UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

# ANALIZA USPEŠNOSTI PROJEKTA ZAGOTAVLJANJA VISOKE RAZPOLOŽLJIVOSTI PODATKOVNE BAZE NA PRIMERU SPLETNEGA PORTALA BANKE

Ljubljana, junij 2016 MITJA MAKOVEC

# IZJAVA O AVTORSTVU

Podpisani Mitja Makovec, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Analiza uspešnosti projekta zagotavljanja visoke razpoložljivosti podatkovne baze na primeru spletnega portala banke, pripravljenega v sodelovanju s svetovalko prof. dr. Mojca Indihar Štemberger,

IZJAVLJAM

1. da sem predloženo delo pripravil samostojno;

2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;

3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;

4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;

5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;

6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;

7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;

8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;

9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;

10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 28.6.2016 Podpis študenta:_____

# KAZALO

**KAZALO TABEL**

## KAZALO SLIK

# UVOD

Bančništvo v svetu in tudi v Sloveniji se spopada z izzivom digitalizacije. Komitenti od svoje banke zahtevajo, da je dostopna vedno in povsod ter da lahko bančne storitve opravijo brez fizičnega obiska bančne poslovalnice. Velik pomen zato banke namenjajo načinu, kako zagotoviti, da njihova informacijska podpora deluje ves čas nemoteno. V drugih panogah je še mogoče govoriti o neprekinjenem poslovanju kot o konkurenčni prednosti, v bančništvu pa ni tako. Namreč ne samo zakonodajni okvir, ki od bank zahteva, da imajo urejene načrte neprekinjenega poslovanja, tudi komitenti zahtevajo, da je storitev na voljo vedno, ko jo potrebujejo.

Zato si banke ne morejo dovoliti, da njihove storitve v določenem trenutku ne delujejo, in svoje informacijske sisteme konfigurirajo tako, da se zagotavljata čim večja razpoložljivost in dostopnost. Prav tako je tudi poslovanje bank skoraj v celoti informacijsko podprto, tudi večina bančnih podatkov je v elektronski obliki. Izguba podatkov bi tako pomenila veliko grožnjo za delovanje banke. S tem namenom se pripravljajo načrti za zagotavljanje neprekinjenega poslovanja v primeru nepričakovanih dogodkov kot načina zagotavljanja neprekinjenega delovanja bank v primeru nesreč.

**Cilj magistrskega dela** je analiza uvedene rešitve zagotavljanja visoke razpoložljivosti podatkovne baze na primeru spletnega portala banke. Teza je, da je od razpoložljivosti posameznega dela celotnega informacijskega sistema odvisno, ali le-ta zagotavlja zahtevano razpoložljivost in obnovljivost po nesreči, ter da je rešitev v skladu z zahtevami načrta neprekinjenega poslovanja banke in bo vplivala na izbor ter uvedbo rešitve tudi pri drugih projektih v banki ter tudi na izboljšanje obstoječe rešitve.

**Metoda dela** je bila proučevanje konkretnega primera in analiza uvedbe visoke razpoložljivosti podatkovne baze na primeru spletnega portala banke ter ugotovitev ustreznosti izbrane rešitve. V magistrskem delu sem uporabil teoretično in praktično znanje, ki sem ga pridobil med dodiplomskim in podiplomskim študijem na ekonomski fakulteti ter na različnih izobraževanjih v banki, predvsem pa z delom na primeru v banki.

V teoretičnem delu sem uporabil metodo študija literature. Usmeril sem se predvsem na tehnično dokumentacijo, članke in strokovno literaturo s področja neprekinjenega poslovanja, visoke razpoložljivosti in delovanja sistemov za upravljanje baz podatkov. Pri pregledu literature sem poskušal opisati problem z vidika več avtorjev.

V praktičnem delu pa sem se usmeril v analizo obstoječega stanja ter s pomočjo literature in teoretičnih dognanj ocenil začetno stanje, nato pa sem z analizo s pomočjo metode tehtanih vsot in analizo prednosti, slabosti ter nevarnosti in priložnosti kritično ocenil uvedeno rešitev visoke razpoložljivosti podatkovne baze spletnega portala. Prav tako pa sem na rezultatih ugotovitev analize predlagal možnosti za izboljšavo rešitve.

**Magistrsko delo** je razdeljeno na šest poglavij. V prvih dveh poglavjih so podana teoretična izhodišča o neprekinjenem poslovanju v podjetjih in vplivu informacijske tehnologije na zagotavljanje neprekinjenega poslovanja in obnove po nesreči, proučevanje področja temelji na študiji literature predvsem tujih avtorjev.

V tretjem poglavju je predstavljeno področje neprekinjenega poslovanja v bančništvu, pregled zakonodaje in smernic s strani regulatorja Banke Slovenije in Evropske centralne banke, sledi pregled zagotavljanja neprekinjenega poslovanja večjih slovenskih bank, pri čemer analiza temelji na prosto dostopnih podatkih slovenskih bank. V nadaljevanju sem opisal področje neprekinjenega poslovanja v proučevani banki. Ker je varovanje informacij v bančništvu zelo strogo, so nekatere podrobnosti v magistrskem delu namenoma spremenjene ali izbrisane, kar pa ne vpliva na bistvo problema, ki ga proučujem.

V nadaljnjih poglavjih je poudarek predvsem na proučevanem primeru; tako v četrtem poglavju predstavim arhitekturo spletnega portala banke in samo rešitev visoke razpoložljivosti podatkovne baze. Zadnje poglavje je namenjeno analizi rešitve, kjer s pomočjo pregledane literature definiram zahteve in kriterije, ki jih po mojem mnenju rešitev mora zagotavljati, nato pa z metodo tehtanih vsot in analizo prednosti in slabosti ter priložnosti in nevarnosti (v nadaljevanju SWOT-analiza) preverim, ali uvedena rešitev res ustreza zahtevam.

Namen magistrskega dela je torej proučiti uvedbo visoke razpoložljivosti podatkovne baze spletnega portala banke kot načina zagotavljanja nemotenega delovanja storitve ter predlog možnih izboljšav rešitve.

# 1 NEPREKINJENO POSLOVANJE PODJETJA

Bančništvo v svetu in v Sloveniji se spopada z izzivom digitalizacije. Komitenti od svoje banke zahtevajo, da je dostopna vedno in povsod ter da lahko bančne storitve opravijo brez fizičnega obiska bančne poslovalnice kadarkoli in kjerkoli. Tradicionalne banke, ki so navajene, da komitenti pridejo v banko, so postavljene pred izziv. Da bi se izognile izgubi veljave in komitentov, se morajo banke nujno začeti prilagajati novim zahtevam, predvsem mlajših generacij komitentov.

Banke morajo zagotavljati, da njihova informacijska podpora deluje neprekinjeno. Če je neprekinjeno delovanje informacijskih sistemov še pred nekaj leti veljalo za konkurenčno prednost podjetij, je v današnjem času pričakovano in samoumevno. Banke si namreč ne morejo dovoliti, da njihove storitve v določenem trenutku ne delujejo. Kot navaja Forrester (2013, str. 1), je danes zahteva trga, da podjetja zagotavljajo storitve 24/7, v nasprotnem primeru jih čakajo finančne izgube in dolgoročne posledice izgube zaupanja. Zaradi velike konkurence na trgu se v primeru nedelovanja storitev podjetja veliko njihovih kupcev preusmeri h konkurenčnim podjetjem, ki lahko storitev zagotavljajo nemoteno. Ugotovitve

kažejo, da je vedno manj ljudi pripravljenih počakati na delujočo storitev oziroma da se je stopnja tolerance do prekinitev delovanja storitev v primerjavi s preteklostjo znižala.

Vedno bolj digitalizirano poslovanje bank pa tudi povečuje pomen in vrednost samih podatkov. Hiles (2002, str. 198) navaja, da je povprečna vrednost 100 mb podatkov vrednih približno 1 milijon dolarjev, zaradi izgube podatkov pa evropska podjetja letno izgubijo več kot 4,5 milijarde dolarjev, brez vštetih posrednih stroškov izgube komitentov, tržnega deleža in zakonskih kazni.

Bančništvo, ki je ena ob bolj reguliranih panog, je podvržena tudi zakonskim zahtevam. Ena izmed zahtev je tudi zahteva po vzpostavitvi sistemov neprekinjenega poslovanja ter zagotavljanja obnovitve po nesrečah. Evropska centralna banka od svojih članic zahteva, da imajo vzpostavljene načrte neprekinjenega poslovanja, da zagotavljajo krizno vodenje ter da stalno preverjajo in testirajo načrte neprekinjenega poslovanja (Evropska Centralna Banka, 2006, str. 3).

Banke ter ostale finančne institucije morajo imeti izdelan načrt, ki varuje zaupnost, dostopnost in integriteto finančnih podatkov. Snedaker in Rima (2014, str. 416) govorita o nujnosti zagotovitve finančnih institucij, da njihovi podatki ostanejo zaupni, dostopni in nespremenjeni, tudi v primeru kriznega dogodka oziroma nesreče.

Uvedbo neprekinjenega poslovanja podjetja naj bi, po navajanju Protiviti (2013, str. 12), podjetja uvajala za obvladovanje naslednjih skupin tveganj:

- **Regulatorna tveganja** so glavni dejavnik za uvedbo neprekinjenega poslovanja veliko panog, med njimi so tudi bančna podvržene zahtevam regulatorjev panoge. Podjetja, ki nimajo uvedenega načrta neprekinjenega poslovanja, lahko doleti kazen ali pa se jim celo prepove opravljanje dejavnosti.
- **Finančna tveganja**, podjetja se z uporabo načrtov neprekinjenega poslovanja poskušajo izogniti izgubam tako, da se usmerjajo na dejavnike, ki manjšajo izgubo in ohranjajo tržni delež. Dejavniki so med drugimi upoštevanje zahtev kupcev, razumevanje odgovornosti ter minimizacija kritičnih točk odpovedi in zunanjih odvisnosti
- **Tveganje izgube ugleda** je tretja pomembna skupina tveganj, ki vpliva na uvedbo neprekinjenega poslovanja. Glavna dejavnika, povezana s tveganjem izgube ugleda, sta zaščita ugleda podjetja pred konkurenco ter vzdrževanje ugleda podjetja z načinom upravljanja v krizi.

## 1.1 Opredelitev neprekinjenega poslovanja podjetja

Termina neprekinjeno poslovanje in obnova po nesreči (angl. *disaster recovery,* v nadaljevanju DR) se velikokrat zamenjujeta ali se uporabljata izmenično, kar povzroča nejasnosti, kaj naj bi sam termin neprekinjenega poslovanja pomenil. Tako se neprekinjeno

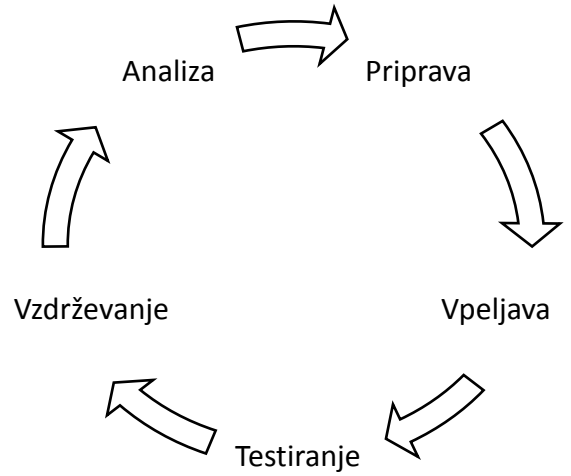
poslovanje definira z zagotavljanjem neprekinjenega delovanja informacijskih sistemov. Ta definicija pa ni pravilna, kajti če želimo, da je načrt neprekinjenega poslovanja uspešen, je treba zajeti veliko večji obseg in vključiti v načrt še druge dele in procese podjetja. Lahko bi rekli, da je obnova po nesreči samo del, in ne sopomenka neprekinjenega poslovanja. Ali drugače, neprekinjeno poslovanje bi lahko opisali tudi kot zbirko sorodnih disciplin: obnovitev po nesreči, krizni management ter načrt neprekinjenega poslovanja (Savage, 2002, str. 255; Watters, 2014, str. 22).

Standard ISO 222300 neprekinjeno poslovanje definira kot sposobnost podjetja, da zagotavlja svoje proizvode in storitve po sprejemljivi vnaprej določeni ravni tudi v primeru nesreče.

Watters (2014, str. 4) pravi, da upravljanje neprekinjenega poslovanja (angl. *business continuity management,* v nadaljevanju BCM) zagotavlja, da ne pride do ovir, ki bi lahko povzročile velike izgube podjetja ali težave pri doseganju postavljenih ciljev podjetja. Za neprekinjeno poslovanje pravi, da je tako proces kot disciplina.

Proces neprekinjenega poslovanja je cikličen in poteka v podobnih korakih kot drugi procesi stalnega izboljševanja. Življenjski cikel neprekinjenega poslovanja predstavlja Slika 1.

*Slika 1: Življenjski cikel procesa neprekinjenega poslovanja*




*Vir: Prirejeno po J. Watters, Disaster recovery, crisis response, and business continuity a management desk reference, 2014, str. 4, slika 1-1.*

Proces je torej ponovljiv in upravljan s strani več oseb z različnimi znanji in izkušnjami, zagotavlja pa stalno obnavljanje in izboljševanje. Uporaba kontroliranega procesa je zaželena, saj prinaša dodano vrednost pri revizijah postopkov, saj so vsi postopki beleženi (Watters, 2014, str. 5).

Disciplino neprekinjenega poslovanja pa definira kot posledično povezavo nalog in ljudi, ki jih opravljajo. Disciplina neprekinjenega poslovanja je skupek politik ter ljudi in timov, ki so zadolženi za cikle neprekinjenega poslovanja.

Watters (2014, str. 5) deli ljudi, ki podpirajo neprekinjeno poslovanje, na:

- planerje, ki definirajo, kateri proces je kritičen in kako zagotoviti njegovo neprekinjeno delovanje,
- informatike, ki so zadolženi za kritične informacijsko tehnološke rešitve (v nadaljevanju IT-storitve), ki morajo biti delujoče in podpirati kritične poslovne aktivnosti,
- krizni menedžment, ki je zadolžen za nadzor poslovanja ter pravočasno ukrepanje v primeru aktivacije načrta neprekinjenega poslovanja.

Protiviti (2013, str. 10) vodstvene vloge pri zagotavljanju upravljanja neprekinjenega poslovanja razdeli na:

- sponzorske – zagotavljajo podporo na organizacijski in finančni ravni,
- lastniške – zadolžene za izvajanje in podporo pri izvajanju načrta,
- skrbniške – zadolžene za koordinacijo nalog neprekinjenega poslovanja v podjetju.

Ob ustanovitvi interne skupine zaposlenih za izvajanje nalog neprekinjenega poslovanja je nujno jasno opredeliti in komunicirati njihove pristojnosti ter naloge. Zaposleni se morajo izobraziti za področje, na katerem bodo delovali, saj bodo le tako lahko suvereno opravljali svojo funkcijo. Napaka, ki jo naredi veliko podjetij, je, da delegirajo zadolžitve in odgovornost za izvajanje in izdelavo načrta neprekinjenega poslovanja na nižje ravni v podjetju, posledično načrti ne odražajo dejanskih zahtev podjetja, in ko je treba uporabiti načrt, se izkaže, da ima veliko prikritih lukenj (Protiviti, 2013, str. 14).

Za zagotavljanje neprekinjenega poslovanja je treba najprej opredeliti aktivnosti oziroma storitve, ki so kritične za določeno podjetje. Definiranje kritičnih aktivnosti je nujno, saj so le-te najpomembnejše za delovanje podjetja in jih je treba ščititi pred nesrečami (Watters, 2014, str. 35).

## 1.2 Načrt neprekinjenega poslovanja podjetja

Načrt neprekinjenega poslovanja je načrt, ki opredeljuje postopke za nemoteno delovanje v primeru različnih nesreč, ki lahko prizadenejo podjetja. Načrt pomaga zagotoviti, da podjetje v zahtevanem času nadaljuje delovanje.

Naravne nesreče, kot so potres, poplave, požari, se zgodijo v najbolj nepričakovanem času. Gregg (2006) pravi, da študije kažejo, da ima samo 50 odstotkov podjetij izdelane celovite

načrte neprekinjenega poslovanja. Za organizacije brez načrtov lahko večja nesreča pomeni ločnico med preživetjem in propadom.

Nesreče delimo na (Gregg, 2006):

- naravne – med njih štejemo potres, poplavo, požar, hurikane, tornade ter plazove,
- sistemske oziroma tehnične – zlonamerna koda, virusi, trojanci ter hekerji,
- infrastrukturne – težave pri dobavi električne energije, težave z opremo, pomanjkanje vode,
- politične – nemiri, nezadovoljni zaposleni, upori, vandalizem, kraja, politična negotovost.

Veliko avtorjev daje precejšen poudarek na naravne nesreče in njihov vpliv na neprekinjeno poslovanje podjetja, zato je zanimiva študija svetovalnega podjetja Forrester (2013, str. 2), ki v svoji raziskavi ugotavlja, da se tveganje nedelovanja povečuje, kar pomeni, da naravne nesreče ne morejo biti edini razlog za prekinitve v delovanju.

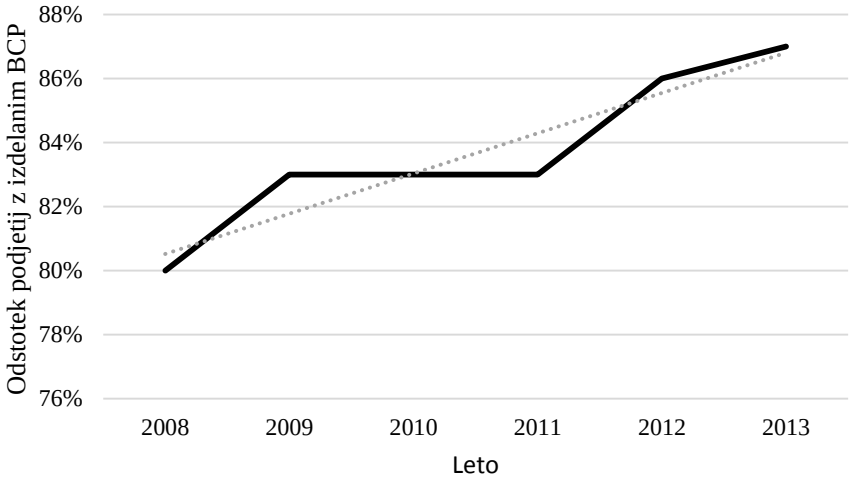
Kot glavne dejavnike povečevanja tveganja so v Forresterju (2013, str. 2) definirali: vedno večja odvisnost od tehnologije, povečana kompleksnost poslovanja in šele na tretjem mestu naravne nesreče. Drugi dejavniki so še odvisnost od zunanjega dobavitelja, povečane regulatorne zahteve, računalniški napadi in starost podatkovnih centrov.

Vsaka od teh lahko povzroči prekinitev v delovanju podjetja. Ko govorimo o prekinitvi delovanja, lahko glede na dolžino nedelovanja ločimo (Gregg, 2006):

- manjše prekinitve – Pomenijo prekinitev v delovanju sistemov podjetja za nekaj ur, vendar manj kot en dan,
- srednje prekinitve – Povzročijo prekinitev, daljšo od enega dneva, podjetje potrebuje aktivacijo rezervne lokacije, da lahko nadaljuje delo,
- večje prekinitve – Pri tem tipu prekinitve lahko že govorimo o katastrofi, saj večja prekinitev po navadi pomeni, da je celotna primarna lokacija podjetja podrta ter da bo podjetje na dolgi rok moralo razmisliti o prestavitvi lokacije ali ponovnem zidanju na obstoječi lokaciji.

Da se podjetja zavedajo, kako pomembni so načrti neprekinjenega poslovanja, kaže tudi raziskava, ki jo je izvedlo podjetje AT&T (v Mattord & Whitman, 2014, str. ix). V raziskavi so primerjali število podjetij, ki imajo vzpostavljen načrt neprekinjenega poslovanja po letih. Med letoma 2008 in 2013 je opazno rastoče število podjetij, ki so izdelala načrte neprekinjenega poslovanja. V letu 2013 je bilo takih podjetij več kot 87 odstotkov (Mattord & Whitman, 2014, str. 2). Trend rasti prikazuje Slika 2.

*Slika 2: Odstotek podjetij z izdelanim načrtom neprekinjenega poslovanja*




*Vir: Prirejeno po H. J. Mattord & M. E. Whitman, Business continuity: state of the industry report, 2014, str. 2, slika 1.*

Barnes (2001, str. 2) navaja, da aktivnosti izdelave načrta neprekinjenega poslovanja obsegajo naslednje korake:

- zagon projekta in tehnično analizo,
- analizo vpliva na poslovanje (angl. *business impact analysis*, v nadaljevanju BIA-analiza),
- izbiro strategije obnove,
- izdelavo načrta,
- testiranje in vzdrževanje.

ISACA deli načrt na naslednje korake (Kramer, 2003, str. 306):

- Vizija projekta – izdelava visokonivojske arhitekture z namenom opredelitve zahtev po obnovi, pogled vodstva in opredelitev področja obnove. Pomembno je, da je vizija vodstva usklajena z načrti obnove in finančnimi okvirji.
- Ocena tveganja – definiranje scenarijev tveganja ter možni učinki, ki potrebujejo načrt obnove.
- Izdelava strategije obnove – kot drugje je tudi pri izdelavi načrta neprekinjenega poslovanja možnih več variant rešitve, zato je treba izbrati tako, ki ustreza pričakovanjem. Priporoča se izdelava analize stroškov in koristi.
- Izdelava načrta – v tej fazi se izdela posamezni proces načrta obnove, ki mora upoštevati vse ugotovitve predhodnih korakov v procesu.
- Vzdrževanje načrta – vzdrževanje načrta je stalna naloga, saj se s testiranjem načrtov ali pa zaradi spremembe procesov lahko ugotovi odstopanja od zastavljenih ciljev, ki jih

opredeljuje načrt strategije. Zato je pomembno, da se redno izvajajo testiranja ter korigirajo morebitna odstopanja.
- Izobraževanje – pomembno je, da se zaposleni na vseh ravneh zavedajo in poznajo načrt ter njihovo vlogo v njem.
- Testiranje – načrt mora biti redno testiran za zagotavljanje pravilnosti postopkov, opisanih v njem.
- Potrditev načrta s strani višjega vodstva oziroma uprave – vodstvo mora potrditi načrt, saj se mu le tako doda večja vrednost in zavedanje, da je to edini pravilni načrt obnove oziroma zagotavljanja neprekinjenega poslovanja, ki se ga morajo držati zaposleni.

Pri načrtovanju neprekinjenega poslovanja je treba analizirati in popisati šest glavnih kategorij virov z namenom, da se opredeli vpliv na neprekinjeno poslovanje ter možnosti obnove. Glavne kategorije so naslednje (Kramer, 2003, str. 308–309):

- informacije in podatki,
- tehnologija in sistemi,
- telekomunikacijski sistemi,
- procesi in procedure,
- ljudje,
- prostori.

V nadaljevanju bo še bolj podrobno predstavljen korak analize vpliva na poslovanje, ker je ta korak pomemben za razumevanje zahtev po neprekinjenem poslovanju, kakor tudi velik dejavnik pri uvedbi rešitve za visoko razpoložljivost sistemov, v mojem primeru podatkovne baze spletnega portala. To pa ne pomeni, da drugi predhodni oziroma nadaljnji koraki v izdelavi načrta neprekinjenega poslovanja niso pomembni. Vsak korak pri izdelavi načrta je pomemben in se ne sme zanemariti, če želimo, da načrt dejansko odraža želeno stanje oziroma želene akcije ob težavah.

**Analiza vpliva na poslovanje**

Analiza vpliva na poslovanje ali BIA-analiza je metoda za ocenjevanje poslovnih procesov. Uporabna je predvsem, ker z analizo podjetje opredeli in popiše svoje kritične poslovne procese, za katere mora zagotoviti načrt neprekinjenega poslovanja, saj lahko nedelovanje ogrozi obstoj podjetja (Barnes, 2001, str. 57). Lahko rečemo, da v BIA-analizi ločimo med kritičnimi in manj kritičnimi procesi v podjetju. Ločitev je nujna, saj le tako lahko svoj fokus pri zagotavljanju neprekinjenega delovanja usmerimo na bolj kritične procese.

ISACA definira BIA-analizo kot »postopek, s katerim opredelimo vpliv izgube podpore katerega koli sredstva«. Ocena BIA-analize pomaga definirati, kako se bo sčasoma stopnjevala izguba, tako da lahko vodstvo podjetja oceni in sprejme ustrezno odločitev za sprejem načrtov za omejitev oziroma preprečitev škode (ISACA, 2012c, str. 23).

Savage (2002, str. 256) kot kritične procese vključuje proizvodnjo, dobavo, distribucijo, prodajo, IT, komunikacije, finance, razvoj in kadrovski proces.

Cilji BIA-analize so (Barnes, 2001, str. 68):

- ocena finančnega in operativnega učinka,
- definirati število zaposlenih in druga sredstva za zagotovitev uspešnosti obnove,
- za vsak identificirani kritični proces mora biti definiran ciljni čas vzpostavitve delovanja sistema oziroma procesa (angl. *Recovery Time Objective*, v nadaljevanju RTO) in ciljna točka obnovitve podatkov (angl. *Recovery Point Objective*, v nadaljevanju RPO).

**Kazalnik RTO** pove, kako hitro je možno obnoviti delovanje sistema ali procesa. RTO je tesno povezan s kritičnostjo procesa, saj imajo najbolj kritični procesi zahtevan nižji RTO kot manj kritični. Tako ima lahko najbolj kritičen proces RTO enak 0, kar pomeni, da je zahteva, da mora ta proces biti dosegljiv brez prekinitev. Watters (2014, str. 12) navaja, da je treba biti pazljiv, saj je razumevanje kazalnika med posameznimi zaposlenimi v podjetju različno. Na primer poslovna stran podjetja lahko razume RTO kot čas, ki je potreben, da se celoten proces ponovno vzpostavi v delovanje, medtem ko zaposleni v službi za informacijsko tehnologijo gledajo na kazalnik kot na čas, ki je zahtevan za vzpostavitev samo podpornih procesov informacijske tehnologije (v nadaljevanju IT-podpora).

