

**Modeliranje poslovnog procesa obračuna i isplata plaća u
Ministarstvu unutarnjih poslova Republike Hrvatske**

Upravljanje poslovnim procesima

Student: Petra Puljić, 0030106935

Nositelj: [Izv. prof. dr. sc. Darko Etinger](#)

Asistent: [Luka Blašković, mag. inf.](#)

Sažetak

Ovaj seminarski rad bavi se analizom, modeliranjem i optimizacijom ključnog poslovnog procesa – Modeliranje poslovnog procesa obračuna i isplata plaća u Ministarstvu unutarnjih poslova Republike Hrvatske. Rad je podijeljen u tri glavna dijela. Prvi dio obuhvaća identifikaciju i detaljnu analizu trenutnog stanja (As-Is) procesa unutar hipotetske organizacije, modeliranog prema BPMN 2.0 standardu. Kroz As-Is analizu identificirane su ključne aktivnosti, sudionici, tokovi informacija te su uočene potencijalne točke za optimizaciju, poput uskih grla, suvišnih koraka i nedostatka automatizacije. Drugi dio rada fokusira se na definiranje ciljanog stanja (To-Be), gdje se na temelju uočenih nedostataka predlažu konkretna poboljšanja s ciljem povećanja učinkovitosti, smanjenja vremena izvođenja i minimiziranja rizika od pogrešaka. Treći, praktični dio rada, prikazuje izradu i izvršavanje model-driven aplikacije korištenjem Camunda platforme. Kroz implementaciju pojednostavljenog dijela procesa, demonstrirana je izvedivost predloženih poboljšanja, uključujući automatizaciju zadataka i integraciju s vanjskim sustavima putem REST API servisa.

Sažetak

1. Uvod

2. Analiza i modeliranje poslovnog procesa

2.1. Zašto baš proces obračuna plaća?

2.2. As-Is analiza

2.3. To-Be analiza

Automatizacija prikupljanja i validacije podataka

Digitalizacija komunikacije i eskalacija problema

Potpuna integracija s bankom i pametno rukovanje ishodima

Jedinstveni izvor istine za podatke

3. Razvoj model-driven aplikacije

3.1. Izvođenje procesa

3.2. Integracija s vanjskim servisima (Connectori)

Slanje naloga za isplatu banci (REST API)

Slanje platne liste zaposleniku putem e-maila

4. Zaključak

1. Uvod

U ovom seminarskom radu bavit ću se područjem upravljanja poslovnim procesima (BPM). To je pristup koji tvrtkama pomaže da svoje svakodnevne operacije učine boljima – od otkrivanja kako trenutno rade, preko analize i modeliranja, sve do poboljšanja i automatizacije. U današnje vrijeme, kada se sve brzo mijenja, sposobnost efikasnog upravljanja procesima ključna je za uspjeh.

Za praktični dio rada odabrala sam **proces obračuna i isplate plaća**. Odlučila sam se za njega jer je to jedan od najvažnijih i najosjetljivijih procesa u svakoj organizaciji. Uključuje suradnju između ljudskih resursa, računovodstva i menadžmenta, podložen je strogim zakonima i direktno utječe na to kako se zaposlenici osjećaju. Zbog svega toga, bio je odličan primjer za analizu i poboljšanje.

Cilj mi je bio proći kroz cijeli BPM ciklus. Prvo sam analizirala postojeće stanje procesa (As-Is) kako bih pronašla probleme poput uskih grla, suvišnih koraka i nedostatka automatizacije. Nakon toga, predložila sam konkretna poboljšanja (To-Be analiza). Drugi, praktični dio rada, bila je izrada funkcionalne aplikacije u Camundi. Time sam htjela pokazati kako se BPMN model može pretvoriti u stvarnu, izvršnu aplikaciju koja automatizira zadatke i olakšava posao ljudima.

2. Analiza i modeliranje poslovnog procesa

U ovom poglavlju detaljno ću prikazati kako sam analizirala i modelirala odabrani poslovni proces.

2.1. Zašto baš proces obračuna plaća?

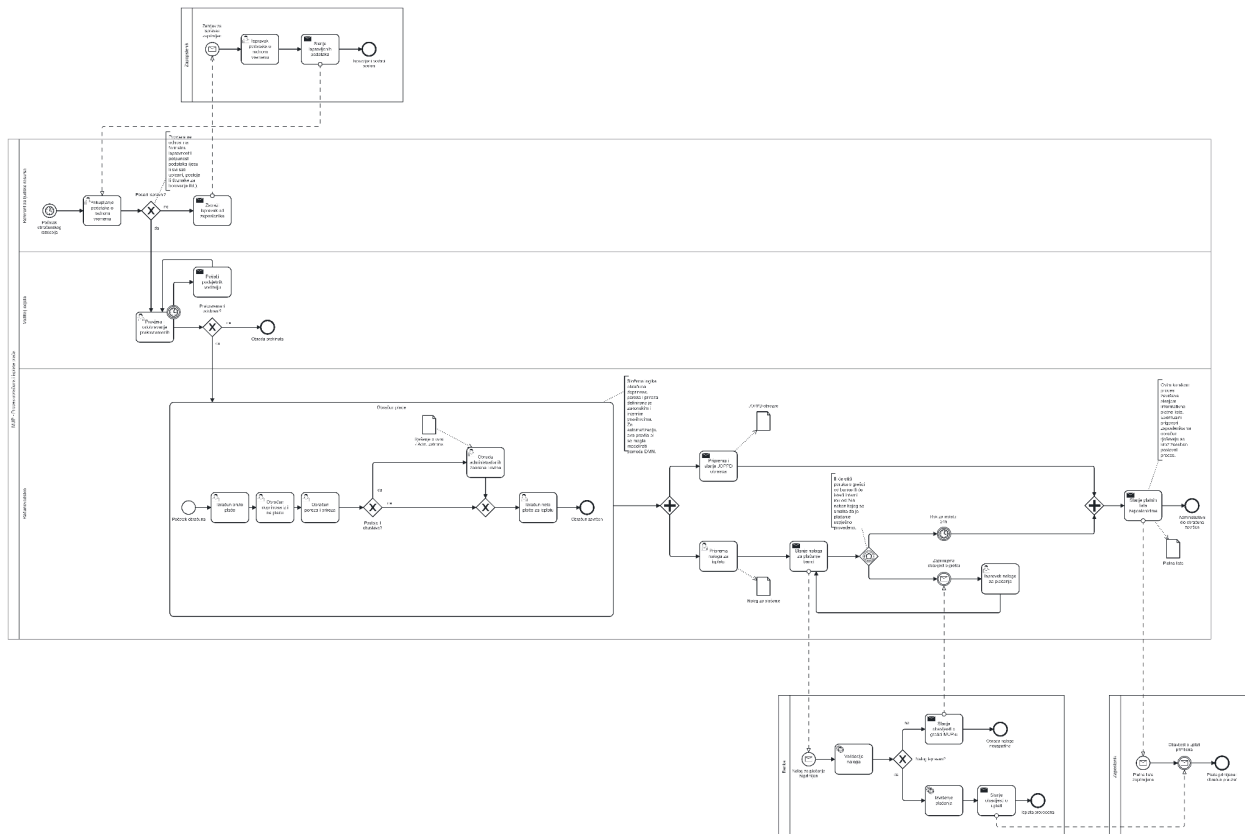
Za svoj projekt odabrala sam proces obračuna i isplate plaća. Smatram da je to odličan primjer jer je istovremeno jako važan i prilično složen. Uključuje puno ljudi iz različitih odjela – od zaposlenika koji dostavljaju podatke, preko ljudskih resursa i voditelja, pa sve do računovodstva koje radi konačni obračun.

