analysis)

Identifikuju se zahtjevi i očekivanja različitih zainteresiranih strana, uključujući krajnje korisnike, menadžment i druge ključne aktere.

2.2.4 4. Izrada poslovnog slučaja i studije izvodljivosti

Poslovni slučaj se bavi ekonomskim aspektima projekta, analizirajući troškove i koristi, dok studija izvodljivosti procjenjuje tehničke i vremenske mogućnosti realizacije projekta.

2.2.5 5. Definisanje opsega projekta (Scope definition)

Definira se šta je tačno uključeno u projekt, kako bi se izbjeglo proširenje opsega (scope creep). Jasno se postavljaju granice onoga što će softver obuhvatiti.

2.2.6 6. Planiranje resursa i budžeta

Procjenjuju se potrebni resursi (ljudski, tehnički, finansijski) i kreira se budžet za projekt.

2.2.7 7. Izrada vremenskog rasporeda (Project Schedule)

Kreira se detaljan raspored koji uključuje ključne prekretnice (milestones) i rokove za svaki korak u procesu razvoja.

2.2.8 8. Analiza rizika (Risk analysis)

Identificiraju se potencijalni rizici koji mogu ugroziti projekt, te se razvijaju planovi za njihovo ublažavanje.

2.2.9 9. Izrada projektne povelje (Project Charter)

Dokument koji formalizuje ciljeve, opseg, raspored, budžet i odgovornosti, te daje ovlaštenja projektnom menadžeru za vođenje projekta.

2.3 Zahtjev za sistemom / Analiza izvodljivosti

2.3.1 Identifikacija zahtjeva za sistemom

- **Prikupljanje informacija:** Angažovanje sa ključnim zainteresovanim stranama (korisnicima, menadžerima, tehničkim timovima) kroz intervjue, ankete, fokus grupe i radionice.
- **Definisanje zahtjeva:** Na osnovu prikupljenih informacija, tim definiše funkcionalne (šta sistem treba da radi) i nefunkcionalne zahtjeve (kako treba da se ponaša, performanse, sigurnost).

2.3.2 Analiza izvodljivosti

Vrste analiza izvodljivosti:

1. Tehnička izvodljivost - Možemo li izgraditi sistem?

- Bliskost sa poslovnom oblasti
- Bliskost sa tehnologijom
- Kompatibilnost sa postojećom tehnologijom

- Veličina projekta i resursi koje projekat zahtjeva
2. **Ekonomska izvodljivost** - Trebamo li izgraditi sistem?
 - Identifikacija koristi i troškova
 - Dodjeljivanje vrijednosti koristima i troškovima
 - Određivanje toka novca
 - Ocjenjivanje ekonomske vrijednosti projekta
 3. **Operativna izvodljivost** - Ako izgradimo sistem, hoće li se koristiti?
 4. **Organizaciona izvodljivost** - Da li menadžment razumije tržište? Koliko je menadžment stručan za pokretanje poduhvata?
 5. **Zakonska izvodljivost** - Da li je predloženi sistem u sukobu sa zakonskim zahtjevima?
 6. **Rasporedna izvodljivost** - Da li je raspored projekta razuman?

2.3.3 Dokumentovanje i validacija

- **Dokumentovanje zahtjeva:** Sve prikupljene informacije i analize se dokumentuju u **SRS (Software Requirements Specification)** dokumentu.
- **Validacija:** Nakon što su zahtjevi definisani, važno je da se oni validiraju sa ključnim zainteresovanim stranama.

2.4 Software Requirements Specification (SRS)

2.4.1 Šta je SRS dokument?

Dokument o specifikaciji softverskih zahtjeva opisuje namjenu, zahtjeve i prirodu softvera koji će se razvijati.

2.4.2 Sadržaj dokumenta

1. UVOD

- Svrha/Cilj/Namjena
- Konvencija dokumenta (Standard izgleda dokumenta)
- Predviđeni korisnici i sugestije pri korištenju
- Opseg projekta
- Reference

2. **Historija izmjena dokumenta** - **OBAVEZNO** koristiti jedan od službenih jezika u Bosni i Hercegovini

2.5 Upoznavanje sa organizacijom / kompanijom

- Upoznati se sa organizacijom za koju gradite IS
- Koristiti različite izvore (web, direktni kontakt, promotivni materijali)
- Navesti kratku historiju organizacije
- Navesti glavne proizvode

2.6 Misija organizacije / kompanije

Misija organizacije definira svrhu organizacije, osnovnu funkciju organizacije odnosno razlog zbog kojeg organizacija postoji.

MISIJA = RAZLOG POSTOJANJA ORGANIZACIJE

Najbolje izjave o misiji su jasne, sažete i korisne. Prosječna dužina je samo 15,3 riječi.

2.7 Vizija organizacije / kompanije

Vizija organizacije je poželjna slika budućnosti organizacije, odnosno ono što poduzeće želi ostvariti.

Imati viziju znači imati odgovore na pitanja:

1. Kuda želimo ići?
2. Šta želimo biti u budućnosti?

2.8 Ciljevi organizacije / kompanije

Ciljevi organizacije predstavljaju stanje, tip ili nivo poslovnih performansi kojima će organizacija težiti u ostvarivanju svoje misije i vizije.

CILJ = KRAJNJA TAČKA DO KOJE TREBA IĆI; KRAJNJI REZULTAT KOJI TREBA POSTIĆI

Ciljevi se mogu podijeliti na:

- **Kratkoročne** (period do 1 godine)
- **Srednjoročne** (period između 1 i 5 godina)
- **Dugoročne** (period veći od 5 godina)

2.9 Stakeholderi

Stakeholderi su pojedinci i organizacije koji su aktivno uključeni u projekt ili na čije interese (rezultat) izvršenje projekta može (pozitivno ili negativno) uticati.