**Kazalnik RPO** definira največjo količino izgube podatkov, ki si jo podjetje še lahko privošči v primeru nesreče. Kazalnik je predvsem pomemben za IT-podporo, saj zahteva po obnovitvi podatkov vpliva na vzdrževanje sistemov (na primer število izvedb varnostnega kopiranja podatkov, zanesljivost opreme), predvsem pa vpliva na samo arhitekturo sistemov informacijske tehnologije (v nadaljevanju IT-sistem). Tako RPO 0 pomeni, da se zahteva restavriranje vseh podatkov, RPO 8 ur pa pomeni, da je zadosti, da se obnovijo podatki iz kopije izpred 8 ur (Watters, 2014, str. 14).

Critchley (2015, str. 32) namesto kazalnika RTO predlaga uporabo stopenj pri oblikovanju zahtev v BIA-analizi, kamor poslovna stran klasificira aplikacije, z namenom, da se definira še dovoljeni čas prekinitve, brez velikega vpliva na poslovanje podjetja:

- stopnja 0 – zahteva po obnovitvi v uri ali manj,
- stopnja 1 – zahteva po obnovitvi v štirih urah,
- stopnja 2 – prekinitev lahko traja tudi dan ali več,
- stopnja 3 – maksimalni čas obnove nekaj dni.

Klasifikacija po stopnjah naj bi pomagala bolje razdeliti poslovne aplikacije in procese, Critchley (2015, str. 32) pa poudarja, da je razporeditev v stopnje poslovna odločitev, kjer velja rek »Kolikor plačaš, toliko dobiš«, kar seveda pomeni, da je poleg zahtev treba vedno upoštevati tudi strošek, ki ga posamezna zahteva prinese.

Barnes (2001, str. 91–92) navaja, da je ne glede na klasifikacijo zahtevane razpoložljivosti v BIA-analizi pomembno, da se analiza opravi za vse kritične sisteme. Tako je potem v naslednjih korakih izdelave načrta neprekinjenega poslovanja možno definirati scenarije obnove, da ustrezajo tako poslovnim procesom ter ne povzročajo finančnih stroškov, večjih od koristi obnovitve poslovanja.

## 1.3 Standardi neprekinjenega poslovanja

Pri upravljanju neprekinjenega poslovanja se podjetja lahko naslonijo na najboljše prakse, nekatere so zajete v standardih, druge v okvirjih (angl. *framework*) ali najboljših praksah (angl. *best practices*) kot na primer ISO, COBIT ali ITIL. Ti standardi definirajo veliko področij delovanja podjetja, med drugim pa opisujejo tudi najboljše prakse pri uvajanju in upravljanju neprekinjenega poslovanja v podjetju.

Ko govorimo o pomembnosti uporabe standardov ali najboljših praks, lahko omenim raziskovalno podjetje IDC (v Spremić, Bajgorić, & Turulja, 2013, str. 190), ki ugotavlja, da lahko podjetja, ki uporabijo eno od metodologij (na primer ITIL, COBIT) za ureditev področja informacijske tehnologije, znižajo število prekinitev za 85 odstotnih točk, kar pripomore k velikemu zmanjšanju dnevnih prekinitev podatkovnih procesov ter procesov, ki zagotavljajo nemoteno delovanje IT-podpore, ter s tem nižjim stroškom operativnega izvajanja.

Svata (2013, str. 28) kot primer standardov, okvirjev in dobrih praks navaja naslednje primere:

- Standardi:
  - ISO 22301, ISO 22313, ki opredeljujeta standard in dobro prakso pri uvajanju in upravljanju neprekinjenega poslovanja v podjetju,
  - ISO 27031, ISO 27001 in ISO 27002, ki se ukvarjajo predvsem z varnostjo poslovanja,
  - PAS 200 opredeljuje krizno vodenje,
  - PD 25666 govori o testiranju načrtov neprekinjenega poslovanja,
  - PD 25111, ki navaja smernice o upravljanju človeških virov v povezavi z neprekinjenim poslovanjem.
- Okvirji:
  - ISACA – COBIT 4.1, COBIT 5, Continuity Management Audit/Assurance Program,
  - BCMM (Business Continuity Maturity Model),
  - ITIL.
- Dobre prakse:
  - navodila in dobre prakse inštituta BCI,
  - navodila in dobre prakse inštituta DRII,
  - ENISA (European Network and Information Security Agency).

V nadaljevanju bom na kratko opisal nekatere od zgoraj naštetih standardov in dobrih praks, usmeril se bom predvsem v pomembnejše standarde, ki so uporabljeni v bančništvu na področju Slovenije in Evropske unije. Med te lahko štejemo ISO (angl. *The international organization for standardization,* v nadaljevanju ISO) standarda ISO 22301 in ISO 22313 kot standard pri uvajanju in upravljanju neprekinjenega poslovanja ter okvir COBIT, ter zbirko dobrih praks upravljanja z IT-storitvami – ITIL.

### 1.3.1 ISO 22301 in ISO 22313

Organizacija za standardizacijo ISO je mednarodno združenje organizacij. Ukvarja se z izdelavo mednarodnih standardov za vsa področja razen za elektrotehniko, kjer sodeluje z IEC (angl. *International Electrotechnical Commision*) (British Standards Institution, 2012, str. 4).

Standard, ki se ukvarja z neprekinjenim poslovanjem podjetja, je ISO 22301 (angl. *Societal security - Business continuity management systems - Requirements*), ki predstavlja zahteve za upravljanje neprekinjenega poslovanja v podjetju, dodatno standard ISO 22313 (angl. *Societal security - Business continuity management systems - Guidance*) pa predstavlja dobre prakse pri upravljanju neprekinjenega poslovanja.

ISO 22301 definira sistem upravljanja neprekinjenega poslovanja kot vzpostavljen proces vodenja, ki zagotavlja okvir za (ISACA, 2011, str. 6):

- razumevanje potreb podjetja ter nujnost vpeljave politik neprekinjenega poslovanja,
- prepoznavanje mogočih groženj in tveganj ter vpeljavo kontrol za njihovo upravljanje,
- merjenje in nadzor ustreznosti politik neprekinjenega poslovanja,
- model nenehnega izboljševanja na osnovi objektivnih meritev, ki je opisan s PDCA-modelom (angl. *Plan-Do-Chek-Act*).

ISO 22313 pa zagotavlja najboljše prakse, ki pomagajo podjetjem pri načrtovanju, vzpostavljanju, upravljanju, nadzoru in vzdrževanju načrtov neprekinjenega poslovanja, ki omogočajo podjetjem pripravo in odziv na možne nesreče v prihodnosti. Razlika med obema standardoma je v tem, da ISO 22301 predstavlja zahteve, ki jih mora podjetje izpolniti, če želi zadostiti standardu, medtem ko standard ISO 22313 predstavlja možne načine oz. najboljše prakse o načinu doseganja zahtev standarda ISO 22301.
ISO standard postavlja neke splošne okvire in je uporaben za vse vrste in velikosti podjetij, tako velika, srednja in mala podjetja v različnih panogah od industrije do neprofitnega sektorja. Namenjen je podjetjem, ki želijo (ISACA, 2011, str. 1):

- vzpostaviti, vzdrževati in izboljševati proces upravljanja neprekinjenega poslovanja,
- zagotavljati skladnost s politikami neprekinjenega poslovanja podjetja,
- certificirati upravljanje neprekinjenega poslovanja podjetja pri organizaciji ISO.

### 1.3.2 BSI

British Standards Institution (v nadaljevanju BSI) je nacionalni organ za standardizacijo v Združenem kraljestvu. Tako kot organizacija ISO, BSI definira standarde za vrsto produktov in storitev. Standard BSI, ki se ukvarja z upravljanjem neprekinjenega poslovanja, je bil BS 25999, ki pa je bil leta 2012 ukinjen ob nastanku standarda ISO 22301. Za upravljanje neprekinjenega poslovanja se BSI od ukinitve BS 25999 v celoti naslanja na standarda ISO 22301 in ISO 22313 (British Standards Institution, 2012).

### 1.3.3 COBIT

COBIT (angl. *Control Objectives for Information and Related Technology*, v nadaljevanju COBIT) je okvir, postavljen s strani organizacije ISACA in namenjen upravljanju ter vodenju informacijske tehnologije. Je podporno orodje, ki omogoča premostiti razkorak med kontrolnimi zahtevami, tehničnimi izzivi in poslovnimi tveganji (ISACA, 2012c, str. 32–33).

COBIT-okvir naj bi pomagal podjetjem ne glede na velikost dosegati optimalno vrednost svoje IT-podpore, z vzdrževanjem razmerja med realiziranimi prednostmi ter obvladovanjem tveganj in optimizacijo stroškov. Temelji na petih načelih: doseganje zahtev udeleženih, podpora celotnemu poslovanju, enoten okvir, celovit pristop, ločitev upravljanja od vodenja (ISACA, 2012a, str. 13).

COBIT-okvir je do sedaj imel pet glavnih izdaj (De Haes & Van Grembergen, 2015, str. 103–104):

- leta 1996 je bila objavljena prva izdaja,
- leta 1998, druga izdaja, dodatno področje – kontrola,
- leta 2000 je bila izdana tretja izdaja,
- leta 2005 je bila izdana četrta izdaja COBIT-okvirja, nato 2007 še 4.1 izdaja,
- leta 2012 pa je bila izdana zadnja verzija, verzija 5, ki združuje COBIT v4.1, ValIT 2.0 standard ter okvir RiskIT.

COBIT 5 je razdeljen na štiri domene in 37 procesov. Slika vseh procesov je predstavljena v Prilogi 2. S področjem neprekinjenega poslovanja se ukvarja domena izvajanja, storitve in podpore (angl. *Deliver, Service and Support*, v nadaljevanju DSS), proces DSS04 – upravljanje neprekinjenosti (angl. *Manage Continuity*) (ISACA, 2012b, str. 185).

Proces DSS04 je namenjen vzpostavitvi in vzdrževanju načrta neprekinjenega poslovanja, ki omogoča odziv na incidente in nesreče z namenom zagotavljanja nemotenega delovanja kritičnih poslovnih procesov in povezanih IT-sistemov.

Glavni cilji procesa so naslednji (ISACA, 2012b, str. 185):

- zagotoviti delovanje poslovno kritičnim procesom z zagotavljanjem minimalnega nivoja storitev,
- zagotoviti zadostno prožnost kritičnih storitev,
- zagotavljanje izvedljivosti planov s testiranjem,
- zagotoviti ažurnost načrtov,
- zagotoviti usposobljenost zaposlenih za izvajanje načrta.

COBIT 5 za vse procese opredeljuje matrike odgovornosti (angl. *RACI matrix*), kjer imajo posamezni profili zaposlenih različne odgovornosti. V matriki je tako definirano, kdo je odgovoren, zadolžen, posvetovan in seznanjen. Z definiranjem odgovornosti je v nadaljevanju lažje opredeliti ključne zaposlene in bolje definirane postopke ter procese.

### 1.3.4 ITIL

Zbirka napotkov za upravljanje in uvajanje storitev informacijske tehnologije (angl. *Information Technology Infrastructure Library*, v nadaljevanju ITIL) je formalna zbirka tehničnih publikacij, ki so pod strogo kontrolo, podobno kot standardi ISO. ITIL predstavlja podroben okvir najboljših praks v upravljanju storitev informacijske tehnologije (v nadaljevanju IT-storitev), z definicijo celovitih seznamov, nalog, postopkov in odgovornosti, ki so zasnovani tako, da se lahko prilagodijo katerikoli funkciji informacijske tehnologije. ITIL je postal de facto standard za opisovanje temeljnih procesov pri upravljanju IT-storitev; kot primer lahko navedem proces upravljanja sprememb. Okvir ITIL se usmerja na stalno merjenje in izboljševanje kakovosti storitev, ta fokus pa je bil pomemben dejavnik, ki je prispeval k večji uporabi okvirja ITIL po svetu (Moeller, 2013, str. 88).

ITIL 2011 vsebuje pet glavnih področij (Moeller, 2013, str. 89):

- strategijo storitev,
- načrtovanje storitev,
- uvajanje storitev,
- izvajanje storitev,
- nenehne izboljšave storitev.

Kot vidimo, se ITIL ukvarja s široko tematiko upravljanja informacijske tehnologije in storitev, vendar se v podpodročju ukvarja s tematiko neprekinjenega poslovanja. Le-ta je opisana v področju načrtovanja storitev, kjer je podpodročje upravljanja neprekinjenosti IT-storitev (angl. *IT Service Continuity Management* – v nadaljevanju ITSCM).

Proces ITSCM, kot ga opisuje ITIL, poteka v naslednjih korakih (Long, 2012, str. 45):

- vzpostavitev načrta neprekinjenega poslovanja,
- definiranje zahtev in strategij neprekinjenega poslovanja,
- uvedba načrta neprekinjenega poslovanja,
- izvajanje načrta neprekinjenega poslovanja,
- v primeru nesreče uporaba načrta neprekinjenega poslovanja.

# 2 NEPREKINJENO POSLOVANJE IN INFORMACIJSKI SISTEMI

Neprekinjeno poslovanje z vidika informacijskih sistemov razumemo kot zagotavljanje razpoložljivosti IT-sistemov in njihovo okrevanje po nesreči. Predvsem na začetku so bili postopki BCP ozko usmerjeni na informacijsko tehnologijo, s pojavom groženj in terorističnih napadov v tem tisočletju pa podjetja ugotavljajo, da je treba razširiti postopke na celotno podjetje (Naujoks, 2002, str. 99).

To poglavje bo tako usmerjeno na področje informacijske tehnologije ter njeno vlogo v celotnem načrtu neprekinjenega poslovanja podjetja.

## 2.1 Obnova delovanja sistemov po katastrofi

Obnova delovanja oziroma obnovitev po nesreči se ukvarja s pripravo in izvajanjem postopkov ob nesrečah. Ker celotno področje presega okvire te naloge, se bom v nadaljevanju osredotočil samo na obnovo delovanja IT-sistemov po nesreči.

Evropski sistem centralnih bank opredeljuje naslednje velikostne razrede nesreč oziroma katastrof (European System of Central Banks, 2007, str. 5):

- Katastrofa velikih razsežnosti – dogodek, ki prizadene široko urbano ali geografsko območje, kar povzroči obsežno prekinitev običajnih poslovnih operacij udeležencev v finančni panogi in drugih ekonomskih subjektov. Taka katastrofa lahko vodi v izpraznitev prizadetih območij znotraj določenega polmera ali v njihovo nedosegljivost.
- Regionalna katastrofa – regionalna katastrofa vpliva na geografsko področje velikosti regije.
- Lokalna katastrofa – njen škodljivi vpliv je omejen na geografsko območje s polmerom največ nekaj kilometrov.

Načine zagotavljanja neprekinjenega poslovanja z vidika sistemov IT lahko opišemo s sedmimi nivoji, ki jih je definirala organizacija SHARE (Warrick, 2003; Žnidar, 2006, str. 8; Brooks, Bedernjak, & Juran, 2002, str. 22):

- nivo 0 – rezervna lokacija ne obstaja, prav tako ne obstaja načrt neprekinjenega poslovanja,
- nivo 1 – varnostne kopije na rezervni lokaciji,
- nivo 2 – varnostne kopije na rezervni lokaciji, ki vsebuje tudi potrebno strojno opremo (»hot site«),
- nivo 3 – elektronski prenos podatkov (»electronic vaulting«),
- nivo 4 – trenutne varnostne kopije (»point-in-time copies«),
- nivo 5 – transakcijska integriteta,
- nivo 6 – zrcalno kopiranje brez izgube podatkov,
- nivo 7 – visoko avtomatizirana poslovno integrirana rešitev.

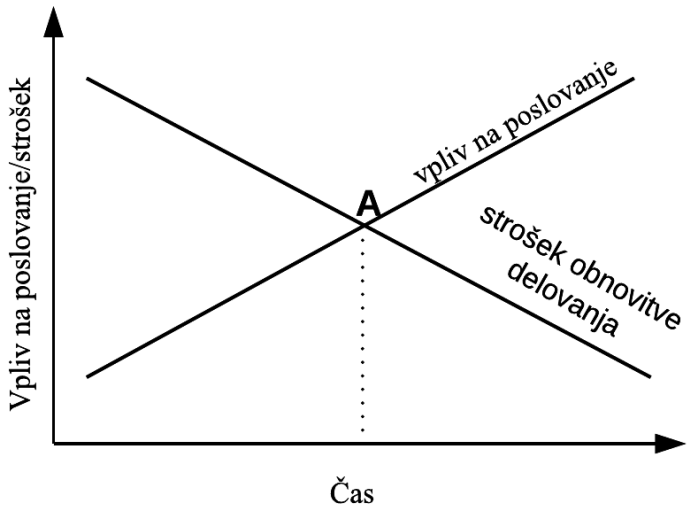
Višji kot je nivo zagotavljanja neprekinjenega poslovanja sistemov IT, višji so stroški za izvedbo rešitve. Prav tako višji nivoji zagotavljajo tudi hitrejšo obnovo delovanja v primeru predvidenih in nepredvidenih dejavnikov, ki vplivajo na razpoložljivost sistemov (Brooks et al., 2002, str. 30).

Sedmi nivo zagotavlja stalno razpoložljivost sistemov z avtomatsko obnovo delovanja, zahteva tudi največji napor pri implementaciji rešitve, poleg tega pa višje stroške kot rešitve na nižjih nivojih. Gartner ugotavlja, da morajo podjetja sama odločiti, katero strategijo bodo izbrala glede na svoje zahteve, saj je možnih strategij veliko, vsaka pa ni dobra za vsako podjetje (Scott, 2005, str. 2).

Tako Brooks et al. (2002, str. 30) svetujejo, da se izbere nivo zagotavljanja obnovitve po nesreči, ki zahteva manj finančnih vložkov, kot je možna višina izgube ob nesreči. Seveda je za to potrebna analiza, da se ovrednotijo procesi v podjetju, prav tako pa moramo poznati strošek rešitve povezane s informacijsko tehnologijo (v nadaljevanju IT-rešitev).

Slika 3 predstavlja povezavo med stroški strategije in vplivom na poslovanje. Izbrati je torej treba strategijo, pri kateri je hitrost obnove delovanja enaka strošku nedelovanja, torej kjer se sekata premici vpliva na poslovanje in stroška obnovitve delovanja, na Sliki 3 prikazano v točki A.

*Slika 3: Strošek strategije obnove proti strošku izpada delovanja*




*Vir: Prirejeno po J. C. Barnes, A guide to business continuity planning, 2001, str. 92, slika 4.2.*

Ko govorimo o zagotavljanju delovanja sistemov, je treba z vidika obnove po nesreči upoštevati naslednjo informacijsko infrastrukturo:

- računalniški center,
- mrežna infrastruktura,
- strežniška infrastruktura,
- aplikacije.

### 2.1.1 Računalniški center

Za izpeljavo kakršnega koli načrta neprekinjenega delovanja IT-sistemov podjetje potrebuje rezervno lokacijo računalniškega centra (v nadaljevanju RC), saj je samo z dodatnim RC, ki je zadosti oddaljen od primarnega, možno zagotoviti uspešno obnovo po nesreči, če je primarni RC uničen.

Pri vprašanju o oddaljenosti med RC v literaturi ni pravega odgovora. Niti v standardih, ki obravnavajo neprekinjeno poslovanje, ni definirane priporočene oddaljenosti.

Kosutic (2016) pravi, da ne glede na to, za kakšno razdaljo se podjetje odloči, je nujno, da se najprej izvede ocena tveganja, kjer se opredelijo posamezni dejavniki, ki vplivajo na odločitev o razdalji med RC. V nadaljevanju ločuje med dejavniki, ki zahtevajo daljšo oddaljenost med centroma, in dejavniki, ki krajšajo zahtevano razdaljo. Med prve štejemo: potrese, poplave, požare, industrijske objekte, nuklearne elektrarne ter vojašnice. Pomembno je, da sta centra na različnih električnih centralah. Dejavniki iz druge skupine, ki zahtevajo, da sta RC čim bližje, pa so: telekomunikacijske povezave ter čas poti zaposlenih do rezervnega centra. V zahteve po nizkem RTO je namreč treba všteti tudi čas, ki ga zaposleni, ki morajo vzpostaviti rezervni RC, potrebujejo, da se pripeljejo na lokacijo.

Glede na dokaj ohlapno definicijo razdalje med RC v literaturi in standardih je zanimiva raziskava Forresterja, v kateri je tudi graf, ki prikazuje odstotek podjetij in oddaljenost med njihovimi RC-ji. Iz grafa je možno sklepati, da ni enotnega mnenja, kako daleč naj bi bil rezervni RC, saj nobena od možnosti razdalj ni dobila večinskega deleža. Vse možnosti so sicer razdeljene med 10 in 20 odstotki. Pri interpretaciji ankete se je treba zavedati, da je bila raziskava narejena v ZDA, tako da so razdalje večje v primerjavi s podjetji v evropskem prostoru. Graf je prikazan v Prilogi 3 (Dines, 2011, str. 16).

Enako ugotavlja tudi Kosutic (2016), ki pravi, da v ZDA razdalja nekaj sto milj ne pomeni omembe vredne razdalje, medtem ko v Evropi lahko pri določenih državah taka razdalja že pomeni drugo državo. Torej, če se podjetje odloči za tako oddaljenost med svojimi RC-ji, mora računati na strošek take razdalje ter tudi kompatibilnost zakonodaje. Tako kot pri izbiri strategije neprekinjenega poslovanja podjetja je tudi pri izboru rezervne lokacije RC treba pretehtati koristi v primerjavi z zahtevanimi stroški vzpostavitve RC.

Barnes (2001, str. 93) navaja naslednje možne strategije RC:

- brez strategije,
- preselitev, ponovna vzpostavitev, obnova,
- hladna lokacija,
- vroča lokacija,
- vroča lokacija z elektronskim prenosom podatkov,
- aktivna rezervna lokacija (angl. *mirrored*).

Avtorji Laan (2011, str. 62), Gregg (2006) in Gregory (2008, str. 147) poleg navedenih vključujejo še toplo lokacijo, ki je vmesna rešitev med hladno in vročo lokacijo. Tabela 1 prikazuje primerjavo med različnimi tipi rezervnih lokacij in zmožnosti, ki jih posamezen tip nudi.

*Tabela 1: Primerjava vroče, tople in hladne rezervne lokacije*

| **Zmožnost** | **Vroča** | **Topla** | **Hladna** |
|---|---|---|---|
| Čas za aktivacijo | Od minute do ure | Največ nekaj dni | Več kot teden |
| Aplikacijski sistemi | Pripravljeni na uporabo | Na voljo, vendar niso pripravljeni na uporabo | Niso na voljo, treba jih je še vzpostaviti |
| Komunikacijska infrastruktura | Pripravljena na uporabo | Zmožna | Ne obstaja oz. obstaja v omejenem obsegu |
| Podatki aplikacij | Pripravljeni na uporabo | Ne čisto sinhronizirani, treba jih je še dopolniti iz logov oz. kopij | So na voljo v varnostnih kopijah, vendar jih je treba v celoti restavrirati |
| Strošek rezervne lokacije | Zelo visok | Srednje visok | Nizek |

*Vir: Prirejeno po P. H. Gregory, IT disaster recovery planning for dummies, 2008, str. 148, tabela 6-1.*

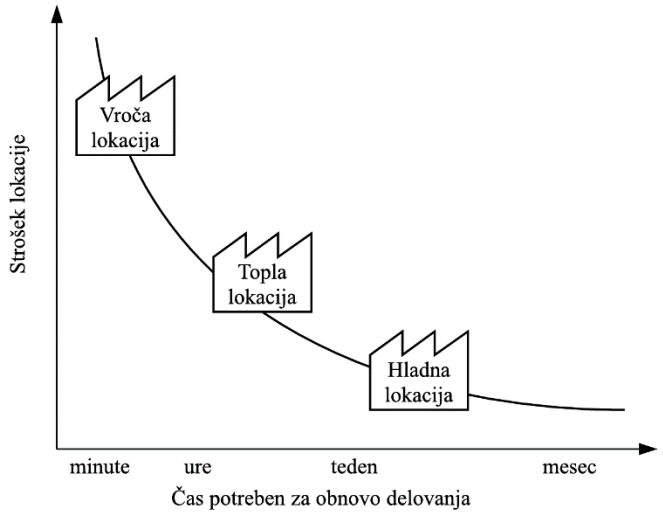
O hladni lokaciji govorimo, ko ima podjetje najete prostore, vendar je v njih malo oziroma skoraj nič informacijske infrastrukture. To pomeni, da mora podjetje v primeru večje nesreče najprej vzpostaviti vso strojno opremo, vzpostaviti programsko opremo (tako sistemsko kot aplikacije) ter mrežno infrastrukturo. Ocena je, da taka aktivnost ne more biti zaključena prej kot v tednu dni. Vseeno pa je hladna lokacija bolje kot nič. Če podjetje torej nima sredstev za vzdrževanje rezervne lokacije, je možnost hladne lokacije še vedno sprejemljiva (Laan, 2011, str. 62).

Vroča lokacija na drugi strani pomeni popolnoma konfigurirano in pripravljeno okolje, ki je pripravljeno prevzeti vlogo aktivne lokacije v zelo kratkem času. V primeru aktivne lokacije je ta čas tudi manj kot minuto. Pomeni pa taka postavitev večji strošek ter bolj kompleksno upravljanje, saj je treba zagotoviti, da sta obe strani med seboj sinhronizirani ves čas.

Topla lokacija pa je vmesna rešitev. Lokacija v taki postavitvi že ima postavljeno strojno opremo, vzpostavljeno mrežno infrastrukturo, nima pa programske opreme ali podatkov. Tako da je tudi čas za vzpostavitev lokacije ocenjen med obema drugima rešitvama. Predvidoma vzpostavitev tople lokacije traja vsaj nekaj dni do vzpostavitve delujočega okolja (Gregory, 2008, str. 148).