Glavni cilj ovog procesa je jednostavan: osigurati da svaki zaposlenik dobije točnu plaću na vrijeme i u skladu sa svim zakonima. No, put do toga je kompliciran. Svaka greška, bilo da je riječ o krivom izračunu, kašnjenju isplate ili problemu s JOPPD obrascem, može stvoriti velike probleme za tvrtku i narušiti povjerenje zaposlenika. Zbog toga je optimizacija ovog procesa ključna.

Da bih shvatila kako proces trenutno izgleda, koristila sam nekoliko metoda. Proučila sam dostupnu dokumentaciju poput pravilnika o radu i zakona, razgovarala sam s maminim kolegama iz računovodstva i ljudskih resursa kako bih dobila uvid u njihove svakodnevne zadatke i probleme.

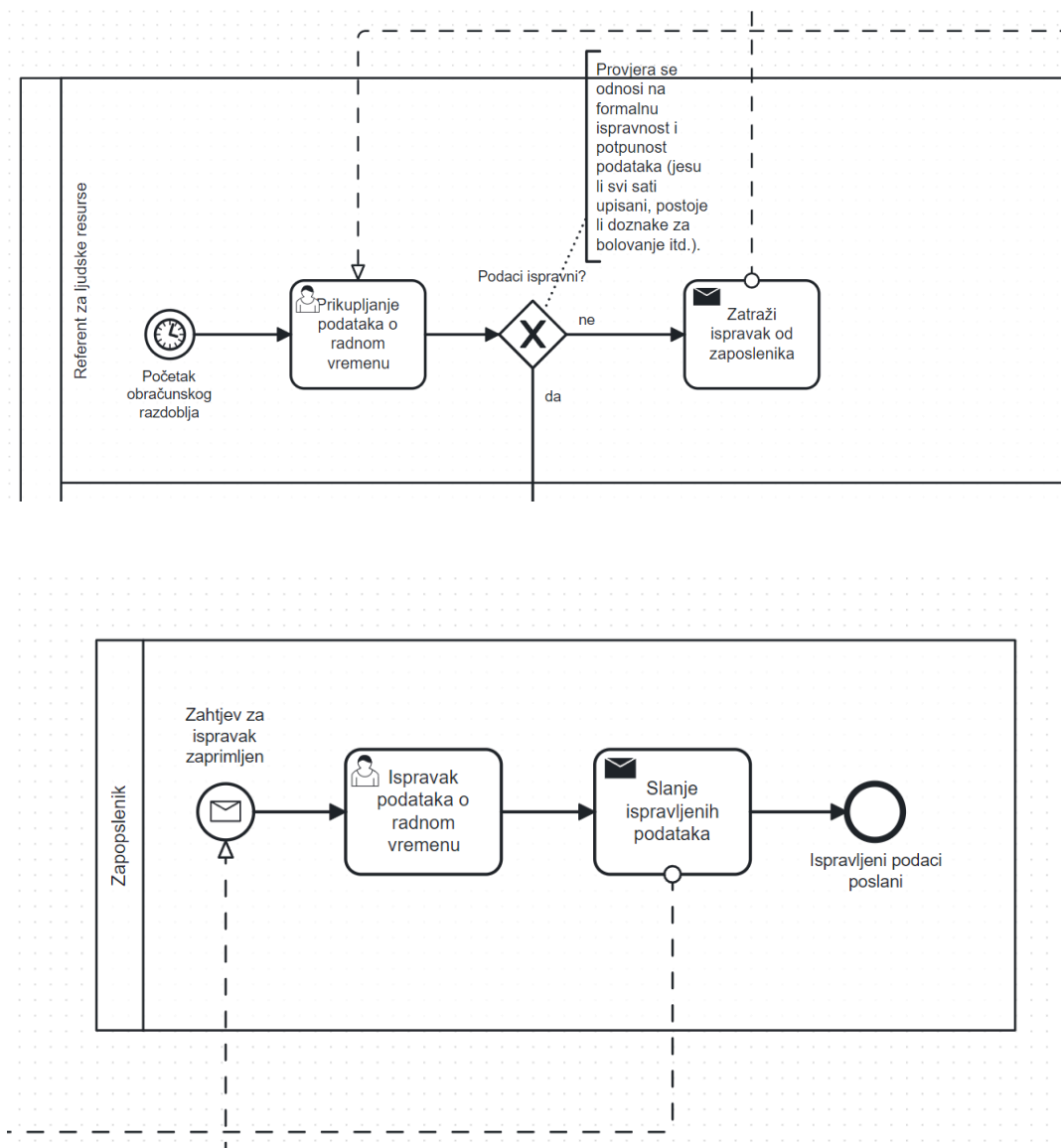
2.2. As-Is analiza

Kako bih dubinski razumjela trenutni način rada, izradila sam detaljan As-Is model procesa u Camunda Modeleru. Odlučila sam se za kolaboracijski dijagram jer on najbolje prikazuje interakcije i podjelu odgovornosti između svih sudionika: Referenta za ljudske resurse, Voditelja odjela, centralnog odjela Računovodstva i vanjskih sudionika, Banke i Zaposlenika.