Stakeholderi projekta:

- Sponzori projekta
- Kupac koji će primiti isporučene proizvode
- Korisnici rezultata projekta
- Voda projekta i projektni tim

2.10 Zahtjev za sistemom

Dokument u kojem su navedeni razlozi za izgradnju sistema i vrijednost za koju se očekuje da će je sistem obezbijediti. Odgovara na pitanje: **Odakle dolazi ideja za novi projekat?**

2.11 Ciljevi projekta

- Navesti ciljeve projekta te preciznije detalje istih (referentne tačke)
- Navesti konkretne rezultate koje projekat donosi, preciznije detalje istih (koji će to biti rezultati i ko je odgovoran za njihovu proizvodnju i primanje)

2.12 Ekonomska izvodljivost - Finansijski pokazatelji

2.12.1 Return on Investment (ROI)

ROI pokazuje koliki je povrat neke investicije. Računa se kao količnik neto koristi (ukupna korist - ukupan trošak) i ukupnog troška.

Formula:

$$ROI = \frac{\text{Ukupna korist} - \text{Ukupan trošak}}{\text{Ukupan trošak}} \times 100\%$$

Visok ROI pokazuje da su koristi projekta daleko veće od ulaganja. ROI se obično računa na svakih 5 godina. Na samom početku projekta često isti bude negativan.

2.12.2 Break-Even Point (BEP)

BEP ili tačka pokrića je ekonomski koncept koji označava nivo prodaje ili aktivnosti pri kojem su ukupni prihodi jednaki ukupnim troškovima. U ovoj tački, preduzeće ne ostvaruje ni profit ni gubitak; ono samo pokriva svoje troškove.

Vrste troškova:

- **Varijabilni troškovi** - troškovi kompanije koji su povezani s brojem robe ili usluga koje proizvodi. Varijabilni troškovi povećavaju se i smanjuju s opsegom proizvodnje.
- **Fiksni troškovi** - troškovi kompanije koji ne variraju s opsegom proizvodnje. Fiksni troškovi ostaju isti bez obzira proizvode li se robe ili usluge.

2.12.3 Payback Period (PB)

Payback period ili period povrata ulaganja je vrijeme potrebno da se povрати inicijalno ulaganje u neki projekat ili investiciju kroz novčane tokove (cash flow) generisane tim projektom.

Za ravnomjeran cash flow:

$$\text{Payback period} = \frac{\text{Investirani novac}}{\text{Neto godišnji novčani priliv}}$$

Za neravnomjeran cash flow: Računa se kumulativno - broj godina potrebnih da poduzeće primi neto novčane prilive koji se zbrajaju s iznosom početnog novčanog ulaganja.

2.12.4 Present Value (PV) i Net Present Value (NPV)

Present Value (PV) predstavlja trenutnu vrijednost buduće sume novca ili tok novčanog toka s obzirom na određenu stopu povrata. Present Value predstavlja vrijednost novca koji ćete primiti u budućnosti, ali izražena u današnjim vrijednostima.

Net Present Value (NPV) je razlika između sadašnje vrijednosti novčanih priliva i sadašnje vrijednosti odliva novca tokom određenog vremenskog razdoblja.

Formula za PV:

$$PV = \frac{\text{Buduća vrijednost}}{(1 + r)^n}$$

gdje je r stopa povrata, a n broj perioda.

Projekat se smatra ekonomski isplativim ukoliko je NPV veći od nule.

3 Tutorijal 3: Upravljanje Projektom

3.1 SDLC faza - Planiranje (detaljno)

Dokument "Planiranja" za projekt koji koristi SDLC metodologiju Waterfall trebao bi sadržavati:

1. **Uvod i sažetak** - Uvodni dio koji opisuje svrhu dokumenta i sažetak projekta
2. **Opis projekta** - Detaljan opis projekta koji uključuje njegovu svrhu, ciljeve i obim
3. **Faze i aktivnosti** - Opis svake faze u SDLC metodologiji Waterfall, počevši s analizom i planiranjem
4. **Raspored projekta (Project Schedule)** - Raspored koji prikazuje vremenski okvir za svaku fazu i aktivnost projekta
5. **Resursi** - Popis resursa potrebnih za projekt, uključujući osoblje, opremu, alate i tehnologije
6. **Budžet** - Financijski plan koji uključuje procijenjeni proračun za projekt
7. **Plan upravljanja rizicima** - Identifikacija potencijalnih rizika i strategije za njihovo upravljanje
8. **Kvaliteta i testiranje** - Planiranje za osiguranje kvalitete u svim fazama projekta
9. **Odgovornosti i ovlasti** - Popis ključnih članova tima i njihovih odgovornosti
10. **Kontrola promjena** - Proces i pravila za upravljanje promjenama u zahtjevima ili planu projekta
11. **Komunikacijski plan** - Strategija za komunikaciju unutar tima i s dionicima projekta
12. **Zaključak i odobrenje** - Zaključak dokumenta u kojem se traži odobrenje od strane relevantnih strana

3.2 Aktivnost i projekat

Aktivnost koja je:

- Privremena
- Posjeduje jasno definiran datum početka i završetka
- Jedinstvena
- Donosi promjene
- Ograničena vremenom, troškovima i resursima
- Se preduzima da bi se isporučio proizvod u skladu sa postavljenim zahtjevima
- Služi za rješavanje problema ili korištenja prednosti prilike

Projekat je jedinstven - to nije rutinska operacija, već određeni skup operacija osmišljenih za postizanje jedinstvenog cilja.

3.3 Često korišteni projektni termini

- **Stakeholder** - Svako ko ima interes za neki projekat, posao ili organizaciju. U pogledu upravljanja projektima, stakeholder je pojedinac ili grupa na koju će utjecati ishod projekta.
- **Milestones (Kontrolna tačka)** - Tačka koja predstavlja mjesto provjere; datumi do kojih su glavne aktivnosti izvršene.
- **Tasks (Zadaci)** - Akcije: radnje poduzete projektom
- **Risks (Rizici)** - Potencijalni problemi koji mogu nastati
- **Issues (Problemi)** - Rizici koji su se dogodili

3.4 Upravljanje projektom

Definicija: Upravljanje projektom je primjena procesa, metoda, vještina, znanja, tehnika i iskustva za postizanje određenih ciljeva projekta prema kriterijima prihvatanja projekta u okviru zadanog vremena i budžeta.

