

# PRIKUPLJANJE I ANALIZA SISTEMSKIH ZAHTJEVA

---

Lozančić, Ana

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:145:177901>

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

Download date / Datum preuzimanja: **2024-07-16**



Repository / Repozitorij:

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Sveučilišni prijediplomski studij Poslovna informatika

Ana Lozančić

**PRIKUPLJANJE I ANALIZA SISTEMSKIH ZAHTJEVA**

Završni rad

Osijek, 2023.

Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Sveučilišni prijediplomski studij Poslovna informatika

Ana Lozančić

## **PRIKUPLJANJE I ANALIZA SISTEMSKIH ZAHTJEVA**

Završni rad

**Kolegij: Oblikovanje i implementacija IS-a**

JMBAG: 0113143366

e-mail: [alozancic1@efos.hr](mailto:alozancic1@efos.hr)

Mentor: doc. dr. sc. Dario Šebalj

Osijek, 2023.

Josip Juraj Strossmayer University of Osijek  
Faculty of Economics and Business in Osijek  
Undergraduate Study Business Informatics

Ana Lozančić

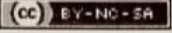
# **SYSTEMS REQUIREMENTS ELICITATION AND ANALYSIS**

Final paper

Osijek, 2023

## IZJAVA

### O AKADEMSKOJ ČESTITOSTI, PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA, SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je \_\_\_\_\_ završni  
(navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o visokom obrazovanju i znanstvenoj djelatnosti, NN 119/2022).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

**Ime i prezime studenta/studentice:** Ana Lozančić

**JMBAG:** 0113143366

**OIB:** 85451810309

**e-mail za kontakt:** lozancic.ana31@gmail.com

**Naziv studija:** Prijediplomski sveučilišni studij, smjer Poslovna informatika

**Naslov rada:** Prikupljanje i analiza sistemskih zahtjeva

**Mentor/mentorica rada:** doc. dr. sc. Dario Šebalj

U Osijeku, \_\_\_\_\_ 20.srpnja 2023. \_\_\_\_\_ godine

Potpis Ana Lozančić

## **Prikupljanje i analiza sistemskih zahtjeva**

### **SAŽETAK**

Sustav je skup međusobno povezanih elemenata koji djeluju zajedno radi postizanja određenog cilja. U kontekstu informacijskog sustava (IS), to je organizirana struktura koja omogućuje upravljanje poslovnim procesima putem prikupljanja, prijenosa i obrade informacija. Prikupljanje i analiza zahtjeva predstavljaju fazu u razvoju informacijskog sustava. Prilikom izgradnje sustava potrebno je razlikovati funkcionalne i nefunkcionalne zahtjeve, ograničenja te navesti sve specifikacije koje se očekuju od sustava. Najčešće metode prikupljanja zahtjeva u analizi sustava uključuju intervjuiranje korisnika i ostalih relevantnih dionika, organiziranje radionica s dionicima te anketiranje i promatranje. Također se spominje važnost prikazivanja prototipa sustava tijekom intervjua kako bi korisnici imali vizualno i interaktivno iskustvo te mogli dati povratne informacije. Iskustvo analitičara može biti korisno u dokumentiranju prethodnih procesa i pregledavanju kako bi se identificirale koristi i propusti za buduće projekte. Pri provedbi analize izvedivosti i postavljanju prioriteta sistemskih zahtjeva, ključno je procijeniti tehničku, financijsku i operativnu izvedivost projekta kako bi se odredili prioriteti zahtjeva. Ova analiza uključuje procjenu resursa, vremenskog okvira, troškova i mogućih rizika kako bi se odredili ključni zahtjevi koji će biti implementirani prvi. Cilj je osigurati da se najvažniji i izvedivi zahtjevi prvo zadovolje, a ostali se planiraju i raspoređuju prema dostupnim resursima.

**Ključne riječi:** informacijski sustav, funkcionalni zahtjevi, nefunkcionalni zahtjevi, analiza izvedivosti, prioritetni zahtjevi

# Systems Requirements Elicitation and Analysis

## ABSTRACT

A system is a collection of interconnected elements that work together to achieve a specific goal. In the context of a Business Information System (BIS), it is an organized structure that enables the management of business processes through the collection, transmission, and processing of information. The elicitation and analysis of system requirements represents a phase of information systems development. When developing a system, it is necessary to distinguish between functional and non-functional requirements, constraints, and specify all expected system specifications. In system analysis, the most common techniques for eliciting system requirements include interviewing users and relevant stakeholders, organizing workshops with stakeholders, and conducting surveys and observations. The importance of presenting a system prototype during interviews is also mentioned to provide users with a visual and interactive experience and to gather feedback. The analyst's experience can be valuable in documenting past processes and reviewing them to identify benefits and shortcomings for future projects. During the feasibility analysis and prioritization of system requirements, it is crucial to evaluate the project's technical, financial, and operational feasibility in order to determine the priorities of the requirements. This analysis involves evaluating resources, time frames, costs, and potential risks to identify key requirements that will be implemented first. The goal is to ensure that the most important and feasible requirements are met first, while the others are planned and allocated based on available resources.

**Keywords:** information system, functional requirements, non-functional requirements, feasibility analysis, requirement prioritization

# Sadržaj

1. Uvod .....	1
1.1. Metodologija rada.....	1
2. Izgradnja informacijskog sustava .....	2
2.1. Definicija sustava.....	2
2.2. Sudionici i razvoj informacijskog sustava.....	3
2.3. Faze životnog ciklusa razvoja IS-a.....	4
3. Sistemski zahtjevi.....	8
3.1. Vrste zahtjeva.....	8
3.2. Funkcionalni zahtjevi.....	8
3.2.1. Iznimke funkcionalnih zahtjeva.....	11
3.3. Nefunkcionalni zahtjevi .....	12
3.4. Ograničenja.....	14
4. Prikupljanje sistemskih zahtjeva.....	16
4.1. Metode prikupljanje sistemskih zahtjeva.....	16
4.1.1. Intervju.....	16
4.1.2. Radionice.....	18
4.1.3. Brainstorming.....	19
4.1.4. Promatranje .....	20
4.1.5. Anketiranje .....	21
4.1.6. Dokumentiranje.....	22
4.2. Postavljanje prioriteta zahtjeva .....	24
5. Osnovni zahtjevi u izradi IS-a .....	26
5.1. Vremenski okvir.....	26
5.2. Troškovi .....	27
5.3. Funkcionalnost.....	27
5.4. Sigurnost.....	28



<b>5.5. Zahtjevi vezani za kulturu tvrtke.....</b>	<b>29</b>
<b>5.6. Legalnost zahtjeva .....</b>	<b>30</b>
<b>5.7. Performanse .....</b>	<b>30</b>
<b>6. Zaključak.....</b>	<b>32</b>
<b>Literatura.....</b>	<b>34</b>
<b>Popis slika.....</b>	<b>36</b>
<b>Popis tablica .....</b>	<b>36</b>

## **1. Uvod**

Svrha ovog završnog rada je prikazati značaj analize i prikupljanja sistemskih zahtjeva te neophodnost koju zahtjevi predstavljaju za razvoj visokokvalitetnih informacijskih sustava. Prikupljanje zahtjeva uključuje identificiranje, razumijevanje i dokumentiranje potreba korisnika i poslovnih procesa koje će informacijski sustav morati podržavati. Ovaj proces obuhvaća interakciju s korisnicima, vođenje intervjuima, analizu postojećih sustava i dokumentacije, kao i primjenu drugih tehnika za prikupljanje relevantnih informacija. Nakon prikupljanja zahtjeva, slijedi analiza koja uključuje detaljniju procjenu i interpretaciju prikupljenih podataka. Cilj analize je razumjeti prirodu zahtjeva, identificirati potencijalne probleme i konflikte te pružiti temelje za dizajn informacijskog sustava koji će ispravno funkcionirati i zadovoljiti potrebe korisnika. Analiza sistemskih zahtjeva uključuje specifikaciju funkcionalnih i nefunkcionalnih zahtjeva, identifikaciju ograničenja sustava, definiranje prioriteta i pravljenje plana za daljnji razvoj informacijskog sustava. Ova faza osigurava da sve potrebne informacije budu jasno definirane i dokumentirane prije nego što se krene u fazu dizajna i implementacije. Prikupljanje i analiza sistemskih zahtjeva su temeljni koraci u stvaranju uspješnih informacijskih sustava koji zadovoljavaju potrebe korisnika i pridonose postizanju poslovnih ciljeva organizacije. Prvi dio ovog rada bit će usmjeren na proces izgradnje informacijskog sustava kako bi se pobliže objasnila početna faza oblikovanja IS-a, tko sve sudjeluje u procesu izrade sustava te koje se metode koriste prilikom komunikacije s klijentom prilikom priprema za razvoj informacijskog sustava. U drugom dijelu rada opisat će se specifikacija sistemskih zahtjeva te će se definirati njihove prednosti, rizici i nedostaci.

### **1.1. Metodologija rada**

U ovom radu, najzastupljenija metoda istraživanja korištena za prikupljanje podataka je analiza znanstvenih članaka, knjiga i internetskih izvora. Osnove informacijskog sustava su objašnjene kroz deskriptivne metode, dok su načini prikupljanja i analize sistemskih zahtjeva detaljnije objašnjeni analizom prikupljenih podataka. Također, kod usporedbe poslova analitičara i vrsta zahtjeva koristila se metoda komparacije.

## **2. Izgradnja informacijskog sustava**

Izgradnja informacijskog sustava je proces stvaranja ili razvoja organizirane strukture koja omogućuje prikupljanje, obradu, pohranu, prijenos i upravljanje informacijama u svrhu podrške poslovnim procesima ili ostvarivanja određenih ciljeva. Ovaj proces uključuje aktivnosti kao što su analiza zahtjeva, projektiranje arhitekture sustava, programiranje, testiranje, implementacija i održavanje sustava. Cilj izgradnje informacijskog sustava je stvoriti funkcionalan i pouzdan sustav koji zadovoljava potrebe korisnika i pridonosi poboljšanju poslovnih procesa i donošenju odluka (Avison i Fitzgerald, 1995).

### **2.1. Definicija sustava**

Informacijski sustav (IS) predstavlja visoko organiziranu strukturu i temelj za upravljanje organizacijom poslovnog subjekta. Njegova glavna svrha je nadziranje i upravljanje poslovnim procesima putem prikupljanja, prijenosa i obrade informacija kako bi se stvorile nove spoznaje koje postaju dio arhive poslovnog subjekta i primjenjuju se u daljnjem poslovanju. Kroz primjenu IS-a, stvara se neprocjenjivo iskustvo i znanje koji postaju nematerijalna imovina poslovnog subjekta, a to se postiže suradnjom između informacijskog sustava i zaposlenika koji njime upravljaju (Ward i Peppard, 2002).

Informacijski sustav omogućava učinkovito prikupljanje, pohranjivanje, analizu i interpretaciju podataka koji se generiraju tijekom izvršavanja poslovnih procesa. Podaci se prikupljaju iz različitih izvora unutar organizacije, kao i iz vanjskih izvora, te se prenose kroz informacijski sustav kako bi se omogućila brza i točna obrada. Na temelju tih podataka, donositelji odluka mogu dobiti nove informacije i uvide koji im pomažu u vođenju organizacije, planiranju, identificiranju trendova i pravovremenom reagiranju na promjene u poslovnom okruženju (Drljača, 2016).