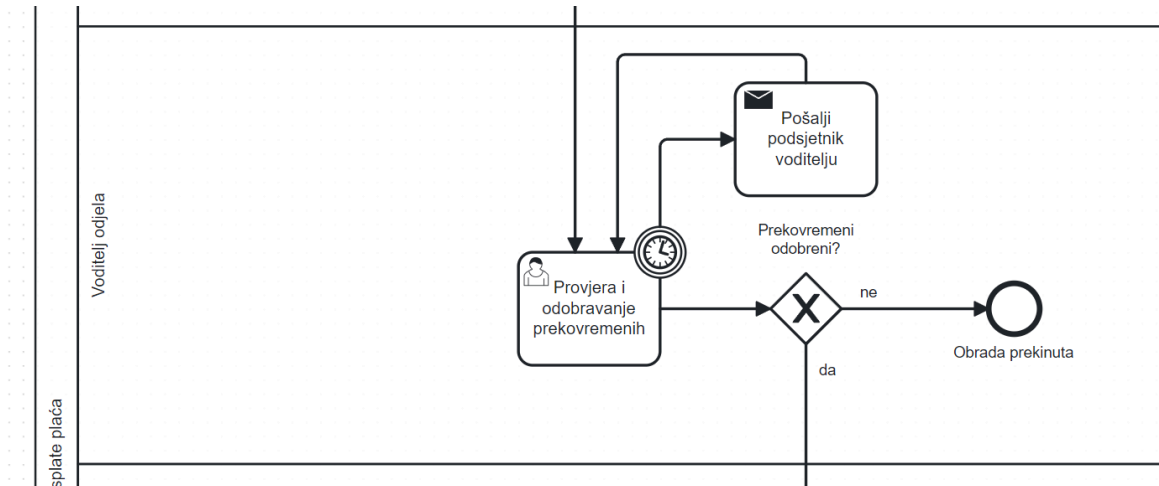


Proces se može podijeliti u nekoliko logičkih faza.

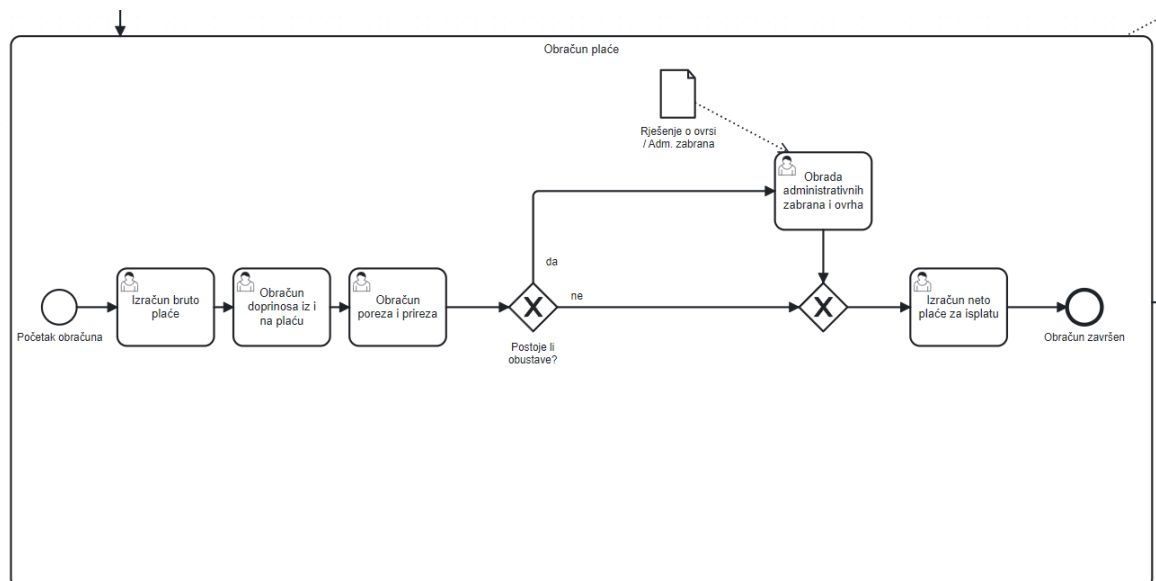
Sve započinje **pripremom i validacijom podataka**. Referent za ljudske resurse prikuplja podatke o radnom vremenu i provjerava njihovu ispravnost. Već ovdje nailazimo na prvo veliko usko grlo. Ako podaci nisu točni, referent mora kontaktirati zaposlenika i zatražiti ispravak. Ta komunikacija, koja se najčešće odvija putem e-maila, stvara petlju koja može zaustaviti proces na nekoliko dana dok se ne dobije odgovor.