Razmerje med stroškom lokacije in časom, potrebnim za obnovitev delovanja, prikazuje Slika 4.

*Slika 4: Razmerje med stroškom lokacije in časom obnovitve računalniškega centra*




*Vir: Prirejeno po J. C. Barnes, A guide to business continuity planning, 2001, str. 93, slika 4.3.*

Oblačno računalništvo vedno bolj prihaja v ospredje tudi pri upravljanju neprekinjenega poslovanja. Tako mnogi svetujejo podjetjem, da razmislijo, ali res potrebujejo lastne vire za zagotavljanje neprekinjenega poslovanja, konkretno obnove IT-sistemov. Če podjetja

ugotovijo, da se jim stroškovno ali kadrovsko ne izplača upravljati še rezervnega računalniškega centra, imajo možnost le-to prepustiti zunanjim ponudnikom.

Critchley (2015, str. 329) navaja, da na trgu obstajajo naslednje možnosti:

- Najetje zunanjega ponudnika in izvajanje (angl. *outsource*) storitve spletnim ponudnikom, ki nudijo storitev upravljanja rezervne lokacije.
- Podjetjem, ki so specializirana v ponudbi rešitev za obnovo po nesreči, obseg storitve je različen, odvisen od dogovora.
- Ponudnikom oblačnih storitev (angl. *Cloud service provider*), ki ponujajo rešitve, imenovane DRaaS (angl. *Disaster recovery as a Service*, v nadaljevanju DRaaS), nekatere nudijo celo izdelavo varnostne kopije celotnega okolja, in ne samo kopije podatkov. Možnost DRaaS je predvsem primerna za podjetja, ki nimajo zadosti znanja ali pa nimajo zadosti zaposlenih, da bi sami uredili področje obnovitve po nesreči, zato je bolje, da najamejo celotno storitev.

### 2.1.2 Mrežna infrastruktura

Delovanja sistemov si v današnjem času skorajda ne moremo predstavljati brez mrežne infrastrukture. Schmidt (2006, str. 234) celo pravi, da je »najbolj pomembna komponenta infrastrukture«. Skoraj vse aplikacije namreč uporabljajo več kot en strežnik, tako da je povezljivost med njimi nujna. Sicer še vedno obstaja nekaj procesov, ki delujejo tudi brez mrežne povezljivosti, vendar jih je vedno manj. V banki so taki procesi paketne obdelave na centralnem računalniku, ki za svoje delovanje uporabljajo kvečjemu lokalno mrežo znotraj centralnega računalnika.

Pri načrtovanju mrežne infrastrukture za namene neprekinjenega poslovanja obstajajo trije glavni načini zagotovitve delovanja oziroma podvajanja (Schmidt, 2006, str. 265):

- Primarni RC in rezervni RC sta v istem lokalnem mrežnem delu (v nadaljevanju LAN-segmentu) – uporabno predvsem, ko je razdalja med RC majhna, na primer ko sta oba v istem mestu. Za ločevanje omrežja se uporablja VLAN (angl. *Virtual Local Area Network*). Slabost take postavitve je v tem, da imata obe strani enake številke, ki določajo računalnik v mreži (v nadaljevanju IP-naslov), tako da če se zaradi nekega razloga startajo storitve na obeh lokacijah, lahko pride do izpada delovanja omrežja, saj se podvojijo IP-naslovi, kar povzroči, da nobena od lokacij ne deluje. V bistvu lahko govorimo o kritični točki odpovedi.
- Primarni RC in rezervni RC sta v ločenih LAN-segmentih, ob aktivaciji rezervne lokacije se IP-naslovi prenesejo – vsaka lokacija ima svoje IP-naslove, za povezavo skrbijo usmerjevalniki. Ob preklopu storitev na rezervni RC se IP-naslovi prenesejo. Tako dobijo strežniki ob startu na rezervni lokaciji enake IP-naslove, kot so jih imeli na

primarni lokaciji. S tem načinom sicer rešimo kritično točko odpovedi, vendar je upravljanje težje, ker je potreben poseg ob izvedbi preklopa med RC.

- Primarni RC in rezervni RC sta v ločenih LAN-segmentih, IP-naslovi se ne prenašajo – gre za tradicionalni pristop, ob preklopu na rezervno lokacijo se startajo strežniki z novimi IP-naslovi, ki so različni od IP-naslovov, ki so jih strežniki imeli na primarni strani. Seveda tudi ta način ni brez slabosti, na primer, če aplikacije pričakujejo drugačne IP-naslove, bodo imele težave z delovanjem na rezervni lokaciji, prav tako je v tem primeru potreben ročna osvežitev storitve sistema domenskih imen (angl. *Domain Name System*, v nadaljevanju DNS-storitev), ki pa ni trenutna, tako da lahko povzroči dodaten čas prekinitve aplikacij.

Glede na kompleksnost mrežne infrastrukture ter dejstva, da je vitalnega pomena za delovanje IT-sistemov, je nujno, da se načrt mrežne infrastrukture temeljito analizira in izbere najbolj primerna rešitev, ki bo ustrezala zahtevam neprekinjenega poslovanja podjetja (Schmidt, 2006, str. 266).

### 2.1.3 Strežniška infrastruktura

Z vidika zagotavljanja zahtev neprekinjenega poslovanja oziroma BIA-analize moramo tudi strežniško infrastrukturo načrtovati tako, da bo dosegala zahteve. Pri strežniški infrastrukturi je treba upoštevati strojno in sistemsko programsko opremo ter zagotoviti tako postavitev, da ustreza zahtevam.

Razlogi za prekinitev v delovanju strežnika so lahko (Schmidt, 2006, str. 149):

- okvara strojne opreme – komponente strojne opreme niso bile podvojene ali pa je rezervna komponenta bila prav tako okvarjena,
- napaka v delovanju operacijskega sistema – možni razlogi so lahko napaka v naslavljanju spomina, težave v mrežni povezljivosti, težave s procesi ali pa napaka na datotečnem sistemu,
- napaka v programski opremi – aplikaciji – na primer uhajanje pomnilnika, mrtve zanke v komunikacijskih procesih ali pa programske napake, ki povzročijo interne napake aplikacije.

Strojna oprema je prva stvar, na katero se pomisli, ko se izdeluje načrt obnove po nesreči za IT-sisteme ali pa uvedbe visoke razpoložljivosti sistemov, saj je strojna oprema osnova za nadaljnje plasti programske opreme. Schmidt (2006, str. 99) navaja, da je strojna oprema tudi najbolj upravljana, saj je samo pri strojni opremi možno statistično izračunati verjetnost odpovedi posameznih komponent. Tako je tudi lažje načrtovati njihov vpliv na razpoložljivost oziroma možnost zagotavljanja neprekinjenega poslovanja.

Za načrtovanje postavitve operacijskih sistemov z namenom zagotavljanja neprekinjenega poslovanja oziroma visoke razpoložljivosti so v uporabi naslednje postavitve (Gregory, 2008, str. 167; Schmidt, 2006, str. 150):

- Rezervni strežniki (angl. *cold standby*)

Rezervni strežnik, ki je s stališča konfiguracije enak primarnemu strežniku, ki je postavljen v rezervnem RC in je v stanju pripravljenosti. V primeru izpada primarnega strežnika le-ta prevzame vlogo, vendar je potreben ročni poseg administratorja.

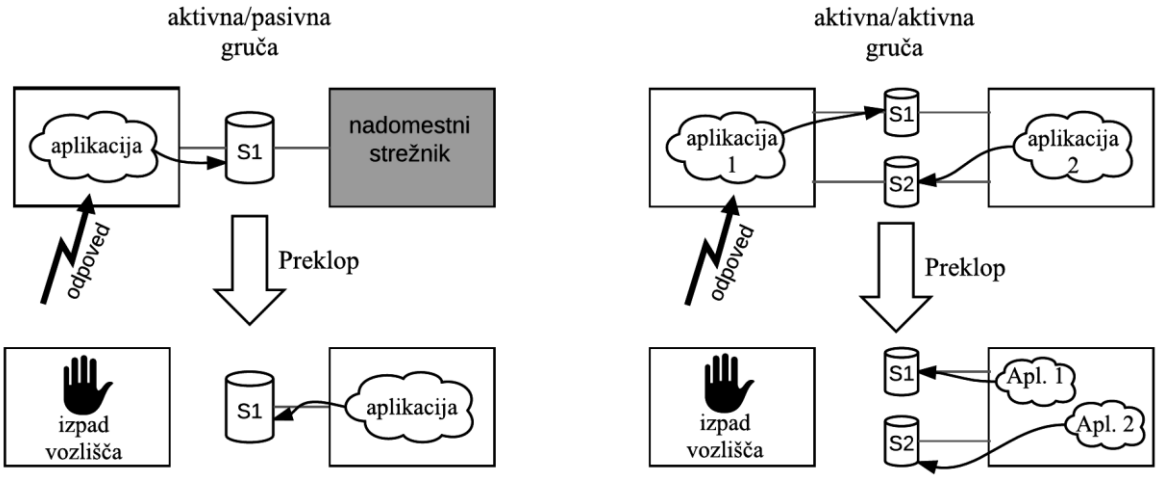
- Gruče strežnikov

Gruča strežnikov je povezava dveh ali več strežnikov, ki so konfigurirani tako, da za aplikacije ali pa končne uporabnike delujejo kot en strežnik, med seboj pa si delijo diskovna polja. Najosnovnejši način je aktivni/pasivni način, kjer sta vključena dva strežnika, kar pomeni, da je vedno aktiven en strežnik, drugi pa je nadomestni strežnik oziroma nadomestno vozlišče (angl. *hot standby*). Posamezni strežnik v gruči lahko poimenujemo tudi vozlišče (angl. *node*). Za nadzor in koordinacijo med vozlišči mora biti vzpostavljena povezava čez omrežje. V primeru izpada aktivnega vozlišča drugi avtomatsko prevzame funkcionalnost izpadlega.

Drugi način je aktivni/aktivni, kar pomeni, da vedno delujeta obe vozlišči v gruči. Postavitev je uporabna, če na gruči teče več aplikacij, ki se razdelijo med obe aktivni vozlišči. S tem se pridobi procesorska moč, saj se tako izkoristita obe vozlišči. V primeru izpada enega od vozlišč vse storitve prevzame drugo, pomembno pri načrtovanju pa je, da je vsako od vozlišč zadosti zmogljivo, da lahko zagotavlja delovanje vseh aplikacij.

Grafično predstavitev delovanja aktivna/pasivna in aktivna/aktivna gruča prikazuje Slika 5.

*Slika 5: Grafična predstavitev delovanja gruče strežnikov*




*Vir: Prirejeno po K. Schmidt, High availability and disaster recovery concepts, design, implementation, 2006, str. 156, slika 6.3.*

Izbira načina uporabe gruč je odvisna od zahtev visoke razpoložljivosti, kot tudi neprekinjenega poslovanja. Seveda pa morajo tudi aplikacije podpirati te različne postavitve, če želimo, da delujejo pravilno.

Na splošno so glavne zahteve, ki naj bi jih strežniška gruča zagotavljala: zagotoviti dostopnost podatkov, obnovitev storitev ter ponovna vzpostavitev delovanja v doglednem času.

- Strežniške farme (angl. *load balancing cluster*)

Strežniške farme so uporabne za aplikacije, ki delujejo neodvisno od predhodnega stanja. To pomeni, da ko enkrat aplikacijska povezava konča svoj proces, naslednja izvaja popolnoma ločen proces in med njima ni povezave oziroma soodvisnosti. Zato imajo vozlišča strežniške farme vsa enako konfiguracijo aplikacij oziroma storitev, ki jih nudijo. Vsa vozlišča v farmi so aktivna in si med seboj porazdeljujejo obremenitev. V večini primerov se pred strežniške farme namesti mrežni delilnik prometa, ki razvršča promet med vozlišči farme (Schmidt, 2006, str. 177).

Delilnik prometa lahko razvršča promet med vozlišči na več načinov (Schmidt, 2006, str. 182):

- naključno izbrano vozlišče,
- round robin – vozlišče izbere zaporedoma, začne s prvim in nadaljuje do zadnjega in ponovno od začetka, kar omogoča enakomerno razdelitev prometa med vozlišči,
- utežen round-robin – posameznemu vozlišču se doda še ocena zmogljivosti, tako delilnik prometa bolje razdeljuje promet med vozlišči, saj pozna zmogljivost posameznega vozlišča,
- metoda hitrejšega – promet se preusmeri k najhitrejšemu vozlišču, preverja se na primer ping odziv,
- najmanj povezav – za vsako vozlišče se beleži število povezav, delilnik prometa preusmerja povezave na vozlišča z najmanj povezavami,
- uteženo najmanj povezav – enako kot predhodna metoda, s tem, da se definira zmogljivost za vsako vozlišče, delilnik prometa upošteva tako zmogljivost kot število povezav,
- prilagojeno usmerjanje – obremenjenost vozlišča, število aktivnih povezav in čas odzivanja vozlišča se upoštevajo pri kalkulaciji delitve prometa.

Vse zgoraj naštete možnosti izvedbe gruč predpostavljajo, da so strežniki lahko na isti lokaciji znotraj istega RC. Za zagotavljanje neprekinjenega poslovanja je v poglavju o računalniškem centru napisano, da je nujno, da ima podjetje tudi rezervno lokacijo, ki je od primarne lokacije ločena, če želimo zagotoviti obnovitev po nesreči. Postavitev infrastrukture informacijske tehnologije (v nadaljevanju IT-infrastrukture) mora tako

upoštevati tudi geografsko komponento, saj morajo biti gruče povezane med seboj pri daljših razdaljah, kot je to znotraj RC.

V takih primerih govorimo o geografsko porazdeljenih gručah. Vse lastnosti gruč so enake, s to razliko, da so vozlišča med seboj ločena tudi geografsko. Torej, da je pri aktivni/pasivni gruči aktivno vozlišče v primarnem RC in pasivno vozlišče v rezervnem RC. Zaradi razdalje med RC lahko prihaja do zakasnitev med posameznimi vozlišči, tako da je možnost izgube podatkov v izrednih okoliščinah. Poudariti je treba, da morajo vozlišča biti čim bolj enaka, tako po konfiguraciji kot tudi zmogljivostih (Watters, 2014, str. 73).

- Konsolidacija in virtualizacija strežnikov

Novejši strežniki imajo zelo zmogljive procesorje, tako da manjše aplikacije ne morejo izkoristiti te moči v celoti. Posledica je, da so strežniki nizko izkoriščeni. Zato je smiselno zmogljive fizične strežnike razkosati na posamezne navidezne ali virtualne strežnike, ki si med seboj delijo fizične procesorje. Tako lahko na enem fizičnem strežniku gostimo več virtualnih strežnikov, vsak izmed njih pa izvaja svojo aplikacijo. Tako postavljenemu strežniku pravimo tudi konsolidiran strežnik.

Konsolidacija strežnikov ni pomembna samo zaradi povečanja izkoriščenosti posameznega fizičnega strežnika, ampak je tudi stroškovno učinkovita metoda, saj zmanjša strošek investicije v strojno opremo. Ker je le-te manj, je tako tudi strošek vzdrževanja nižji, kar privede do nižjega TCO (angl. *total cost of ownership*).

Za izvedbo virtualizacije je treba na fizični strežnik namestiti upravljavca virtualnih strežnikov, tako imenovanega hipernadzornika (angl. *hypervisor*). Njegova naloga je, da upravlja povezave med fizično strojno opremo ali gostiteljem (angl. *host*) in posameznim virtualnim strežnikom (Portnoy, 2012, str. 19).

Poznamo več vrst hipernadzornikov (Critchley, 2015, str. 301; Portnoy, 2012, str. 21):

- Polna virtualizacija – pri polni virtualizaciji ločimo med dvema tipoma hipernadzornikov. Tip 1 hipernadzornika teče neposredno na fizičnem strežniku brez vmesnega operacijskega sistema. Tip 2 hipernadzornika pa se izvaja na operacijskem sistemu, ki je nameščen na fizičnem strežniku. Ker je pri tipu 1 hipernadzornika manj plasti med virtualnim strežnikom in fizičnim strežnikom, so tip 1 hipernadzorniki bolj učinkoviti.
- Paravirtualizacija – podobna polni virtualizaciji tipa 1, s to razliko, da je hipervizor okleščen. Prav tako na virtualnih strežnikih ne teče poln operacijski sistem, ampak samo prilagojeno jedro operacijskega sistema. Ker virtualni strežnik uporablja enak operacijski sistem kot gostitelj, je taka virtualizacija učinkovitejša, saj ni potrebe po emulaciji na virtualnem nivoju.

V zadnjih letih se v ospredje prebija nova oblika virtualizacije – virtualizacija na nivoju zabojnikov (angl. *container-based virtualization*). Razlika s hipernadzorniki je v tem, da zabojniki, namesto da bi tekli v svojem virtualnem strežniku, tečejo v uporabniškem prostoru jedra operacijskega sistema. Kot taka omogoča več med seboj ločenih instanc, ki lahko tečejo vzporedno na istem fizičnem strežniku. Med slabosti takega tipa virtualizacije predvsem štejemo nižjo varnost kot pri polni virtualizaciji ter manjši fleksibilnosti, saj lahko zabojnik uporablja samo isti operacijski sistem kot gostitelj. Na primer, če je na fizičnem strežniku nameščen operacijski sistem Linux, se lahko v zabojniku izvaja samo ena od distribucij Linuxa, ne pa na primer Windows (Turnbull, 2014, str. 6; Scheepers, 2014, str. 2).

Da podjetja vedno več uporabljajo virtualizacijo svoje strežniške infrastrukture, ugotavlja tudi raziskava podjetja Forrester (2013, str. 3), kjer so vprašana podjetja potrdila pomembnost virtualizacije v povezavi z neprekinjenim poslovanjem oziroma obnovo po nesreči. Tako je več kot polovica podjetij odgovorila, da je vpliv virtualizacije zelo pomemben, medtem ko samo dva odstotka vprašanih podjetij ne vidi povezave med virtualizacijo infrastrukture in omogočanjem obnove po nesreči.

### 2.1.4 Aplikacije

Za večino podjetij je uspešen načrt neprekinjenega poslovanja, če so na voljo njihove storitve. Torej aplikacije, za katere je treba načrtovati celotno infrastrukturo z namenom, da dosegajo zahteve neprekinjenega poslovanja oziroma da je možna obnova delovanja po nesreči v zahtevanem časovnem roku.

Obnova po nesreči z vidika aplikacij je odvisna od arhitekture aplikacije. Poznamo dva načina delovanja aplikacij:

- Aplikacije z možnostjo hranjenja podatkov (angl. *stateful application*) – je arhitektura aplikacije, ki hrani podatke o preteklem delovanju, predhodne transakcije si aplikacija shrani, le-te pa lahko vplivajo na trenutno transakcijo. Primer take aplikacije bi lahko bila spletna banka, katere aplikacija si mora shraniti stanje med plačilom, da bi v primeru odpovedi lahko vzpostavila enako stanje na rezervnem vozlišču.
- Aplikacije brez hranjenja podatkov (angl. *stateless application*) – arhitektura tega tipa aplikacije ne hrani podatkov o preteklosti, deluje samo s podatki, ki jih ima v seji. Ko se seja zaključi, se ponovno vzpostavi začetno stanje in nova transakcija nima nobene povezave s prejšnjo.

Schmidt (Schmidt, 2006, str. 217) navaja, da je treba izbor načina obnove delovanja strežnikov po nesreči načrtovati v odvisnosti od delovanja aplikacije oziroma njene arhitekture. Tako predlaga uporabo strežniške farme za aplikacije, ki jim ni treba shranjevati stanj sej ter ne hranijo podatkov. Treba pa je imeti vzpostavljen proces upravljanja verzij aplikacij (angl. *release management*), da imamo vedno enako verzijo na vseh vozliščih

farme. Nasprotno je treba za aplikacije, ki hranijo podatke in seje, uporabiti strežniško gručo, saj mora biti sklop usklajen z aplikacijo, da se ne izgubijo podatki.

Erder in Pureur (2016, str. 128) svetujeta uporabo arhitekture aplikacij brez hranjenja podatkov, saj je ta tip aplikacije bolj skalabilen, torej ima boljšo zmožnost prilagajanja zmogljivosti zahtevam. Skalabilnost ima velik pomen predvsem pri spletnih aplikacijah, saj se tam najpogosteje pojavljajo obdobja povečanega prometa, to pa aplikacije brez hranjenja podatkov lažje prestanejo, ker je možno v farmo strežnikov brez težav vključiti nova vozlišča in s tem povečati zmožnost celotne farme.

## 2.2 Visoka razpoložljivost sistemov

Pojem razpoložljivost lahko opredelimo kot nekaj, kar je na voljo za uporabo. Torej bi na primeru informacijske tehnologije lahko rekli, da je le-ta razpoložljiva, ko nemoteno deluje.

Marcus in Stern (2003, str. 34) definirata visoko razpoložljivost (angl. *high availability*, v nadaljevanju HA) kot stopnjo razpoložljivosti, za katero se pričakuje, da dosega ali presega poslovne zahteve, za katere je bil računalniški sistem postavljen.

Schmidt (2006, str. 22) definira visoko razpoložljivost kot značilnost sistema, da se lahko v kratkem času zaščiti ali pa obnovi ob manjših težavah z uporabo avtomatskih metod. Prav tako ni pomembno, ali je razlog znotraj sistema, v okolju ali pa je vzrok človeški dejavnik.

Razpoložljivost merimo kot odstotek časa, ko je bil sistem razpoložljiv v odvisnosti od celotnega časa. Razpoložljivost lahko matematično opišemo z naslednjo enačbo (1) (Schmidt, 2006, str. 24):

$$razpoložljivost = \frac{neprekinjeno\ delovanje}{čas\ izpada + neprekinjeno\ delovanje} \tag{1}$$

Tabela 2 predstavlja razpoložljivost sistema ali aplikacije, katere zahteva je, da deluje nemoteno, torej v režimu 24/7. Tako vidimo, da je za dosego 99,999 odstotka ali petih devetk lahko prekinitve na letnem nivoju samo 5,3 minute. Le-to pomeni, da je brez dobrega načrtovanja sistema in obvladovanja sprememb taka razpoložljivost težko dosegljiva.

*Tabela 2: Trajanje prekinitve glede na odstotek dosežene razpoložljivosti*

| Razpoložljivost v % | Mesečna prekinitev | Letna prekinitev |
|---|---|---|
| 99,0 | 7,3 ure | 3,7 dneva |
| 99,5 | 3,7 ure | 1,8 dneva |
| 99,9 | 43,8 minute | 8,8 ure |
| 99,99 | 4,4 minute | 52,6 minute |
| 99,999 | 26,3 sekunde | 5,3 minute |

*Vir: K. Schmidt, High availability and disaster recovery concepts, design, implementation, 2006, str. 30, tabela 2.3.*

Velikokrat se pojem obnovitve po nesreči meša s pojmom visoke razpoložljivosti. Med njima je razlika v tem, da je za obnovitev po nesreči zadosti vzpostaviti arhitekturo, kjer je rezervni strežnik ali pa rezervna storitev v mirovanju in niti ni potrebe, da ima enako zmogljivost kot strežnik ali storitev na primarni strani, saj se pričakuje, da bo rezervna lokacija imela aktivno vlogo samo do ponovne vzpostavitve primarne lokacije in sistemov (Acharya et al., 2016, str. 85).

Obstajajo tri temeljna načela sistemskega načrtovanja za doseganje visoke razpoložljivosti (Acharya et al., 2016, str. 86):

- odpraviti je treba vse kritične točke odpovedi (angl. *single point of failure*) z namenom, da odpoved ene komponente ne vpliva na delovanje celotnega sistema,
- zagotoviti je treba zanesljivo selitev na rezervne sisteme,
- odkrivanje izpadov, treba je zagotoviti vzdrževanje sistemov za preprečitev odpovedi v delovanju.

Na razpoložljivost vplivajo tako nenapovedane prekinitve kot napovedane. Pri tem je zanimivo, da vedno, ko govorimo o visoki razpoložljivosti, pomislimo na nenapovedane izpade in to, kako se jim izogniti. Zato je presenetljiv podatek študije, ki pravi, da je samo 13 odstotkov nenapovedanih prekinitev in 87 odstotkov napovedanih prekinitev. Sklepamo lahko, da je treba za doseganje visoke razpoložljivosti torej zagotoviti, da so načrtovane prekinitve izvedene čim hitreje in brez napak, priporočljivo je, da je vzpostavljen nadzor nad spremembami, saj se samo tako lahko izogne napakam, oziroma da se načrtovane prekinitve ne podaljšajo v nenapovedane (Critchley, 2015, str. 125).

Marcus in Stern (2003, str. 75–104) navajata 20 ključnih načel pri zagotavljanju visoke razpoložljivosti sistemov. Med ključna načela tako štejemo pristop KISS (akronim za angl. *Keep it simple, stupid*), nato princip »en problem – ena rešitev«, izkoriščanje zunanjih virov, ponovna uporaba konfiguracije itd. Vseh 20 ključnih načel je naštetih v Prilogi 4. Pomembno je, da se iz teh ključnih načel lahko vidi, da je možnih veliko možnosti, ki povečujejo visoko razpoložljivost, ki niso niti drage niti kompleksne. Na primer pisanje dokumentacije, ki je velikokrat opredeljeno kot nepotrebno, vendar je le z obsežno dokumentacijo možno dobro

upravljati sisteme. Velika verjetnost je namreč, da ljudje, ki so postavljali rešitev, ne bodo sodelovali ves čas, enako je z izkušnjami.

## 2.3 Sistemi za upravljanje baz podatkov

Sistem za upravljanje podatkovnih baz (v nadaljevanju SUBP) je programska oprema, ki omogoča dostop do podatkov aplikacijam v kontroliranem procesu. Z ločitvijo strukture podatkov od podatkov SUBP omogočajo lažje programiranje, saj težaven del upravljanja podatkov prevzema SUBP ter tako ni treba le-tega definirati v aplikacijah (Liu & Özsu, 2009, str. 714).

Glavne prednosti in slabosti SUBP so prikazane v Tabeli 3.