Upravljanje informacijskim sustavom uključuje dizajn, razvoj, implementaciju i održavanje sustava. Zaposlenici koji upravljaju IS-om igraju ključnu ulogu u osiguravanju pouzdanosti, sigurnosti i integriteta podataka. Oni su odgovorni za postavljanje i prilagođavanje sustava prema specifičnim potrebama organizacije, kao i za pružanje podrške korisnicima, obuku i rješavanje problema koji se mogu pojaviti tijekom korištenja sustava (Bourgeois, 2019).

Kroz sinergiju između informacijskog sustava i zaposlenika, organizacija stvara vrijednost i konkurentne prednosti. Sustav omogućava brže donošenje odluka, optimizaciju poslovnih

procesa, poboljšanje učinkovitosti i produktivnosti te bolje iskorištavanje raspoloživih resursa. Uz to, informacijski sustav podržava rast i inovacije organizacije, omogućavajući prilagodljivost i konkurentnost na tržištu (Bourgeois, 2019).

## **2.2. Sudionici i razvoj informacijskog sustava**

Iz tradicionalne perspektive, dostupne su dvije opcije. Tvrtka, ili posjeduje vlastito razvijeni informacijski sustav, tzv. *in-house* sustav, ili je kupljen gotov sustav, odnosno softverski paket, od dobavljača. Danas, izbor je puno složeniji. Opcije uključuju internetske aplikacijske usluge, outsourcing, prilagođena rješenja od IT konzultanata i strategije softvera za cijelo poduzeće. Bez obzira na način razvoja, razvoj novog informacijskog sustava nosi rizike i koristi. Najveći rizik javlja se kada tvrtka pokušava odlučiti kako će sustav biti konstruiran prije nego što utvrdi što sustav treba raditi. Ključno je da poslovni subjekt mora započeti s definiranjem svojih poslovnih potreba i identificiranjem mogućih IT rješenja. Ovaj važan posao obično obavljaju analitičari sustava i drugi IT stručnjaci. Tvrtka ne bi trebala razmatrati opcije implementacije dok nema jasan skup ciljeva. Kasnije, kako se sustav razvija, uloga analitičara sustava će varirati ovisno o odabranoj opciji implementacije. U samom procesu razvoja informacijskog sustava, klijent u dogovoru s dobavljačem predstavlja zahtjeve koji obuhvaćaju definiranje ciljeva, resursa, vremenskih rokova i aktivnosti koje su potrebne za razvoj aplikacije. Iza svega također stoje uprava i vlasnici koji postavljaju viziju, ciljeve i strategiju za informacijski sustav te osiguravaju financijska sredstva i podršku za njegov razvoj (Robertson i Robertson, 2013).

Sistem analitičar i poslovni analitičar dio su tima koji razvija informacijski sustav. Glavna razlika između sistem analitičara i poslovnog analitičara leži u njihovim fokusima. Sistem analitičar se usredotočuje na tehničke aspekte, poput definiranja funkcionalnosti i izrade tehničkih specifikacija, dok poslovni analitičar ima naglasak na razumijevanju poslovnih procesa, definiranju poslovnih zahtjeva i postizanju poslovnih ciljeva, što je detaljnije prikazano u tablici 1. Upravo zbog različitih fokusa, sistem analitičar i poslovni analitičar često surađuju kako bi osigurali da informacijski sustav zadovoljava tehničke i poslovne zahtjeve organizacije. Njihova suradnja je ključna za uspješan razvoj i implementaciju informacijskog sustava koji pruža vrijednost organizaciji i korisnicima (de Bree, n.d.)

## SISTEM ANALITIČAR

## POSLOVNI ANALITIČAR

Detaljna analiza tehničkih zahtjeva.	Razumijevanje poslovnih procesa i potreba tvrtke.
Dizajn arhitekture sustava.	Identifikacija načina na koje informacijski sustav može podržati i unaprijediti poslovne procese.
Identifikacija potrebnih tehnologija i alata za izgradnju sustava.	Suradnja s korisnicima sustava kako bi se identificirali zahtjevi i specifikacije.
Uspostavljanje usklađenosti između tehničkih rješenja i poslovnih zahtjeva.	Dokumentiranje poslovnih pravila i procesa.
Suradnja s poslovnim analitičarima radi osiguravanja da tehničko rješenje odgovara poslovnim zahtjevima.	Priprema specifikacija za razvoj sustava.

Tablica 1. Razlika sistemskog i poslovnog analitičara

Izvor: de Bree (n.d.)

Predmet ovog završnog rada je proces prikupljanja i analize sistemskih zahtjeva te je u tom kontekstu bitno naglasiti važnost sistem i poslovnog analitičara. Njihovom suradnjom zahtjevi se identificiraju i analiziraju kako bi se mogle identificirati potrebne funkcionalnosti, performanse i karakteristike informacijskog sustava.

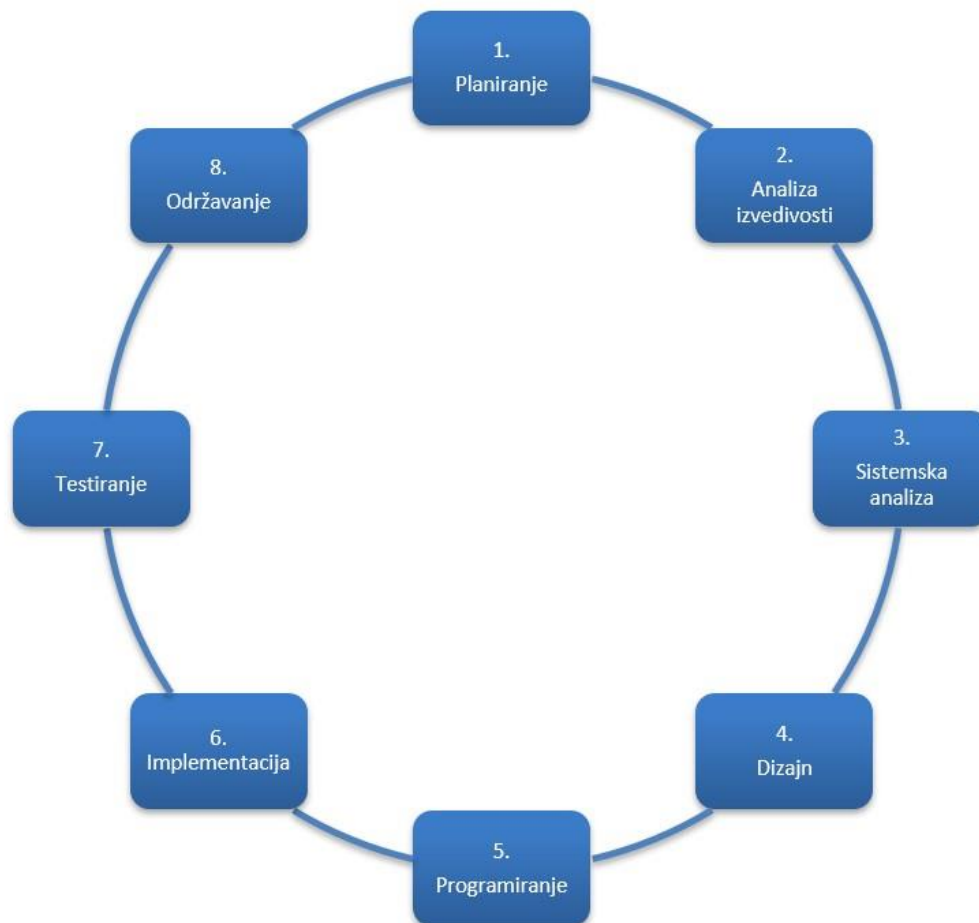
### 2.3. Faze životnog ciklusa razvoja IS-a

Životni ciklus razvoja softvera (eng. *Software Development Life Cycle - SDLC*) je temelj za strukturiranje, planiranje i izvođenje zadataka koji su uključeni u razvoj informacijskog sustava. Zbog razlika u zahtjevima sustava postoje varijacije SDLC-a, te će u ovom radu biti prikazana proširena verzija zbog naglaska na prikupljanje i analizu zahtjeva (Lemke, 2018).

Na slici 1. prikazane su faze životnog ciklusa razvoja softvera. Faza planiranja u razvoju informacijskog sustava je ključna faza koja postavlja temelje projekta. U ovoj fazi se identificiraju ciljevi projekta, analiziraju zahtjevi korisnika i definira opseg projekta. Također

se planiraju resursi potrebni za uspješno izvođenje projekta, uključujući ljude, financije, opremu i infrastrukturu. Cilj ove faze je stvoriti sveobuhvatan plan koji će osigurati uspješnu izvedbu projekta i postizanje željenih rezultata. Planiranje resursa i definiranje jasnog opsega projekta su ključni elementi u ovoj fazi. U fazi analize izvedivosti provodi se pregled zahtjeva kako bi se utvrdila izvedivost projekta. Svrha je doći do zaključka o mogućnosti stvaranja adekvatnog rješenja za zadovoljenje zahtjeva te se istražuju alternative. Izvedivost ovisi o tehničkom, ekonomskom i pravnom kriteriju. Ako se utvrdi da je projekt izvediv i ima pozitivne ishode, može se započeti s daljnjim fazama razvoja informacijskog sustava. Zatim slijedi sistemska analiza. Jedan ili više analitičara sustava surađuju s različitim skupinama dionika kako bi identificirali specifične zahtjeve za novi sustav. Važno je napomenuti da se u ovoj fazi ne provodi programiranje. Umjesto toga, provodi se dokumentiranje postupaka, intervjuiranje ključnih sudionika i razvijanje zahtjeva za podacima kako bi se stekao cjelovit uvid u ono što sustav treba postići. Cilj ove faze je stvoriti dokument sa svim zahtjevima, što uključuje funkcionalnosti, korisnička sučelja, poslovne procese, sigurnost, performanse i druge aspekte koji su bitni za razvoj novog sustava. Iduća faza je faza dizajna, gdje dizajner preuzima dokument koji je izrađen u prethodnoj fazi te na osnovu identificiranih zahtjeva razvija specifične tehničke detalje potrebne za implementaciju sustava. U ovoj fazi se poslovni zahtjevi pretvaraju u konkretne tehničke zahtjeve. Nakon toga slijedi faza programiranja, koristeći dokument o dizajnu sustava kao smjernicu, tim programera započinje razvoj programskog koda. U ovoj fazi se piše kod koji će implementirati funkcionalnosti i logiku definiranu u prethodnim fazama. Korištenjem programskog jezika i razvojnih alata dizajn sustava se pretvara u stvaran radni program. Ovaj program će se temeljiti na zahtjevima i specifikacijama identificiranim u analizi sustava te će biti u skladu s dizajnom razvijenim u fazi projektiranja sustava. Prije implementacije sustava potrebno je provesti fazu testiranja, pri čemu razvijeni program polazi kroz niz testova. Prvo se zasebni dijelovi koda testiraju na pogreške, zatim se testiraju komponente sustava kako bi se ustanovio rad cjelokupnog sustava. Konačni test obavljaju korisnici, a njime se utvrđuje zadovoljava li sustav njihove standarde. Pogreške i problema pronađeni prilikom testiranja se rješavaju te nakon toga ponovo testiraju. Nakon što je sustav razvijen i testiran, implementira se u organizaciju. Kako bi se osiguralo uspješno uvođenje sustava, potrebno je provesti nekoliko faza implementacije. Jedna od tih aktivnosti je obuka korisnika. Korisnici sustava trebaju biti upoznati s funkcionalnostima i načinom rada novog sustava kako bi ga mogli učinkovito koristiti. Obuka može uključivati radionice, treninge ili individualnu podršku, ovisno o potrebama korisnika. Također je važno pružiti dokumentaciju koja opisuje funkcionalnosti, upute za korištenje i rješavanje problema vezanih uz sustav.