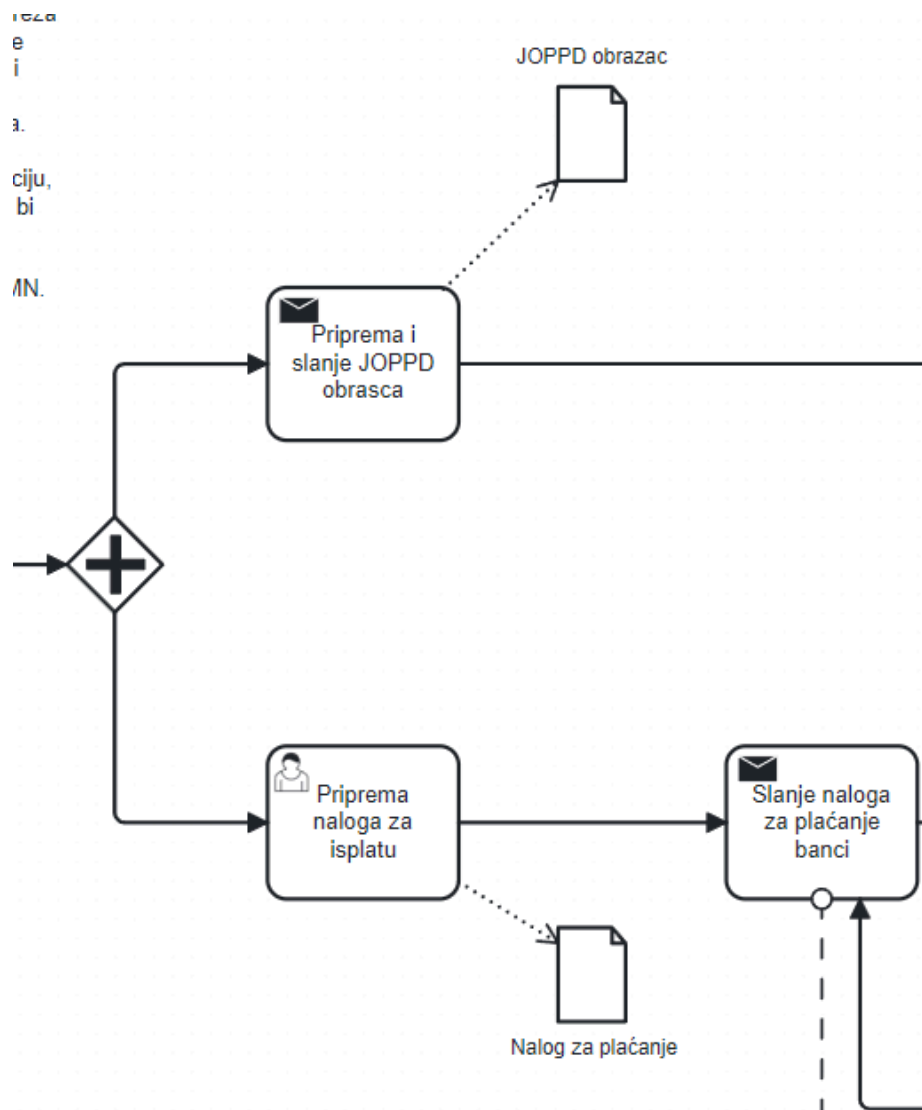


Nakon što su podaci ispravni, proces prelazi na **fazu odobrenja prekovremenih sati**. Ovdje voditelj odjela, nakon primljenog podsjetnika, mora ručno pregledati i odobriti prekovremene sate. Zanimljivo je da u slučaju neodobravanja, proces završava u stanju "Obrada prekinuta".



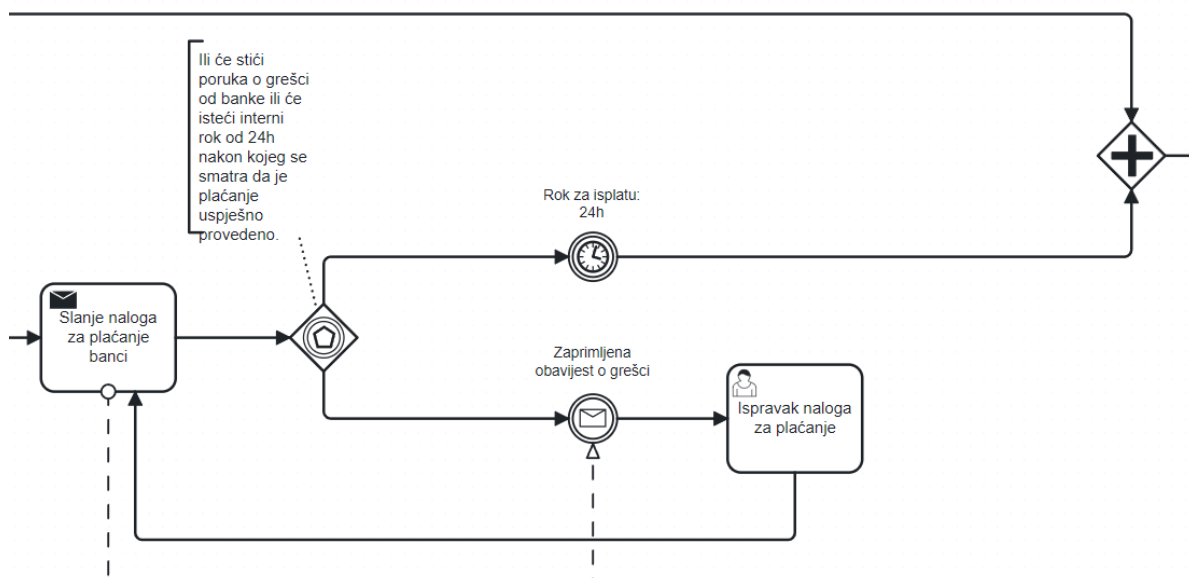
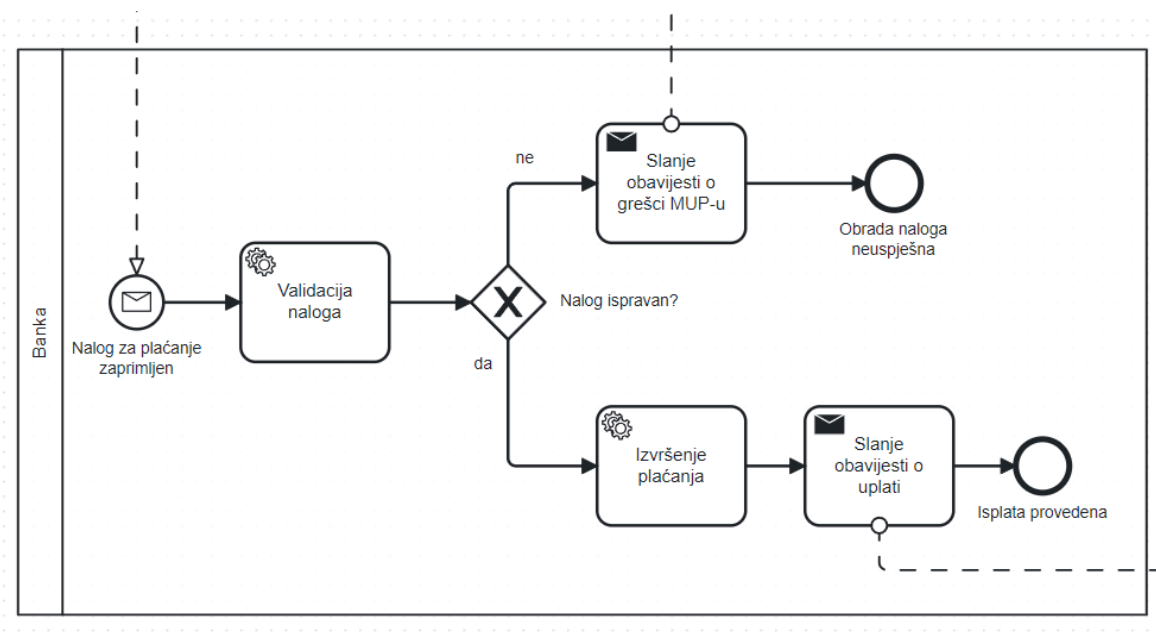
Kada su svi podaci prikupljeni i odobreni, ulazimo u **centralnu fazu obračuna plaće** unutar računovodstva. Ovdje se sekvencijalno odvijaju ključni zadaci: obračun bruto plaće, doprinosa, poreza i prireza. Nakon toga, provjerava se postojanje obustava kod zaposlenika. U ovoj fazi uočila sam i jedan primjer dobre prakse. Nakon što se izračuna neto iznos, proces koristi paralelno grananje (Parallel Gateway) kako bi se istovremeno pripremali JOPPD obrazac i nalog za isplatu. Time se efikasno štedi vrijeme jer se dva zadatka mogu obavljati neovisno jedan o drugome.