Gljučni faktor koji razlikuje upravljanje projektima od samog procesa "menadžmenta" je taj što ima konačni rezultat i ograničen vremenski period, za razliku od menadžmenta koji je tekući proces.

Zašto upravljanje projektom?

- Unapređenje unutarne i vanjske komunikacije
- Jednostavno surađivanje
- Optimizirano upravljanje vremenom
- Jasnije razumijevanje ciljeva
- Praćenje projektnih zadataka i procesa

- Učinkovito upravljanje proračunom projekta
- Unapređenje donošenja odluka
- Smanjenje neizvjesnosti

3.5 Projekt menadžer i projektni tim

Projekt menadžer:

- Osoba odgovorna za vođenje projekta od njegovog početka do izvršenja
- Dužan osigurati uspjeh projekta minimiziranjem rizika kroz životni ciklus razvoja sistema
- Osoba sa raznolikim setom vještina: menadžment, liderstvo, tehnička potkovanost, rješavanje sukoba, upravljanje i odnos sa kupcima, upravljanje vremenom, planiranje, komunikativnost, upravljanje rizikom, pregovaranje
- Projekt menadžer pravi plan projekta

Projektni tim:

- Tim je grupa pojedinaca koji rade zajedno na postizanju određenog cilja
- Dobra saradnja projektnog tima ključna je za uspjeh projekta
- Posebno je efikasno i učinkovito komuniciranje važno kako bi se osiguralo da svi imaju potrebne informacije
- U idealnom slučaju, mišljenja, rješenja i sukobi se razmatraju, raspravljaju i rješavaju u sklopu projektnog tima

3.6 Plan projekta

Prvi korak u kreiranju plana projekta jeste identifikacija taskova. Identifikacija taskova slijedi **top-down pristup**.

3.7 Work Breakdown Structure (WBS)

Definicija: Work Breakdown Structure (WBS) je hijerarhijska dekompozicija posla koji treba izvesti projektni tim radi ostvarenja projektnih ciljeva i stvaranja potrebnih rezultata.

Karakteristike:

- To je način za podjelu i osvajanje velikih projekata, radi bržeg i efikasnijeg rada
- WBS dijeli projekt u manje elemente, koji su upravljivi i mjerljivi
- Svaki element opisan je detaljima poput: naziv, datum početka i završetka, predviđeno i stvarno trajanje, status, ovisnosti, potrebni resursi

3.8 Gantt Chart (Gantogram)

Definicija: Gantogram je tip horizontalnog stupčanog grafika za grafički prikaz rasporeda projekta.

Gantogram nudi grafički prikaz rasporeda koji pomaže u planiranju, koordinaciji i praćenju određenih zadataka u projektu.

3.9 Upravljanje osobljem

Prilikom raspoređivanja članova na taskove, treba imati na umu da li oni posjeduju odgovarajuće tehničke i komunikacijske sposobnosti.

Matrica sposobnosti je alat koji se koristi za dokumentiranje i uspoređivanje potrebnih kompetencija za radno mjesto s trenutnom razinom vještina zaposlenih.

Skale:

- **INTERES:** 0 - nema interesa, 1 - ima interesa
- **ZNANJE:** 0 - nema znanja, 1 - osnovni nivo, 2 - srednji nivo, 3 - visok nivo znanja
- **ISKUSTVO:** 0 - nema iskustva, 1 - mora raditi uz nadzor, 2 - može raditi samostalno uz povremenu kontrolu, 3 - može obučavati ostale

3.10 Mrežno planiranje

Mrežni dijagram (plan) je graf, kojim je prikazan redoslijed aktivnosti jednog projekta i njihove međusobne zavisnosti, čime se postiže uvid u način, redoslijed i vrijeme izvršenja aktivnosti.

Veze se prikazuju linijama orijentisanim strelicama. Podrazumijeva se da je veza orijentisana od lijevo na desno, te da je lijevo aktivnost prethodna, a desno naredna.

Osnovni termini:

- **DOGAĐAJ:** Stanje koje indicira kada aktivnost počinje ili završava. U mrežnim dijagramima su predstavljeni sa krugovima. Obično se označavaju brojevima: i - početni događaj aktivnosti, j - krajnji događaj aktivnosti.
- **AKTIVNOST:** Predstavlja izvršenje nekog taska. U mrežnim dijagramima se predstavljaju strelicama između dva kruga - događaja. Usmjerenje strelice pokazuje tok aktivnosti. Obično se označavaju sa $i - j$, gdje su i i j početni i krajnji događaj aktivnosti.
- **TRAJANJE:** Vrijeme potrebno da se izvrši neka aktivnost. Izražava se u vremenskim jedinicama (sati, dani, sedmice, godine).
- **VREMENSKA REZERVA:** Pokazuje koliko se može produžiti ili odgoditi izvršavanje neke aktivnosti. Aktivnost ima vremenski rezervu veću od nule ako joj je vrijeme trajanja manje od raspoloživog vremena.
- **KRITIČNI PUT:** Skup aktivnosti čije se izvršavanje ne može produžiti niti odgoditi, a da se to ne odrazi na trajanje projekta.

3.10.1 Proračun naprijed

Proračunom naprijed se odredi najranije vrijeme u kojem se može dogoditi svaki događaj u rasporedu.

Najraniji početak aktivnosti:

$$EF_i = \max[EF(PA) + t_i]$$

gdje je PA oznaka za prethodnu aktivnost, a t_i trajanje aktivnosti i .

3.10.2 Proračun nazad

Proračun "nazad", na osnovu koga se određuje vrijednost najkasnijeg početka (LF) svake aktivnosti:

$$LF_i = \min[LF(NA) - t_i(NA)]$$

gdje je NA oznaka za narednu aktivnost.