Kvalitetna dokumentacija omogućava korisnicima da se bolje upoznaju s sustavom te služi kao referenca za buduće upite ili potrebe (Shelly i Rosenblatt, 2012).



Slika 1. Životni ciklus razvoja softvera

Izvor: Izrada autorice prema Shelly i Rosenblatt (2012)

Ukoliko je tvrtka već posjedovala sustav i uvodi se novi, potrebno je provesti konverziju podataka. Potrebno je prebaciti podatke i postaviti novu konfiguraciju kako bi novi sustav funkcionirao s postojećim podacima i procesima. Cilj ove faze je uspješno implementirati novi sustav u organizaciju i osigurati osposobljavanje korisnika za korištenje sustava. Nakon implementacije potrebno je održavati sustav i osigurati dugoročnu održivost, stabilnost i unapređenje. Redovito rješavanje grešaka, implementacija novih značajki i ažuriranje sustava osiguravaju da sustav bude pouzdan, siguran i prilagodljiv u skladu s promjenama potreba i tehnološkim zahtjevima organizacije. Redovito ažuriranje sustava također je važan dio ove faze. To uključuje primjenu sigurnosnih zakrpa, nadogradnje softverskih komponenti i održavanje operativne i tehničke infrastrukture. Održavanje sigurnosnih kopija podataka

također se provodi redovito kako bi se osigurala zaštita podataka i mogućnost oporavka u slučaju neželjenih događaja (Shelly i Rosenblatt, 2012).



### **3. Sistemski zahtjevi**

Sistemski zahtjevi su definirani skupovi uvjeta, karakteristika i funkcionalnosti koje informacijski sustav mora zadovoljiti kako bi ispunio svoju svrhu. Oni precizno opisuju očekivanja i specifikacije vezane za tehničke aspekte sustava. Glavna svrha sistemskih zahtjeva je osigurati da informacijski sustav bude razvijen i implementiran na način koji zadovoljava potrebe korisnika, poslovnih zahtjeva i tehničkih ograničenja. Oni pružaju smjernice i temelj za dizajn, razvoj, testiranje i održavanje sustava (Grady, 2016).

#### **3.1. Vrste zahtjeva**

Prikupljanje zahtjeva odvija se fazi analize, a dijele se na funkcionalne i nefunkcionalne. Funkcionalni zahtjevi opisuju specifične funkcionalnosti i zadatke koje informacijski sustav treba obavljati kako bi podržao poslovne procese ili zadovoljio korisničke potrebe. Ovi zahtjevi opisuju što sustav mora raditi, koje radnje treba izvršavati, na koji način treba obraditi podatke itd. Primjeri funkcionalnih zahtjeva uključuju stvaranje korisničkih računa, unos podataka, pretraživanje i filtriranje podataka i slično. Nefunkcionalni zahtjevi odnose se na karakteristike i ograničenja informacijskog sustava koja nisu povezana s njegovim izravnim funkcionalnostima. Obuhvaćaju aspekte performansi, sigurnosti, pouzdanosti, dostupnosti, skalabilnosti, upotrebljivosti, održivosti i drugih sličnih atributa sustava. Primjeri nefunkcionalnog zahtjeva vezani su za intuitivnost sučelja, minimalno vremensko kašnjenje, sigurnost, održivost i slično (Grady, 2016).

#### **3.2. Funkcionalni zahtjevi**

Funkcionalni zahtjevi određuju radnje i ponašanja koja proizvod mora pokazivati kako bi ispunio namjeravanu svrhu. Oni opisuju što proizvod treba učiniti kako bi ispunio temeljne ciljeve za koje je dizajniran. Ovi zahtjevi važni su u razvojnom procesu jer prenose programerima što treba izgraditi. Funkcionalni zahtjevi usmjereni su na funkcionalnost proizvoda i idealno su izraženi neovisno o bilo kojoj specifičnoj tehnologiji koja se koristi za implementaciju. Ovo odvajanje omogućuje poslovnom analitičaru da artikulira što bi tehnološko rješenje trebalo postići, dok detalje o tome kako se to postiže ostavlja dizajneru. Kako bi učinkovito vodili razvojni proces, funkcionalni zahtjevi moraju pružati dovoljno detalja kako bi programer mogao točno izraditi proizvod uz minimalna pojašnjenja ili

objašnjenja. Važno je napomenuti da određena razina pojašnjenja ipak može biti potrebna, jer potpuni izostanak pitanja od strane programera može ukazivati na pretjerano detaljne specifikacije. Jasnim definiranjem funkcionalnih zahtjeva, poslovni analitičar osigurava razvoj ispravnog proizvoda i zadovoljenje potreba dionika, istovremeno dopuštajući fleksibilnost pri odabiru dizajna i implementacije koje donosi razvojni tim (Davies, n.d.).

U nastavku je prikazan primjer funkcionalnog zahtjeva (Robertson i Robertson, 2013):

*„Sustav treba moći analizirati povijesne podatke o prodaji vozila, uključujući informacije o modelima, godinama, regijama i ostalim relevantnim faktorima.“*

Na primjeru ovog zahtjeva će biti objašnjen scenarij slučaja korištenja.

Funkcionalnost proizvoda može se opisati pomoću nekoliko stavki. Najbitnija stavka je scenarij. Scenarij se dobije podjelom posla na dijelove pomoću poslovnih događaja koji utječu na njih. Za svaki poslovni događaj postoji slučaj korištenja (eng. *Use case*) koji se podudara s poslovnim događajem te opisuje scenarij poslovnog slučaja korištenja. Iz ovog scenarija proizlazi ponekad jedan, a ponekad više slučaja korištenja.

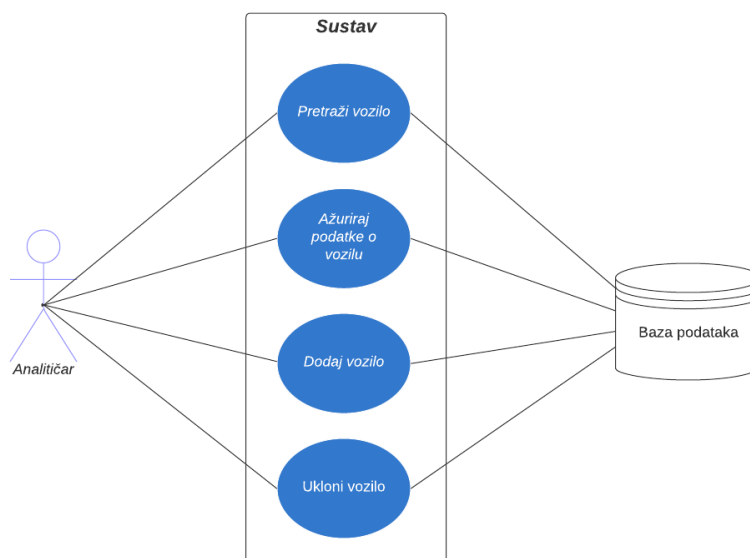
Scenarij je najprikladniji način rada sa sudionicima kako bi se odredila potrebna funkcionalnost za slučaj korištenja proizvoda. Svaki korak scenarija sastoji se od više koraka koji predstavljaju funkcionalne zahtjeve. Kada se skupe svi zahtjevi dobivamo krajnju svrhu proizvoda i što sve mora odraditi kako ispunio slučaj korištenja proizvoda.

Na prethodno navedenom zahtjevu bit će objašnjeno što sve proizvod mora obaviti kako bi zadovoljio potrebe korisnika. Idući scenarij objašnjava slučaj korištenja proizvoda (Robertson i Robertson, 2013):

- Analitičar prikuplja povijesne podatke o prodaji vozila iz baze podataka automobilske tvrtke. Podaci obuhvaćaju informacije o prodajnim transakcijama, uključujući modele vozila, godine, regije, cijene i ostale relevantne parametre.
- Analitičar provodi obradu prikupljenih podataka kako bi uklonio nedostajuće ili nevaljane vrijednosti.
- Na temelju pripremljenih podataka, analitičar primjenjuje statističke metode, analitičke algoritme i vizualizacijske tehnike za identificiranje prodajnih trendova.

- Analitičar provodi analizu korelacija između prodajnih podataka i ostalih faktora poput marketinških aktivnosti, cijena, konkurencije ili gospodarskih uvjeta.
- Analitičar uređuje bazu podataka, gdje pretražuje, dodaje, briše i ažurira vozila ili podatke o vozilima.

U analizi scenarija poželjno je imati najmanje tri, a najviše deset koraka jer se na taj način daje dovoljno detalja bez pretjerane kompliciranosti. U idućem dijagramu korištenja prikazan je zadnji korak analize scenarija.



Slika 2. Dijagram korištenja na osnovi analize scenarija

Izvor: Izrada autorice

U nastavku će biti prikazano nekoliko primjera pravilnog definiranja funkcionalnih zahtjeva (Robertson i Robertson, 2013).

Ukoliko je potrebno napraviti informacijski sustav za bolnicu, medicinski stručnjak treba imati pristup povijesti bolesti, dijagnozama, terapijama i ostalim relevantnim podacima o pacijentu.

*„Sustav bolnice treba omogućiti upisivanje i spremanje medicinske povijesti pacijenata.“*

Sustav bolnice treba omogućiti unos i pohranu tih podataka kako bi se osigurala cjelovitost i dostupnost informacija o pacijentu. Funkcionalni zahtjev u ovom slučaju glasi:

Ako se radi informacijskom sustavu za e-trgovinu, potrebno je omogućiti informacije o trenutnom statusu isporuke pri naručivanju proizvoda. Sustav e-trgovine treba omogućiti

praćenje isporuke putem dostavnog broja ili drugih relevantnih podataka. Funkcionalni zahtjev glasi:

*„Korisnik e-trgovine treba moći pratiti status isporuke svoje narudžbe.“*

U slučaju da se radi o informacijskom sustavu kojem će moći pristupiti samo djelatnici poslovnog subjekta i njegovi sudionici koji dobiju pristup, potrebno je postaviti idući funkcionalni zahtjev:

*„Sustav treba omogućavati korištenje samo od strane autoriziranih osoba.“*

U slučaju da se radi o fakultetu, studenti trebaju moći pristupiti ispitu. Kako se ispitu pristupa prijavljivanjem u informacijski sustav fakulteta, pri njegovoj izradi potrebno postaviti iduće funkcionalne zahtjeve:

*„Korisnik se treba moći prijaviti u sustav koristeći svoje korisničko ime i lozinku.“*

U sustavu automobila treba biti dostupna mogućnost korištenja karte i navigacije prilikom putovanja. Funkcionalni zahtjev glasi:

*„Sustav treba pružati navigacijske usluge i upute za vožnju, što uključuje prikazivanje karte, praćenje trenutne lokacije vozila, davanje uputa za skretanje i prikazivanje prometnih informacija.“*

Iz primjera se može vidjeti kako funkcionalni zahtjevi opisuju ponašanje i informacije uspješnog rješenja, te mogućnosti koje će sustav moći izvesti u smislu ponašanja ili operacije (Robertson i Robertson, 2013).