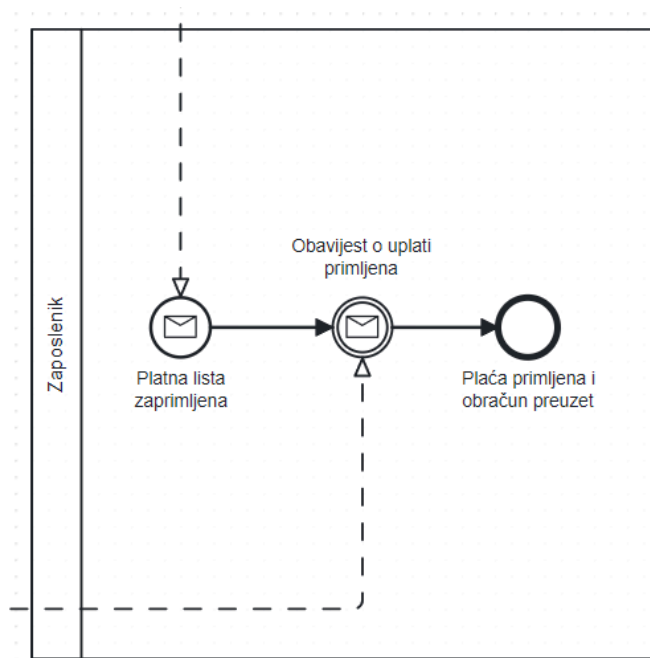
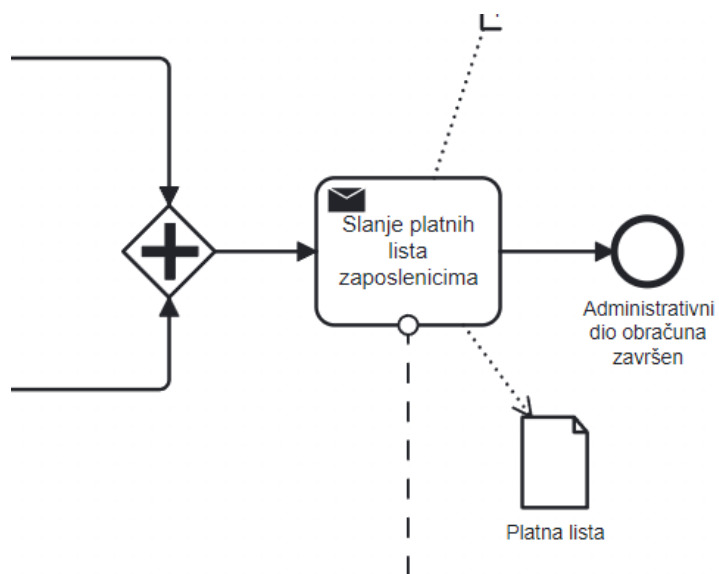


Posljednja ključna faza je **interakcija s bankom i sama isplata**. Računovodstvo šalje nalog za plaćanje, a zatim proces ulazi u stanje čekanja pomoću Event-Based Gatewaya. Ovo je napredniji dio modela, gdje proces čeka na jedan od dva moguća događaja. Prvi je istek vremenskog roka od 24 sata, nakon kojeg proces nastavlja dalje pod pretpostavkom da je sve prošlo u redu. Drugi događaj je primanje poruke o grešci od banke, što odmah pokreće zadatak ispravka naloga. Iako je mehanizam za obradu greške pohvalan, oslanjanje na timer kao potvrdu uspjeha je rizično. Ako banka ne javi

grešku, a nalog iz nekog razloga nije proveden, računovodstvo će imati pogrešnu informaciju. Idealno, proces bi trebao čekati eksplicitnu potvrdu o uspješnoj transakciji.



Na samom kraju, nakon što su svi tokovi završeni, zaposleniku se šalje platna lista i proces je gotov. Kroz ovu detaljnu analizu, iako sam uočila neke dobre prakse poput paralelizacije i rukovanja greškama, postalo je jasno da je proces opterećen ručnim radom, neefikasnom komunikacijom i rizičnim pretpostavkama, što ga čini idealnim kandidatom za optimizaciju.