*Tabela 3: Prednosti in slabosti SUBP*

| Prednosti | Slabosti |
|---|---|
| Podatkovna neodvisnost | Prostor, ki ga zasedejo podatki |
| Konsistenca podatkov | Višji stroški |
| Varnost podatkov | Kompleksnost |
| Integriteta podatkov | Višje zahteve za strojno opremo |
| Ekonomija obsega | Velik vpliv ob nedelovanju |

*Vir: R. P. Anjard, The Basics of Database Management Systems (DBMS), 1994, str. 12, tabela 1.*

Kot navajata Liu & Özsu (2009, str. 715), je bilo v zgodovinskem razvoju razvitih več vrst SUBP:

- hierarhične SUBP,
- mrežne SUBP,
- relacijske SUBP,
- objektno orientirane SUBP.
- NoSQL SUBP.

Relacijske SUBP so prevladovale v zadnjih 50 letih, v zadnjem desetletju pa se pojavlja vedno več NoSQL SUBP, predvsem zaradi potrebe po shranjevanju vedno večjih količin podatkov, ki predstavljajo velik zalogaj za relacijske SUBP. Glavna značilnost teh sistemov je, da ne uporabljajo relacijskega modela ter ne podpirajo ACID transakcij (Kuznetsov & Poskonin, 2014, str. 323).

Gartner (2015) v svoji študiji navaja, da 91 odstotkov vprašanih podjetij uporablja relacijske SUBP, se pa število uporabnikov novih NoSQL SUBP povečuje, kar kaže tudi podatek, da je zahteva po podprtosti ACID pomembna samo še 69 odstotkom vprašanih.

V primeru spletnega portala banke je uporabljen IBM DB2 in temelji na relacijskem modelu, čeprav trenutne verzije že podpirajo tako XML shrambo dokumentov kot tudi poizvedovanje po njih in urejanje le-teh. Podpora pa je tudi že za JSON shrambo dokumentov. Tako lahko rečemo, da je DB2 ne več le relacijska, ampak že hibridna SUBP, saj poleg relacijskega modela že vključuje nekatere elemente objektnih SUBP, kot tudi novejših NoSQL SUBP (Cochrane & McKnight, 2013).

Po predvidevanjih Gartnerja (2015) bodo do leta 2017 vsi večji ponudniki SUBP vključili tako relacijski kot tudi model NoSQL v enotni SUBP ter tako pokrili potrebo po celotnem segmentu trga SUBP, saj se zahteve uporabnikov SUBP menjajo in relacijski model ni več edini sprejemljiv. Predvideva se celo, da se bo termin NoSQL, ki nakazuje razliko med relacijskim in novim modelom, prenehal uporabljati, saj bodo obstoječe SUBP podpirale NoSQL model, zato termin ne bo več razločujoč.

Gartner (2016) postavi v magični kvadrant določen produkt ali rešitev glede na uspešnost z vidika izvedljivosti in vizije za določeno časovno obdobje. Magični kvadrant je razdeljen na štiri kvadrate:

- vodje (angl. *Leaders*) – so uspešni pri izvajanju trenutne vizije ter dobro pripravljeni na prihodnost,
- vizionarji (angl. *Visionaries*) – razumejo in sprejemajo spremembe na trgu in imajo vizijo, manjka jim uspešnost izvedljivosti vizije,
- nišni ponudniki (angl. *Niche Players*) – usmerjeni na manjše, nišne segmente trga ali pa nimajo usmeritve in niso inovativnejši od drugih,
- izzivalci (angl. *Chalengers*) – uspešni v izvajanju trenutne vizije, vendar jim manjka razumevanje gibanja trga v prihodnje.

DB2 zaseda položaj vodilnih v magičnem kvadrantu, je pa v letu 2015 opazen zaostanek za glavnima konkurentoma, Oraclom in Microsoftom. Grafični prikaz magičnega kvadranta za leto 2015 za področje SUBP je prikazan v Prilogi 5.

## 2.4 Visoka razpoložljivost podatkovnih baz

SUBP so samo ena od storitev, ki se izvajajo na strežnikih. Tako kot strežnik ali pa končna aplikacija mora tudi SUBP biti načrtovan, da zagotavlja visoko razpoložljivost, saj v nasprotnem primeru predstavlja kritično točko odpovedi. V nadaljevanju sledijo nekatere možnosti zagotavljanja visoke razpoložljivosti podatkovnih baz. Nekatere so tesno povezane z operacijskim sistemom, na katerem je nameščena SUBP, druge metode zagotavljanja visoke razpoložljivosti pa so neodvisne od nameščenega operacijskega sistema.

Za zagotavljanje visoke razpoložljivosti se lahko enako kot pri strežnikih uporabi eden od načinov strežniških gruč. Zavedati se je treba, da sistemska programska oprema gruč ni

nujno popolno kompatibilna s SUBP. Zato lahko v skrajnih primerih pride tudi do izgube podatkov. Na primer podatkovna baza ima podatke shranjene na mrežnem disku, ki zaradi hitrosti shranjuje podatke še v svojem predpomnilniku, ob napaki v delovanju SUBP in preklopu med vozlišči se tako lahko zgodi, da so bili podatki zapisani samo v predpomnilniku mrežnega diska, niso pa bili dejansko zapisani. Posledično le-to pomeni izgubo podatkov (Schmidt, 2006, str. 199).

Načini zagotavljanja visoke razpoložljivosti na nivoju operacijskega sistema (Schmidt, 2006, str. 200–204):

- Varnostna kopija celotnega strežnika – za način varnostnega kopiranja celotnega strežnika težko rečemo, da gre za pravo visoko razpoložljivost, saj je v primeru odpovedi podatkovne baze ali celotnega strežnika treba ponovno vzpostaviti stanje iz varnostne kopije.
- Aktivna/pasivna strežniška gruča – pri tem načinu je podatkovna baza shranjena na diskih, ki so sinhronizirani med posameznimi vozlišči. Ko pride do odpovedi primarnega strežnika, se izvede preklop na drugo vozlišče, ki prevzame nalogo prej aktivnega, SUBP pa se starta z enakimi podatki iz povezanega mrežnega diska. Tudi v tem primeru gre bolj za zagotavljanje visoke razpoložljivosti samega strežnika kot pa SUBP. Ko pride do preklopa med vozlišči, mora SUBP ob ponovnem zagonu izvesti obnovitev iz baznih logov (angl. *crash recovery*), kar pa lahko traja kar nekaj časa, predvsem pri velikih bazah.
- Aktivna/aktivna strežniška gruča – prednost pred prejšnjim načinom je v tem, da je tukaj na vseh vozliščih SUBP startana in zapisuje podatke iz vseh vozlišč v deljeno diskovno polje (angl. *shared storage*). Ob padcu enega od vozlišč ni prekinitve na nivoju podatkovne baze, ker se samo transakcije vozlišča z napako razveljavijo, druga vozlišča pa delujejo nemoteno naprej.
- Strežniška gruča z več primarnimi podatkovnimi bazami (angl. *multi-master database cluster*) – enako kot pri aktivni/aktivni gruči imamo več vozlišč z aktivnimi SUBP na vsakem od vozlišč, ves bazni promet pa se replicira med SUBP na posameznem vozlišču (angl. *peer-to-peer replication*). Način je uporaben predvsem za podatkovne baze, ki nimajo veliko vstavljanja novih zapisov, saj se morajo le-ti replicirati med vozlišči, in če je zahteva po konsistentnosti med posameznimi SUBP, mora biti ta replikacija sinhrona, kar pa pomeni, da transakcija traja dlje, saj se mora vsakič zapisati na vse aktivne podatkovne baze.

Zgoraj opisane metode zagotavljanja visoke razpoložljivosti podatkovne baze so seveda tudi možne metode zagotavljanja neprekinjenega poslovanja oziroma z IT-vidika obnove po nesreči. Seveda pa mora konfiguracija takih gruč upoštevati, da so vozlišča razdeljena tudi geografsko, tako da se varuje tudi pred večjimi nesrečami, na primer potresom, ko pride do odpovedi celotne lokacije RC.

Na nivoju podatkovne baze pa poznamo tudi kar nekaj načinov zagotavljanja visoke razpoložljivosti. Pri tem se je treba zavedati, da so rešitve narejene specifično za posamezno SUBP, tako da rešitve, ki delujejo z Oracle SUBP, ne delujejo z DB2 ali obratno, je pa način izvedbe podoben. V nadaljevanju bodo opisane posamezne možnosti, ki naj bi bile skupne vsem relacijskim bazam, bolj podrobno pa bodo metode zagotavljanja visoke razpoložljivosti DB2 SUBP opisane v poglavju analize.

Visoko razpoložljivost na nivoju SUBP lahko zagotavljamo na naslednje načine (Critchley, 2015, str. 272–281; Schmidt, 2006, str. 306–318; Bartkowski, De Buitlear, Kalicki, & Loster, 2012, str. 6–12):

- Varnostna kopija podatkovne baze – najpočasnejša metoda, ob težavah je treba obnoviti podatke iz varnostne kopije, paziti pa je treba tudi, da so varnostne kopije varno spravljene, najbolje na več lokacijah.
- Zrcaljenje diskov – metoda, pri kateri istočasno zapisujemo podatke na dve različni diskovni polji. Za zagotavljanje visoke razpoložljivosti ter čim manjšega vpliva na hitrost delovanja sta večinoma obe diskovni polji na isti lokaciji, velikokrat celo v istem diskovnem sistemu. Če želimo zagotoviti obnovo po nesreči, moramo konfigurirati zrcaljenje diskov med različnima lokacijama. Pri tem se je treba zavedati, da na daljše razdalje prihaja do zakasnitve in s tem upočasnitve tudi aktivne lokacije. Zrcalni diski so med seboj povezani s temnim vlaknom (angl. *dark fiber*) ali podobnimi tehnologijami, ki minimizirajo vpliv razdalje na zakasnitve.
- Replikacija podatkov – pri replikaciji podatkov lahko uporabljamo različna orodja, ki znajo prenašati podatke med podatkovnimi bazami. Replikacija je lahko enosmerna – od aktivne podatkovne baze proti sekundarni, rezervni podatkovni bazi, ali pa dvosmerna, kjer sta obe podatkovni bazi aktivni, promet med njima pa se replicira.
- Pošiljanje baznih logov na sekundarno lokacijo (angl. *log shipping*) z namenom, da pošiljamo vse spremembe na sekundarno podatkovno bazo, ki je v posebnem statusu, ki omogoča uvažanje podatkov iz logov v podatkovno bazo (angl. *rollforward*). Možni sta dve metodi:

    - Pošiljanje arhivskih logov (angl. *archive log shipping*) – pri tej metodi se bazni logi pošiljajo v paketih, ko se aktivni log primarne podatkovne baze arhivira. Glede na velikost arhivskih logov je možna večja ali manjša izguba podatkov, saj ob izpadu primarnega strežnika sekundarna baza ne prejme zadnjih podatkov iz takrat še aktivnega loga.
    - Pošiljanje logov v realnem času (angl. *redo shipping*) – ponavadi je to dodatna zmožnost SUBP, ki zna brati aktivne bazne loge in pošilja spremembe takoj, ko se zgodijo na primarni bazi. Tudi tu je možno definirati, ali želimo sinhron ali asinhron proces, kot vedno je to odločitev med zagotavljanjem konsistentnosti podatkov in vplivom na delovanje aktivne podatkovne baze.

- Uporaba baznih funkcij za zagotavljanje visoke razpoložljivosti in obnove po nesreči, imena funkcionalnosti so različna glede na proizvode SUBP, tako na primer Oracle svojo rešitev imenuje Oracle DataGuard, IBM svojo rešitev poimenuje DB2 HADR (angl. *High Availability and Disaster Recovery,* v nadaljevanju HADR), Microsoft SQL Server pa AlwaysOn. Ne glede na poimenovanje pa vse uporabljajo v ozadju metodo pošiljanja baznih logov v realnem času ter arhitekturo brez deljenja virov (angl. *shared nothing architecture*).
- Stalna razpoložljivost – v zadnjem času so zahteve po visoki razpoložljivosti že zelo visoke, saj vedno več storitev mora delovati neprekinjeno. Če želimo izpolniti zahteve, mora biti rešitev izvedena tako, da pokrije čim več možnih kritičnih točk odpovedi. S tem namenom se postavljajo gruče podatkovnih baz, posamezna vozlišča si med seboj delijo diskovna polja ter so vsa aktivna. Taka konfiguracija podatkovnih baz se navzven kaže kot ena podatkovna baza, tako da za aplikacije ne pomeni spremembe v delovanju, so pa velike spremembe v ozadju, ki pa jih nadzoruje in upravlja funkcionalnost SUBP. Kot primer lahko navedem rešitve IBM in Oracla: IBM DB2 Purescale in Oracle RAC.

Načini zagotavljanja visoke razpoložljivosti se lahko med seboj prepletajo, tako da je za zagotavljanje čim višje razpoložljivosti priporočeno uporabiti mešanico načinov. Ena od možnosti bi lahko bila: na nivoju operacijskega sistema se vzpostavi aktivna/pasivna gruča, na nivoju SUBP pa se izbere način repliciranja podatkov med bazami na posameznem vozlišču. Tako bo ob preklopu med vozlišči baza že sinhronizirana in aktivna, kar zelo skrajša čas za obnovo podatkov iz baznih logov ob ponovnem zagonu.

## 2.5 Odpornost proti okvaram

Odpornost proti okvaram je zmožnost računalnika, da deluje nemoteno, tudi če se katera od komponent okvari. Zato strežnik, ki je načrtovan za večjo odpornost proti okvaram, vključuje uporabo podvojenih elementov strojne opreme. Nekateri modeli strežnikov (centralni računalnik IBM, računalnik Tandem, HP Nonstop) imajo podvojeno večino elementov strojne opreme z namenom, da rezervna komponenta prevzame delo ob odpovedi primarne, brez vpliva na delovanje aplikacij.

Za zagotavljanje HA pa ni nujno, da so strežniki visoko odporni proti okvaram, saj lahko z gručami strežnikov nadomestimo izpade posamezne strojne opreme. Torej visoka odpornost proti okvaram sicer zagotavlja večjo HA, ni pa edini način doseganja HA.

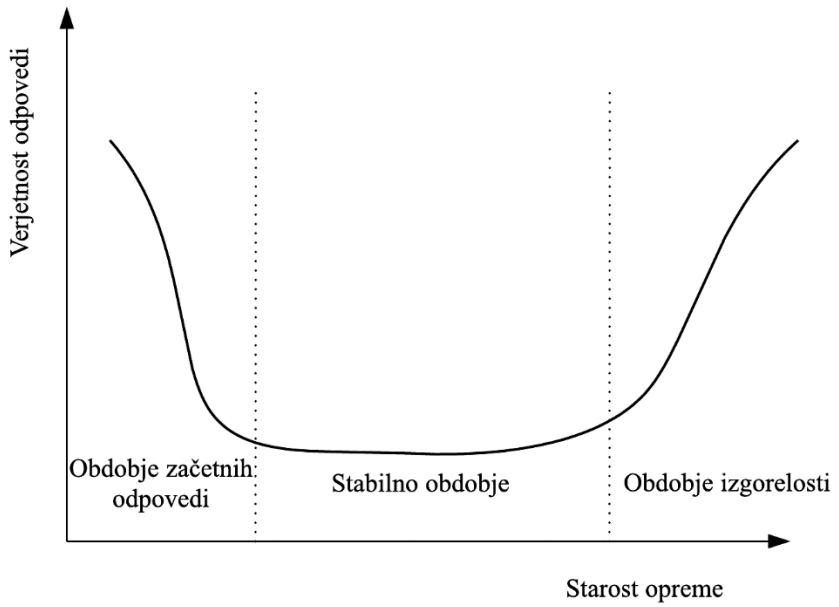
Četudi je strojna oprema visoko odporna proti okvaram, pa je treba upoštevati, da je vsaka komponenta sistema podvržena okvaram v obdobju uporabe. Delovanje strojne opreme v odvisnosti od časa uporabe je možno predvidevati. Za analizo delovanja se uporabljata kazalnika »povprečni čas med odpovedma« (angl. *mean time between failures*, v nadaljevanju MTBF) ter letna stopnja odpovedi (angl. *annual failure rate*, v nadaljevanju AFR).

Slaba lastnost kazalnikov MTBF in AFR je v tem, da predpostavljata, da so odpovedi enakomerno razdeljene in neodvisne od starosti opreme. Vendar ni tako, obstaja namreč tako imenovana krivulja kadi (angl. *bathtube curve*), ki ponazarja število odpovedi v odvisnosti od starosti strojne opreme (Critchley, 2015, str. 109).

Krivulja kadi, predstavljena na Sliki 6, deli okvare strojne opreme na tri različna obdobja (Schmidt, 2006, str. 375):

- obdobje začetnih odpovedi (angl. *burn-in period*) – odpovedi so posledica težav oz. nepravilnosti v izdelavi, oz. če gre za novo tehnologijo, tudi posledica zadostnega testiranja s strani proizvajalca,
- stabilno obdobje ali obdobje uporabe (angl. *stable period*) – v tem obdobju so odpovedi manjše, oprema odpoveduje glede na specificiran parameter MTBF,
- obdobje izgorelosti (angl. *burnout period*) – ocenjuje se, da je strojna oprema, ki deluje več kot pet let, že na koncu svoje življenjske dobe, saj po tem času število prekinitev oz. okvar začne ponovno naraščati.

*Slika 6: Tipična oblika krivulje kadi za strojno opremo*




*Vir: Prirejeno po K. Schmidt, High availability and disaster recovery concepts, design, implementation, 2006, str. 375, slika A.5.*

Bistveno je, da strojno opremo obnavljamo, saj se samo tako lahko zagotavlja, da bo stopnja nenapovedanih odpovedi nadzorovana in najnižja. Seveda pa je zanašanje samo na preverjeno opremo premalo. Oprema mora biti tudi sestavljena tako, da zagotavlja zahtevan nivo tolerance pri okvarah. Critchley (2015, str. 72) sicer pravi, da način podvajanja strojne opreme v samem strežniku kot način zagotavljanja visoke razpoložljivosti izgublja pomen, saj je strošek take strojne opreme previsok, in se je bolje zanašati na druge tipe podvojitev,

veliko vlogo pa igra tudi uporaba virtualizacije, tako na nivoju sistemskih slik kot tudi kontejnerizacija.

## 2.6 Stalna razpoložljivost

Vedno večje zahteve po dostopnosti storitev podjetij je privedlo do meje, ki jo lahko zagotavlja visoka razpoložljivost sistemov. Če je še pred nekaj leti zadostovalo, da je bila storitev na voljo večino časa, se zdaj zahteva stalna razpoložljivost, kar pomeni, da ne sme biti nobene načrtovane, kot tudi nenačrtovane prekinitve v delovanju. Termin stalne razpoložljivosti v literaturi navajajo tudi kot »Five 9s«, torej 99,999-odstotna razpoložljivost, kar pomeni zahtevo, da sistem ne sme biti nedostopen več kot 5 minut na leto (Alcott, 2010, str. 1).

Z vidika zagotavljanja neprekinjenega poslovanja podjetja je arhitekturna postavitev sistemov v način stalne razpoložljivosti najvišji možen nivo, ki zagotavlja, da bo storitev neprekinjeno delovala tudi v primeru nesreč, seveda le, če so povezani strežniki tudi geografsko ločeni med seboj.

# 3 NEPREKINJENO POSLOVANJE V BANČNIŠTVU

Kot omenjeno, je bančništvo ena izmed bolj reguliranih panog v gospodarstvu. Tako je za banke, poleg drugih predpisov, zahtevano tudi zagotavljanje neprekinjenega poslovanja. V naslednjem podpoglavju bodo predstavljene zakonodaja in smernice pri upravljanju neprekinjenega poslovanja v bankah, nato pa sledi pregled področja neprekinjenega poslovanja in glavne aktivnosti v letu 2015, pri čemer bo osredotočenost na pregledu letnih poročil petih največjih poslovnih bank, ki delujejo v Sloveniji. Pričakujem, da se bo iz poročil bank razbralo, da banke resno jemljejo priporočila oziroma zahteve regulatorjev, v primeru slovenskih bank sta to Evropska centralna banka (v nadaljevanju ECB) in Banka Slovenije (v nadaljevanju BS).

## 3.1 Zakonodaja in smernice neprekinjenega poslovanja

Smernice in zakonodaja v bančništvu zahtevajo od posameznih bank, da upravljajo operativna tveganja. Med le-ta štejemo tudi tveganje izpada delovanja sistema zaradi zunanjega dejavnika. Zato regulatorne institucije zahtevajo, da imajo banke opredeljene in analizirane operativna tveganja ter postopke njihovega upravljanja.

Basel II, ki ga je izdelal Baselski odbor, je zbirka priporočil, katerih namen je priprava mednarodnega standarda, na podlagi katerega nato lokalni bančni regulatorji (v primeru Slovenije je to Banka Slovenije) predpišejo smernice in zahteve za posamezne banke glede zagotovitve potrebnega kapitala oz. kapitalske ustreznosti. Cilji sporazuma so: pravilna alokacija kapitala v povezavi s stopnjo tveganja, ločitev operativnega tveganja od kreditnega tveganja in upravljanje obeh ter poskusiti približati regulatorni kapital dejanskemu

ekonomskemu tveganju bank z namenom zmanjšanja kapitalske arbitraže (Basel 2 — ENISA, 2016).

Basel II temelji na treh stebrih (Chernobai, Rachev, & Fabozzi, 2007, str. 38–49):

- 1. steber: minimalne kapitalske zahteve za tveganja – definira zahtevan minimalni nivo kapitalske ustreznosti,
- 2. steber: regulatorni nadzor – definira pravila in politike z namenom, da banke opredelijo zadostni nivo kapitala, ki ustreza njihovi tveganosti in strategiji,
- 3. steber: tržna disciplina – preglednost poslovanja bank, letno morajo banke poročati o izvedenih ukrepih in poslovanju. V naslednjem poglavju bodo tako predstavljena letna poročila večjih slovenskih bank za področje operativnega tveganja, upravljanja neprekinjenega poslovanja.

V Sloveniji je, ko govorimo o zakonodaji v bančništvu, pomemben Zakon o bančništvu (Ur. l. RS, št. 25/15, v nadaljevanju ZBan-2), ki v šestem poglavju govori o upravljanju operativnega tveganja, konkretno 162. člen pa govori o načrtu neprekinjenega poslovanja:

162. člen
(načrt neprekinjenega poslovanja)
(1) Banka mora izdelati načrt neprekinjenega poslovanja za primer kriznih razmer, ki določa ukrepe za zagotavljanje nemotenega poslovanja banke, da se ustrezno omejijo izgube banke
(2) zaradi teh motenj.

Nadalje Banka Slovenije s podzakonskim aktom v Sklepu o ureditvi notranjega upravljanja upravljalnemu organu in procesu ocenjevanja ustreznega notranjega kapitala za banke in hranilnice (Ur. l. RS, št. 73/15) v prilogi 3 sklepa opredeljuje dodatne zahteve glede upravljanja operativnega tveganja, ki jih morajo banke izpolnjevati v zvezi z načrti neprekinjenega poslovanja, kot so opisani v 162. členu Zban-2:

(1) Banka mora vzpostaviti načrte neprekinjenega poslovanja in krizne načrte za primer delovanja v pogojih hudih motenj poslovanja. Načrt neprekinjenega poslovanja vključuje postopke zagotavljanja neprekinjenega poslovanja pri pomembnih procesih in sistemih. Krizni načrt je sestavni del načrta neprekinjenega poslovanja ter določa tehnične in organizacijske ukrepe za ponovno vzpostavitev delovanja ter zmanjšanja posledic motenj poslovanja.
(2) Načrt neprekinjenega poslovanja mora ob nastopu hudih motenj poslovanja zagotoviti, da so pomožne zmogljivosti za nadaljevanje poslovnih dejavnosti čim prej na razpolago. Krizni načrt mora ob nastopu hudih motenj poslovanja v primernem časovnem obdobju zagotoviti ponovno vzpostavitev normalnega delovanja motenih dejavnosti banke.
(3) Načrti neprekinjenega poslovanja in krizni načrti morajo med drugim določati:

1. pristojnosti in odgovornosti glede začetnega odziva na dogodke, ki se odrazijo v večji motnji ali prekinitvi bistvenih sistemov in procesov;
2. pristojnosti in odgovornosti za izvedbo aktivnosti za obnovitev bistvenih sistemov in procesov;
3. časovne okvire za okrevanje bistvenih sistemov in procesov;
4. ključne zaposlene in postopke za zagotavljanje neprekinjenega delovanja bistvenih sistemov in procesov;
5. komunikacijske tokove, ki se uporabljajo v pogojih hudih motenj poslovanja.
(4) Banka mora zagotoviti, da so z načrti neprekinjenega poslovanja in kriznimi načrti seznanjeni odgovorni zaposleni.
(5) Banka mora zagotoviti redno, ter najmanj enkrat letno, testiranje načrtov neprekinjenega poslovanja in kriznih načrtov.

Tudi na nivoju plačilnih sistemov je zahteva Banke Slovenije po neprekinjenem poslovanju, obveznost zagotavljanja le-te je določena v Sklepu o plačilnih sistemih (Ur. l. RS, št. 73/09 in 5/11), ki določa, da morajo upravljavec in udeleženci plačilnih sistemov zagotoviti (1) ustrezne nadomestne rešitve in postopke za zagotavljanje neprekinjenega delovanja plačilnega sistema tudi v izrednih okoliščinah; (2) ažurno dokumentirane nadomestne rešitve za zagotavljanje neprekinjenega delovanja ter (3) njihovo redno testiranje in ustrezno usposobljenost osebja.

Na nivoju evropske unije ECB na področju neprekinjenega poslovanja definira naslednje smernice oz. uredbe:

- uredba Evropske centralne banke (EU) št. 795/2014 o pregledniških zahtevah za sistemsko pomembne plačilne sisteme (ECB/2014/28),
- smernice Evropske centralne banke (EU) 2015/930 o spremembah Smernice ECB/2012/27 o transevropskem sistemu bruto poravnave v realnem času (TARGET2),
- pričakovanja preglednikov glede neprekinjenosti poslovanja sistemsko pomembnih plačilnih sistemov.