### 3.2.1. Iznimke funkcionalnih zahtjeva

Ukoliko zahtjev dolazi u obzir samo u specifičnim okolnostima, potrebno je dodati uvjet zahtjevu. Ovaj pristup se vrši ukoliko se uvjeti rijetko pojavljuju u specifikaciji. Korištenje riječi „ako“ u uvjetu je nezgodno, pa je preporučeno pisati zahtjeve u imperativu, poput „proizvod mora...“. O ovome pristupu treba upozoriti programere jer je potrebno da se zahtjevi čitaju zajedno sa scenarijem kako bi se mogli razlikovati uvjetni zahtjevi od onih koji to nisu (Robertson i Robertson, 2013).

Tehnološki zahtjevi su zahtjevi koji proizlaze iz odabrane tehnologije i služe za osiguranje ispravnog funkcioniranja implementacije. Oni su neovisni o poslovnim razlozima i nužni su za podržavanje odabrane tehnološke infrastrukture. Kako bi se jasno razumjelo zašto se ti zahtjevi pojavljuju, preporučuje se da budu zabilježeni u posebnoj specifikaciji ili jasno označeni kao tehnološki zahtjevi, zajedno s poslovnim zahtjevima. Važno je da dionici u potpunosti razumiju poslovne zahtjeve prije nego što se uvode tehnološki zahtjevi. Ukratko, tehnološki zahtjevi su specifični za odabranu tehnologiju i osiguravaju ispravno funkcioniranje implementacije (Robertson i Robertson, 2013).

Grupiranje funkcionalnih zahtjeva prema slučajevima korištenja omogućuje otkrivanje povezanih skupina zahtjeva i testiranje cjelovitosti funkcionalnosti. Međutim, ponekad se može pokazati korisnim grupiranje prema značajkama. Značajka može biti manja ili veća i imati različit stupanj vrijednosti za organizaciju. Grupiranje zahtjeva po značajkama olakšava njihovu manipulaciju i prilagodbu specifikacije kada se tržište ili zahtjevi marketinga promijene. Važno je napomenuti da značajka obično uključuje zahtjeve iz više različitih slučajeva korištenja proizvoda. Atomički zahtjevi predstavljaju najnižu razinu specifikacije zahtjeva i mogu se grupirati u hijerarhiju zahtjeva iz nekoliko razloga. To omogućuje uključivanje dionika s različitim interesima i dubinom razumijevanja, pomaže u otkrivanju atomske zahtjeve i nosi se s volumenom i složenošću zahtjeva. Preporučuje se raditi s nesubjektivnom hijerarhijom zahtjeva koja uključuje sljedeće razine: radni kontekst, poslovni događaj, slučaj korištenja proizvoda, korak slučaja korištenja proizvoda i atomski zahtjev. Ova praksa izbjegava sukobe i argumente koji proizlaze iz subjektivnih "visokih" i "niskih" razina zahtjeva. Ukratko, grupiranje zahtjeva prema značajkama i rad s nesubjektivnom hijerarhijom zahtjeva omogućuje bolje upravljanje i razumijevanje zahtjeva, olakšava prilagodbu specifikacije i osigurava dosljednost i jasnoću u specifikaciji zahtjeva (Robertson i Robertson, 2013).

### **3.3. Nefunkcionalni zahtjevi**

Nefunkcionalni zahtjevi predstavljaju svojstva ili kvalitetu koje proizvod mora imati kako bi bio prihvatljiv vlasniku i operatoru. U nekim slučajevima, nefunkcionalni zahtjevi - koji specificiraju svojstva poput performansi, izgleda, upotrebljivosti, sigurnosti i zakonskih atributa - ključni su za uspjeh proizvoda (Robertson i Robertson, 2013).

Ranije navedeni funkcionalni zahtjev odnosio se na sustav koji analizira podatke. Nefunkcionalni zahtjev bio bi da sustav mora analizirati podatke unutar tri sekunde, a analizirane podatke ne može mijenjati nitko osim analitičara. Ova zadatka su nefunkcionalni, prvi predstavlja zahtjev performanse, a drugi predstavlja zahtjev sigurnosti. Ne predstavljaju funkcionalni dio proizvoda, no neophodni su za njegovu funkcioniranje. Ne postoji dogovorena podjela nefunkcionalnih zahtjeva, može ih raditi ovisno o situaciji i projektu. Iduća podjela koja će biti navedena služi za bolje razumijevanje, otkrivanje i snalaženje u nefunkcionalnim zahtjevima (Robertson i Robertson, 2013):

- Zahtjevi o izgledu i osjećaju – izgled proizvoda.
- Zahtjevi upotrebljivosti i prilagođenosti – jednostavnost korištenja proizvoda i faktori koji utječu na korisničko iskustvo.
- Zahtjevi performansi – brzina, dostupnost i točnost funkcionalnosti.
- Operativni zahtjevi – radno okruženje proizvoda i faktori koji su bitni za to okruženje.
- Zahtjevi održivost i podrške – promjene i vrijeme potrebno za njihovo izvršenje te specifikacija podrške koja će biti pružena proizvodu.
- Zahtjevi o sigurnosti – pristup i povjerljivost proizvoda.
- Kulturološki i politički zahtjevi – zahtjevi specifični za kulturu ljudi koji će doći u dodir s proizvodom.
- Pravni zahtjevi – zakoni i standardi koji se odnose na proizvod.

U nastavku će biti prikazano nekoliko primjera pravilnog definiranja nefunkcionalnih zahtjeva (Robertson i Robertson, 2013)

Sustav bolnice treba pružati visoku razinu sigurnosti i zaštite podataka kako bi se osigurala povjerljivost, integritet i dostupnost medicinskih informacija. To uključuje zaštitu osobnih podataka pacijenata, medicinske povijesti, dijagnoza, terapija i drugih osjetljivih informacija od neovlaštenog pristupa, krađe, manipulacije ili gubitka podataka. Nefunkcionalni zahtjev u ovom slučaju glasi:

*„Sustav bolnice treba pružati visoku razinu sigurnosti podataka.“*

U slučaju e-trgovine, potrebnih su nefunkcionalni zahtjevi poput idućih:

*„Sustav e-trgovine treba imati intuitivno sučelje koje omogućava jednostavnu navigaciju, pregled proizvoda, postavljanje narudžbi i plaćanje.“*

*„Sučelje sustava treba biti prilagođeno različitim uređajima i veličinama zaslona.“*

Informacijski sustav fakulteta treba biti dostupan korisnicima tijekom cijelog kako bi mogli pristupiti podacima i obavljati radnje u bilo koje vrijeme. Funkcionalni zahtjev za ovim karakteristikama glasi:

*„Sustav treba biti dostupan 24/7 kako bi korisnici (studenti, nastavnici, administrativno osoblje) mogli pristupiti podacima i obavljati svoje zadatke u bilo koje vrijeme.“*

Prilikom izgradnje bilo kojeg proizvoda, moguće ga je učiniti jednostavnim i omogućiti da bude svima na raspolaganju za korištenje čak i osobama s ograničenim poznavanjem korištenja kompjutera ili laptopa.

*„Proizvod će biti moguće koristiti korisnicima s ograničenim iskustvom korištenja računala.“*

Iz navedenih primjera može se primijetiti kako nefunkcionalni zahtjevi prikazuju opis okoline uvjeta pod kojima rješenje mora ostati učinkovito (Robertson i Robertson, 2013).

### **3.4. Ograničenja**

Ograničenja su jedan od oblika zahtjeva. Mogu se odnositi na ograničavanje samog projekta ili na proizvod. Ograničenja određuju što sistemski zahtjevi ne smiju premašiti ili kršiti. Ograničenja mogu biti tehničke, resursne, vremenske, operativne ili zakonske prirode. Navedene su neki primjeri gdje se ograničenja mogu pojaviti (Shelly i Rosenblatt, 2012):

- Tehnička ograničenja: Specifični hardverski ili softverski zahtjevi, kompatibilnost s postojećim tehnologijama, maksimalni kapacitet ili performanse koje sustav treba podržati.
- Resursna ograničenja: Raspoloživost određenih resursa poput procesorskog vremena, memorije, mrežne propusnosti ili prostora za pohranu.

- Vremenska ograničenja: Rokovi za dostavu sustava ili njegovih komponenti, vremenski okviri za odgovor na zahtjeve ili minimalno vrijeme potrebno za izvršavanje određenih operacija.
- Operativna ograničenja: Pravila i politike koje reguliraju korištenje sustava, sigurnosni zahtjevi, propisi vezani uz privatnost ili zaštita podataka.
- Zakonska ograničenja: Zakonski propisi ili standardi koji moraju biti poštovani prilikom dizajna i izgradnje sustava, kao što su zahtjevi za zaštitu korisničkih podataka, autorska prava ili pravila o sigurnosti.



## **4. Prikupljanje sistemskih zahtjeva**

Prikupljanje sistemskih zahtjeva je proces identificiranja, dokumentiranja i razumijevanja zahtjeva koji se odnose na razvoj novog informacijskog sustava. Sistemske zahtjeve definiraju korisnici, poslovni analitičari ili analitičari sustava kako bi osigurali da sustav ispunjava potrebne funkcionalnosti, performanse, sigurnosne zahtjeve i druge karakteristike.

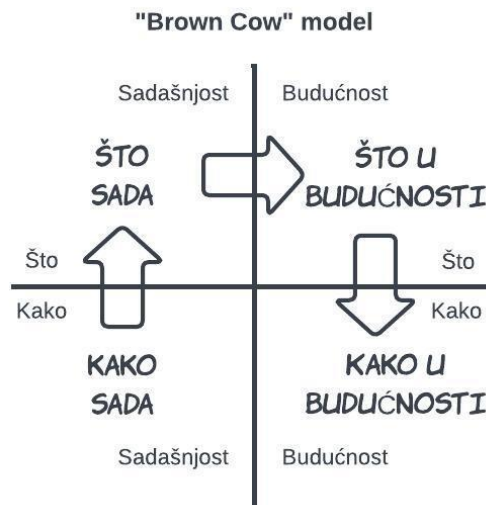
### **4.1. Metode prikupljanje sistemskih zahtjeva**

Najčešća metoda prikupljanja zahtjeva je upravo intervju. Analitičar obavlja razgovor sa korisnicima, djelatnicima i relevantnim osobama kako bi se mogle identificirati potrebe, očekivanja i zahtjevi kako bi se primijenili na sustav. Kroz razgovor i dodatna pitanja, najčešće unaprijed pripremljena analitičar dolazi do okvirne slike sustava koji je potrebno izraditi za klijenta. Ukoliko je malo vremena na raspolaganju, postoji i opcija radionica. Organiziraju se sastanci s različitim dionicima kako bi se raspravljalo o zahtjevima, identificirale potrebe i rješavale eventualne konflikte ili nesporazume. Radionice omogućuju grupno sudjelovanje i doprinose više perspektiva. Postoji i mogućnost anketiranja, ali se preporučuje prikupljanje zahtjeva konkretnim razgovorom. Dodatan način uz razgovor je promatranje. Promatranjem analitičar će steći uvid u stvarne potrebe, radne uvjete i probleme. Time može doći do vrijednih informacija o stvarnim korisničkim zahtjevima koji se možda ne izražavaju verbalno. Prilikom intervjua/sastanka pokazivanje prototipa sustava također može olakšati posao. Korisnicima je omogućeno vizualno i interaktivno iskustvo funkcionalnosti i sučelja sustava te otvara mogućnost povratne informacije i prijedloga za poboljšanje. Ukoliko analitičar ima prethodno iskustvo, može ga iskoristiti na način da dokumentira prijašnje procese i pregledava ih na svakom novom projektu, jer time može imati uvid u koristi i propuste koji su se već bili zastupljeni (Shelly i Rosenblatt, 2012).