2.3. To-Be analiza

Nakon detaljne analize trenutnog (As-Is) stanja procesa i identifikacije njegovih ključnih slabosti, sljedeći korak je definiranje ciljanog, poboljšanog (To-Be) stanja. Cilj nije samo ispraviti pojedinačne probleme, već redizajnirati proces tako da postane brži, pouzdaniji, jeftiniji i transparentniji za sve uključene. Prijedlozi se temelje na principima automatizacije, digitalizacije komunikacije i pametnije integracije sa sustavima.

Ključni ciljevi poboljšanja su:

- Povećanje učinkovitosti: Značajno skratiti vrijeme potrebno za cijeli obračun, od prikupljanja podataka do isplate.
- Smanjenje rizika od pogreške: Minimizirati ljudske greške koje nastaju zbog ručnog unosa i prepisivanja podataka.
- Povećanje transparentnosti: Omogućiti svim sudionicima da u svakom trenutku znaju u kojoj je fazi proces i tko je na potezu.

Na temelju ovih ciljeva, predlažem sljedeće ključne promjene u procesu:

Automatizacija prikupljanja i validacije podataka

Prvi i najočitiji korak je eliminacija ručnog prikupljanja podataka. Umjesto da referent za ljudske resurse ručno skuplja i unosi podatke, novi proces bi trebao započeti **automatskim povlačenjem podataka**. To bi se realiziralo pomoću Service Task koji bi se putem API-ja spajao direktno na elektronički sustav za evidenciju radnog vremena. Time se ne samo ubrzava proces, već se i eliminira rizik od pogrešaka pri prijepisu. Osnovnu validaciju, poput provjere jesu li svi dani pokriveni, također može odraditi sustav automatski.

Digitalizacija komunikacije i eskalacija problema

Problematičnu petlju za ispravak podataka i čekanje na odobrenje voditelja treba u potpunosti digitalizirati. Kada sustav detektira neispravnost podataka, automatski bi kreirao zadatak direktno za zaposlenika s jasnim uputama što treba ispraviti. Slično

tome, zahtjev za odobrenje prekovremenih sati postao bi zadatak u listi voditelja odjela. Moguće je definirati rokove za izvršenje zadataka i, ako se rok prekorači, sustav može automatski poslati podsjetnik ili čak eskalirati problem na višu razinu. Time se nestrukturirana i spora e-mail komunikacija zamjenjuje sljedivim i upravljivim procesom.

Potpuna integracija s bankom i pametno rukovanje ishodima

Interakciju s bankom treba redizajnirati kako bi se eliminirao rizik i nepotrebno čekanje. Ručno slanje naloga i oslanjanje na 24-satni timer su zastarjele prakse.

U To-Be procesu, slanje naloga banci bio bi zadatak koji direktno komunicira s API-jem banke. Nakon slanja, proces više ne bi čekao na fiksni timer. Umjesto toga, koristio bi Message Intermediate Catch Event. Proces bi bio u stanju čekanja sve dok od banke ne primi **eksplicitnu poruku** o ishodu transakcije, bilo da je ona uspješna ili neuspješna.

Ovakav pristup donosi ogromne prednosti. Prvo, eliminira se nepotrebno čekanje – proces se nastavlja onog trenutka kad stigne potvrda. Drugo, eliminira se rizik od pogrešne pretpostavke; proces točno zna je li plaćanje prošlo. U slučaju greške, proces se može odmah preusmjeriti na rješavanje problema, bez gubitka vremena.

Jedinstveni izvor istine za podatke

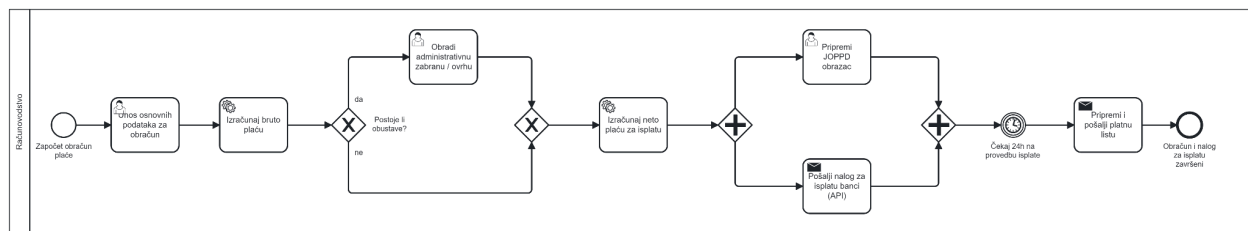
Naposljetku, kako bi se osigurala konzistentnost, svi podaci relevantni za obračun – od sati rada, preko podataka o zaposleniku do iznosa za isplatu – trebali bi biti pohranjeni kao **procesne varijable** unutar same instance procesa u Camundi. Time se izbjegava korištenje nepouzdatih i nepovezanih Excel tablica i osigurava da svi sudionici u svakom trenutku barataju istim, ažurnim i točnim podacima.

Ukratko, predložene promjene transformirale bi trenutni, većinom ručni i reaktivni proces u visoko automatiziran, proaktivan i robustan sustav koji značajno smanjuje operativni napor i povećava pouzdanost jedne od najkritičnijih funkcija u organizaciji.

3. Razvoj *model-driven* aplikacije

Nakon teorijske analize i prijedloga za poboljšanje, drugi, praktični dio projekta bio je usmjeren na izradu funkcionalne, model-driven aplikacije korištenjem Camunda 7 procesnog enginea. Cilj je bio demonstrirati kako se BPMN model može pretvoriti u izvršnu aplikaciju koja automatizira zadatke, upravlja ljudskim interakcijama i integrira se s vanjskim sustavima.

Zbog složenosti cjelokupnog As-Is modela, za potrebe implementacije definirala sam pojednostavljenu verziju procesa. Fokusirala sam se na ključne aktivnosti koje se odvijaju unutar odjela računovodstva. Ovaj sažeti model implementiran je kao izvršni proces unutar jednog bazena (poola), što je u skladu sa zadatkom koji dopušta takvo pojednostavljenje.