## 3.2 Neprekinjeno poslovanje v slovenskih bankah

Ob pregledu letnih poročil večjih bank, ki delujejo v Sloveniji, lahko povzamem, da je področje neprekinjenega poslovanja urejeno, predvsem pa kaže na zavedanje vodstva bank, kako pomembno je to področje. Največji doprinos k urejenemu okolju pa po mojem mnenju predstavljajo zakonske zahteve regulatorjev po zagotovitvi upravljanja neprekinjenega poslovanja v banki oz. upravljanja operativnega tveganja.

Banke tako letno poročajo o operativnih tveganjih, med njih štejemo tudi upravljanje neprekinjenega poslovanja. Banke o področju neprekinjenega poslovanja ne poročajo samo

regulatorju, ampak v letnih poročilih tudi svojim lastnikom (delničarjem) in splošni javnosti, kar kaže na to, kako pomembno je področje zagotavljanja neprekinjenega poslovanja.

Pomembno je, da večina bank v svojem letnem poročilu poroča tudi o izvedenih testih neprekinjenega poslovanja, nekatere banke pa tudi o izvedenih izobraževanjih. Dejstvo je, da samo testiran načrt v realnem okolju lahko pokaže na pomanjkljivosti in možnosti za izboljšavo. Enako pomembno je tudi zavedanje zaposlenih o pomembnosti neprekinjenega delovanja banke, zato je treba pohvaliti zagnanost bank, ki namenjajo tolikšen pomen izobraževanju svojih zaposlenih s področja neprekinjenega poslovanja.

V največji slovenski banki NLB v svojem letnem poročilu navajajo (NLB d.d., 2016, str. 218): »Upravljanje neprekinjenega poslovanja se v NLB Skupini izvaja z namenom obvarovanja življenj, dobrin in ugleda. Za primere naravnih nesreč, IT-nesreč in nezaželenih vplivov okolja so pripravljeni načrti neprekinjenega poslovanja, ki omilijo posledice nesreč.«

V poročilu v nadaljevanju navajajo (NLB d.d., 2016, str. 218) »Izhodišče za posodobitev načrtov neprekinjenega poslovanja je vsakoletna redna analiza vpliva na poslovanje (BIA). Na njeni podlagi se preveri vzdržnost načrtov za poslovne stavbe in IT-načrtov. Najboljši kazalnik vzdržnosti načrtov neprekinjenega poslovanja so testiranja. V letu 2015 je bilo v NLB d.d. izvedenih 43 testiranj (33 internih in 10 z zunanjimi poslovnimi partnerji), pri katerih niso bila ugotovljena večja neskladja. Za učinkovitejše delovanja sistema upravljanja neprekinjenega poslovanja v NLB Skupini se izvajajo tudi izobraževanja in obiski posameznih bančnih članic. V letu 2015 je bilo tako v NLB d.d. izvedeno izobraževanje za člane kriznega štaba, poslovno operativne krizne skupine ter člane kriznih skupin poslovnih stavb. Ob IT-nesrečah/prekinitvah je banka uspešno uporabila IT-načrte in navodila za ročne postopke ter tako zagotovila poslovanje tudi v izrednih razmerah.«

V Novi KBM v svojem letnem poročilu (Nova KBM d.d., 2016, str. 37) pravijo, da imajo »vzpostavljen sistem upravljanja neprekinjenega poslovanja za primere izrednih dogodkov, tako ob izpadih informacijsko-komunikacijskih virov banke, naravnih nesrečah kot tudi ob drugih neželenih vplivih iz okolja. Namen sistema je učinkovito ukrepanje s ciljem celovito zaščititi storitve, kritične procese, opravila in sredstva banke. Izhodišče sistemu predstavlja regulatorni okvir, v banki pa sprejeta Politika upravljanja neprekinjenega poslovanja in pripadajoči akti. Področje korporativnega upravljanja v okviru sistema upravljanja izvaja letno analizo vpliva na poslovanje (BIA), oceno tveganj, vzdržuje in preizkuša načrte neprekinjenega poslovanja ter izvaja poročanje o izrednih dogodkih in zaznanih tveganjih odboru za operativna tveganja.«

Abanka je v sklopu poročanja o operativnih tveganjih zapisala, da »ima pripravljene načrte neprekinjenega poslovanja in okrevalne načrte za kritične aktivnosti, ki se redno posodabljajo in testirajo. V letu 2015 je Abanka posodobila ključne načrte neprekinjenega

poslovanja ter pripravila dodatne načrte in analize tveganj za nekatere storitve« (Abanka d.d., 2016, str. 67).

V Unicredit banki enako poudarjajo pomembnost načrtovanja neprekinjenega poslovanja (Unicredit Banka Slovenija d.d., 2016, str. 39): »Načrtovanje neprekinjenega poslovanja se je bistveno izboljšalo in več strokovnjakov za poslovanje je bilo del procesa izboljšav, kar je še dodatno pripomoglo k povečanju osveščenosti o tem, kako pomembna je ta tema.«

Nadalje navajajo (Unicredit Banka Slovenija d.d., 2016, str. 133), da je »z okrepljeno kadrovsko zasedbo na področju neprekinjenega poslovanja v letu 2015 posodobila načrt neprekinjenega poslovanja za primer kriznih razmer v skladu s politikami matične skupine UniCredit. Načrt neprekinjenega poslovanja določa ukrepe za zagotavljanje nemotenega poslovanja Banke, da se ustrezno omejijo izgube Banke zaradi teh motenj. Glavni namen načrta je opisati postopke, ki varujejo identificirane poslovno kritične procese pred naslednjimi učinki, ki so neposredno vezani na Banko in imajo morebitni vpliv tudi na matično skupino UniCredit.«

Pri načrtovanju neprekinjenega poslovanja upoštevajo šest kriznih scenarijev (Unicredit Banka Slovenija d.d., 2016, str. 133):

- nerazpoložljivost/nedostopnost poslovnih prostorov,
- nerazpoložljivost kritičnega osebja,
- nerazpoložljivost IT sistemov,
- nerazpoložljivost javnih dobrin/storitev,
- nerazpoložljivost kritične dokumentacije in
- nerazpoložljivost kritičnih dobaviteljev oz. ponudnikov storitev.

Kot izhaja iz objav proučevanih letnih poročil bank, se vse zavedajo pomembnosti področja neprekinjenega poslovanja. Prav tako vse banke opredeljujejo področje širše kot samo informacijska tehnologija, torej se zavedajo, da je za zagotavljanje neprekinjenega poslovanja treba upravljati in načrtovati tudi druga področja, to so: poslovne stavbe in delovni prostori, kritično osebje ter postopki in dokumentacija, potrebna za nemoteno poslovanje tudi v primeru nesreč.

Tudi glede načrtovanja neprekinjenega poslovanja vidimo, da banke letno obnavljajo analize vpliva na poslovanje in na podlagi rezultatov le-te obnavljajo same načrte neprekinjenega poslovanja.

## 3.3 Obnovitev sistemov IT po nesreči v banki

Kot je razvidno iz letnih poročil bank, vse resno jemljejo področje neprekinjenega poslovanja. Banka ima tako vzpostavljen načrt neprekinjenega poslovanja, ki obsega naslednje vrste načrtov (Klarič, 2016, str. 2):

- ročni postopki,
- poslovne stavbe,
- IT načrti,
- kadri,
- krizno komuniciranje.

Eden od načrtov je tudi načrt obnovitve IT-sistemov po nesreči. Klarič (2016, str. 8) navaja, da načrt IT-sistemov obsega dva sklopa dokumentov oz. načrtov: tehnična navodila in navodila za aktiviranje rezervnega dela IT-sistema.

Tehnična navodila vsebujejo sezname virov, seznam povezav, seznam aplikacij, opis podvojenosti sistemov, logično in fizično shemo, zmogljivosti strojne opreme in funkcionalnosti na rezervni lokaciji, zahtevan čas, ki je potreben za vzpostavitev delovanja na rezervni lokaciji, ter skrbnike, ki so odgovorni za delovanje sistemov (Klarič, 2016, str. 8).

Navodila za aktiviranje rezervnega dela IT-sistema pa vsebujejo splošen opis postopkov aktivacije, nadalje natančen opis postopkov za aktivacijo, razdeljenih po posameznih korakih, ter seznam kadrov, ki so zadolženi za posamezen korak (Klarič, 2016, str. 8).

Pomemben del zagotavljanja obnovitve sistema po nesreči je vsakoletno testiranje, ki je del letnega načrta BCM, ki ga potrdi uprava banke. Za vsak test se pripravijo scenariji, izvede se testiranje in pripravi poročilo o testiranju z oceno izvedbe in sklepi ter možnimi priporočili za izboljšanje (Klarič, 2016, str. 9–12).

# 4 ARHITEKTURA »SPLETNEGA PORTALA BANKE«

V nadaljevanju magistrskega dela se bom usmeril na konkreten problem zagotavljanja visoke razpoložljivosti podatkovne baze. Teza je, da z uvedbo visoke razpoložljivosti podatkovne baze omogočamo doseganje visoke razpoložljivosti celotnega sistema spletnega portala ter s tem tudi zadovoljimo zahtevam po neprekinjenem poslovanju ter obnovi po nesreči.

Kot sem že v prejšnjem poglavju opisal, je zahteva po nemotenem delovanju v banki regulatorna, kar pomeni, da je banka primorana zagotoviti rešitve, ki delujejo neprekinjeno tudi po nesrečah. Ni pa regulatorna zahteva edina zahteva, saj tudi notranji uporabniki,

predvsem pa komitenti banke, zahtevajo delujoče rešitve in pričakujejo, da bodo storitve dostopne, kadarkoli jih bodo potrebovali. Zato bom v nadaljevanju proučeval možne načine uvedbe visoke razpoložljivosti podatkovne baze kot načina zagotavljanja nemotenega delovanja.

Kot metodo dela bom uporabil pregled stanja pred uvedbo rešitve visoke razpoložljivosti podatkovne baze, kritično ocenil pomanjkljivosti, nato pa opisal in analiziral uvedeno rešitev. Pri tem si bom pomagal z literaturo s področja zagotavljanja visoke razpoložljivosti podatkovnih baz ter izkušnjami, ki sem jih pridobil z delom s podatkovnimi bazami v banki.

Pri definiranju kriterijev bom zahteve črpal iz izkušenj s sodelovanjem pri projektu uvedbe visoke razpoložljivosti podatkovne baze ter predvsem pregleda literature s področja podatkovnih baz in visoke razpoložljivosti. Tako bom definiral nujne in zaželene kriterije, ki naj bi jih rešitev zagotavljala, nato pa z metodo tehtanih vsot analiziral nekaj pogosto uporabljenih rešitev zagotavljanja visoke razpoložljivosti. Analize se bom lotil v treh korakih: definiranje kriterijev in možnih scenarijev, ocena posameznega scenarija glede na kriterij, izračun tehtane vsote posameznega scenarija. Po metodi tehtanih vsot je najboljša rešitev tista, ki dobi najvišjo vsoto ocen.

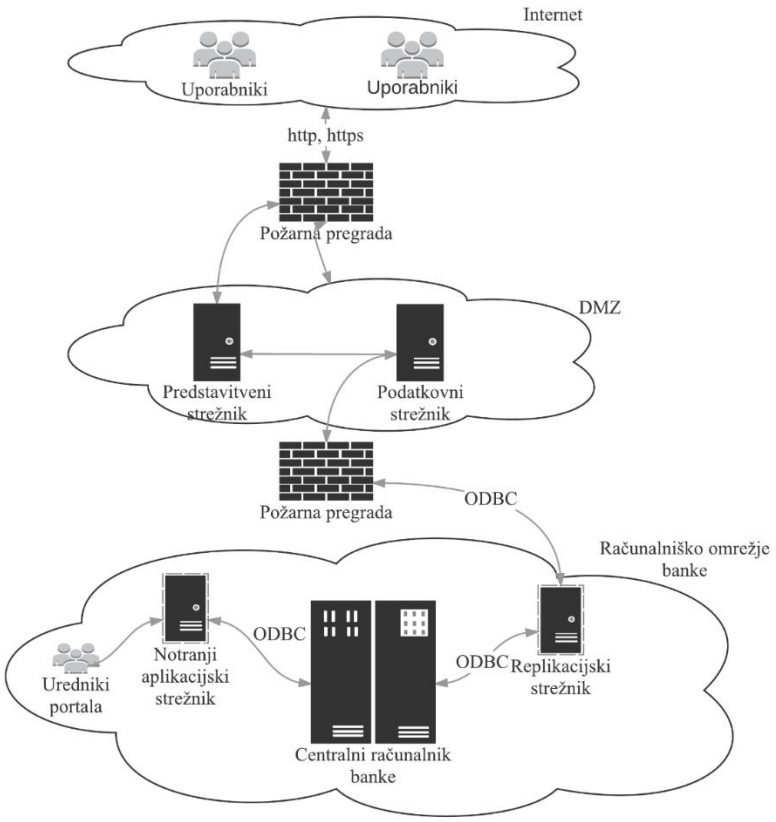
Nato bom izvedel še analizo prednosti, slabosti in priložnosti, nevarnosti uvedene rešitve, kot načina globlje analize problema, z namenom, da kritično opredelim njegove dobre in slabe lastnosti. Pri oceni bom prek podrobnejše analize tehnične realizacije ovrednotil prednosti in slabosti izbrane metode visoke razpoložljivosti ter navedel morebitne negativne in pozitivne posledice na delovanje podatkovne baze portala in neprekinjenega poslovanja. Predvsem slabe lastnosti bom v zadnjem poglavju poskušal odpraviti s predlaganimi izboljšavami uvedene rešitve.

Ker je sistem spletnega portala banke z vidika informacijske podpore kompleksen in sestavljen iz več med seboj povezanih delov, bom v nadaljevanju opisal posamezne sklope ter tudi proces nastajanja novih vsebin za prikaz na spletnem portalu banke.

Informacijska arhitektura spletnega portala je sestavljena iz naslednjih delov in predstavljena na Sliki 7:

- predstavitveni strežnik ozirom zunanji aplikacijski strežnik,
- podatkovni strežnik z zunanjo podatkovno bazo aplikacije,
- notranji aplikacijski strežnik,
- replikacijski strežnik,
- centralni bančni računalnik z zaledno podatkovno bazo aplikacije.

*Slika 7: Arhitektura spletnega portala banke*




*Vir: Prirejeno po U. Šavli, Razvoj informacijskega sistema EnKa za spremljanje sheme ugodnosti, 2008, str. 61, slika 21; R. Lavrič, Analiza replikacije na primeru portala NLB, 2004, str. 43, slika 14.*

## 4.1 Spletna aplikacija portala banke

Aplikacija spletnega portala je sestavljena iz dveh med seboj ločenih delov, zaledne aplikacije (angl. *back-end*) za urejanje spletnih strani in administracijo uporabnikov ter spletne čelne aplikacije (angl. *front-end*) za prikaz strani na internetu. Med enim in drugim delom obstaja logična povezava prek replikacije podatkov med SUBP IBM DB2 (v nadaljevanju DB2) podatkovnima bazama, replikacija podatkov se uporablja za povečanje varnosti podatkov v omrežju banke (Lavrič, 2004, str. 41–42).

### 4.1.1 Zaledna aplikacija

Dostop do zaledne aplikacije je dovoljen samo v omrežju banke in samo zaposlenim, ki imajo dovoljenje za urejanje posameznih vsebinskih sklopov portala. (Šavli, 2008, str. 32)

Za administracijo uporabnikov ter definiranje pooblastil in dostopa do aplikacije skrbi skupina za administracijo uporabnikov v banki. Aplikacija je namenjena izdelavi, urejanju in objavljanju vsebin, ki se nato prikazujejo v spletnem delu aplikacije.

Kreiranje nove vsebine oziroma novega dokumenta poteka v več korakih. Skrbnik kreira ali uredi obstoječi dokument v zaledni aplikaciji in ga shrani v zaledno podatkovno bazo aplikacije na centralnem bančnem računalniku. Preden se dokument dokončno objavi na spletnem portalu, ga pregleda in potrdi še drugi skrbnik. S tem se zagotovi, da je dokument ustrezen za objavo na internetu. Potrjen dokument se nato prek procesa replikacije podatkovne baze DB2 prepiše v podatkovno bazo na podatkovnem strežniku čelne aplikacije (Lavrič, 2004, str. 11).

### 4.1.2 Čelna spletna aplikacija

Spletni portal predstavlja vstopno točko do informacij, ki jih želi banka posredovati svojim komitentom. Na njem banka objavlja tekoče podatke, kot so obrestne mere, tečajnice valut, borzne podatke ter tudi vse informacije v zvezi z delovanjem banke kot tudi celotno ponudbo. Obiskovalci portala se lahko naročijo tudi na prejem tečajnih list in elektronskih novic prek elektronske pošte. Poleg tega so na voljo informativni izračuni za najem posojila, varčevanje, menjavo valut ter zemljevid poslovne mreže banke.

Čelna spletna aplikacija je nameščena na predstavitvenem strežniku, na katerem teče programska oprema spletnega strežnika, ki servira spletne strani uporabnikom. Aplikacija se preko odprte podatkovne povezljivosti (angl. *open database connectivity*, v nadaljevanju ODBC) povezuje na podatkovni strežnik, na katerem teče podatkovna baza DB2. Zaradi varnosti ima aplikacijski uporabnik čelne spletne aplikacije omogočeno samo branje podatkov iz baze (angl. *read-only*), za spremembe v podatkih skrbi zaledna aplikacija.

## 4.2 Spletni aplikacijski strežnik

Spletni aplikacijski strežnik aplikacije spletnega portala banke je postavljen na operacijskem sistemu Unix, za zagotavljanje spletnih strani portala pa se uporablja programska oprema spletnega strežnika. Spletni aplikacijski strežnik je namenski strežnik, ki je v uporabi za spletni portal banke. Sama postavitev in konfiguracija strežnika sta bili izvedeni z zavedanjem, da mora strežnik zagotavljati visoko stopnjo varnosti, saj je strežnik dostopen prek interneta, kar omogoča dostop vsakemu, ki ima dostop do interneta, s tem pa tudi tako imenovanim hekerjem oziroma skupinam, ki se ukvarjajo z vdiranjem v spletne strežnike. Zato je bilo okolje Unix na strežniku spletnega portala postavljeno z najmanjšo možno konfiguracijo, ki še omogoča nemoteno delovanje operacijskega sistema, prav tako pa se na strežniku redno izvajajo namestitve varnostnih popravkov (Lavrič, 2004, str. 54).

## 4.3 Spletni podatkovni strežnik

Spletni podatkovni strežnik je enako kot aplikacijski strežnik postavljen na operacijskem sistemu Unix s podobno minimalno konfiguracijo, da zagotavlja čim višji nivo varnosti pred zunanjimi vdori v strežnik. Na podatkovnem strežniku je nameščena SUBP DB2.

## 4.4 Replikacijski strežnik

Replikacijski strežnik je vmesni povezovalni člen med SUBP na spletnem podatkovnem strežniku in centralno podatkovno bazo na centralnem računalniku banke. Kot navaja Lavrič (2004, str. 39), je bila replikacija kot način prenosa podatkov med bazama izbrana kot najzanesljivejše sredstvo za prenos podatkov. Izbor je temeljil na strogih varnostnih zahtevah v banki ter predhodnih izkušnjah z uporabo replikacije.

Glede na to, da z replikacijo ustvarjamo podvojene podatke oziroma repliko podatkovne baze v dveh okoljih, bom za razumevanje procesa v nadaljevanju opisal načina repliciranja (dodajanja) podatkov v DB2. Kot bo opisano v nadaljevanju, je ponovna hladna replikacija namreč ena od možnosti zagotavljanja obnovitve podatkovne baze po nesreči oziroma tudi možnost zagotavljanja visoke razpoložljivosti.

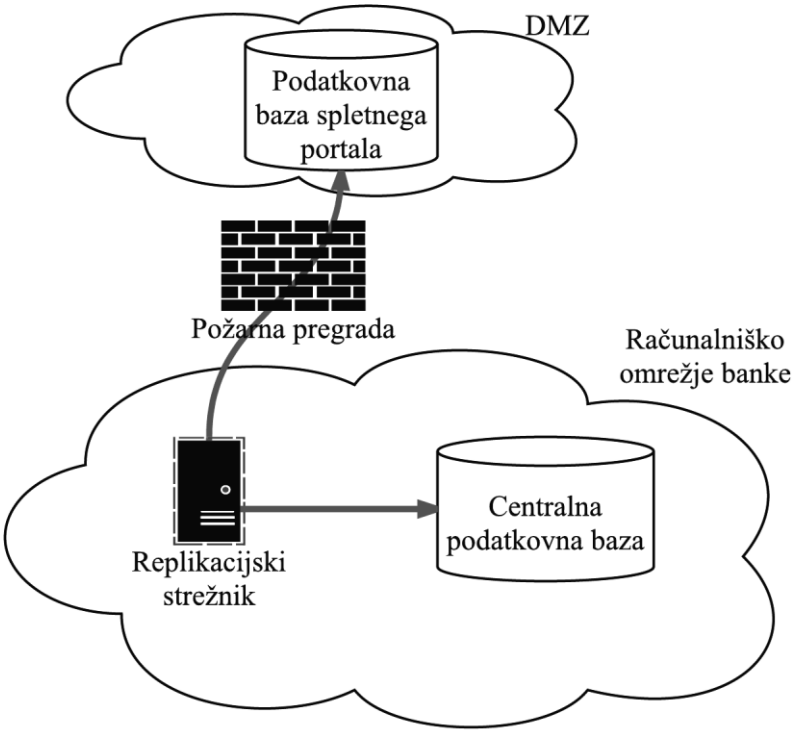
Replikacija podatkov poteka z orodjem, ki je del paketa DB2, in sicer SQL replikacija. Leta temelji na branju podatkovnega loga na izvorni podatkovni bazi, na izvornem strežniku mora biti aktiven proces zajemanja (angl. *capture program*), ki beleži vse spremembe v tabelah v vmesne tabele replikacije. Proces dodajanja (angl. *apply program*) pa prenaša podatke v ciljno podatkovno bazo (IBM, 2011).

Proces dodajanja vpliva na samo delovanje replikacije, saj v odvisnosti od njegove namestitve na izvorni, ciljni ali vmesni kontrolni strežnik vplivamo na izbiro metode prenosa podatkov med podatkovnima bazama. Lavrič (2004, str. 38) navaja, da proces dodajanja lahko podatke prenaša iz izvorne podatkovne baze v ciljno z naslednjima metodama:

- metoda potiskanja podatkov (angl. *push*) – proces dodajanja najprej prebere podatke iz izvorne baze, nato se prijavi na ciljno bazo in pošlje spremembe podatkov,
- metoda vlečenja podatkov (angl. *pull*) – proces dodajanja se izvaja na ciljnem strežniku, zato se mora najprej prijaviti na izvorno podatkovno bazo, prebrati podatke, nato pa se izvede lokalna prijava na bazo in izvedejo spremembe podatkov v bazi.

Proces replikacije podatkov spletnega portala banke poteka prek vmesnega replikacijskega strežnika, ki je v tej postavitvi kontrolni strežnik, na katerem je nameščen proces dodajanja, z namenom, da se na obe podatkovni bazi podatki potiskajo, kar je tudi varnostno sprejemljivo, saj se v tem primeru vsa komunikacija vedno prične na replikacijskem oz. kontrolnem strežniku in ni potrebne neposredne povezave na centralno podatkovno zbirko (Lavrič, 2004, str. 42). Tok podatkov med podatkovnima bazama je prikazan na Sliki 8.

*Slika 8: Proces replikacije podatkov spletnega portala banke*




*Vir: Prirejeno po R. Lavrič, Analiza replikacije na primeru portala NLB, 2004, str. 43, slika 14.*

## 4.5 Notranji aplikacijski strežnik

Tako notranji aplikacijski strežnik kot replikacijski strežnik sta virtualna strežnika, ki delujeta na hipernadzorniku Microsoft Hyper-V. Na aplikacijskem strežniku za spletni portal je nameščen operacijski sistem Windows server s programsko opremo spletnega strežnika. Namenjen je gostovanju zaledne aplikacije spletnega portala banke. Dostop do zaledne aplikacije je omogočen zaposlenim v banki, ki imajo pravice za dostop do storitve.

## 4.6 Centralni računalnik banke

Podatkovna zbirka zaledne aplikacije je na centralnem računalniku banke IBM, SUBP je tako kot na čelnem sistemu DB2. Na centralnem računalniku je nameščen operacijski sistem z/OS (Lavrič, 2004, str. 16–17).

Čeprav gre za centralni računalnik (angl. *mainframe, mainframe computer*), katerega razvoj se je pričel že pred več kot 50 leti, je razvoj platforme s strani dobavitelja IBM še vedno zelo aktiven, zadnja generacija centralnega računalnika z13 je prišla na trg v letu 2015. V banki je v uporabi prav zaradi prednosti, ki jih nudi platforma centralnega računalnika, to so: zanesljivost, razpoložljivost, varnost in stalna združljivost programske opreme. Ta tip platforme se je obdržal v banki kot osrednji sistem. Zaradi načina delovanja, je lahko

uporabljen ne samo kot podatkovni strežnik, ampak tudi aplikacijski strežnik za mnogo bančnih transakcijskih aplikacij, ter tudi za analitične aplikacije – podatkovno skladišče, poročevalske aplikacije, upravljanje odnosov s strankami, itd.

Banka tudi v svetu ni izjema, ki še uporablja to platformo, po trditvah IBM (2016) naj bi kar 92 bank od 100 najboljših bank v svetu uporabljalo centralni računalnik IBM.

## 4.7 Visoka razpoložljivost spletnega portala

Čeprav je za delovanje spletnega portala treba zagotoviti delovanje vseh med seboj povezanih komponent, ki so bile predstavljene v prejšnjih poglavjih, se bom v nadaljevanju naloge usmeril predvsem v analizo zagotavljanja visoke razpoložljivosti podatkovne baze. Za razumevanje pa bodo v tem podpoglavju podane informacije za druge dele sistema, predvsem zaradi razumevanja celotne umestitve rešitve v sistem zagotavljanja neprekinjenega poslovanja banke in načina izvedbe visoke razpoložljivosti posameznega dela, v tem primeru podatkovne baze.