3.11 Critical Path Method (CPM)

One aktivnosti čija su vremena najranijeg i najkasnijeg početka, odnosno završetka, jednaka imaju vremensku rezervu jednaku 0. To su tzv. kritične aktivnosti, koje tvore kritični put.

Critical Path određuje trajanje projekta. Ukoliko bilo koja aktivnost koja je na kritičnom putu bude odložena, cijeli projekat će kasniti!

3.12 DiSC Profili

DiSC profili se tiču samospoznaje i komunikacije. Osmišljeni su kako bi pomogli poboljšati komunikaciju i suradnju na radnom mjestu, poboljšati performanse timova i poboljšati učinkovitost vođa i menadžera.

DiSC je akronim koji označava četiri glavna profila ličnosti:

- **D - Dominance (Dominacija):** Tendencija ka samopouzdanju i stavljanje naglasaka na postizanje krajnjih rezultata. Samopouzdana su, iskreni i zahtjevni.
- **I - Influence (Utjecaj):** Tendencija ka pristupačnosti i stavljanje naglasaka na odnose, utjecaj ili uvjeravanje drugih. Entuzijastični, optimistični, povjerljivi i energični.
- **S - Steadiness (Stabilnost):** Tendencija ka pouzdanosti i naglasak na suradnji i iskrenosti. Skloni smirenoj naravi i ne vole da ih se požuruje.
- **C - Conscientiousness (Savjesnost):** Naglasak na kvalitetu, tačnosti, stručnosti i kompetenciji. Uživaju u svojoj neovisnosti, zahtijevaju detalje i često se boje da ne pogriješe.

3.13 Iron Triangle

Tri ograničenja u Iron Triangle-u služe i kao mjerilo uspjeha ili neuspjeha projekta:

- **Opseg** - šta projekt treba da isporuči
- **Vrijeme** - kada projekt treba da bude završen
- **Troškovi** - koliko će projekt koštati

Da bi projekt bio uspješan, navedena tri faktora moraju biti uravnotežena.

3.14 Upravljanje kvalitetom

Kvaliteta je četvrto ograničenje (jednako važno, ali često zaboravljeno).

Upravljanje kvalitetom projekta obuhvata procese i aktivnosti koje se koriste da bi se utvrdila i postigla kvaliteta rezultata projekta.

Kvaliteta je jednostavno ono što kupcu ili stakeholderima treba od rezultata projekta.

3.14.1 Koncepti upravljanja kvalitetom

Tri su ključna koncepta upravljanja kvalitetom:

1. **Zadovoljstvo kupaca** - Nekada i kada su svi zahtjevi i ciljevi ispunjeni, a sam proces nije bio zadovoljavajući, kupac će projekat obilježiti kao neuspješan.
2. **Prevenција prije inspekcije** - Trošak kvalitete (Cost of Quality - COQ) je novac koji se troši za bavljenje problemima tokom projekta, a potom i nakon projekta, kako bi se riješili eventualni propusti.

Vrste troškova:

- Troškovi prevencije i procjene
 - Troškovi neusklađenosti/neuspjeha (unutarnji i vanjski troškovi kvara)
3. **Kontinuirano poboljšanje** - Neprekidni napor poboljšanja proizvoda, usluga ili procesa tokom vremena

3.14.2 PDCA / Deming Cycle

PDCA (Plan-Do-Check-Act ili Plan-Do-Check-Adjust) je iterativna metoda upravljanja u četiri koraka koja se koristi u poslovanju za kontrolu i kontinuirano poboljšavanje procesa i proizvoda.

Koraci:

1. **Plan** - Planiranje
2. **Do** - Izvršavanje
3. **Check** - Provjera
4. **Act/Adjust** - Djelovanje/Podešavanje

4 Tutorijal 4: skip

5 Tutorijal 5: skip

6 Tutorijal 6: Analiza zahtjeva

6.1 Definicija analize zahtjeva

Zahtjevi su izjave koje određuju šta sistem treba raditi zarad pružanja funkcionalnosti.

Analiza zahtjeva je proces definiranja očekivanja korisnika za aplikaciju koja treba biti izgrađena ili izmijenjena.

Analiza zahtjeva se provodi radi prepoznavanja potreba različitih stakeholdera. Stoga, analiza zahtjeva znači analizu, dokumentiranje, potvrđivanje i upravljanje softverskim ili sistemskim zahtjevima.

Dobro definirani zahtjevi su presudni za uspjeh projekta. Svi sudionici u projektu (programeri, krajnji korisnici, menadžeri) moraju postići zajedničko razumijevanje o tome kakav će proizvod biti i šta će isti raditi.

6.2 4 koraka analize zahtjeva

1. **Prikupljanje zahtjeva:** Komuniciranje sa korisnicima u cilju saznanja njihovih zahtjeva
2. **Analiziranje zahtjeva:** Utvrđivanje jesu li navedeni zahtjevi nejasni, nepotpuni, dvosmisleni, a zatim rješavanje ovih pitanja
3. **Modeliranje zahtjeva:** Zahtjevi se mogu dokumentirati u različitim oblicima, kao što su dokumenti, slučajevi upotrebe, korisničke priče ili specifikacije procesa
4. **Pregled i retrospektiva:** Članovi tima analiziraju cjelokupan proces te predlažu eventualna unapređenja

6.3 Metode prikupljanja zahtjeva

6.3.1 Tradicionalne metode

- Intervjuiranje pojedinačnih korisnika
- Intervjuiranje grupa korisnika
- Promatranje rada (observiranje)
- Proučavanje poslovnih dokumenata

6.3.2 Savremene metode

- **Joint Application Design (JAD):** Okuplja ključne korisnike, menadžere i analitičare sistema. Namjena: sakupiti zahtjeve o sistemu simultano od ključnih ljudi, intenzivna grupno orijentirana tehnika određivanja zahtjeva. Članovi tima sastaju se u izolaciji na duže vrijeme.