3.1. Izvođenje procesa

Nakon što je model deployan na Camunda engine, pokrenula sam novu instancu procesa kako bih simulirala stvarni obračun plaće. Cijeli tijek izvođenja može se pratiti kroz dva ključna sučelja Camunde: Cockpit i Tasklist.

Camunda Tasklist je sučelje namijenjeno krajnjem korisniku, u ovom slučaju djelatniku računovodstva. Proces započinje korisničkim zadatkom *Unos osnovnih podataka za obračun*. Za ovaj, kao i za druge korisničke zadatke, koristila sam **Generated Task Form**. Kao što je vidljivo na slici, forma za ovaj konkretni zadatak sadrži polja osnovnaPlaca i postojeObustave, koja korisnik unosi ručno.

postojeObustave

ID

postojeObustave

Refers to the process variable name

Label

Postoje li obustave na plaći?

Type

boolean

Default value

Constraints

+

Properties

+

osnovnaPlaca

ID

osnovnaPlaca

Refers to the process variable name

Label

Unesite osnovnu plaću:

Type

long

Default value

Constraints

+

Properties

+

Unos osnovnih podataka za obračun

racunovodstvo

Set follow-up date

Set due date

Add groups

Demo Demo

Form

History

Diagram

Description

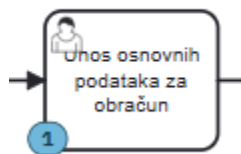
Postoje li obustave na plaći?

☐

Unesite osnovnu plaću:

Save

Complete



Jedan od primjera automatizacije unutar samog procesa je zadatak *Izračunaj neto plaću za isplatu*. Ovaj zadatak je implementiran kao Service Task koji koristi **JUEL izraz (Expression)** za izračun. Izraz dinamički postavlja vrijednost procesne varijable `netoPlaca` na temelju prethodno unesenih podataka:

```
${execution.setVariable('netoPlaca', postojeObustave ? osnovnaPlaca - iznosOvrhe : osnovnaPlaca)}
```

Ovaj jednostavan primjer pokazuje kako se poslovna logika može direktno ugraditi u procesni model, smanjujući potrebu za ručnim izračunima.

3.2. Integracija s vanjskim servisima (Connectori)

Slanje naloga za isplatu banci (REST API)

Zadatak *Pošalji nalog za isplatu banci (API)* modeliran je kao Send Task i koristi ugrađeni **HTTP Connector**. Ovaj konektor je konfiguriran da pošalje HTTP POST zahtjev na endpoint jednostavnog REST API servisa koji sam razvila korištenjem **Node.js** i **Express** okvira.

Connector inputs + 4

url

Local variable name

url

Assignment type

String or expression

Value

http://host.docker.internal:3000/api/banka/nalog

Start typing "\${}" to create an expression.

method

Local variable name

method

Assignment type

String or expression

Value

POST

Start typing "\${}" to create an expression.

headers

Local variable name

headers

Assignment type

Map

Map entries + 1

Content-Type

Key

Content-Type

Value

application/json

payload

Local variable name

payload

Assignment type

String or expression

Value

{
 "primatelj": "Zaposlenik MUP-a",
 "iznos": "\${netoPlaca},
 "valuta": "EUR",
 "opis": "Isplata place",
 "hitno": false
}

Start typing "\${}" to create an expression.

Command Prompt - node index.js

```
Microsoft Windows [Version 10.0.19045.5965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Petra>cd C:\Users\Petra\Desktop\FIPU\3\UPP\PROJEKT\camunda-api-servis
C:\Users\Petra\Desktop\FIPU\3\UPP\PROJEKT\camunda-api-servis>node index.js
Poslužitelj sluša na adresi http://localhost:3000
-----
API POZVAN: Zaprimljen nalog za isplatu!
> Primatelj: Zaposlenik MUP-a
> Iznos za isplatu: 1440 EUR
> Opis plaćanja: Isplata place
-----
```

API servis, koji se izvršava lokalno, simulira sustav banke. On prima podatke o nalogu (iznos, primatelj, opis), ispisuje ih u konzoli radi demonstracije.

```
app.post('/api/banka/nalog', (req, res) => {
  console.log('-----');
  console.log('API POZVAN: Zaprimljen nalog za isplatu!');

  const nalog = req.body;

  console.log(` > Primatelj: ${nalog.primatelj}`);
  console.log(` > Iznos za isplatu: ${nalog.iznos} ${nalog.valuta}`);
  console.log(` > Opis plaćanja: ${nalog.opis}`);

  console.log('-----');

  res.status(200).json({
    status: 'uspjeh',
    poruka: `Nalog za isplatu u iznosu od ${nalog.iznos} ${nalog.valuta} je uspješno zaprimljen.`
  });
});
```

Slanje platne liste zaposleniku putem e-maila

Slično kao i slanje naloga, zadatak *Pripremi i pošalji platnu listu* također je implementiran kao Send Task s HTTP konektorom. Međutim, ovaj konektor poziva drugi, zasebni mikroservis koji sam razvila za slanje e-mailova.

Taj mikroservis, također baziran na **Node.js** i **Expressu**, djeluje kao posrednik. On prima podatke o platnoj listi od Camunde, a zatim koristi vanjski servis **EmailJS** kako bi te podatke formatirao u e-mail i poslao ga na definiranu adresu (u ovom slučaju, moju).

```


app.post("/send-email", async (req, res) => {
  try {
    const dataFromCamunda = req.body;
    console.log("Priljeni podaci od Camunde:", dataFromCamunda);

    const emailJsPayload = {
      service_id: SERVICE_ID,
      template_id: TEMPLATE_ID,
      user_id: PUBLIC_KEY,
      accessToken: PRIVATE_KEY,
      template_params: dataFromCamunda
    };

    await axios.post(
      "https://api.emailjs.com/api/v1.0/email/send",
      emailJsPayload
    );

    console.log("Email uspješno poslan!");
    res.status(200).json({ message: "Email uspješno poslan!" });
  } catch (error) {
    console.error("Greška prilikom slanja emaila: ", (error.response && error.response.data) || error.message);
    res.status(500).json({ error: "Greška prilikom slanja emaila!", details: (error.response && error.response.data) || error.message });
  }
});