Pri postavitvi spletnega aplikacijskega strežnika se je upoštevala zahteva po visoki razpoložljivosti aplikacije oziroma celotnega sistema spletnega portala, zato je spletni aplikacijski strežnik postavljen v načinu strežniške farme, kar je standardna praksa za spletne aplikacije, ki ne hranijo podatkov med sejami, le-to pa so v veliki večini spletne aplikacije.

Poleg same podvojitve strežnikov pa sta tudi fizično ločena na dve med seboj oddaljeni lokaciji z namenom zagotavljanja neprekinjenega poslovanja, katere zahteva je regulatorna. Strežniki v farmi so med seboj neodvisni, tako da težave v delovanju enega strežnika ne vplivajo na delovanje drugih. V primeru težav mrežni delilnik prometa le-tega preusmeri na delujoče strežnike, poudariti pa je treba, da v tem primeru pride do degradacije storitve, saj se v takem primeru celoten promet servisira z še delujočimi strežniki.

Strežniška farma vsebuje najmanj dve vozlišči, vendar se lahko število vozlišč v farmo dodaja, na primer, če se promet toliko poveča, da obstoječa spletna infrastruktura ne bi bila zmožna servisirati vseh zahtev. Z dodajanjem novih strežnikov se povečuje zmogljivost same spletne aplikacije, za distribucijo prometa pa skrbi mrežni delilnik prometa (Marcus & Stern, 2003, str. 352).

Podatkovni strežnik je prav tako fizični strežnik, nameščen v demilitarizirano področje (v nadaljevanju DMZ), na njem je nameščen operacijski sistem Unix ter konfiguriran z enakimi varnostnimi zahtevami kot spletni aplikacijski strežnik.

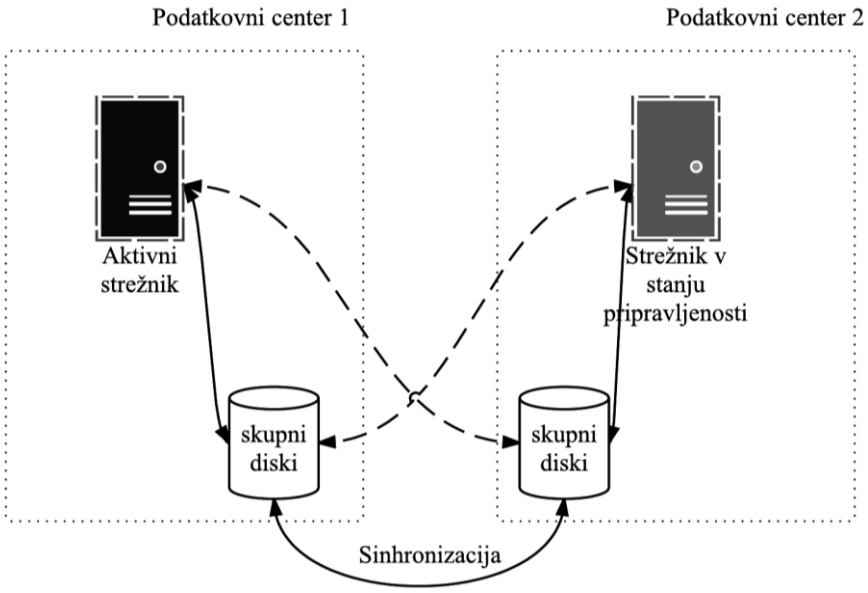
Drugi strežniki v konfiguraciji spletnega portala so tudi postavljeni tako, da omogočajo hitro obnovitev; ob nesrečah torej zagotavljajo neprekinjeno poslovanje v skladu z BIA-analizo. Razlika med spletnim aplikacijskim strežnikom in drugimi je v tem, da so druge postavitve

izvedene v načinu aktivne/pasivne gruče (angl. *active-passive cluster*) z uporabo virtualizacije Microsoft Hyper-V (Pogačar, 2010, str. 21).

Vsak virtualni strežnik ima svoj operacijski sistem ter je popolnoma ločen tako od gostitelja/hipernadzornika kot tudi od drugih virtualnih strežnikov. Tako da je edina razlika med fizičnim strežnikom in virtualnim v tem, da si virtualni strežnik deli sistemska sredstva z drugimi virtualnimi strežniki (Finn, 2013, str. 5).

Slika 9 predstavlja način izvedbe postavitve gruče tipa aktivna/pasivna, glavna značilnost je, da ima gruča na diskovnem nivoju zrcaljenje diskovne kapacitete, tako da se v primeru odpovedi ali gostiteljskega strežnika Hyper-V ali težav na virtualnem strežniku izvede avtomatski preklop gruče, v primeru banke tudi na sekundarno lokacijo podatkovnega centra, tako da je v primeru virtualnih strežnikov poskrbljeno tako za zahteve neprekinjenega poslovanja kot tudi za visoko razpoložljivost sistemov.

*Slika 9: Aktivna/pasivna gruča med dvema podatkovnima centroma*




*Vir: Prirejeno po M. Pogačar, Rešitev iz prakse: Uvedba Hyper-V in Microsoft System Center Virtual Machine Manager v NLB d.d., 2010, str. 21.*

Centralni računalnik banke je prav tako podvojen, z vidika zagotavljanja nemotenega poslovanja oziroma obnove po nesreči je prav tako vzpostavljena rezervna strojna oprema, ki naj bi bila enako zmogljiva kot na primarni lokaciji, z vidika zagotavljanja visoke razpoložljivosti pa je postavitev okrnjena, saj je strojna oprema centralnega računalnika na rezervni lokaciji v stanju pripravljenosti. Preklop na rezervni sistem se mora izvesti po predpisanih ročnih postopkih. Ker pa ročni postopki zahtevajo daljšo prekinitev, tukaj ne moremo govoriti o visoki razpoložljivosti sistema, postavitev pa ustreza zahtevam neprekinjenega poslovanja podjetja.

Arhitekturna postavitev centralnega računalnika z vidika zagotavljanja neprekinjenega poslovanja je različna glede na druge postavitve sistemov tudi zaradi posebnosti platforme, ki zagotavlja visoko toleranco pri okvarah, saj je večina strojne opreme podvojene in se avtomatično aktivira ob težavah (White et al., 2016, str. 88).

Kot navaja White et al. (2016, str. 71), naj bi bila prav visoka toleranca pri okvarah eden od večjih adutov centralnega računalnika.

Drugi pomemben dejavnik, ki vpliva na arhitekturno umestitev centralnega računalnika v informacijski sistem banke z vidika zagotavljanja neprekinjenega poslovanja, je stroškovni vidik. Licenčni pogoji za uporabo centralnega računalnika so specifični in zahtevajo velike investicije za nakup dodatnih licenc, zato se v večini primerov uporabe centralnega računalnika ne uporablja možnost visoke razpoložljivosti, ampak se zagotovi, da je postavitev primerna zahtevam BIA-analize aplikacij, ki uporabljajo to platformo, ter da je možno vzpostaviti rezervni sistem v zahtevanem času.

## 4.8 Analiza začetnega stanja visoke razpoložljivosti podatkovne baze

V analizi izhajam iz zahteve, ki so jo podali poslovni uporabniki oziroma uporabniški skrbniki aplikacije spletnega portala in je predstavljena z BIA-analizo. V banki se izvaja BIA-analiza na nivoju celotnega sistema oziroma aplikacije, in ne posameznega dela, v tem primeru samo podatkovne baze. Vseeno mora biti vsak del sistema načrtovan tako, da lahko celoten sistem zagotovi zahteve iz BIA-analize.

V nadaljevanju bom na kratko predstavil BIA-analizo spletnega portala, saj je bila arhitekturna postavitev IT-sistemov odvisna tudi od zahtev, ki so izhajale iz nje.

Skupna ocena je sestavljena iz več ocen področij, ocenjuje pa se vpliv na poslovanje v primeru nesreče in to, kakšen je (Klarič, 2014, str. 7):

- finančni vpliv,
- vpliv na ugled banke,
- pravni vpliv,
- vpliv s stališča tajnosti,
- vpliv na konkurenčno prednost.

Ocene so razvrščene od A do E, ocena E pomeni majhno škodo, medtem ko ocena A pomeni pomembno škodo za banko, enako se ocene definira tudi na ostalih področjih, kot je možna pravna odgovornost ter vpliv na ugled banke. Poleg vpliva na posamezna področja pa se ločujejo ocene še na obdobje nedelovanja, ocena ena predstavlja nedelovanje ene ure, dva štiri ure, tri tekoči dan, štiri en teden in pet izpad za več kot en teden (Klarič, 2014, str. 7).

Zahteva oziroma ocena, ki je bila podana v BIA-analizi za spletni portal banke, je bila postavljena nizko. Razlog je bil verjetno, da nedelovanje spletnega portala nima finančnega, pravnega ter vpliva na varnost, zaradi česar je bila tudi skupna ocena nizka.

Naslednji možni dejavnik je nepravilna ocena dejavnikov tveganja s strani uporabnikov v banki. Tako ocenjujem, da so se podcenili dejavniki, na katerih se najbolj občuti nedelovanje, tukaj je mišljen predvsem vpliv na ugled banke, nekaj pa tudi vpliv na konkurenčno prednost.

Ob izpadih delovanja spletnega portala banke se je izkazalo, da že njegovo krajše nedelovanje vpliva na ugled banke, saj je spletni portal prva stran, ki jo uporabniki, ki poizvedujejo o bančnih storitvah, obiščejo. Nedelovanje se tako odrazi v javnosti kot težave v delovanju IT-sistemov banke, čeprav vitalni deli IT-sistema nimajo povezave z delovanjem samega spletnega portala.

Kot primer lahko navedem dogodek, ki je vplival na delovanje spletnega portala banke, pri katerem je prišlo do kratkotrajne preobremenitve spletnih strežnikov aplikacije in nedelovanja spletnega portala. O dogodku so poročali v nekaterih dnevnih časopisih in novičarskih portalih, kar kaže na to, kako pomembno za ugled banke je, da njen spletni portal deluje neprekinjeno.

Tudi zaradi nizkih ocen iz BIA-analize se ob postavitvi spletnega portala ni načrtovala postavitev, ki bi zagotovila visoko razpoložljivost spletnega portala, zato je SUBP bil nameščen na strežnik, ki sta si ga delili tako spletna aplikacija kot tudi podatkovna baza. Takšna postavitev je v večini primerov neustrezna, saj je verjetnost, da bo ob povečanem prometu, torej povečanem številu obiska spletnega portala, strežniku zmanjkalo sistemskih sredstev.

Prvi korak je zato ločitev storitev, aplikacije in SUBP, vsakega na svoj strežnik. Postavitev podatkovne baze je bila izvedena brez uporabe rezervnega strežnika. V primeru težav na podatkovnem strežniku se v takem primeru izvede obnovitev podatkovne baze iz varnostne kopije, ki se dnevno izvaja tako na nivoju podatkovne baze kot tudi celotnega strežnika oziroma operacijskega sistema.

Zagotavljanje varnostnih kopij podatkovne baze je osnovni koncept obnavljanja podatkov v primeru težav ali odpovedi v delovanju. Varnostna kopija zagotavlja obnovitev podatkovne baze v primeru sistemskih težav, ki bi vplivale na konsistentnost podatkov, transakcijskih napak v aplikaciji, napak na diskih, ki bi povzročile nedostopnost podatkov, ter tudi za primere nesreč, ko je treba vzpostaviti delovanje na rezervni lokaciji (Chong, 2008, str. 722).

Izdelavo varnostnih kopij podatkovne baze lahko izvedemo na več načinov; kot polno varnostno kopijo (angl. *full backup*), prirastno varnostno kopijo (angl. *incremental backup*)

ter diferencialno varnostno kopijo (angl. *differential backup*). Vsako izmed varnostnih kopij pa lahko izvedemo v načinu hladnega varnostnega kopiranja ali pa v online načinu, brez prekinitve delovanja podatkovne baze (Wang, Li, & Xu, 2007, str. 765).

DB2 omogoča vse naštete načine izdelave varnostnih kopij, dodatno pa omogoča večjo granularnost pri izvajanju varnostnih kopij, saj lahko poleg polne varnostne kopije celotne podatkovne baze izvedemo varnostno kopijo posameznega tablepacea, torej prostora na disku, rezerviranega za podatke v tabelah, ki so povezane v posamezne tablespace podatkovne baze (Chong, 2008, str. 738).

V primeru podatkovne baze spletnega portala je bil izbran način izdelave online dnevnih polnih varnostnih kopij, glavna razloga sta bila neprestana dostopnost podatkovne baze ter velikost podatkovne baze, ki je relativno majhna, tako da je izvajanje diferencialnih oziroma prirastnih varnostnih kopij nepotrebno, saj se s tem ne bi prihranil porabljeni čas za izvedbo kopij, niti ne na diskovnem prostoru. Prav tako je obnovitev iz več kopij časovno zamudnejša, poleg tega pa tudi bolj tvegana, saj je treba za obnovitev uporabiti več varnostnih kopij skupaj.

Bartkowski et al. (2012, str. 11) navajajo, da je izdelovanje varnostnih kopij temeljna metoda zagotavljanja obnovitve podatkovnih baz po nesrečah. Varnostne kopije zadostujejo za zagotavljanje obnove po nesreči, zagotovo pa niso metode zagotavljanja visoke razpoložljivosti.

Vseeno niso vsi avtorji enakega mnenja, tako Marcus in Stern (2003, str. 108) navajata, da je uporaba varnostnih kopij, če so seveda vpeljani pravilni postopki tako varnostnega kopiranja kot tudi obnovitve podatkov iz varnostnih kopij, zagotovo ena izmed metod zagotavljanja visoke razpoložljivosti. Prav s pravilnimi postopki se namreč skrajša čas obnovitve podatkov iz nekaj dni v samo nekaj ur.

V primeru odpovedi podatkovnega strežnika spletnega portala banke sta tako na voljo dva načina obnove podatkovne baze:

- vzpostavitev ponovno delujočega okolja podatkovne baze iz varnostnih kopij,
- hladni zagon replikacije – podatki na ciljni podatkovni bazi se v celoti pobrišejo in ponovno prenesejo iz izvorne podatkovne baze.

Ne glede na to, da sta za obnovitev podatkovne baze spletnega portala na voljo celo dva načina, pa bi čas, ki bi bil potreben za obnovitev podatkovne baze ali iz varnostnih kopij ali s hladnim zagonom replikacije, presegel zahtevani čas za obnovitev delovanja, ne glede na to, ali bi bil strežnik že postavljen in pripravljen na delo (topla lokacija).

Ker podatkovna baza, kot del sistema spletnega portala, ni podvojena, zato predstavlja velik vpliv na visoko razpoložljivost spletnega portala banke. SUBP v tem primeru predstavlja kritično točko odpovedi. Marcus in Stern (2003, str. 78) postavljata odpravo kritične točke odpovedi kot enega od 20 ukrepov, ki jih je treba izvesti za zagotovitev visoke razpoložljivosti.

Iz napisanega zato sklepam, da je bilo začetno stanje visoke razpoložljivosti podatkovne baze neustrezno, saj dejansko ni obstajala možnost zagotavljanja visoke razpoložljivosti. Poleg tega je izbran način postavitve predstavljal tveganje ogrozitve sporazuma SLA na nivoju aplikacije spletnega portala banke, kot tudi težave pri zagotavljanju obnovitve po nesreči, saj omenjene metode vzpostavitve ponovnega delovanja podatkovne baze v celoti ne ustrezajo podanim zahtevam. Zato menim, da je bila odločitev za uvedbo visoke razpoložljivosti podatkovne baze pravilna.

# 5 VISOKA RAZPOLOŽLJIVOST PODATKOVNE BAZE SPLETNEGA PORTALA

V poglavju bom najprej predstavil izbor SUBP in razloge za izbor, sam SUBP je že bil opisan v teoretičnem poglavju magistrskega dela, nato pa se bom usmeril v predstavitev in razlago razlogov ter možnih scenarijev, ki so na voljo pri zagotavljanju visoke razpoložljivosti podatkovne baze za spletni portal banke. V zadnjem podpoglavju bom opisal projekt uvedbe rešitve, ter izbrano in uvedeno rešitev.

## 5.1 Predstavitev sistema za upravljanje baze podatkov

SUBP, ki je uporabljen za sistem spletnega portala banke, je IBM DB2. Na izbor SUBP za aplikacijo spletnega portala banke je vplivalo več dejavnikov. Zanesljivo najpomembnejši dejavnik pri izboru DB2 je bilo dejstvo, da omogoča hitro, zanesljivo in cenovno ugodno rešitev zagotovitve izbranega načina prenosa podatkov med centralno podatkovno bazo in bazo spletnega portala – replikacijo podatkov.

Vzpostavitev replikacije podatkov tako iz smeri »centralna podatkovna baza proti spletni podatkovni bazi« kot obratno je najlažje izvedljiva, če je na vseh strežnikih enaka SUBP, poleg tega pa je v ceni licence DB2 že všteta licenca, ki omogoča vzpostavitev replikacije med DB2 ne glede na uporabljeno platformo.

Pomembna postavka pri odločitvi je bila tehnična podpora, dodatna prednost je uporaba v svetu ter analiza podjetja Gartner (2016), saj se sklepa, da če je rešitev oziroma proizvod v kvadratu vodij, ima podjetje moč, zmožnost in vizijo ter s tem zagotavlja zanesljivo podporo za svoj produkt. Le-to je tudi izpostavljeno v omenjeni analizi, in sicer med glavne prednosti IBM-a Gartner šteje globalno prisotnost, saj podjetje IBM zagotavlja podporo, storitve in implementacijo na več področjih. V primeru DB2 je podpora za reševanje težav s SUBP zagotovljena s strani proizvajalca IBM.

Menim, da je podpora proizvajalca SUBP izjemnega pomena, saj vsaka prekinitev delovanja vpliva na delovanje vseh aplikacij, ki za shranjevanje podatkov uporabljajo podatkovno bazo, in prav podpora proizvajalca v veliko primerih olajša iskanje napake oziroma skrajša čas nedelovanja.

Na izbor pa je vplivalo tudi dejstvo, da je v banki centralna podatkovna zbirka tudi DB2; za ta SUBP je tudi največ izkušenj in znanja v sektorju za upravljanje informacijske tehnologije v banki, tako pri razvijalcih kot tudi upravljavcih podatkovnih zbirk (v nadaljevanju DBA). To znanje je bilo ob uvedbi aplikacije spletnega portala banke izjemno pomembno, ker je skrajšalo čas odprave začetnih performančnih težav ter tudi omogočalo optimalno prilagoditev podatkovnega modela aplikacije.

## 5.2 Razlogi ter možni scenariji uvedbe visoke razpoložljivosti

Pomembno je, da so razlogi za uvedbo visoke razpoložljivosti znani in ocenjeni tako finančno kot tudi z vidika zagotavljanja skladnosti z zakonodajo ter zahtevami uporabnikov. Samo tako je namreč možno izdelati IT-rešitev, ki bo na eni strani ustrezala uporabnikom in regulativi, po drugi strani pa ne bo pomenila nepotrebnega finančnega stroška.

Razloge za uvedbo visoke razpoložljivosti podatkovne baze spletnega portala banke lahko razdelim na naslednje sklope:

- Poslovne zahteve po zagotavljanju nemotenega delovanja spletnega portala, kot so opredeljene v izdelani BIA-analizi.
- Sporazum o nivoju zagotavljanja storitev (angl. s*ervice level agreement*, v nadaljevanju SLA) med sektorjem za upravljanje informacijske tehnologije kot ponudnikom storitev ter končnimi uporabniki na poslovni strani banke kot uporabniki storitve: V banki so vzpostavljeni sporazumi SLA za vse pomembne aplikacije, tako da v ta sklop štejemo tudi aplikacijo spletnega portala banke. Prikaz tabelaričnega sporazuma SLA je predstavljen v Tabeli 4.

*Tabela 4: Primer sporazuma SLA za aplikacijo*

| Aplikacija | Zahtevan čas delovanja | Razpoložljivost | Okrevalni čas |
|---|---|---|---|
| Aplikacija spletni portal banke | Čas delovanja v posameznem dnevu npr. 00.00–24.00 | Razpoložljivost v % | Zahtevan čas obnove delovanja po prekinitvi |

Poleg zahtevanega obratovalnega časa in razpoložljivosti aplikacije se v sporazumu SLA lahko definira še dodatne pogoje, kot na primer način in čas trajanja načrtovanih prekinitev delovanja ter časovni termin za izvedbo načrtovanih sprememb. Lahko pa se definira tudi različne zahteve za manjše težave in večje katastrofe. V primeru spletnega

portala je sporazum SLA definiran na nivoju delovanja aplikacije, tako da morajo biti zahteve za posamezne dele IT-podpore – strojna oprema, aplikacija, spletni strežnik, podatkovna baza – postavljene tako strogo, da zagotavljajo doseganje zahtev za celotno aplikacijo. Razpoložljivost je namreč seštevek razpoložljivosti vseh vključenih komponent.

- Arhitektura aplikacije spletnega portala banke: Aplikacija in podatkovna baza sta med seboj neločljivo povezani, saj aplikacija podatke bere neposredno iz podatkovne baze, prav tako pa vanjo zapisuje podatke o statistiki dostopov, prijavo uporabnikov v posamezne dele spletnega portala, ter vse podatke, ki nastajajo ob interakciji obiskovalcev portala z aplikacijo, na primer nagradne igre, vprašanja banki, pohvale, pripombe itd. Zaradi tega je posledica nedelovanja podatkovne baze tudi nedelovanje ali okrnjeno delovanje čelnega dela spletnega portala.
- Vpliv na visoko razpoložljivost spletnega portala: Vsi deli spletnega portala so imeli izveden način zagotavljanja visoke razpoložljivosti ali prek podvojitve strojne in programske opreme ali pa v izvedbi aktivne/pasivne gruče strežnikov. Pri podatkovni bazi pa je bilo izvedeno samo varovanje podatkov na nivoju varnostnih kopij.
- Ugled banke: Ne glede na zahtevani čas obnove delovanja po nesreči vpliv nedelovanja spletnega portala predstavlja možno izgubo ugleda in zaupanja v banko, če se pogosto dogaja, da so spletne strani banke nedostopne. Spletne strani ne samo bank, ampak večine podjetij so namreč prva stična točka oziroma že pravi prodajni kanal, kjer banka oglašuje svoje storitve in kot taka mora biti vedno dostopna. Predvidevam, da bodo z večjo digitalizacijo bank spletne strani bank pridobile še večjo veljavo, saj se predvideva, da se bodo prodajni kanali bank poenotili, ne glede na to, od kod dostopa komitent. V središču bo komitent, ki bo odločal, kdaj in kako želi uporabljati bančne storitve. S tem bodo zahteve po visoki razpoložljivosti oz. verjetno kar stalni razpoložljivosti narasle.

Našteti razlogi so bili osnova za spoznanje, da je uvedba visoke razpoložljivosti podatkovne baze spletnega portala banke nujno potrebna. Le z ukrepi visoke razpoložljivosti je možno zagotoviti ponovno vzpostavitev delovanja podatkovne baze v primeru težav v načrtovanem času.

Ob načrtovanju načina doseganja visoke razpoložljivosti pa je treba upoštevati tudi strošek uvedbe in delovanja rešitve, tako da bo pri analizi opredeljen tudi finančni vidik, saj ni nujno, da je najboljša tehnična rešitev tudi res najboljša rešitev, ko se vključi še finančni vidik.

Z namenom nižjih stroškov rešitve in dejstvom, da je rezervni center banke oddaljen manj kot 50 kilometrov (Gril, 2003, str. 83), menim, da je sprejemljivo, da se združi rešitev visoke razpoložljivosti in obnove po nesreči v skupno rešitev, kot je geografsko razdeljena gruča. Le-ta predstavlja prihranek, saj je vmesna rešitev med pravo arhitekturo visoke razpoložljivosti (dve vozlišči v istem RC za zagotavljanje visoke razpoložljivosti in rezervno vozlišče v rezervnem RC za zagotavljanje obnove po nesreči) in rešitvijo, ki zagotavlja samo

obnovitev po nesreči (mrzla lokacija ali topla lokacija). V prihodnosti lahko pride do težave pri taki postavitvi, predvsem če se banka odloči za rezervno lokacijo, ki bo oddaljena več kot 50 km, saj se na večjih razdaljah že pojavlja zakasnitev pri sinhronizaciji, ki lahko vpliva na hitrost delovanja primarne podatkovne baze in s tem aplikacije.

## 5.3 Uvedba rešitve visoke razpoložljivosti podatkovne baze

Od možnih scenarijev oziroma metod, ki jih podpira DB2, je bila izbrana rešitev z uporabo DB2 HADR-funkcionalnosti SUBP.

Zahteve, ki naj bi jih izbrana rešitev zagotavljala:

- Kazalnik RTO, rešitev mora zagotoviti zahtevo iz BIA-analize ter imeti čas obnove delovanja nižji, kot je zahteva, izražena s kazalnikom RTO.
- Kazalnik RPO = 0, torej brez izgube podatkov, v procesu odločanja se je izkazalo, da je potreben nadzor nad visoko razpoložljivostjo na nivoju podatkovne baze, saj se samo tako lahko zagotovi, da sta obe instanci podatkovne baze konsistentni in zagotavljata ničelno izgubo podatkov v primeru preklopa na rezervno lokacijo.
- Zahteve iz sporazuma SLA za aplikacijo spletnega portala banke.
- Finančni vidik, HADR-funkcionalnost SUBP je že vključena v osnovno licenco, tako da uporaba le-te ne vpliva bistveno na strošek uporabljene SUBP. Potrebna pa je dodatna licenca za rezervno lokacijo, število licenc je odvisno od uporabljenih možnosti HADR-funkcionalnosti (predstavljene bodo v naslednjem poglavju).
- Enostavnost uvedbe in vzdrževanja, rešitev mora biti čim enostavnejša za uvedbo in delovanje. Administracija same rešitve ne sme biti kompleksna, da lahko DBA svoj čas nameni drugim nalogam, in ne vzdrževanju podatkovne baze spletnega portala.
- Možnost zagotavljanja visoke razpoložljivosti tudi ob napovedanih prekinitvah ali nadgradnjah.