- **CASE alati:** Analiza postojećih sistema, domena je softverskih alata za izradu i implementaciju aplikacija. Osnovna ideja CASE alata je da ugrađeni programi mogu pomoći u analizi sistema u razvoju kako bi se poboljšala kvaliteta i pružili bolji ishodi. Primjeri: Accept 360, Accompa, CaseCompleat.
- **Sistem prototipova:** Iterativni razvojni proces, brzo pretvara zahtjeve u radnu verziju sistema. Nakon što korisnik vidi zahtjeve pretvorene u sistem, on će tražiti izmjene ili će generisati dodatne zahtjeve.

6.4 Analysis paralysis

Analysis paralysis odnosi se na situaciju u kojoj se pojedinac ili skupina nisu u stanju pomjeriti naprijed s odlukom zbog prevelike količine podataka. Opisuje stanje kada se zaglavi u procesu analize. Trošak njihovog prikupljanja i struktuiranja može biti velik i vremenski i finansijski. **Previše analize nije produktivno.**

6.5 Intervju

Intervju je tehnika prikupljanja zahtjeva putem razgovora.

Vrste intervju:

- **Strukturirani/formalni intervju:** Pitanja se postavljaju po zadanom/standardiziranom redoslijedu (nefleksibilan intervju bez odstupanja)
- **Nestrukturirani/slobodni intervju:** Usmjereni razgovor

Proces intervju:

1. **Priprema za intervju:** Spisak ispitanika, svrha intervju, mjesto i vrijeme, informacije o ispitaniku, poslu, priprema koncepta, određivanje pitanja
2. **Korak prije intervju:** Dolazak ranije sa spremnim konceptom
3. **Intervjuiranje:** Postavljanje pitanja, kreiranje zabilježki
4. **Zaključivanje intervju:** Zahvala sugovorniku, finalna pitanja i komentari
5. **Evalvacija:** Analiziranje prikupljenog i zapisanog, pregledati bilješke u roku od 48h

Tipovi pitanja:

- **OTVORENA:** Ispitaniku daje priliku da slobodno izrazi svoje mišljenje. Uključivanjem pitanja otvorenog formata u svoj upitnik, možete dobiti istinite, pronicljive i čak neočekivane prijedloge.
- **ZATVORENA:** Vrsta pitanja koja ispitanicima daju mogućnost da odaberu iz različitog skupa unaprijed definiranih odgovora
- **OPIPAVANJA:** Dodatna pitanja, koja se nadovezuju na prethodna otvorena ili zatvorena pitanja. Njihov cilj je da potvrde odgovor ili da ga prošire sa dodatnim informacijama.

6.6 Analiza dokumenata

Analiza dokumenata je tehnika koja se koristi za prikupljanje zahtjeva tokom faze iznošenja zahtjeva projekta. Opisuje akt pregleda postojeće dokumentacije kako bi se izvukli podaci koji su relevantni za tekući projekat.

6.7 Upitnici

Upitnik je instrument za prikupljanje podataka, koji gotovo uvijek uključuje traženje od ispitanika da odgovori na set usmenih ili pismenih pitanja.

Karakteristike upitnika:

- Upitnici omogućavaju da se isti set pitanja postavi većoj grupi ljudi
- Upitnici su efikasni kada sistemski analitičar traži malu količinu podataka od velike grupe ljudi
- Upitnik mora biti pažljivo formuliran kako bi značenje svakog pitanja bilo jasno većini ispitanika
- Upitnike je teško sastaviti - isto pitanje može imati različita značenja za različite ispitanike
- Efikasan upitnik je kratak, jasan i dobro testiran
- Jedna od prednosti upitnika u sistemskoj analizi je to što omogućava velikoj grupi ljudi da osjete da učestvuju u razvoju budućih sistema kompanije

6.8 Funkcionalni vs nefunkcionalni zahtjevi

6.8.1 Funkcionalni zahtjevi

Opisuju funkcije koje softver mora da obavlja. Funkcionalni softverski zahtjevi pomažu da zabilježite predviđeno ponašanje sistema. Ovo ponašanje se može izraziti kao funkcije, usluge ili zadaci koje sistem obavlja da bi ispunio korisnička očekivanja.

Primjer: Softver mora omogućiti spašavanje izvještaja.

6.8.2 Nefunkcionalni zahtjevi

Definiraju attribute kvaliteta softverskog sistema. Predstavljaju skup standarda koji se koriste za prosuđivanje konkretnog rada sistema. Neophodni za osiguravanje upotrebljivosti i efikasnosti cjelokupnog softverskog sistema.

Obuhvataju:

- Političke faktore i zakonske procedure
- Ko ima autorizovan pristup i pod kojim okolnostima
- Fizičko i tehničko okruženje u kojem će sistem raditi
- Brzina, kapacitet i pouzdanost sistema

Primjer: Web lokacija bi se trebala učitati za 3 sekunde kada je broj istodobnih korisnika > 10000.

6.9 Use Case - Model slučaja upotrebe

Slučajevi upotrebe su detaljni opisi kako korisnici koriste sistem za postizanje cilja. Oni opisuju sistem iz perspektive korisnika, tako da korisnik ima aktivnu ulogu u njihovom kreiranju.