```

 Command Prompt - node index.js

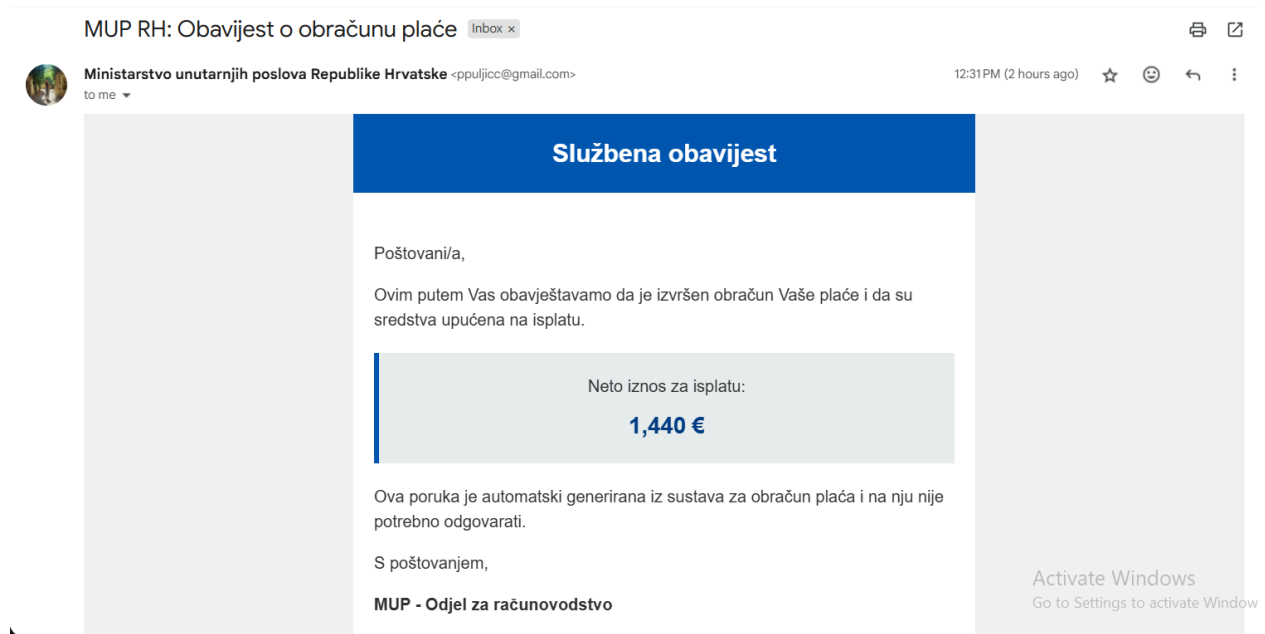
```

Microsoft Windows [Version 10.0.19045.5965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Petra>cd C:\Users\Petra\Desktop\FIPU\3\UPP\PROJEKT\express-email-server

C:\Users\Petra\Desktop\FIPU\3\UPP\PROJEKT\express-email-server>node index.js
[dotenv@17.0.0] injecting env (4) from .env -  encrypt with dotenvx: https://dotenvx.com
Email servis sluša na http://localhost:3002
Priljeni podaci od Camunde: { neto_iznos: '1,440', to_email: 'ppuljicc@gmail.com' }
Email uspješno poslan!

```



Ovaj primjer demonstrira kako se BPM sustav može koristiti za potpunu automatizaciju komunikacije sa zaposlenicima, zamjenjujući ručno slanje platnih lista brzim i pouzdanim automatskim rješenjem.

Kroz izradu ove model-driven aplikacije, uspješno sam pokazala kako se teorijski BPMN model može pretvoriti u živu, izvršnu aplikaciju koja kombinira ljudske zadatke, automatiziranu poslovnu logiku i integraciju s modernim REST API servisima.

4. Zaključak

Ovaj seminarski rad obuhvatio je cjeloviti ciklus upravljanja poslovnim procesima, primijenjen na konkretnom primjeru procesa obračuna i isplate plaća. Kroz kombinaciju teorijske analize i praktične implementacije, cilj je bio demonstrirati kako se sustavnim pristupom i primjenom modernih BPM alata mogu postići značajna poboljšanja u ključnim poslovnim operacijama.

Analiza trenutnog stanja (As-Is) otkrila je da je odabrani proces, unatoč svojoj kritičnoj važnosti, opterećen inherentnim neefikasnostima. Identificirani su ključni problemi poput visokog udjela ručnog rada, nestrukturirane komunikacije koja uzrokuje zastoje te rizičnih pretpostavki u interakciji s vanjskim sustavima. Ove slabosti ne samo da produljuju vrijeme izvršenja procesa, već i povećavaju operativni rizik.

Na temelju provedene analize, u To-Be modelu predložena su konkretna rješenja usmjerena na automatizaciju, digitalizaciju i sistemsku integraciju. Prijedlozi poput automatskog dohvata podataka, upravljanja zadacima unutar BPM platforme i direktne API komunikacije s vanjskim servisima predstavljaju jasan put prema optimiziranom, pouzdanijem i transparentnijem procesu.

Središnji dio rada bila je izrada funkcionalne *model-driven* aplikacije na Camunda 7 platformi. Implementacijom pojednostavljenog, ali reprezentativnog dijela procesa, potvrđena je praktična izvedivost predloženih poboljšanja. Kroz kombinaciju korisničkih formi za unos podataka, automatskih servisnih zadataka za izvršavanje poslovne logike te integracije s vanjskim REST API servisima, uspješno je demonstrirano kako se apstraktni BPMN model može transformirati u izvršnu poslovnu aplikaciju.

U konačnici, ovaj projekt pružio je sveobuhvatan uvid u praktičnu primjenu BPM metodologije. Potvrđeno je da sustavni pristup analizi i redizajnu procesa, podržan moćnim alatima kao što je Camunda, omogućuje organizacijama da ostvare mjerljive koristi u vidu smanjenih troškova, povećane učinkovitosti i, najvažnije, minimiziranja rizika u ključnim segmentima poslovanja.