### 5.3.1 Opis izbrane rešitve – DB2 HADR

SUBP DB2 vsebuje funkcionalnost, imenovano HADR (angl. *High Availability and Disaster Recovery*), ki zagotavlja tako obnovitev SUBP po nesreči kot tudi visoko razpoložljivost. Zato lahko rečemo, da nudi rešitev obnovitev delovanja ob delnem ali pa popolnem izpadu vozlišča (McInnis, Zhuge, Rockwood, & Causley, 2011, str. 4).

Deluje na način izmenjave logov oziroma repliciranja logov med dvema ali več povezanimi podatkovnimi bazami DB2. Ko je funkcionalnost HADR aktivna, le-ta stalno izvaja replikacijo logov DB2 med povezanima podatkovnima bazama, vendar ne čaka na celoten log, ampak konstantno pošilja posamezne transakcije. Prednost pred uporabo standardne metode izmenjave logov je v tem, da ni treba ročno ali z avtomatiziranjem procesa zagotoviti funkcije dodajanja podatkov. Funkcionalnost HADR tako omogoča, da imamo povezane podatkovne baze ne samo med različnimi fizičnimi ali virtualnimi strežniki, ampak tudi med

različnimi lokacijami RC (Bartkowski et al., 2012, str. 13; Chong, 2008, str. 784; Baklarz & Zikopoulos, 2008, str. 850; Hrvatin, 2007, str. 27).

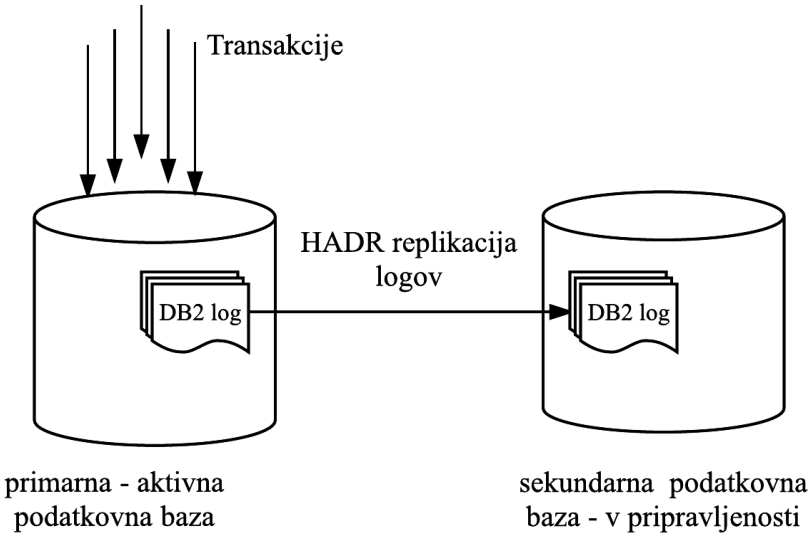
Čeprav sta lahko podatkovni bazi, ki sta v HADR-povezavi na istem strežniku, je za zagotavljanje obnove po nesreči nujno, ne samo da sta na različnih strežnikih znotraj istega RC, ampak morata biti strežnika tudi geografsko ločena. Vsaka podatkovna baza v HADR-konfiguraciji je popolnoma samostojna, gre za arhitekturo brez deljenja virov (McInnis et al., 2011, str. 5).

Poznamo dve glavni vrsti arhitekture uporabe in deljenja virov (Critchley, 2015, str. 338):

- Arhitektura vzajemne uporabe diskov (angl. *shared disk architecture*), primer je strežniška gruča, ki je lahko aktivna/pasivna ali pa aktivna/aktivna. Zaradi odprave kritične točke odpovedi lahko to arhitekturo nadgradimo v arhitekturo zrcaljenih diskov (angl. *mirrored disk architecture*), kjer so diski, ki so vzajemno uporabljeni, tudi zrcaljeni.
- Arhitektura brez deljenja virov (angl. *shared nothing architecture*), kot primer lahko navedem strežniško farmo.

V arhitekturi vzajemne uporabe diskov so vsa diskovna polja deljena med vozlišči, medtem ko je za arhitekturo brez deljenja virov značilno, da gre za popolnoma samostojne enote ali strežnike, ki si ne delijo med seboj nobenih virov, lahko pa so povezani na višjih nivojih, kot na primer v HADR-postavitvi SUBP DB2. Prednost arhitekture brez deljenja virov je v tem, da so strežniki popolnoma samostojni, tako da ne obstaja kritična točka odpovedi, saj imamo lahko dva ali več enakih strežnikov, ki so vsak zase samostojni. Grafični prikaz arhitekture HADR z dvema neodvisnima podatkovnima bazama je na Sliki 10.

*Slika 10: Arhitektura postavitve DB2 HADR*




*Vir: Prirejeno po R. F. Chong, Understanding DB2: learning visually with examples, 2008, str. 784, slika 14.31.*

V povezavi z licenčnimi zahtevami IBM loči med tremi možnostmi, ki so odvisne glede na izbiro metode visoke razpoložljivosti (Zikopoulos & Astorino, 2011, str. 3):

- vroča lokacija – polna licenca DB2,
- topla lokacija – delna licenca DB2,
- hladna lokacija – ni potrebne dodatne licence.

*Tabela 5: Licenčne zahteve za uporabo HADR-funkcionalnosti SUBP DB2*

| Vroča lokacija | Topla lokacija | Hladna lokacija |
|---|---|---|
| Programska oprema SUBP je nameščena na rezervni lokaciji | Programska oprema SUBP je nameščena na rezervni lokaciji | Programska oprema SUBP je nameščena na rezervni lokaciji |
| Instanca podatkovne baze je aktivna in sprejema podatke od vozlišča primarne podatkovne baze, instanca je tudi na voljo za končne uporabnike, ki lahko izvajajo bralne operacije na podatkovni bazi | Instanca podatkovne baze je aktivna in sprejema podatke od vozlišča primarne podatkovne baze, instanca ni na voljo za končne uporabnike, namenjena je izključno zagotavljanju visoke razpoložljivosti | HADR-funkcionalnost ni uporabljena, rezervna instanca podatkovne baze je neaktivna, strežnik, na katerem je nameščena SUBP, pa je lahko ali startan ali pa ugasnjen. |

*Vir: P. Zikopoulos & S. Astorino, Licensing distributed DB2 9.7 servers in a high availability (HA) environment, 2011, str. 3, tabela 1.*

Tabela 5 prikazuje glavne razlike med posameznimi možnimi scenariji uporabe DB2 HADR-funkcije, ki pa se ne razlikuje samo glede na licenčni model, ampak tudi glede na zmožnosti, ki jih omogoča. Za delovanje aplikacije spletnega portala sta na voljo dve možnosti: vroča lokacija in topla lokacija, hladna lokacija pa ne pride v poštev, saj ne zagotavlja dovolj visoke

razpoložljivosti. Aplikacija poleg branja podatkov iz podatkovne baze zapisuje tudi nove zapise v bazo. Zato je najprimernejša izbira tople lokacije, saj ni potrebe po branju iz rezervne podatkovne baze, tako da bi postavitev z aktivno sekundarno bazo bila odveč. Uporabna bi bila, če bi se izkazalo, da je podatkovna baza tako velika, da bi bila izdelava varnostnih kopij težavna. V takem primeru bi lahko sekundarno bazo uporabili za izvajanje varnostnih kopij, ne da bi vplivali na delovanje primarne.

Licenčni pogoji v primeru tople lokacije zahtevajo nakup dodatne licence, vendar je število dodatnih licenc neodvisno od procesorske moči rezervnega strežnika, tako da je možno strežnik izkoristiti za druge namene poleg podatkovne baze.

Ob postavitvi HADR-rešitve se je treba tudi odločiti za enega od tipov sinhronizacije. Bartkowski et al. (2012, str. 164) navajajo naslednje možne načine sinhronizacije med dvema podatkovnima bazama:

- Super asinhrona – šteje se, da je transakcija uspešna, ko se zapiše v log datoteko primarne baze. Ker primarna baza ne čaka na potrditev sekundarne baze, so transakcije uspešne ne glede na stanje sekundarne baze. Možne so izgube podatkov ob preklopu med lokacijama.
- Asinhrona – šteje se, da je transakcija uspešna, ko se zapiše v log datoteko primarne baze ter je poslana na sekundarno lokacijo. Pri tem načinu lahko pride do izgube podatkov, če se izgubijo mrežni paketi v trenutku težav na primarni bazi.
- Skoraj sinhrona – šteje se, da je transakcija uspešna, ko je zapisana v log datoteko primarne baze ter ko primarna baza prejme potrditev sekundarne baze, da je transakcija v njenem pomnilniku.
- Sinhrona – šteje se, da je transakcija uspešna, ko je zapisana v log datoteko sekundarne baze. V tem načinu sinhronizacije ne more priti do izgube podatkov, saj primarna baza čaka na to, da je transakcija potrjena s strani sekundarne baze.

Če imamo postavitev z eno aktivno in več sekundarnimi podatkovnimi bazami, lahko povezavo med prvo sekundarno bazo in primarno seveda izvedemo kot sinhrono, med drugo sekundarno bazo pa kot asinhrono. Tako lahko zagotavljamo visoko razpoložljivost med sinhronima bazama, asinhrona metoda bi bila uporabna za sinhronizacijo med bazama pri večjih razdaljah, ko se že pojavljajo zakasnitve zaradi razdalje.

Pri izbiri tipa sinhronizacije se moramo vprašati, katera zahteva je najpomembnejša. Če je pomembno, da replika ne vpliva na delovanje primarne baze in smo pripravljeni izgubiti nekaj podatkov, izberemo eno od asinhronih možnosti. Če so podatki pomembni in si ne moremo dovoliti izgube le-teh, izberemo enega od sinhronih tipov povezave.

Pomembno pri uporabi HADR-funkcionalnosti je, da preverjamo status sekundarne oziroma rezervne podatkovne baze. Samo stalno povezani podatkovni bazi namreč omogočata visoko

razpoložljivost in preklop med njima v primeru težav na aktivni bazi. Status povezave je tudi pomemben podatek, ki ga moramo nadzorovati pri uporabi HADR-načina.

Možni statusi so naslednji (Bartkowski et al., 2012, str. 346–347):

- lokalno dohitevanje (angl. *local catchup*),
- oddaljeno dohitevanje (angl. *remore catchup*),
- povezana (angl. *peer*),
- nepovezana (angl. *disconnected peer*).

Za vzpostavitev HADR-povezave med dvema ali več bazami DB2 je treba izvesti naslednje korake (Stedman & Lee, 2014, str. 7–11):

- postavitev rezervnega strežnika in namestitev SUBP,
- kreiranje nove instance baze na rezervnem strežniku,
- popravek konfiguracije ter vpis HADR-parametrov – IP-naslovi, porti, HADR-nastavitve – seznam konfiguracijskih parametrov je v Prilogi 6,
- izvedba varnostne kopije primarne baze,
- obnovitev podatkovne baze iz varnostne kopije na rezervni lokaciji,
- popravek konfiguracije na sekundarni instanci baze,
- start HADR na sekundarni bazi,
- start HADR na primarni bazi.

Pri zagotavljanju visoke razpoložljivosti je pomembno, da sta bazi v statusu povezani, saj samo takrat lahko zagotavlja, da so podatki sinhronizirani med bazama. Nadzorni ukazi in pregled statusa so prikazani v Prilogi 7.

Med glavne prednosti HADR štejemo (Baklarz & Zikopoulos, 2008, str. 850):

- zelo hiter preklop med baznima vozliščema, brez izgube podatkov,
- enostavna postavitev in administracija,
- majhen vpliv na zmogljivosti,
- omogoča nadgradnje brez prekinitev v delovanju,
- preklop ne vpliva na delovanje aplikacij.

Ko sta bazi med seboj povezani, lahko med njima preklapljamo z ukazom:

- db2 takeover hadr on db <ime baze>.

Ko izvedemo ukaz, se vlogi baz med seboj zamenjata in tako sekundarna baza postane aktivna, primarna pa postane nova sekundarna. Ta funkcionalnost je uporabna pri

nadgradnjah SUBP z rednimi popravki. DB2 HADR namreč omogoča, da izvedemo nadgradnjo, brez vpliva na razpoložljivost. Proces poteka tako, da se nadgradi najprej sekundarna baza, nato se izvede preklop in naredi nadgradnjo tudi primarne baze. Postopek lahko izvajamo v več korakih, kar pomeni, da nekaj časa sekundarna baza deluje z novejšimi popravki kot primarna. Ta funkcionalnost nam omogoča, da lahko najprej preverimo delovanje SUBP po naloženih popravkih in nato dokončno nadgradimo obe instanci baze (IBM, 2012, str. 194).

V primeru odpovedi primarne lokacije, strežnika ali samo SUBP pa moramo preklop HADR izvesti na silo (angl. *by force*), saj bazi v takem primeru nista več povezani (IBM, 2012, str. 195).

**Vpliv preklopa na aplikacijo**. Ob preklopu med HADR parom se aktivne povezave na podatkovno bazo prekinejo, aplikacija pa mora ponoviti povezavo. Za upravljanje zamenjave IP-naslova podatkovne baze sta na voljo dve možnosti (IBM, 2012, str. 22; Stedman & Lee, 2014, str. 12):

- uporaba virtualnega IP-naslova – aplikacija se vedno povezuje na virtualni naslov, pravi IP-naslov strežnika pa se ob preklopu menja,
- avtomatski preklop klientov (angl. *automatic client reorute*) – skrbi, da imajo klienti vedno podatek o statusu obeh podatkovnih baz. V primeru preklopa klient pozna tudi alternativni IP-naslov podatkovne baze ter v primeru nedostopnosti primarnega strežnika izvede ponovno povezavo na sekundarni strežnik/podatkovno bazo. Tako za aplikacijo ni prekinitve, saj se ponovna povezava zgodi avtomatično.

Tako HADR-metoda zagotavlja visoko razpoložljivost pri napovedanih prekinitvah, saj se preklop izvede v nekaj sekundah, kar v primeru aplikacije spletnega portala pomeni, da je baza nedosegljiva zelo kratek čas, ki ga končni uporabniki niti ne občutijo.

Treba je poudariti, da preklop med bazama ne poteka avtomatično, tako da mora DBA izvesti ukaz za preklop ročno, zato je za zagotavljanje visoke razpoložljivosti v primeru odpovedi primarne lokacije treba preklop HADR avtomatizirati. V nasprotnem primeru je čas trajanja prekinitve veliko daljši, saj je potreben ročni poseg DBA-jev.

Pri avtomatizaciji lahko uporabimo orodje Tivoli System Automation (v nadaljevanju TSA), ki je je namenjeno avtomatizaciji delovanja HADR-gruče. Lahko pa uporabimo tudi druge podobne produkte, ki pa so odvisni tudi od operacijskega sistema, kjer je nameščena SUBP, na primer Windows Server Cluster, HACMP itd.

### 5.3.2 Projekt uvedbe visoke razpoložljivosti

Projekt uvedbe visoke razpoložljivosti podatkovne baze spletnega portala banke je bil razdeljen v več faz, poudariti pa je treba, da sam projekt ni bil voden kot projekt, ampak se je izvajal kot linijska naloga, po razvojni metodologiji, kot je bila v času uvedbe v uporabi v banki.

Faze v razvoju rešitve so potekale po naslednjem vrstnem redu:

- izdelava terminskega načrta uvedbe HADR-rešitve,
- postavitev in testiranje v testnem okolju,
- postavitev produkcijskega podatkovnega strežnika na rezervno lokacijo,
- namestitev operacijskega sistema in SUBP na sekundarnem strežniku,
- konfiguracija DB2 SUBP v HADR-načinu,
- vzpostavitev nadzornih procesov za nadzor delovanja HADR,
- priprava dokumentacije za delovanje in postopki ročnega preklopa podatkovne baze na rezervno lokacijo.

Pri projektu uvedbe je treba izpostaviti nujnost testiranja funkcionalnosti. Čeprav postavitev ni težavna, je nujno, da se pred uvedbo v produkcijo izdelajo scenariji testiranja in dosledno testira delovanje v testnem okolju. Samo v takem primeru smo lahko prepričani, da bo ob pravi prekinitvi postavitev res delovala tako, kot se od nje pričakuje. V ta namen so se izdelali možni scenariji prekinitev, vsak izmed scenarijev se je testiral in zapisale so se ugotovitve. Dodatno testno okolje sicer res pomeni dodatno delo s pripravo in vzdrževanjem, poleg dela pa je še dodaten strošek infrastrukture in programske opreme, vendar brez testnega okolja ne bi bilo možno testirati delovanja, tako SUBP in HADR funkcionalnosti kot tudi novih verzij aplikacije.

Čeprav je priprava dokumentacije po navadi zadnji korak, v tem projektu ni bilo tako. Dokumentacija se je namreč pripravljala tudi med predhodnimi koraki projekta, saj je bistveno, da je dokumentacija uporabna, in ne sama sebi namen. Dokumentacija je pomembna tudi pri rednem vzdrževanju, saj brez nje lahko pride do nepričakovanih težav, ki vplivajo na razpoložljivost. DBA, ki upravlja SUBP, mora poznati postavitev in dokumentacija je edina pomoč, če sam ni sodeloval pri postavitvi.

Critchley (2015, str. 65) navaja, da je ena tretjina prekinitev v delovanju IT-podpore posledica človeške napake. Nadalje Schmidt (2006, str. 289) v povezavi z izgubo podatkov navaja, da je človeška napaka glavni razlog izgube podatkov, tako je krivec za 45 odstotkov vseh primerov izgube podatkov v podjetju, drugi razlogi so šele programske in strojne napake ter preostale nesreče.

Barnes (2001, str. 38) navaja, da v današnjem času, ko podjetja varčujejo pri stroških in je število zaposlenih na minimumu, obstaja veliko večja nevarnost, da pride do človeških napak pri delovanju sistemov in posledično težav v razpoložljivosti. Med glavne dejavnike spadajo: zmanjševanje števila zaposlenih, prenova procesov ter izdvajanje (angl. *outsourcing*).

Zato je nujno, da se postopki dosledno testirajo ter dokumentirajo. Z namenom minimiziranja vpliva človeških napak pa se je uvedel sistem avtomatskega stalnega nadzora delovanja SUBP, in sicer tako, da so se izdelale nadzorne skripte, ki preverjajo delovanje in javljajo alarm DBA-jem v primeru težav. Izsek skripte je prikazan v Prilogi 8. Pri tem bi rad opozoril, da to ni edini način, kako nadzirati delovanje, ukazov za nadzor je namreč več (Priloga 7), tudi skripte lahko napišemo na več možnih načinov, tako da je skripta v Prilogi 8 samo prikaz ene od možnosti.

# 6 ANALIZA USPEŠNOSTI ZAGOTAVLJANJA VISOKE RAZPOLOŽLJIVOSTI PODATKOVNE BAZE

Način uvedbe visoke razpoložljivosti podatkovne baze spletnega portala mora izpolniti pričakovanja oziroma zahteve neprekinjenega poslovanja banke, zahteve, ki izhajajo iz BIA-analize ter vse druge zahteve, predstavljene v prejšnjem poglavju. Namen analize je ugotoviti, ali je bila uvedena rešitev visoke razpoložljivosti spletnega portala ustrezna, ali pa je morda na voljo kakšna alternativna rešitev, ki bolje izpolni zahtevane pogoje.

V nadaljevanju bo s tem namenom izvedena analiza možnih rešitev visoke razpoložljivosti z uporabo metode tehtanih vsot, kjer bom med seboj primerjal izbrano rešitev ter še nekaj alternativnih rešitev zagotavljanja visoke razpoložljivosti podatkovne baze. Izdelana bo tudi SWOT-analiza uvedene rešitve, z namenom opredelitve prednosti, slabosti, priložnosti in nevarnosti.

## 6.1 Analiza možnih scenarijev uvedbe visoke razpoložljivosti podatkovne baze

Pri izboru načina zagotavljanja visoke razpoložljivosti podatkovne baze za spletni portal banke je na voljo več možnih scenarijev uvedbe visoke razpoložljivosti. V analizo bom vključil scenarije, ki so podprti na DB2 SUBP. Nekateri scenariji so že opisani v poglavju o visoki razpoložljivosti podatkovnih baz, tako da na tem mestu ne bodo še enkrat opisane vse njihove značilnosti, bodo pa pri posamezni rešitvi opisane glavne značilnosti rešitve, saj bodo v analizi konkretne rešitve, ki so vezane na SUBP DB2 in veljajo samo za ta SUBP. Funkcionalnost sicer lahko zagotavljajo tudi druge SUBP, vendar sta lahko razumevanje in delovanje različni med posameznimi SUBP.

Večkriterijsko odločanje (angl. *multiple criteria decision* analysis, v nadaljevanju MCDA) lahko opredelimo kot proces ocenjevanja posameznih možnosti glede na naše zahteve, z

namenom, da najdemo najboljšo možno alternativo. Če je v procesu odločanja več kriterijev, ki jih moramo zadovoljiti, govorimo o večkriterijskem odločanju. Bistveno pri uporabi metod MCDA je, da omogočajo odločevalcu organiziranje in sestavljanje informacij z namenom boljšega in bolj samozavestnega odločanja. Seveda to ne pomeni, da je ta metoda popolnoma objektivna, saj se že pri določanju kriterijev in njihove pomembnosti kažejo subjektivne ocene odločevalca. Pomembno je torej, da se odločevalec zaveda subjektivnosti metode in tako tudi interpretira spoznanja pri analizi (Belton & Stewart, 2002, str. 2–5).

Za metode MCDA je značilno, da proces odločanja obravnavajo po korakih, tako Triantaphyllou (2000, str. 5–6) navaja naslednje tri korake v procesu odločanja:

- opredelitev relevantnih kriterijev in alternativ,
- definiranje numeričnih vrednosti, ki pomenijo relativno vrednost posameznega kriterija,
- izračun vrednosti za razvrščanje možnih alternativ oz. scenarijev.

Med metode večkriterijskega odločanja štejemo še naslednje metode: metodo tehtanih vsot, metodo tehtanega produkta, analitično hierarhično metodo, ELECTRE metodo in TOPSIS metodo (Triantaphyllou, 2000, str. 5).

S pomočjo analize z metodo tehtanih vsot (angl. *weighted sum method*) bom preveril, ali je bila izbrana in uvedena rešitev res najboljša. Za izbrano metodo sem se odločil, ker je med vsemi metodami večkriterijskega odločanja največkrat uporabljena in tudi najlažje razumljiva.

## 6.2 Analiza z metodo tehtanih vsot

Metodo tehtanih vsot lahko opišemo kot iskanje najboljše alternative, če imamo na voljo m alternativ in n kriterijev. Izračun najboljše rešitve lahko opišemo z naslednjo enačbo (Triantaphyllou, 2000, str. 6):

$$A = \max_i \sum_{j=1}^{n} a_{ij} w_j, \quad i = 1, 2, 3, \ldots, m. \tag{2}$$

V enačbi 5 $A$ predstavlja najboljšo rešitev, n število kriterijev, $a_{ij}$ je dejanska vrednost i-te alternative s kriterijem $j$, $w_j$ pa je utež oz. pomembnost kriterija j.

Proces analize bo potekal v naslednjem vrstnem redu:

- definicija in opis kriterijev, ki jih rešitev mora zagotavljati, in kriterijev, ki so zaželeni, da jih zagotavlja,
- identifikacija in opis možnih alternativnih rešitev,

- kriteriji, ki jih rešitev mora zagotavljati, so izločitveni kriteriji, če jih rešitev ne zagotavlja, izpade iz nadaljnje obravnave,
- zaželeni kriteriji se v nadaljevanju ponderirajo po pomembnosti – seštevek kriterijev mora biti enak 1,
- s pomočjo metode tehtanih vsot se izvede razvrščanje možnih rešitev od najboljše do najslabše.
- za najboljšo alternativno rešitev se v nadaljevanju izdela SWOT-analiza.

**Nujni kriteriji.** Med nujnimi kriteriji, ki jim morajo zadostiti rešitve, sem definiral naslednje kriterije:

- delovanje rešitve s SUBP DB2,
- alternativna rešitev mora delovati na trenutni infrastrukturi,
- ciljna točka obnovitve podatkov – kazalnik RPO.

Prva dva kriterija sta po mojem mnenju nujna, saj bi v nasprotnem primeru uvedba alternativne rešitve pomenila prevelik vpliv na delovanje podatkovne baze spletnega portala. Poleg tega bi taka alternativa pomenila spremembo IT-infrastrukture, kar bi vplivalo na delovanje aplikacije, saj bi jo bilo treba prilagoditi delovanju z drugo SUBP. To pa je že preveč obsežna naloga, da bi lahko izvedla kot linijska aktivnost znotraj oddelka, in bi že zahtevala vzpostavitev projekta za zamenjavo SUBP, pri čemer bi se morali držati razvojne metodologije, ki je v veljavi v banki. Menim, da bi se morala iskati taka rešitev, če bi se izkazalo, da trenutna postavitev arhitekture spletnega portala ni zadovoljiva, ali pa bi se ugotovilo, da ne omogoča zagotavljanja nadaljnjega razvoja funkcionalnosti spletnega portala. Na primer manjkajoče funkcije, neoptimalno delovanje določenih funkcij, ki jih aplikacija nujno potrebuje itd. V takem primeru bi bilo treba začeti analizo zamenjave načina zagotavljanja visoke razpoložljivosti podatkovne baze ter tudi zamenjave SUBP.