Model slučaja upotrebe sastoji se od dva artefakta:

- **Dijagram slučaja upotrebe:** Grafički prikaz korisnika i slučajeva upotrebe vezanih za iste
- **Opis slučaja upotrebe:** Detaljan tekst koji korak po korak opisuje interakciju i dijalog između aktera i sistema

6.9.1 Elementi slučaja upotrebe

- **Osnovne informacije:** Naziv, ID, prioritet, učesnik (osoba, drugi sistem ili uređaj koji je u interakciji sa sistemom), opis, trigger (događaj koji uzrokuje slučaj upotrebe), učestalost upotrebe
- **Preduvjeti:** Stanja koja trebaju biti uspostavljena prije nego što počne slučaj upotrebe
- **Postkondicioni uvjeti:** Postkondicioniranje uspjeha označava šta se dešava kada se proces uspješno završi. Postkondicija neuspjeha je suprotna; ona određuje šta se događa kada postupak ne završi uspješno.
- **Tok:**
 - **Normalni tok (happy path):** Ruta koja opisuje korake po planu bez neočekivanih koraka do ispunjenja cilja
 - **Alternativni tok:** Ruta koja opisuje alternativne korake do ispunjenja cilja
 - **Tok izuzetaka:** Ruta koja opisuje korake koji vode do neostvarivanja cilja
- **Rezultati:** Definišu finalni proizvod slučaja upotrebe
- **Ulazi i izlazi:** Sumiraju osnovne ulaze i izlaze svakog koraka u slučaju upotrebe

6.9.2 Dijagram slučaja upotrebe

Dijagrami slučaja upotrebe:

- Pokazuju samo ono što sistem treba da uradi (funktionalni zahtjevi)
- Važno je otkriti i predstaviti funkcionalne zahtjeve na početku projekta (uštediti vremena i novca kasnije)
- Prikazuje ponašanje sistema kako ga vidi korisnik
- Aktivnosti na visokom nivou
- Odgovara na pitanja: Šta se opisuje? (Sistem), Ko je u interakciji sa sistemom? (učesnici), Šta učesnici mogu da urade? (slučajevi upotrebe)

6.9.3 Definiranje sistemskih zahtjeva

Prvi korak: Odredite ko/šta je u interakciji sa sistemom! → **UČESNICI**

Učesnik: Eksterni entitet povezan sa sistemom, ali nije dio sistema. Pokreće neke radnje. Može biti stvarna osoba ili drugi sistem (npr. aplikacija). Imaju imena koja ne bi trebala biti vezana za organizaciju kompanije. Napiši kao imenicu u jednini.

Drugi korak: Pronađite slučajeve u kojima se sistem koristi za obavljanje određenih zadataka učesnika! → **SLUČAJEVI UPOTREBE**

Slučajevi upotrebe se identifikuju analizom dokumenata sa zahtevima. Ovi dokumenti uključuju specifikacije napisane na prirodnom jeziku koje objašnjavaju šta korisnik želi od sistema.

Treći korak: Opišite slučajeve upotrebe!

Dijagrami ne mogu pružiti dovoljno detalja dizajnerima sistema. Najčešće se dodatno opisuju složeniji sistemi. Opisan je SVAKI slučaj upotrebe u modelu.

6.9.4 Scenario

Da bi se kroz analizu zahtjeva lakše identifikovali slučajevi upotrebe, potrebno je razviti scenario.

SCENARIO - niz koraka koji opisuje interakciju između učesnika i sistema.

Primjer scenarija: Web shop

Kupac pretražuje web katalog proizvoda i dodaje željene proizvode u košaricu. Prilikom kupovine kupac odabire lokaciju dostave i daje podatke o kreditnoj kartici i potvrđuje kupovinu. Sistem zatim provjerava valjanost kreditne kartice, potvrđuje kupovinu i šalje e-mail kojim potvrđuje kupovinu.

Alternativni scenario: Kreditna kartica odbijena! Kupovina proizvoda je obustavljena!

6.9.5 Veze između elemenata modela

U dijagramu slučaja upotrebe, definirane su četiri vrste veza:

1. **Asocijacija (Association):** Predstavlja komunikacijsku putanju između aktera i slučaja upotrebe u kojem sudjeluje. U slučaju da se želi naglasiti koji akter inicira određeni obrazac upotrebe (inicijator) to se može napraviti dodatkom strelice na vezu asocijacije.
2. **Proširenje (Extend):** Relacijom proširenja se označava dodavanje dodatnih funkcionalnosti osnovnom slučaju upotrebe. Prošireni slučaj upotrebe predstavljaju izvršavanje dodatnih funkcionalnosti osnovnog slučaja upotrebe ili funkcionalnosti koje se izvršavaju samo ako su zadovoljeni prethodno definirani uslovi. Osnovni slučaj upotrebe mora moći funkcionirati samostalno bez upotrebe proširenog slučaja.
3. **Poopćavanje (Generalization):** Relacija poopćavanja ili relacija nasljeđivanja se koristi kada postoji podslučaj upotrebe (dijete) koji koristi svojstva, operacije i odnose od višeg slučaja uporabe (roditelj). Poopćavanje se može primijeniti i između aktera sistema.
4. **Obuhvaćanje/Sadržajne (Include):** Relacija obuhvaćanja se primjenjuje kada osnovni slučaj upotrebe izričito uključuje funkcionalnosti drugog slučaja upotrebe.

Obuhvaćeni slučaj upotrebe ne postoji samostalno, već ovisi o jednom ili više osnovnih slučajeva upotrebe. Njime se izdvajaju zajedničke funkcionalnosti više slučajeva upotrebe.

Granice sistema: Oni služe za jasno odvajanje učesnika (koji nisu dio sistema) od samog sistema. Nacrtani su na način da su svi slučajevi upotrebe stavljeni u jedan kvadrat. Kvadratu se najčešće daje ime samog sistema.

7 Tutorijal 7: Modeliranje podataka i procesa

7.1 Uvod u modeliranje

Model je aproksimacija stvarnosti.

7.2 Modeliranje podataka

Modeliranje podataka je proces kreiranja modela podataka za podatke koji se pohranjuju u bazi podataka. Modeliranje podataka pomaže u vizualnom predstavljanju podataka i provodi poslovna pravila, regulatorne usklađenosti i politike o tim podacima. Modeli podataka osiguravaju dosljednost u imenovanju, zadanim vrijednostima, semantici, sigurnosti, istovremeno osiguravajući kvalitetu podataka.

7.2.1 Nivoi modeliranja podataka

Modeliranje podataka odvija se na tri nivoa: fizičkom, logičkom i konceptualnom.