#### **4.1.1. Intervju**

Intervju predstavlja najčešći oblik prikupljanja zahtjeva. Pitanja i teme koje je potrebno postaviti i raspraviti mogu se također primijeniti na radionice i sastanke. Kako bi se osiguralo postavljanje pravih pitanja, potrebno je imati osnovu za ta pitanja. Jedan od modela je tzv. model „*Brown Cow*“. Ovaj model prikazuje istraživanje sa četiri točke gledišta. Odnosi se na

dvije dimenzije, sadašnjost i budućnost. Za obje perspektive postavljaju se dva pitanja, a to su „Što?“ i „Kako?“ (Robertson i Robertson, 2013).



Slika 3. "Brown Cow" model

Izvor: Robertson i Robertson (2013)

Prema Robertson i Robertsonu (2013), ovaj model promatra sadašnjost koja predstavlja kakvo je trenutno stanje i predviđanje kakvo će stanje biti u budućnosti. „*Brown Cow*“ model služi kao osnova za prikupljanje zahtjeva jer ima cilj postavljanja trenutnih i mogućih problema te postavljanje strategija za njihovo rješavanje. Ovaj model je napredni oblik modela koji prikazuje situaciju onakvom kakva je i kakva će biti. No, te dvije perspektive uglavnom nisu dovoljne, pa „*Brown Cow*“ model daje prošireni pogled i daje više mogućnosti. Početna točka je kvadrant „*How-Now*“ predstavlja implementaciju trenutnog poslovanja. Ovaj pogled se koristi kada je potrebna osnova za postavljanje novih pitanja. Cilj je stvoriti sustav koji ne kopira prijašnji i uvidjeti čistu sliku poslovnog problema. Kvadrant „*What-Now*“ daje sliku o trenutnom poslovanju tvrtke. Ovo je neutralan pogled koji prikazuje poslovanje bez strojeva, opreme, ljudi i organizacije. „*Future-What*“ kvadrant prikazuje poslovanje kakvo klijent želi imati, no i dalje bez tehnologije koja bi mogla biti implementirana u poslovanju. Zadnji kvadrant, „*Future-How*“ prikazuje idealizirani pogled na buduće poslovanje koji se zatim nadograđuje sa tehnologijom i ljudima potrebnim za njegovo stvaranje.

Prilikom postavljanja pitanja potrebno je biti dobar slušatelj, razumjeti odgovore i znati ih protumačiti. Prilagodba sudionicima prilikom intervjuiranja, radionice ili sastanka, te mogućnost razumijevanja tuđe perspektive povećavaju šanse razvitka točnijeg informacijskog sustava (Robertson i Robertson, 2013).

Shelly i Rosenblatt (2012) smatraju kako bi intervju trebao sadržavati nekoliko vrsta pitanja: otvorena pitanja, zatvorena pitanja ili pitanja s rasponom odgovora. Pri formuliranju pitanja, treba izbjegavati upitna pitanja koja sugeriraju odgovor. Na primjer, umjesto: *"Koje prednosti vidite u predloženom sustavu?"*, bolje je pitati: *"Vidite li neke prednosti u predloženom sustavu?"*.

Otvorena pitanja potiču spontane odgovore. Takva pitanja su korisna kada želite razumjeti veći proces, izvući mišljenja i prijedloge ispitanika. Evo nekoliko primjera otvorenih pitanja: *„Što korisnici kažu o novom sustavu?“*, *„Kako se obavlja ova zadaća?“*, *„Zašto izvršavate zadatak na taj način?“*. Također, mogu se koristiti otvorena pitanja za dodatna istraživanja: *„Postoji li nešto dodatno što možete reći o ovoj temi?“*.

Zatvorena pitanja ograničavaju odgovor. Zatvorena pitanja koriste se kada je potrebno saznati specifične informacije ili provjeriti činjenice. Primjeri zatvorenih pitanja uključuju: *„Koliko osobnih računala imate u ovom odjelu?“*, *„Pregledavate li izvještaje prije nego što ih pošaljete?“*, *„Koliko sati obuke prima službenik?“*, *„Je li postupak korištenja opisan u priručniku?“*, *„Koliko je kupaca prošlog mjeseca naručilo proizvode putem web stranice?“*.

Pitanja s rasponom odgovora su zatvorena pitanja koja traže da osoba ocijeni nešto pružanjem ograničenih odgovora na određene reakcije ili numeričkoj skali. Ovaj pristup olakšava odgovaranje i interpretaciju rezultata. Pitanja s rasponom odgovora mogu uključivati sljedeće: *„Na skali od 1 do 10, pri čemu je 1 najniža ocjena, kako biste ocijenili ozbiljnost problema - niska, srednja ili visoka?“*, *„Događa li se gašenje sustava nikada, ponekad, umjereno, često ili uvijek?“* (Shelly i Rosenblatt, 2012).

#### 4.1.2. Radionice

Radionica za prikupljanje sistemskih zahtjeva je metoda koja se koristi u procesu razvoja softvera kako bi se identificirali, analizirali i dokumentirali zahtjevi koji se odnose na informacijski sustav ili softversku aplikaciju. Ova metoda uključuje aktivno sudjelovanje različitih dionika, kao što su klijenti, korisnici, poslovni analitičari, programeri i ostali relevantni članovi tima. Prije same radionice, potrebno je organizirati sastanak kako bi se definirao cilj radionice, identificirali ključni sudionici i osigurala prikladna dokumentacija i materijali za radionicu. Na početku radionice, voditelj daje uvodni govor u kojem se objašnjava svrha radionice, njezini ciljevi i očekivani rezultati. Također se može pregledati trenutno stanje

projekta i kontekst u kojem se razvija softver. Sudionici radionice aktivno sudjeluju u identifikaciji i prikupljanju zahtjeva. To se može postići različitim tehnikama, poput diskusija, grupnih aktivnosti i kreativnih tehnika. Sudionici mogu iznositi svoje ideje, prijedloge i potrebe, a voditelj radionice ih bilježi. Nakon identifikacije zahtjeva, provodi se analiza kako bi se razumjelo njihovo značenje, prioriteti i moguće implikacije na sustav. Ovdje se mogu koristiti tehnike kao što su analiza uzroka i posljedica, identifikacija rizika i tehničke mogućnosti. Na temelju identificiranih i analiziranih zahtjeva, sudionici radionice mogu zajedno razmišljati o mogućim rješenjima i dizajnirati arhitekturu sustava. Ovo može uključivati rasprave, brainstorming i izradu skica. Na kraju radionice, voditelj sumira prikupljene zahtjeve, dogovorene dizajne i sljedeće korake. Također je važno pružiti sudionicima priliku za dodatne komentare i povratne informacije (Robertson i Robertson, 2013).

Prednost radionica je mogućnost aktivnog sudjelovanja svih sudionika u procesu prikupljanja zahtjeva, što rezultira širokim spektrom perspektiva i ideja. Kroz diskusije i interakciju sudionika, dolazi do boljeg razumijevanja konteksta projekta i njegovih ciljeva. Radionica ubrzava proces donošenja odluka jer se svi ključni sudionici nalaze na jednom mjestu, za razliku od intervjua i manjih sastanaka koji zahtijevaju više vremena (Robertson i Robertson, 2013).

S druge strane, radionice, iako su brže, mogu biti i vremenski intenzivne jer uključuju velik broj sudionika. Može biti problema i po samoj organizaciji zbog upitnog rasporeda i mogućnosti dolaska sudionika. Tijekom radionice može doći do dominacije određenih pojedinaca i mogućnosti da usmjere diskusiju u njihove interese (Robertson i Robertson, 2013).

#### 4.1.3. Brainstorming

Brainstorming je metoda koja se često koristi u radionicama za prikupljanje sistemskih zahtjeva kako bi se generirale nove ideje, prijedlozi i zahtjevi. Ova metoda potiče sudionike da slobodno izraze svoje misli i kreativno razmišljaju o problemu ili potrebama sustava. Važno je napomenuti da se brainstorming metoda može prilagoditi potrebama i specifičnostima projekta. Sudionici u sesiji brainstorminga trebaju dolaziti iz različitih disciplina, s što širim rasponom iskustva. Ova mješavina različitih pozadina donosi mnogo više kreativnih ideja na vidjelo.

Na početku brainstorming sesije, voditelj definira cilj i temu, u ovom slučaju to mogu biti funkcionalnosti, performanse ili korisničko iskustvo prilikom korištenja sustava. Sudionicima je potrebno stvoriti okruženje u kojem trebaju slobodno izraziti svoje ideje. Uglavnom su

tijekom sesije zabranjena kritiziranja i ocjenjivanja ideja. Cilj je potaknuti kreativnost i otvorenost te da budu inspirirani idejama drugih. U osmišljanju ideja sudionici imaju određeno vrijeme, a za ideje mogu koristiti razne tehnike, poput asocijacije, slobodnog pisanja, skica i crteža. Nakon generiranja ideja, sudionici zajedno dijele svoje ideje s ostalima. Ideje se bilježe na zajedničkoj ploči ili alatu za vizualizaciju kako bi svi mogli vidjeti i razumjeti ideje drugih sudionika. Sudionici imaju priliku raspravljati o generiranim idejama, postavljati pitanja i nuditi nadogradnje na ideje drugih sudionika. Ovo može potaknuti dodatne ideje i poboljšati razumijevanje zahtjeva. Nakon završetka brainstorminga, ideje se mogu kategorizirati prema određenim kriterijima, poput funkcionalnosti, korisničkih grupa ili važnosti. Sudionici mogu zajedno odabrati prioritete ili odlučiti koje ideje dalje istražiti ili implementirati (Robertson i Robertson, 2013).

#### 4.1.4. Promatranje

Metoda promatranja je još jedna korisna tehnika koja se može primijeniti u prikupljanju sistemskih zahtjeva. Ova metoda uključuje pažljivo promatranje korisnika, radnih procesa i okoline kako bi se prikupili relevantni podaci o postojećem sustavu ili problemu koji treba riješiti. Prvo treba jasno definirati ciljeve i područje promatranja. To može uključivati specifične aspekte sustava koji se trebaju promatrati, poput korisničkog interakcije, radnih procesa, performansi ili problema koji se pojavljuju. Odabiru se relevantni sudionici, poput korisnika sustava ili osoba koje su uključene u radne procese koji se promatraju. Također je važno odabrati prikladno okruženje za promatranje, kao što su radna mjesta, prostorije ili stvarne situacije u kojima se sustav koristi. Tijekom promatranja, treba pažljivo promatrati aktivnosti sudionika, interakcije, probleme ili izazove s kojima se susreću. Bilješke se mogu izrađivati ručno ili koristiti digitalni alati za bilježenje kako bi se zabilježili detalji i važni uvidi. Osim promatranja, postoji opcija postavljanja pitanja sudionicima kako bi se dobile dodatne informacije i postiglo dublje razumijevanje njihovih potreba, stavova i iskustava. Razgovor s sudionicima može pružiti kontekst i omogućiti dodatne detalje o radnim procesima i izazovima. Nakon završetka promatranja, prikupljeni podaci trebaju biti analizirani i interpretirani kako bi se identificirali ključni zahtjevi i problemi. Ovi uvidi mogu se dalje koristiti za dizajniranje ili poboljšanje sustava. Promatranje omogućuje prikupljanje stvarnih podataka o tome kako korisnici rade i suočavaju se s izazovima u stvarnom vremenu. To pruža konkretne uvide u stvarne potrebe, preferencije i ponašanja korisnika. Otvaraju se nove mogućnosti, poput otkrivanja skrivenih potreba i problema koje korisnici možda ne izražavaju otvoreno ili nisu

svjesni. Ova metoda omogućuje uočavanje nedostataka ili neefikasnosti u postojećem sustavu koje korisnici možda nisu svjesni. Omogućuje se dublje razumijevanje konteksta u kojem se sustav koristi. Može se vidjeti interakcija korisnika s okolinom, s drugim korisnicima i s drugim tehnologijama. To pomaže u oblikovanju rješenja koje je usklađeno s konkretnim potrebama i zahtjevima korisnika. Pošto se promatranje ne provodi u uobičajenim okolnostima, promatrači mogu imati utjecaj na ponašanje korisnika ili radnih procesa koje promatraju. Sudionici se mogu ponašati drugačije ili biti svjesni promatrača, što može dovesti do iskrivljenih podataka ili nerealnih situacija. Promatranje može biti vremenski i resursno zahtjevno, pogotovo ako se radi o složenim radnim procesima ili dugotrajnim promatranjima. Potrebno je osigurati dovoljno vremena i resursa za obavljanje kvalitetnog promatranja (Shelly i Rosenblatt, 2012).