1. **Konceptualni nivo:** Ovaj model podataka definira šta sistem sadrži. Ovaj model obično kreiraju poslovni stakeholderi i arhitekti podataka. Namjena je organizirati, obuhvatiti i definirati poslovne koncepte i pravila.
 - Najviši, apstraktni nivo modeliranja
 - Fokusira se na što bolje razumijevanje podataka i njihovih veza iz poslovne perspektive
 - Nema tehničkih detalja, poput vrsta podataka ili tehnologija
 - Koristi dijagrame entiteta i veza (ER dijagram)
 - Fokusiran je na potrebe korisnika
2. **Logički nivo:** Definira kako sistem treba implementirati bez obzira na DBMS. Ovaj model obično kreiraju arhitekti podataka i poslovni analitičari. Namjena je razviti tehničku kartu pravila i strukture podataka.
 - Prelazni nivo između apstraktnog konceptualnog modela i tehničkog fizičkog modela
 - Fokusira se na strukturiranje podataka u obliku koji baza podataka može obraditi, ali i dalje nezavisno od specifične tehnologije
 - Precizno definiše attribute entiteta (npr. ISBN za knjige, ID za člana)
 - Uključuje primarne i strane ključeve (identifikatori za relacije)

- Normalizacija se koristi za smanjenje redundancije
3. **Fizički nivo:** Ovaj model podataka opisuje kako će se sistem implementirati pomoću određenog DBMS sistema. Ovaj model obično kreiraju DBA i programeri. Svrha je stvarna implementacija baze podataka.
- Najdetaljniji i tehnički nivo modeliranja
 - Definiše kako će podaci biti stvarno skladišteni u bazi podataka
 - Fokusira se na implementaciju baze u konkretnom sistemu
 - Definiše konkretne tipove podataka (npr. VARCHAR, INT)
 - Određuje indekse, particije i pravila o pristupu
 - Fokusiran na performanse i tehničke detalje

Podela omogućava postupno prelazak od apstraktnog razmišljanja (poslovni zahtevi) ka tehničkoj implementaciji, što olakšava komunikaciju između poslovnih analitičara, dizajnera baza podataka i inženjera.

7.2.2 Vrste modela podataka

- **Hijerarhijski model:** Koristi hijerarhiju za strukturiranje podataka u obliku poput stabla
- **Relacijski model:** Podaci predstavljeni u obliku tablica
- **Mrežni model:** Inspirisan hijerarhijskim modelom. Za razliku od hijerarhijskog modela, ovaj model olakšava prenošenje složenih odnosa, jer se svaki zapis može povezati s više matičnih zapisa
- **Objektno orijentiran model:** Sastoji se od kolekcije objekata, svaki sa svojim atributima i metodama. Ova vrsta modela baze podataka naziva se i post-relacijski model baze podataka
- **Entity Relationship model:** Predstavlja entitete i njihove odnose u grafičkom formatu. Entitet može biti bilo šta - koncept, podatak ili objekt

7.3 ER model

7.3.1 Entitet

Entitet: Objekt ili komponenta podataka; osnovni gradivni blok za model podataka. Može predstavljati osobu, mjesto, događaj ili objekat o kojem se prikupljaju podaci. Naziv entiteta treba biti u obliku **Imenica**.

Slabi entitet je onaj koji ne posjeduje primarni ključ.

7.3.2 Atributi

Atributi: Karakteristike ili svojstva entiteta.

Vrste atributa:

- **Key attribute:** Jedinstvena karakteristika entiteta (matični broj, broj lične karte, pasoša, ...)

- **Composite attribute:** Kombinacija drugih atributa poznata je kao složeni atribut. (adresa: ulica, broj, država)
- **Multivalued:** Atribut koji može sadržavati više vrijednosti (Osoba može imati više telefonskih brojeva)
- **Derived:** Izvedeni atribut je onaj čija je vrijednost dinamička i izvedena je iz drugog atributa. (Starost osobe je izvedeni atribut te se s vremenom mijenja i može se izvesti iz drugog atributa (Datum rođenja))

7.3.3 Veze i kardinalnost

Veza: Ovisnost ili povezanost između dva entiteta. Veze se označavaju glagolima, koji trebaju pojasniti odnos između dva entiteta.

KARDINALNOST je odnos broja instanci entiteta i broja instanci drugog entiteta.
Vrste kardinalnosti:

- **One to One (1:1):** Jedna instanca entiteta povezana s jednom instancom drugog entiteta
- **One to Many (1:N):** Jedna instanca entiteta povezana s više instanci drugog entiteta
- **Many to One (N:1):** Više instanci entiteta povezano s jednom instancom drugog entiteta
- **Many to Many (M:N):** Više od jedne instance entiteta povezano je s više instanci drugog entiteta

7.4 Normalizacija

Normalizacija je tehnika koja organizuje attribute na način da formiraju neredundantne, stabilne, fleksibilne i prilagodljive entitete.

Model podataka treba biti: fleksibilan, neredundantan, jednostavan
 Normalizacija uključuje tri koraka:

1. Prva normalna forma (1NF)
2. Druga normalna forma (2NF)
3. Treća normalna forma (3NF)

7.4.1 Prva normalna forma (1NF)

Cilj: Postići da tablica sadrži samo atomske vrijednosti u svakoj ćeliji (tj. vrijednosti koje se ne mogu dalje podijeliti) i da svaki stupac ima jedinstven naziv.

Karakteristike:

- Svaka ćelija tabele treba sadržavati jednu vrijednost
- Svaki zapis mora biti jedinstven

Koraci za postizanje 1NF:

1. Ukloniti višestruke vrijednosti: Pobrinuti se da svaki stupac sadrži samo jednu vrijednost po retku
2. Osigurati jedinstvenost stupaca: Svaki stupac treba imati jedinstven naziv i jasno definiranu domenu (tip i format podataka)

7.4.2 Druga normalna forma (2NF)

Cilj: Ukloniti parcijalne zavisnosti, tj. zavisnosti gdje neki atributi zavise samo od dijela primarnog ključa u tablici sastavljenoj od više atributa.