#### 4.1.5. Anketiranje

Metoda anketiranja jedna je od brzih metoda u prikupljanju sistemskih zahtjeva. Ankete se koriste kako bi se prikupili kvantitativni podaci, mišljenja i preferencije korisnika ili zainteresiranih strana o određenim aspektima sustava.

Kao i kod svake metode, potrebno je definirati ciljeve ankete. Temeljem ciljeva ankete, sastavlja se upitnik koji sadrži pitanja koja će se postaviti sudionicima. Pitanja mogu biti otvorena (gdje sudionici slobodno izražavaju svoje mišljenje) ili zatvorena (gdje se sudionici izražavaju putem odabira određenih opcija). Treba odabrati relevantnu skupinu sudionika koji će popuniti anketu. Sudionici mogu biti korisnici sustava, interni timovi, zainteresirane strane ili bilo koja druga skupina koja ima relevantno znanje ili iskustvo. Anketa se distribuira sudionicima na način koji odgovara kontekstu projekta. To može uključivati slanje elektroničkih anketa putem e-pošte, korištenje web anketa, distribuciju tiskanih upitnika ili druge metode komunikacije. Kada sudionici ispune anketu, prikupljeni podaci se analiziraju. Ovisno o prirodi ankete, to može uključivati statističku analizu odgovora, kategorizaciju i tumačenje rezultata. Na temelju analize podataka, rezultati ankete se interpretiraju kako bi se identificirali ključni zahtjevi, trendovi ili preferencije sudionika. Ovi uvidi mogu se dalje koristiti u oblikovanju sustava ili donošenju odluka (Shelly i Rosenblatt, 2012).

Prednosti metode anketiranja uključuju mogućnost bržeg prikupljanja podataka od većeg broja sudionika, standardizaciju prikupljenih informacija, anonimnost sudionika te mogućnost brze obrade i analize podataka. Međutim, nedostaci uključuju mogućnost iskrivljenih odgovora zbog subjektivnosti sudionika, mogućnost niske stope odgovora te ograničenje na predložene

odgovore, što može ograničiti raznolikost informacija koje se prikupljaju (Shelly i Rosenblatt, 2012).

#### 4.1.6. Dokumentiranje

Dokumentiranje je metoda koja uključuje zapisivanje i dokumentiranje svih relevantnih informacija vezanih za zahtjeve kako bi se osigurala njihova jasnoća, preciznost i razumljivost. Identificiraju se ključni zahtjevi sustava putem interakcije s korisnicima, dionicima i drugim relevantnim osobama. Razumijevanje poslovnih potreba i ciljeva te njihovo pretvaranje u konkretnije zahtjeve je ključno. Nakon identifikacije zahtjeva, analitičar ih detaljno opisuje i dokumentira. Specifikacije obuhvaćaju funkcionalne zahtjeve (što sustav treba raditi), nefunkcionalne zahtjeve (kako sustav treba raditi) i ostale relevantne informacije poput ograničenja i performansi. Dokumentirani zahtjevi prolaze kroz proces validacije i pregleda kako bi se osigurala njihova ispravnost i potpunost. Ovo uključuje suradnju s klijentima i ostalim dionicima kako bi se provjerila usklađenost s poslovnim potrebama i otkrile eventualne nedostatke ili nejasnoće (Robertson i Robertson, 2013).

Dokumentiranje sistemskih zahtjeva ima nekoliko prednosti. Omogućuje jasnoću i transparentnost u komunikaciji između različitih dionika. Pruža temelj za dizajn, razvoj i testiranje informacijskog sustava. Dokumentacija služi kao referenca tijekom cijelog životnog ciklusa sustava, olakšavajući održavanje i nadogradnje u budućnosti. No, dokumentiranje zahtjeva može biti izazovno jer se ponekad informacije mogu nedovoljno ili netočno prenijeti. Nedostatak detaljnosti ili nejasnoća u zahtjevima može dovesti do nesporazuma i pogrešnog razumijevanja. Tijekom procesa razvoja informacijskog sustava, zahtjevi se mogu mijenjati ili dodavati. Ako dokumentacija nije fleksibilna ili se ne ažurira pravovremeno, moguće je da ne odražava stvarne potrebe ili promjene u poslovnim zahtjevima (Robertson i Robertson, 2013).

## 4.2. Pregled izvedivosti zahtjeva

Izvedivost je ponekad jednostavna i može se potvrditi unutar nekoliko sati. No, ukoliko su potrebne velike promjene ili potpuno novi sustav, potrebno je napraviti detaljno istraživanje. Potrebno je uložiti određenu količinu angažmana i vremena, a to ovisi o kompleksnosti zahtjeva. Koliko je zahtjev jednostavan poput promjene rasporeda već postojećih narudžbi i njihovih detalja, izvedivost se može odrediti brzo. No ukoliko je potrebno napraviti veliku

promjenu i razviti sasvim novi sustav koji će istraživati tržište kako bi se mogla odrediti potreba i želja za proizvodom, bit će potrebna veća količina napora i uloženog vremena. Prvi korak u procjeni izvedivosti je identificirati i ukloniti zahtjeve sustava koji nisu izvedivi. Na primjer, zahtjev ne bi bio izvediv ako zahtijeva hardver ili softver koji je tvrtka već odbacila. Na primjer, zahtjev koji zahtijeva hardver ili softver koji je već odbačen od strane tvrtke nije izvediv. Čak i ako je zahtjev izvediv, možda nije nužan. Na primjer, zahtjev za više verzija izvješća može zahtijevati značajan napor u dizajnu i programiranju. Bolja alternativa može biti omogućiti korisnicima preuzimanje podataka na njihova osobna računala i koristiti programski paket koji im omogućuje izradu vlastitih izvješća. U tom slučaju, ulaganje u obuku korisnika može biti bolja opcija od izrade izvješća za njih (Shelly i Rosenblatt, 2012).

Kako bi se donijela ispravna odluka, potrebno je diskutirati o nekoliko aspekata (Hudaib i sur., 2019):

- O pristupu problemu i njegovoj mogućnosti da tvrtka postigne željeni cilj.
- O tehničkoj izvedivosti, dostupnosti sredstava i ljudi.
- O poželjnosti u operativnom smislu.
- O predviđenim troškovima i uštedi, te o ekonomskoj poželjnosti
- O postojanju nematerijalnih čimbenika, popust zadovoljstva kupaca ili imidža tvrtke.
- O činjenici spada li projekt u dobro poslovno ulaganje i je li vrijedan rješavanja.
- O vremenskom okviru i može li se prijedlog ostvariti unutar njega.

Najbitnije su izvedivosti u tehničkom, ekonomskom i vremenskom aspektu. Tehnička izvedivost odnosi se na tehničke resurse potrebne za izradu, kupnju, instaliranje i upravljanje sustavom. Ekonomska izvedivost znači da su koristi od predloženog sustava veće nego procijenjeni troškovi i obično se smatraju ukupnim troškom vlasništva, koji uključuje trenutne troškove podrške, održavanja i nabave. Uključuje ljude, hardver, softver, osposobljavanje, licence, troškove savjetovanja te potencijalni trošak mogućeg odgađanja projekta. Vremenska izvedivost znači da se projekt može provesti u određenom vremenskom okviru. Prilikom procjene vremenske izvedivosti, analitičar treba procijeniti odnos troškova i vremena. Ukoliko se projekt mora ubrzati, može ostati izvediv, ali će postati izuzetno skup (Hudaib i sur., 2019).

Prvi korak u procjeni izvedivosti je identificirati i eliminirati zahtjeve sustava koji nisu izvedivi. Na primjer, zahtjev ne bi bio izvediv ako zahtijeva hardver ili softver koji je tvrtka već odbila. Čak i ako je zahtjev izvediv, možda nije nužan. Na primjer, zahtjev za više verzija izvješća



mogao bi zahtijevati značajan napor dizajna i programiranja. Bolju alternativu predstavlja preuzimanje podataka s poslužitelja na osobno računalo i pokazivanje korisnicima kako da sami generiraju izvješća. U tom slučaju, ulaganje u obuku korisnika bilo bi bolje od stvaranja izvješća za njih. Kako projekt napreduje, uvjeti se često mijenjaju. Troškovi nabave mogu se povećati, a projekt može postati skuplji nego što se očekivalo. Osim toga, menadžeri i korisnici ponekad gube povjerenje u projekt. Iz tih razloga, analiza izvedivosti je kontinuirani zadatak koji se mora provoditi tijekom cijelog procesa razvoja sustava (Hudaib i sur., 2019).

### 4.3. Postavljanje prioriteta zahtjeva

Nakon odbijanja zahtjeva sustava koji nisu izvedivi, odbor za pregled sustava mora uspostaviti prioritete za preostale stavke. Najviši prioritet dodjeljuje se projektima koji pružaju najveću korist uz najniže troškove i u najkraćem vremenskom razdoblju. Međutim, mnogi čimbenici utječu na procjenu projekta, te su prikazani u tablici 2. Idućim redoslijedom se određuje prioritet zahtjeva:

Korist	Prednost koju tvrtka ostvaruje kao rezultat implementacije zahtjeva
Kazna	Predstavlja posljedicu neimplementiranja zahtjeva.
Trošak	Napor i resursi koji su potrebni za implementaciju zahtjeva.
Rizik	Vjerojatnost da zahtjev neće donijeti očekivanu vrijednost.
Ovisnosti	Odnos između zahtjeva. Zahtjev će zahtijevati dovršetak drugog zahtjeva radi njegove implementacije.
Vremenski okvir	Svaki zahtjev treba imati svoj datum isteka.
Stabilnost	Vjerojatnost da će zahtjev ostati nepromijenjen.

Regulatorna/usklađenost s politikom	Predstavlja zahtjeve koji se moraju implementirati kako bi se ispunili regulatorni zahtjevi.
-------------------------------------	--

Tablica 2. Kriterij rangiranja prioriternih zahtjeva

Izvor: de Bree (n.d.)

## **5. Osnovni zahtjevi u izradi IS-a**

U ovom dijelu rada navesti će se osnovne stavke koje treba uzeti u obzir prilikom prikupljanja zahtjeva za izradu informacijskog sustava. Prednosti i rizici ovih stavki daju uvid u ključne prednosti i rizike koji oni nose sa sobom, a koje treba uzeti u obzir prilikom analiziranja zahtjeva.

### **5.1. Vremenski okvir**

Definiranje vremenskog okvira kod izrade informacijskog sustava nosi određene prednosti i rizike. Ono omogućuje planiranje i organizaciju aktivnosti tijekom razvojnog procesa te na taj način pomaže u postizanju strukture, usmjeravanju napora i pravovremenom izvršavanju zadataka. Vremenski okvir omogućuje procjenu potrebnih resursa, uključujući ljudske resurse, financijska sredstva i materijale. To olakšava upravljanje resursima i smanjuje rizik od nedostatka resursa. Postavljanjem rokova i definiranjem očekivanja sudionicima projekta doprinosi usklađivanju poslova na više razina, razumijevanju ciljeva i izbjegavanju određenih grešaka i nesporazuma. Identifikacijom kritičnih putova i aktivnosti koje imaju najveći utjecaj na ukupno vrijeme trajanja projekta omogućeno je fokusiranje napora na ključne aspekte i optimizaciju resursa. Koliko god vremenski okvir ima prednosti, toliko ima i rizika. Koliko god projekt djeluje organizirano i predvidivo, od početka treba definirati neizvjesnost projekta. Izrada informacijskog sustava može biti kompleksna i nepredvidljiva, posebno ako postoje tehnički izazovi ili promjene zahtjeva. Definiranje strogog vremenskog okvira može rezultirati neadekvatnim vremenom za prilagodbe ili izmjene, što može dovesti do kašnjenja ili kompromisa kvalitete (International Institute of Business Analysis (IIBA), 2009).

Postavljanje prekratkih rokova može doći do preopterećenosti resursa, kao što su timovi ili pojedinci. To može rezultirati prekomjernim radom, stresom, pogreškama ili lošijom kvalitetom rada. Može doći i do neispravne ili nepotpune procjene. Vremenski okvir zahtijeva procjenu različitih aktivnosti i njihovog trajanja. Nepotpuna ili netočna procjena može dovesti do neuspjeha u ispunjavanju rokova ili prekoračenja budžeta. Strogi vremenski okvir također može ograničiti fleksibilnost u prilagodbi novim zahtjevima ili promjenama tijekom razvojnog procesa (International Institute of Business Analysis (IIBA), 2009).

## **5.2. Troškovi**

Definiranje troškova omogućuje temeljito financijsko planiranje projekta uključujući procjenu i identifikaciju potrebnih resursa, proračun troškova razvoja, implementacije, održavanja i podrške sustava.

Praćenje i kontroliranje proračuna projekta omogućuje usporedbu stvarnih troškova s planiranim troškovima, identifikaciju odstupanja i poduzimanje korektivnih mjera kako bi se osigurala financijska održivost projekta. Troškovnom analizom omogućuje se procjena isplativosti projekta. Izborom prioriteta i zatim definiranjem troškova koje projekt donosi te pomaže donositeljima odluka u procjeni povrata ulaganja i donošenju odluka o projektu. S druge strane, tu su i nepredviđeni troškovi. Definiranje troškova može biti izazovno jer su neki troškovi teško predvidljivi ili nepredvidivi. Nepredviđeni troškovi mogu se pojaviti tijekom implementacije projekta, kao što su promjene u zahtjevima, tehnički problemi ili potreba za dodatnim resursima. U slučaju nepotpune procjene ili podcjenjivanja troškova, može doći do prekoračenja proračuna ili nedostatnih resursa za završetak projekta. Budžet i njegova ograničenja te mogućnost prilagodbe i promjene tijekom projekta mogu biti u kontradikciji. Ako se pojave neočekivani troškovi ili se pojave potrebe za dodatnim resursima, može doći do promjena u kvaliteti ili do kašnjenja u isporuci. Ako se previše naglasi troškovni aspekt projekta, postoji rizik da se kvaliteta rada ili funkcionalnosti informacijskog sustava žrtvuju kako bi se smanjili troškovi. To može rezultirati manje zadovoljavajućim rješenjem ili potencijalnim problemima u budućnosti. Važno je napomenuti da je važno provesti temeljitu analizu troškova, uzeti u obzir potencijalne rizike i biti fleksibilan kako bi se prilagodili promjenama tijekom procesa izrade informacijskog sustava (International Institute of Business Analysis (IIBA), 2009).

## **5.3. Funkcionalnost**

Funkcionalnosti daju jasan uvid u smjernice sustava, odnosno što sustav treba postići i koje funkcionalnosti treba podržavati. Time se olakšava usklađivanje ciljeva sudionika projekta i stvara se razumijevanje u kojem smjeru ide projekt, a kakav sustav se stvara. Jasno definirane funkcionalnosti donose veću vjerojatnosti za zadovoljavanje potreba korisnika sustava. Korisničko zadovoljstvo jedna je od bitnih stavki uspješnosti projekta, a ukoliko su funkcionalnosti konkretno i jasno definirane, šansa za to se povećava. Potrebno je učinkovito

planiranje resursa, odnosno ljudi, materijala i vremena. Time se olakšava procjena i planiranje rasporeda aktivnosti (Robertson i Robertson, 2013).

#### **5.4. Sigurnost**

Kako bi se osigurala sigurnost projekta, potrebno je definirati jasne sigurnosne zahtjeve. To najčešće uključuje zahtjeve za autentifikacijom, autorizacijom, enkripcijom podataka, nadzorom pristupa itd.

Prva faza osiguranja što veće sigurnosti projekta je analiza prijetnji. Potrebno je identificirati prijetnje koje bi mogle utjecati na sustav, poput neovlaštenog pristupa podacima, krađe podataka, zlonamjernih softvera i slično. Analiza prijetnji omogućava razvoj strategija koje će se primijeniti tijekom razvojnog procesa informacijskog sustava. Kako bi sigurnost projekta bila dugotrajna, potrebna je odgovornost svih članova tima. Sudionici i razvojni tim trebaju biti educirani i imati razumijevanje sigurnosne prakse i držati se standarda vezanih za sigurnost u razvojnem procesu. Potrebna je suradnja i komunikacija svih sudionika kako se osigurala dosljednost sigurnosnih zahtjeva. Tim treba biti svjestan osjetljivosti projekta te provesti redovita testiranja pojedinih dijelova informacijskog sustava i sigurnosnih propusta uz ispravljanje grešaka i provoditi redovita ažuriranja kako bi se osigurala sigurnosti. Također, od koristi mogu biti automatizirana testiranja koda i određeni alati koji se mogu koristiti u tu svrhu. Kada se sam sustav implementira, važno je osigurati redovito praćenje i nadzor kako bi se na vrijeme prepoznale prijetnje i nepravilnosti. Edukacija korisnika će također smanjiti rizik od neovlaštenog pristupa i zloupotrebe sustava. Sigurnost je jedna od najtežih vrsta zahtjeva za specifikaciju, a uz to predstavlja i najveći rizik ukoliko se ne rukuje ispravno. Potrebno je uzeti u obzir softver koji se primjenjuje i tome prilagoditi zahtjeve. Sigurnosti se svodi na tri aspekta, a to su pristup podacima, privatnost i integritet. Pristup označava dostupnost podataka, odnosno da proizvod pohranjuje podatke kako bi im korisnici mogli pristupiti, razumjeti ih te na kraju koristiti. U kontekstu sigurnosti, podaci trebaju biti dostupni ovlaštenim osobama i trebaju biti dostupni na vrijeme te trebaju biti usmjereni na onemogućavanje pristupa osobama koje nemaju ovlasti. Kako bi se podaci zaštitili, često se koriste softverska „zaključavanja“ kako bi se pristup neovlaštenih korisnika spriječio. Pri pisanju sigurnosnih zahtjeva potrebno je navesti hijerarhiju pristupa, odnosno tko ima ovlašten pristup, u kojim okolnostima je važeća autorizacija, koja je dostupnost podataka i funkcija svakoj ovlaštenoj osobi. Također treba uzeti u obzir korisnike koji nemaju konstantan pristup, odnosno one koji imaju pristup tijekom određenog perioda ili

moraju surađivati s drugim ovlaštenim korisnikom kako bi imali pristup. Privatnost u poslovanju postaje sve veća briga. Iako se privatnost opisuje u kontekstu sigurnosti, većinom se veže za pravne zahtjeve. Popriličan broj zemalja ima zakone koji reguliraju privatnost podataka koji su čuvani unutar računalnog sustava. U pisanju zahtjeva odnosi se na pisanje zahtjeva za ono što proizvod mora učiniti kako bi se osigurala privatnost podataka. Svrha je da djelomično kontroliranje pristupom kako bi se spriječilo slanje i distribuiranje podataka neovlaštenim osobama. Ako se u sustavu nalaze podaci koji su strogo povjerljivi, u zahtjeve treba dodati zabranu ispisa strogo povjerljivih podataka. Iako oni u sustavu nisu direktno dostupni neovlaštenim korisnicima, ukoliko se isprintaju i dođe do gubitka tih podataka, sigurnost nije ispunila svoju svrhu. Integritet znači da su podaci koji se nalaze u sustavu jednaki onima u izvoru. Zahtjevi vezani za integritet odnose se na prevenciju gubitka ili oštećenja podataka. Ukoliko dođe do oštećenja podatka, potrebno je osigurati oporavak podataka. Kada se sustav implementira, sadržavati se podatke važne za tvrtku koja ih posjeduje, a tu dolazi do zahtjeva za integritetom podataka. Integritet se mora osigurati ukoliko dođe do neočekivanih događaja, poput nestanka struje (International Institute of Business Analysis (IIBA), 2009).

### **5.5. Zahtjevi vezani za kulturu tvrtke**

Kulturološki zahtjevi određuju specifične faktore koji mogu dovesti do neprihvatljivosti proizvoda zbog običaja, vjere, jezika, predrasuda ili drugih aspekata ljudskog ponašanja. Ovi zahtjevi posebno dolaze do izražaja kada se pokušava plasirati proizvod na tržištu druge zemlje koja se razlikuje po kulturi i jeziku od one u kojoj se proizvod razvija. Često uzimamo svoju vlastitu kulturu zdravo za gotovo i rijetko razmišljamo o tome kako naši proizvodi i mi sami možemo biti doživljeni u drugim zemljama. Kada se proizvod namjerava prodavati izvan vlastite zemlje, pa čak i izvan lokalne regije ili države, važno je razmotriti angažiranje stručnjaka za kulturu. Kulturološki zahtjevi često su neočekivani i na prvi pogled mogu izgledati iracionalno. Međutim, treba imati na umu da razlog iza tih zahtjeva leži izvan naše vlastite kulture. Ukoliko dođe do pitanja "Zašto to rade na taj način?", vrlo vjerojatno je riječ o kulturološkom zahtjevu (Robertson i Robertson, 2013).

Potrebno je voditi riječ o idućim aspektima (Robertson i Robertson, 2013):

- Religija: potrebno je uzeti u obzir religije kako bi proizvod bio prihvatljiv, te da ne sadržava uvredljive riječi i simbole.

- Politička korektnost: potrebno je istražiti određene pojmove i osigurati da proizvod ne sadržava uvredljive ikone i pojmove.
- Forsiranje političkog ili kulturnog svrstavanja: potrebno je osigurati dostupnost proizvoda na svim jezicima i prilagodljiv svim regijama svijeta.