Karakteristike:

- 1NF
- Ako su svi atributi, koji nisu primarni ključ, ovisni od cijelog primarnog ključa

Koraci za postizanje 2NF:

1. Identificirati i ukloniti parcijalne zavisnosti: Ako tablica ima složeni primarni ključ (sastavljen od više stupaca), svaki stupac koji nije dio ključa mora ovisiti o cijelom ključu, a ne samo o dijelu
2. Razdvajanje tablica: Ako neki atributi ne ovise o cijelom ključu, trebali bi biti izdvojeni u zasebnu tablicu gdje će formirati funkcionalnu cjelinu s dijelom ključa o kojem zavise

7.4.3 Treća normalna forma (3NF)

Cilj: Ukloniti tranzitivne zavisnosti, gdje atributi zavise od drugih atributa koji nisu ključ.

Karakteristike:

- 2NF
- Ako su vrijednosti njegovih atributa, koji nisu primarni ključ, neovisne od drugih atributa, koji nisu primarni ključ

Koraci za postizanje 3NF:

1. Identificirati i ukloniti tranzitivne zavisnosti: Ako neki atribut zavisi od drugih neključnih atributa, te zavisnosti treba ukloniti
2. Razdvajanje tablica: Atributi koji imaju tranzitivne zavisnosti trebaju biti izdvojeni u zasebne tablice

7.4.4 Kako kreirati ERD?

1. Identifikacija entiteta
2. Identifikacija atributa za svaki entitet
3. Kreiranje veza između entiteta

7.4.5 Primjer: ERD za kompaniju za osiguranje automobila

Kreirati ERD za kompaniju za osiguranje automobila čiji klijent posjeduje po jedan ili više automobila. Svaki automobil može prouzrokovati 0 ili više nezgoda.

- Automobil je opisan modelom, godinom proizvodnje, tablicama
- Osoba opisana ID vozača, adresom i imenom
- Nezgoda opisana brojem štete, lokacijom i datumom
- Također, nezgoda će u odnosu na automobil opisati i iznos štete

7.5 Modeliranje procesa

Model procesa je način prikazivanja kako bi poslovni sistem trebao raditi. On ilustruje procese ili aktivnosti, te kako se podaci razmjenjuju između njih.

7.6 Dijagrami aktivnosti

Dijagram aktivnosti je dijagram ponašanja, tj. prikazuje ponašanje sistema.

7.6.1 Elementi dijagrama aktivnosti

- **Inicijalno stanje:** Početak procesa
- **Akcija ili stanje aktivnosti:** Aktivnost predstavlja izvršavanje akcije na objektima. Predstavljamo aktivnost pomoću pravougaonika sa zaobljenim uglovima. U osnovi svaka radnja ili događaj koji se odvija predstavljen je pomoću aktivnosti.
- **Akcijski tok ili Kontrolni tokovi:** Akcijski tokovi ili Kontrolni tokovi također se nazivaju putanje i rubovi. Koriste se za prikazivanje prelaska iz jednog stanja u drugo.
- **Čvor odluke i grananje:** Kad moramo donijeti odluku prije nego što odlučimo o protoku, koristimo čvor odluke.
- **Čuvari:** Čuvar se odnosi na izjavu napisanu pored čvora odluke na strelici, unutar uglastih zagrada.
- **Fork:** Čvorovi Fork koriste se za podršku paralelnim aktivnostima. Odluka se ne dijeli prije podjele izvršavanja paralelnih aktivnosti. Oba dijela se moraju izvršiti u slučaju fork-a.
- **Join:** Čvorovi pridruživanja koriste se za podršku istodobnim aktivnostima konvergiranjem u jednu.
- **Spajanje ili spajanje događaja:** Scenariji nastaju kada aktivnosti koje se ne izvršavaju istovremeno moraju biti spojene. Za takve scenarije koristimo oznaku spajanja. Dvije ili više aktivnosti možemo spojiti u jednu ako kontrola prede na sljedeću aktivnost bez obzira na odabrani put.

- **Swimlanes:** Koristimo swimlanes za grupiranje povezanih aktivnosti u jednoj koloni. Swimlanes mogu biti vertikalni i horizontalni. Nije obavezna upotreba swimlanes-a. Obično daju više jasnoće dijagramu aktivnosti.
- **Vremenski signal:** Možemo imati scenario u kojem je nekom događaju potrebno određeno vrijeme.
- **Signali:** Signal ukazuje na to da aktivnost prima događaj iz spoljnog procesa.
- **Kraj toka:** Označava kraj određenog toka, ali ne i kraj kompletne aktivnosti.

7.6.2 Primjeri dijagrama aktivnosti

Primjer 1: Login akcija sistema

Nacrtati dijagram aktivnosti za login akciju sistema. Korisnik unosi username i password i kada su oni ispravni se uspješno loguje na sistem. U suprotnom mu se otvara prozor sa porukom o nevalidnosti kredencijala te ga se ponovo vraća na login stranicu. Kada se korisnik uspješno loguje prikazane su mu njegove postavke.

Primjer 2: Email komunikacija

Kreirati dijagram aktivnosti za email komunikaciju. Prvobitno se uspostavlja email komunikacija te se nakon toga paralelno izvrše radnje slanja emaila i primanja odgovora. Ukoliko je sadržaj privatan, šalje se enkriptovan email, u suprotnom sistem šalje standardan email. Nakon slanja bilo koje vrste emaila, sistem jednako nastavlja sa radom. Kada je riječ o primanju odgovora, sistem čeka 2 sata na odgovor te provjerava da li je odgovor validan ili ne. Ukoliko je odgovor nevalidan ili ne postoji prekida se izvršenje, u suprotnom izvršenje se nastavlja. Nakon završetka svih pomenutih paralelnih aktivnosti, sistem potvrđuje uspostavu email komunikacije.