## **5.6. Legalnost zahtjeva**

Trošak sudskih postupaka predstavlja jedan od glavnih rizika sustava za prodaju, ali može biti i relevantan za druge vrste sustava. Potrebno je biti upoznat s zakonima i napisati zahtjeve kako bi se osiguralo da proizvod ispunjava ove zakonske zahtjeve. Čak i ako se sustav gradi za interno korištenje u vlastitoj organizaciji, treba imati na umu da se primjenjuju zakoni koji se odnose na radno mjesto, što može biti relevantno. Kada se radi o zakonima, možda će biti potrebno savjetovanje s odvjetnikom. Kada su poznati svi potrebni zakoni, moguće je izdvojiti najbitnije i primijeniti ih prema specifikacijama informacijskog sustava (Robertson i Robertson, 2013).

## **5.7. Performanse**

Zahtjevi vezani uz performanse uglavnom su kratki i jednostavni, ali poprilično bitni jer su zastupljeni u svakom koraku korištenja sustava. Očekivano trajanje sustava je jedan od zahtjeva koji često ostane zaboravljen, a izrazito je potreban. Brzina sustava predstavlja količinu vremena koja je na raspolaganju sustavu za obavljanje zadataka. Ovi zahtjevi se često odnose na vrijeme odaziva, ali se mogu odnositi i na sposobnost sustava da radi brzinom koja je prikladna za radno okruženje. Sustavi moraju moći izvesti određene radnje i funkcije i određenom vremenskom intervalu. Ovaj zahtjev, ukoliko nije odrađen, može prouzročiti čak i nesreće. Ako se radi o sustavu za zrakoplove, a radar zrakoplova ne može na vrijeme prikazati podatke o nadolazećoj planini, doći će do katastrofe. Potrebno je postaviti zahtjeve na osnovu očekivane preciznosti od strane klijenta. Ukoliko se unutar sustava nalaze novčani podaci, zahtjev se može odnositi na brojeve koji trebaju biti prikazani sa dvije decimale. Pouzdanost sustava se izražava kao dopušteno vrijeme između kvarova ili dopuštena stopa kvarova. U zahtjevima se definira očekivana dostupnost sustava. Za neke sustave je ključno da se greške svedu na minimum. Potrebno je istražiti mogućnost neuspjeha te definirati realne razine usluge. Ove informacije je potrebno prezentirati klijentu i napraviti kompromis između realnih mogućnosti i očekivanja klijenta. Tolerancija je mogućnost sustava da tolerira neočekivanu i

ekstremnu situaciju, te da radi u takvim uvjetima. Sustavi mogu biti toliko složeni da uvijek postoji šansa da neka komponenta neće ispravno raditi. Ovi zahtjevi sprječavaju potpuni kvar sustava. Cilj je da, ukoliko dođe do kvara, sustav može i nakon toga uspostaviti prihvatljiv način rada. Kapacitet predstavlja količinu podataka s kojom se sustav treba moći nositi i koju može obraditi. Prvi korak u planiranju kapaciteta je razviti model na temelju trenutnog opterećenja sustava, te zatim uzeti u obzir zahtjeve korisnika te poboljšati sustav, ukoliko je to moguće. Skalabilnost pokazuje kako će sustav funkcionirati s većom količinom informacija i podataka. Kako posao raste (ili se očekuje da će rasti), softverski proizvodi moraju imati mogućnost rukovanja s povećanim opsegom podataka. Potrebno je da softverski proizvodi povećaju svoje kapacitete kako bi se nosili s novim zahtjevima i povećanim obujmom podataka. Glavni cilj je osigurati da dizajneri uključe dovoljno kapaciteta za budući rast i skalabilnost sustava sustava (Robertson i Robertson, 2013).



## 6. Zaključak

Prilikom razvoja informacijskog sustava, neophodno je napraviti kvalitetnu analizu zahtjeva i dionika sustava. Izgradnja informacijskog sustava sastoji se od više faza, a početna faza prikupljanja i analize zahtjeva osigurava postavljanje temelja za njegovu izgradnju. Posao analitičara ključan je u razvoju informacijskog sustava. Potrebno je imati znanja i iskustvo, pretpostaviti koji je oblik najbolji za prikupljanje zahtjeva te koja pitanja treba postaviti i razumjeti ih. Iskustvo analitičara posebno je potrebno u trenutku kada klijent ne zna točno navesti ili ne poznaje terminologiju potrebnu za formiranje zahtjeva. Ukoliko analitičar ima iskustva i poznaje prijašnje ishode, može prikazivanjem prošlih slučajeva i zahtjeva s klijentom dogovoriti nove zahtjeve. Vremenski okvir je važan čimbenik koji omogućuje planiranje, organizaciju i strukturiranje aktivnosti tijekom razvojnog procesa informacijskog sustava.

Definiranje vremenskog okvira može pružiti prednosti u usklađivanju poslova, procjeni resursa i fokusiranju napora na ključne aspekte projekta. Međutim, potrebno je biti svjestan rizika, kao što su nepredviđeni tehnički izazovi ili promjene zahtjeva, koji mogu dovesti do kašnjenja ili kompromisa kvalitete ako se strogi vremenski okvir ne prilagodi. Definiranje troškova je ključno za financijsko planiranje projekta i procjenu isplativosti. Praćenje proračuna omogućuje usporedbu stvarnih troškova s planiranim te identifikaciju odstupanja i poduzimanje korektivnih mjera. Međutim, nepredviđeni troškovi i nepotpuna procjena mogu dovesti do prekoračenja proračuna ili nedostatnih resursa, stoga je važno provesti temeljitu analizu troškova. Funkcionalnost sustava treba biti jasno definirana kako bi se olakšalo usklađivanje ciljeva sudionika projekta i zadovoljavanje potreba korisnika. Jasne i konkretne funkcionalnosti povećavaju šanse za uspješan projekt i korisničko zadovoljstvo. Učinkovito planiranje resursa također je važno kako bi se osiguralo pravilno izvršavanje aktivnosti. Definiranje jasnih sigurnosnih zahtjeva omogućuje razvoj strategija i osvješćivanje prijetnji koje mogu utjecati na sustav. Timski rad važan je za razvoj sustava, edukacija sudionika bitna je za njegovu implementaciju i korištenje, a redovito održavanje i ažuriranje sustava potrebno je kako bi se osigurala konzistentnost sustava. Pristup podacima, privatnost i integritet trebaju biti pažljivo promišljeni i implementirani kako bi se osigurala dugoročna sigurnost projekta. Nakon odluke o izvedivosti, analize i postavljanja prioriteta zahtjeva, potrebno je kreirati dokument sa zahtjevima i prezentirati ga razvojnom timu. Tim zatim stvara jasnu sliku o tome što korisnici očekuju od informacijskog sustava. To uključuje identificiranje poslovnih procesa, potreba korisnika, ciljeva sustava te specifičnih funkcionalnosti i ograničenja. Komunikacija s relevantnim dionicima, kao što su korisnici, menadžeri i stručnjaci, ključna je za dobivanje

kvalitetnih zahtjeva. Kvalitetno prikupljanje zahtjeva omogućuje timu za razvoj donošenje pravodobnih odluka tijekom razvojnog procesa.

## Literatura

1. Avison, D., Fitzgerald, G. (1995). *Information systems development: methodologies, techniques and tools*. Dostupno na: [https://archive.org/details/informationssystem0000avis\\_h9z4/page/n1/mode/2up](https://archive.org/details/informationssystem0000avis_h9z4/page/n1/mode/2up) [Pristupljeno: 18. lipnja 2023.]
2. Bourgeois, D. (2019). *Information Systems for Business and Beyond*. Dostupno na: <https://opentextbook.site/informationssystem2019/> [Pristupljeno: 18. lipnja 2023.]
3. de Bree, T. (n.d.). *Difference Between Systems Analyst and Business Analyst*, Modern Analyst. Dostupno na: <https://www.modernanalyst.com/Resources/Articles/tabid/115/ID/185/Difference-Between-Systems-Analyst-and-Business-Analyst.aspx> [Pristupljeno: 27. lipnja 2023.]
4. Davies, A. (n.d.). *What are Functional Requirements? [With Examples]*. Dostupno na: <https://www.devteam.space/blog/what-are-functional-requirements/> [Pristupljeno: 20. lipnja 2023.]
5. Drljača, M. (2006). *Model informacijskog sustava za upravljanje poslovnim procesom*. Dostupno na: <https://www.bib.irb.hr/520681> [Pristupljeno: 20. lipnja 2023.]
6. Grady, J. O. (2016). *System Requirements Analysis*. Dostupno na: [https://www.google.hr/books/edition/System\\_Requirements\\_Analysis/o8kIR7lvrRQC?hl=en&gbpv=1&dq=system+requirements&pg=PA605&printsec=frontcover](https://www.google.hr/books/edition/System_Requirements_Analysis/o8kIR7lvrRQC?hl=en&gbpv=1&dq=system+requirements&pg=PA605&printsec=frontcover) [Pristupljeno: 27. lipnja 2023.]
7. Hudaib, A., Masadeh, R. M. T., Qasem, M. H., Alzaqebah A. I. (2019). *Requirements Prioritization Techniques Comparison*. Dostupno na: [https://www.researchgate.net/publication/322487895\\_Requirements\\_Prioritization\\_Techniques\\_Comparison](https://www.researchgate.net/publication/322487895_Requirements_Prioritization_Techniques_Comparison) [Pristupljeno: 27. lipnja 2023.]
8. International Institute of Business Analysis (IIBA) (2009). *A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 2.0*. Dostupno na: <http://paginaspersonales.unam.mx/app/webroot/files/1674/general/curriculum.pdf?rand=20210414232705000000> [Pristupljeno: 3. srpnja 2023.]
9. Lemke, G. (2018). *The software development life cycle and its application*. Dostupno na: <https://commons.emich.edu/honors/589/> [Pristupljeno: 25. lipnja 2023.]
10. Robertson, S., Robertson, J. (2013). *Mastering the requirements process, Third edition*. Dostupno na: [http://www.pkt.edu.my/pdf\\_sys/home/pdf/145](http://www.pkt.edu.my/pdf_sys/home/pdf/145) [Pristupljeno: 18. lipnja 2023.]

11. Shelly, G. B., Rosenblatt, H. J. (2012). *System Analysis and Design, Ninth edition*.  
Dostupno na: <https://www.auhd.edu.ye/upfiles/elibrary/Azal2020-01-22-01-16-49-66429.pdf> [Pristupljeno: 18. lipnja 2023.]
12. Ward J., Peppard J. (2002). *Strategic Planning for Information Systems* (3. izdanje).  
Dostupno na: [https://repository.dinus.ac.id/docs/ajar/ebooksclub.org\\_StrategicPlanningofInformationSystems.pdf](https://repository.dinus.ac.id/docs/ajar/ebooksclub.org_StrategicPlanningofInformationSystems.pdf) [Pristupljeno: 27. lipnja 2023.]

## **Popis slika**

Slika 1. Životni ciklus razvoja softvera.....	6
Slika 2. Dijagram korištenja na osnovi analize scenarija.....	10
Slika 3. "Brown Cow" model.....	17

## **Popis tablica**

Tablica 1. Razlika sistemskog i poslovnog analitičara.....	4
Tablica 2. Kriterij rangiranja prioriternih zahtjeva.....	25