Zadnji kriterij – ciljna točka obnovitve podatkov oziroma kazalnik RPO – je zahteva, da je kazalnik enak 0 oz. ne več kot 5 minut, torej da ne sme priti do izgube podatkov oziroma je le-ta čim manjša. To je po mojem mnenju predvsem pomembno pri podatkih, ki se generirajo na spletnem portalu (vprašanja banki, vpis v osebni del portala, naročilo na obvestila in ponujene podatke o tečajih, borznih podatkih itd.). Glede na to, da ti podatki vseeno niso pravi transakcijski podatki, katerih izguba bi pomenila finančni vpliv, je izguba nekaj minut podatkov v primeru katastrofe še sprejemljiva, ni pa sprejemljiva pri kontroliranem preklopu baz, pri nadgradnjah oziroma drugih napovedanih prekinitvah.

**Zaželeni kriteriji.** Zaželeni kriteriji so predstavljeni v Tabeli 6, v kateri so tako opisi posameznih kriterijev kot tudi relativna vrednost oziroma ponder posameznega kriterija, ki je odvisen od pomembnosti, ki jo predstavlja. Torej večjo ko ima relativno vrednost, večja je vrednost posameznega kriterija, večji ima tudi vpliv na končno oceno rešitve.

*Tabela 6: Zaželeni kriteriji za ocenjevanje scenarijev visoke razpoložljivosti*

| Ime kriterija | Opis kriterija | Relativna vrednost kriterija (ponder) |
|---|---|---|
| K1 | Hitrost obnove delovanja sistema ob nesrečah | 0,10 |
| K2 | Kompleksnost uvedbe rešitve | 0,10 |
| K3 | Kompleksnost vzdrževanja rešitve | 0,10 |
| K4 | Kompleksnost nadzora rešitve | 0,10 |
| K5 | Vpliv na HA-aplikacije ob preklopu vozlišč podatkovne baze | 0,40 |
| K6 | Strošek uvedbe rešitve | 0,20 |

S kriterijem K1 je mišljen čas, ki je potreben za vzpostavitev delovanja podatkovne baze na sekundarni lokaciji v primeru večjih ali manjših nenapovedanih prekinitev v delovanju. Pri tem je treba upoštevati vse že naštete razloge za uvedbo visoke razpoložljivosti, med drugim BIA-analizo, SLA, vpliv na ugled banke …

S kriterijem K2 je mišljena kompleksnost rešitve, torej čas, ki je potreben, in število administratorjev za postavitev, test in uvedbo izbrane rešitve.
Kriterij K3 nam pove, kako kompleksna je rešitev za vzdrževanje in nadzor. Pri tem mislimo na zahtevan čas in število administratorjev, ki so potrebni za izpolnjevanje nalog vzdrževanja in nadzora rešitve.

Kriterij K4 ocenjuje vpliv na delovanje čelne aplikacije spletnega portala ob napovedanih prekinitvah oziroma administratorskih posegih na podatkovni bazi. Zaželeno je, da napovedani posegi oziroma prekinitve ne vplivajo na delovanje čelne aplikacije.

Kriterij K5 pa ocenjuje možno rešitev glede na strošek uvedbe rešitve. Pri tem sta kot strošek mišljena investicija v programsko in strojno opremo ter strošek dela in storitev.

Glede na stanje celotne panoge si velikih investicij v rešitve ne moremo privoščiti, tako da je finančni vidik rešitve med pomembnejšimi kriteriji pri ocenjevanju.

Predpostavka pri ceni je, da sta pred analizo na voljo že aktiven primarni strežnik, pasiven sekundarni strežnik, na obeh strežnikih je že postavljen operacijski sistem in na primarnem strežniku je že nameščena SUBP ter kupljene vse licence za delovanje primarnega strežnika. Prav tako je mrežna povezljivost neodvisna od posamezne alternativne rešitve. Predpostavke so nujne zaradi zagotovitve objektivne ocene, saj je v nasprotnem primeru že uvedena rešitev – DB2 HADR – v boljšem izhodiščnem položaju.

Posamezne rešitve bodo ocenjene z ocenami od 1 do 5 pri vsakem od kriterijev, pri čemer je za vsak kriterij definirana drugačna zaloga vrednosti, saj nimajo vsi kriteriji enakih merskih enot. Tako bo pri stroškovnem kriteriju ocena odvisna od vrednosti investicije in stroška v EUR, pri kriteriju čas vzpostavitve delovanja pa bodo razredi podani v časovnem intervalu. Vsi razredi ocen kriterijev so zbrani v Prilogi 9.

Scenariji oziroma alternativne možnosti, ki jih bom uporabil pri analizi obstoječe rešitve, so naslednji:

- Visoka razpoložljivost na nivoju operacijskega sistema
  - varnostna kopija celotnega strežnika,
  - gruča strežnikov v aktivno/pasivnem načinu z deljenjem diskov.

- Visoka razpoložljivost na nivoju podatkovne baze
  - varnostna kopija na nivoju podatkovne baze,
  - ustavitev I/O operacij na podatkovni bazi in razdružitev zrcalnih diskov,
  - replikacija podatkov med centralnim računalnikom in dvema podatkovnima strežnikoma,
  - replikacija podatkov med dvema podatkovnima strežnikoma,
  - pošiljanje db logov na sekundarno lokacijo,
  - uporaba funkcije HADR,
  - avtomatizacija HADR v povezavi s TSA,
  - stalna razpoložljivost – DB2 Purescale – aktivna/aktivna gruča.

Vsaka alternativna rešitev mora izpolniti nujne pogoje, tako da bom najprej preveril, ali posamezne alternativne rešitve ustrezajo le-tem. Če ne ustrezajo, jih bom izločil iz nadaljnje analize. Tabela 7 prikazuje ustreznost rešitve glede na doseganje nujnih kriterijev.

*Tabela 7: Analiza ustreznosti rešitev glede na nujne kriterije*

| Kratko ime rešitve | Opis rešitve | Ustreza nujnim kriterijem |
|---|---|---|
| R1 | Varnostna kopija celotnega strežnika | NE |
| R2 | Aktivna/pasivna gruča strežnikov z deljenjem diskov | DA |
| R3 | Varnostna kopija na nivoju podatkovne baze | DA |
| R4 | Ustavitev I/O operacij na podatkovni bazi in razdružitev zrcalnih diskov | DA |
| R5 | Replikacija podatkov med centralnim računalnikom in dvema podatkovnima bazama spletnega portala - SQL replikacija | DA |
| R6 | Replikacija podatkov med dvema podatkovnima strežnikoma - Q replikacija | DA |
| R7 | Pošiljanje baznih logov na sekundarno lokacijo | DA* |
| R8 | DB2 funkcija HADR | DA |
| R9 | Avtomatizacija DB2 HADR v povezavi s TSA | DA |
| R10 | DB2 Purescale – aktivna/aktivna gruča | DA |

Od desetih opredeljenih scenarijev samo en scenarij – R1, ne izpolnjuje nujnih pogojev, zato bo izločen iz nadaljnje analize. Razlog za neustreznost je, da ne zagotavlja doseganja kazalnika RPO pod 5 minut, torej je možnost izgube podatkov lahko večja. To je predvsem zaradi razloga, ker se varnostnih kopij na nivoju celotnega sistema ne da izvajati tako pogosto (frekvenca 5 minut), poleg tega ni smiselno stalno izvajati varnostne kopije celotnega sistema. Zato lahko zaključim, da zagotavljanje visoke razpoložljivosti podatkovne baze z izvajanjem varnostnih kopij na nivoju strežnika ni primerno.

Iz nadaljnje analize bom izključil tudi scenarij R7 – Pošiljanje baznih logov na sekundarno lokacijo, predvsem zato, ker je DB2 HADR-funkcionalnost primerljiva oziroma uporablja enako metodo sinhronizacije podatkov med povezanima bazama, s tem da je pri HADR sam preklop med instancami baze veliko hitrejši kot pri metodi pošiljanja baznih logov.

*Tabela 8: Rezultat analize z metodo tehtanih vsot*

| Kratko ime rešitve | K1 | K2 | K3 | K4 | K5 | K6 | Skupaj |
|---|---|---|---|---|---|---|---|
| | (0,1) | (0,1) | (0,1) | (0,1) | (0,4) | (0,2) | (1) |
| R2 | 3 | 2 | 5 | 4 | 2 | 2 | 2,6 |
| R3 | 3 | 5 | 5 | 4 | 1 | 5 | 3,1 |
| R4 | 4 | 2 | 5 | 3 | 3 | 2 | 3 |
| R5 | 4 | 3 | 2 | 3 | 3 | 3 | 3 |
| **R6** | **5** | **3** | **4** | **3** | **3** | **3** | **3,3** |
| R8 | 5 | 4 | 4 | 3 | 4 | 4 | **4** |
| **R9** | **5** | **3** | **4** | **5** | **4** | **4** | **4,1** |
| **R10** | **5** | **2** | **4** | **5** | **5** | **1** | **3,8** |

Primer izračuna za R2 (vse druge rešitve se izračunajo po enakem postopku):
Končna ocena = (3 x 0,1) + (2 x 0,1) + (5 x 0,1) + (4 x 0,1) + (2 x 0,4) + (2 x 0,2) = 2,60

Tabela 8 prikazuje rezultat analize tehtanih vsot posameznih možnih scenarijev uvedbe visoke razpoložljivosti. Iz analize vidimo, da je najboljša uporaba rešitve avtomatizacije DB2 HADR v povezavi s TSA, sledi ji trenutno uporabljena rešitev, to je DB2 HADR, in nato DB2 Purescale oziroma aktivna/aktivna gruča. V nadaljevanju bom primerjal prve tri alternativne rešitve, pri čemer bosta rešitvi DB2 HADR in avtomatizacija DB2 HADR v primerjavi združeni, ker gre za nadgradnjo že uvedene rešitve, v primerjavo pa bom vključil še četrto uvrščeno rešitev, to je replikacijo podatkov s pomočjo Q replikacije.

Tabela 9 prikazuje glavne značilnosti treh najbolje ocenjenih rešitev za zagotovitev visoke razpoložljivosti.

*Tabela 9: Primerjava treh najbolje ocenjenih rešitev*

| **Q replikacija** | **HADR s TSA** | **DB2 PureScale** |
|---|---|---|
| Aktivna/aktivna postavitev, podatki se replicirajo med obema podatkovnima bazama. | Aktivna/pasivna postavitev, sekundarna baza samo sledi primarni, možna je aktivacija sekundarne baze za branje. | Aktivna/aktivna postavitev, vsaka od baz je samostojna, povezujejo se na nivoju predpomnilnika, ki ga vsaka od baz popravlja in s tem zagotavlja konsistenca. |
| Preklop med bazami je trenuten, ker sta obe aktivni, težava je lahko pri aplikaciji, ker se mora tudi na aplikaciji izvesti nova katalogizacija baze. | Preklop med bazami traja manj kot minuto, aplikacija deluje naprej nemoteno, ker ni potrebne nove katalogizacije baze. | Preklop med bazami je trenuten, ker sta obe aktivni, omogoča najhitrejši preklop, manj kot 30 sekund, tako da je za aplikacijo skoraj neviden. |
| Kompleksna za postavitev, treba je poznati način delovanja, uporablja MQ sporočila. | HADR-postavitev je enostavna, vzpostavitev zahteva samo definiranje nekaj parametrov v DB2, dodatno TSA zahteva nekaj več dela, ker je treba konfigurirati tudi preostale strežniške vire, kot na primer mrežne IP-naslove, virtualni IP-naslov … | Kompleksna postavitev, zahteva deljene diskovna polja, za katera je treba tudi zagotoviti podvojenost, da se odpravi kritična točka odpovedi. Pri postavitvi je nujno sodelovanje sistemskih administratorjev, DBA-jev in mrežnih administratorjev. |
| Strošek uvedbe je velik, potreben je nakup licence za Q replikacijo, dodatno pa je treba tudi licencirati DB2 na sekundarnem vozlišču, ker je aktiven, četudi ga aplikacija ne uporablja. | Strošek uvedbe je zanemarljiv oz. je enak uvedbi HADR. Tako HADR-funkcionalnost kot TSA sta del paketa DB2 in sta na voljo ob nakupu licenc za DB2. Sekundarno vozlišče se, če ne potrebujemo branja na sekundarni bazi, licencira s toplo licenco, kar pomeni 1 licenco ne glede na zmogljivost strežnika. | Strošek uvedbe v primeru spletnega portala banke je najvišji, potrebna sta nakup diskov SAN ter vzpostavitev replikacije med diski na dveh lokacijah za zagotovitev obnove po nesreči. Prav tako je treba polno licencirati vsako od baz, ker sta obe aktivni. Dodatno je treba dokupiti strežnika, na katerem se hrani predpomnilnik baz (angl. *cluster caching facility*). |

**K1 - Hitrost obnove delovanja sistema ob nesrečah**
Pri prvem zahtevanem kriteriju so se prve tri primerjane alternative izkazale kot odlične, saj je čas, potreben za obnovitev delovanja po nesreči, pri vseh manjši od 1 ure, kar je najvišja zahteva, ki izhaja iz BIA-analize.

**K2 - Kompleksnost uvedbe rešitve**
Uvedba DB2 PureScale je po mojem mnenju najbolj kompleksna, saj predstavlja čisto novo arhitekturno postavitev podatkovne baze. Trenutna rešitev HADR namreč deluje na principu nedeljenih virov, torej dveh povsem ločenih okolij, kjer sta samo podatkovni bazi povezani med seboj. DB2 Purescale pa zahteva deljene diske, prav tako pa še bazni predpomnilnik. Pri Q replikaciji je sicer arhitektura enaka DB2 HADR, vendar je kompleksnost večja, ker rešitev ni integrirana v SUBP, ampak je ločena komponenta.

**K3 - Kompleksnost vzdrževanja rešitve**
Vzdrževanje je pri Q replikaciji in DB2 HADR približno enako zahtevno, z razliko, da je v primeru težav pri DB2 HADR le-to zapisano v db log, pri Q replikaciji pa v log replikacije. Tako da je pri replikaciji potreben dodaten nadzor, prav tako pa niso podprte vse spremembe v shemi, zaradi česar je replikacija dobila nižjo oceno pri kriteriju K3. DB2 PureScale je za vzdrževanje najlažji, saj so vse bazne akcije podprte in ni potrebe po intervenciji. Seveda pa je kompleksnost nastavitev v primeru DB2 PureScale veliko večja, saj se lahko pojavijo težave pri zagotavljanju konkurenčnosti med bazami na istih objektih oziroma v tabelah.

**K4 - Kompleksnost nadzora rešitve**
Pri Q replikaciji je potreben nadzor, saj je rešitev samostojna in ni integrirana v SUBP. Zato brez nadzora baza ne ve, ali replikacija deluje ali ne, če pa ne deluje, se pojavi težava, kjer je možna izguba podatkov. Nadzor sicer ni kompleksen, je pa še dodatni dejavnik, ki lahko poslabša rešitev visoke razpoložljivosti. Pri preostalih dveh rešitvah je težava manjša, ker gre za integrirane rešitve, z uvedbo avtomatike TSA pa le-ta že vsebuje skripte za avtomatsko preverjanje delovanja in izvedbo akcij v primeru težav v delovanju enega od baznih vozlišč. Pri DB2 PureScale pa so vsa vozlišča aktivna, tako da težava enega od vozlišč ne pomeni prekinitve v delovanju aplikacije. Nadzor je avtomatiziran, seveda pa je pri obeh rešitvah še vedno nujno, da DBA preverja alarme in delovanje.

**K5 - Vpliv na HA-aplikacije ob preklopu vozlišč podatkovne baze**
Pri DB2 PureScale ne pride do preklopa vozlišč, ker sta obe vozlišči stalno aktivni. Tako da odpoved enega od vozlišč ne pomeni prekinitve za aplikacijo. Pri DB2 HADR v povezavi s TSA je za aplikacijo podobno, s tem, da ob preklopu pride do krajše prekinitve, ki pa je krajša od 1 minute. V aplikaciji se le-to pozna kot krajša ustavitev delovanja aplikacije. Q replikacija sicer na nivoju podatkovne baze izvede preklop brez prekinitve, saj sta obe vozlišči aktivni, vendar je potrebna nova konfiguracija povezave do baze na aplikaciji, kar

pa pomeni prekinitev v delovanju. V odvisnosti od avtomatizacije postopkov traja ta prekinitev lahko od nekaj minut do nekaj ur.

**K6 - Strošek uvedbe rešitve**
Pri oceni stroška sem upošteval tako nakupe strojne opreme kot licenc za programsko opremo. Najvišji strošek predstavlja uvedba DB2 PureScale, saj je potreben nakup diskov SAN, programske opreme za zrcaljenje diskov ter dodatnih licenc za SUBP. Pri Q replikaciji ni potrebne investicije v nakup SAN diskov, vendar dodaten strošek predstavlja nakup licenc za programsko opremo Q replikacije in dodatne licence za SUBP, ki jo je treba polno licencirati na obeh vozliščih. Poleg kompleksnosti pa je DB2 PureScale vsaj z vidika aplikacije preobsežna rešitev, namenjena je kompleksnim transakcijskim sistemom, tako da bi bila uvedba take rešitve za spletni portal banke neustrezna tako z vidika porabe sistemskih virov kot stroška, ki bi ga uvedba zahtevala. Pri uvedbi TSA je dodatni strošek nižji, investicija bi obsegala predvsem dodatna izobraževanja ter morebitni dokup posameznih komponent na strežniku, na primer mrežna kartica.

## 6.3 Analiza prednosti in slabosti ter priložnosti in nevarnosti uvedene rešitve visoke razpoložljivosti

Analizo prednosti in slabosti ter priložnosti in nevarnosti oziroma SWOT-analizo poznamo v ekonomiji predvsem v povezavi s strateškim managementom in celovitim ocenjevanjem podjetja. SWOT-analiza naj bi tako podjetjem pomagala oblikovati celovito strategijo nadaljnjega poslovanja (Dimovski & Penger, 2008, str. 53).

Lahko pa se SWOT-analiza uporabi tudi pri preostalem odločanju, saj omogoča enostaven vpogled v notranje in zunanje dejavnike, ki lahko v prihodnosti vplivajo na analiziran produkt, storitev ali strategijo. Pri delitvi na notranje in zunanje dejavnike se nadalje razdeli še na dejavnike, ki pozitivno vplivajo na proučevano področje, in dejavnike, ki so negativni.

SWOT-analizo sem izvedel na primeru uvedenega načina zagotavljanja visoke razpoložljivosti z namenom, da povzamem vse do sedaj ugotovljene značilnosti uvedene rešitve, z namenom, da na pregleden način zapišem vse prednosti in slabosti, nevarnosti in priložnosti rešitve ter poskušam objektivno opredeliti vse lastnosti rešitve. Pri izdelavi sem se opiral na dosedanje izkušnje, ki sem jih pridobil pri projektu ter med rednim delovanjem, deloma pa tudi na izkušnje iz podobnih projektov v banki. Predvsem pri možnih priložnostih in nevarnostih sem se opiral tudi na literaturo s področja podatkovnih baz in upravljanja le-teh, ter na spletne vire, ki obravnavajo področje podatkovnih baz in visoke razpoložljivosti.

V SWOT-analizi razdelimo notranje dejavnike na prednosti (angl. *strengths*) in slabosti (angl. *weaknesses*), zunanje dejavnike pa na priložnosti (angl. *opportunities*) in nevarnosti (angl. *threats*), tako da na koncu dobimo 2 x 2 matriko, ki jo prikazuje Tabela 10.

*Tabela 10: SWOT-analiza visoke razpoložljivosti podatkovne baze spletnega portala*

| **Prednosti** | **Slabosti** |
|---|---|
| - Uvedba HADR je enostavna<br>- Omogoča nadgradnje SUBP brez prekinitev za aplikacijo<br>- Deluje na vseh podprtih operacijskih sistemih ter vseh edicijah DB2<br>- Več možnih načinov sinhronizacije<br>- Hiter preklop brez izgube podatkov<br>- Preklop vozlišč ne vpliva na delovanje aplikacij | - Nekatere bazne akcije niso podprte<br>- Sekundarna baza, namenjena samo za DR<br>- Omogoča uporabo sekundarne baze za branje, vendar je le-to povezano z dodatnim stroškom licenc<br>- Dvojna poraba diskovnega prostora<br>- Brez avtomatizacije – daljši čas prekinitve |

| **Priložnosti** | **Nevarnosti** |
|---|---|
| - Avtomatizacija HADR<br>- HA-vozlišče v oblaku<br>- Varovanje pred aplikacijskimi napakami | - Odvisnost od proizvajalca SUBP<br>- Ukinitev podpore s strani proizvajalca SUBP |

Med glavne prednosti rešitve lahko štejem enostavnost uvedbe, koraki za uvedbo so bili predstavljeni v poglavju 5.3.1. Naslednja velika prednost je možnost nadgradenj brez prekinitve v delovanju podatkovne baze. Nadgradnja s pomočjo HADR poteka po naslednjem vrstnem redu:

- nadgradnja sekundarnega vozlišča s popravki, nedelovanje sekundarne baze ne vpliva na delovanje aplikacije, ki uporablja primarno bazo, saj je preklop za aplikacijo neviden in brez izgube podatkov,
- preklop med vozliščema, sekundarna lokacija ima sedaj primarno podatkovno bazo, na primarni lokaciji je podatkovna baza pripravljena na nadgradnjo,
- nadgradnja še podatkovne baze na primarni lokaciji,
- po posegu se lahko ponovno preklopi med vozliščema, da je ponovno primarna baza na primarnem vozlišču.

Seveda ima rešitev tudi slabosti, na katere je treba biti pozoren, saj neupoštevanje le-teh lahko privede do nepričakovanih težav in s tem vpliva na razpoložljivost. Ena od večjih slabosti so nepodprte bazne akcije. Tako na primer niso podprti veliki objekti (angl *Large Objects*, v nadaljevanju LOB), če le-ti niso logirani. Ob vzpostavitvi HADR-povezave je tako treba vsem lob poljem spremeniti logiranje, kar lahko povzroči večjo porabo diskovnega prostora, potrebnega za bazne loge. Prav tako se med baznimi vozlišči ne prenašajo spremembe v bazni konfiguraciji. DBA, ki je zadolžen za administracijo, mora paziti, da spremembe v konfiguraciji izvaja na vseh vozliščih. Zadnja nepodprta akcija so bazni postopki in funkcije. Definicije le-teh se sicer prenašajo med vozlišči, vendar brez povezanih objektov in knjižnic. Le-te je treba prenašati ročno. V takih primerih lahko pride do težav, saj so ročni posegi povsod težavni, ker hitro pride do človeške napake.

Naslednja slabost take arhitekture zagotavljanja visoke razpoložljivosti je redundanca tako pri strojni opremi kot pri potrebnih licencah za SUBP. Če imamo postavitev tipa topla lokacija, to pomeni, da ni dostopov aplikacije do sekundarne baze. Strojna oprema pravzaprav stoji neuporabljena. Če pa želimo imeti dostop do sekundarne baze, potrebujemo dodatne licence, ki so povezane z dodatnim stroškom. Prav tako pomeni postavitev, kjer diski niso deljeni med vozliščema, podvojeno porabo diskovnega prostora. Res pa je, da se v primeru baze spletnega portala uporabljajo lokalni diski, ki so veliko cenejši kot na primer SAN diski z eno od metod replikacije med diski.

Zadnja slabost je odzivnost rešitve na težave v delovanju. Če je HADR postavljen brez dodatne programske opreme za avtomatizacijo, je lahko čas preklopa občutno daljši. DBA mora namreč ročno ob alarmu o nedelovanju preklopiti bazo na drugo lokacijo.

Avtomatizacija procesov v največji meri vpliva na visoko razpoložljivost. Če imamo procese avtomatizirane in nadzorovane, bo avtomatika poskrbela, da se bo takoj ob prekinitvi iz katerega koli razloga izvedla prej definirana akcija. V primeru visoke razpoložljivosti podatkovne baze je to preklop med vozlišči. Tako je to velika priložnost, če obstaja možnost avtomatizacije, da se preveri, ali je v primeru podatkovne baze spletnega portala taka rešitev možna.

V zadnjem času je vedno več podjetij in IT-rešitev, ki za svoje delovanje uporabljajo oblačne storitve. V uporabi oblačnih storitev vidim priložnost tudi za primer visoke razpoložljivosti, predvsem pa zagotavljanja obnove po večjih nesrečah. Eden od predlogov je, da se naredi analiza delovanja rezervnega vozlišča v oblaku, saj bi s tem zagotovili, da ob primeru večje nesreče (potres, poplave) ne bi izgubili obeh vozlišč in bi lahko zagotavljali nemoteno delovanje. V primeru DB2 obstaja več možnosti. Ena od teh je uporaba storitve IBM Bluemix – »IBM DB2 on Cloud«. IBM tako ponuja možnost uporabe DB2 SUBP, ki je nameščena v njihovem oblaku, vse funkcionalnosti pa so enake kot v primeru namestitve SUBP DB2 v našem RC. Enako možnost nudijo tudi drugi ponudniki oblačnih storitev, na primer Microsoft Azure ali pa Amazon AWS.

Med glavne nevarnosti bi štel odvisnost od proizvajalca SUBP. Če se proizvajalec odloči, da ne bo več podpiral in razvijal SUBP, tako lahko ostanemo s sicer delujočo rešitvijo, vendar brez nadaljnjega razvoja ter potencialno nevarnostjo pojava novih varnostnih lukenj ali pa nekompatibilnosti z novejšimi verzijami operacijskega sistema. Sicer je tveganje enako pri vseh zaprtih rešitvah, vseeno pa bi bilo verjetno smiselno razmisliti o možnih scenarijih prehoda na druge SUBP, če proizvajalec ukine podporo ali pa SUBP postane licenčno predraga.

## 6.4 Predlogi za izboljšave

V prejšnjih podpoglavjih analize sem predstavil možne alternativne rešitve ter tudi SWOT-analizo izbrane rešitve. Izkazalo se je, da uvedena rešitev sicer izpolnjuje pogoje, tako zahteve BIA-analize kot tudi SLA. Sicer pa postavitev predstavlja nevarnost, da v posameznih primerih, predvsem pri nenapovedanih prekinitvah, pride do daljše prekinitve, kot je pričakovana. Razlog je predvsem v potrebi po ročnem posegu, saj zaenkrat ni uvedena nobena možnost avtomatizacije.

Prav tako je trenutni bazni strežnik fizični strežnik, ki je v celoti namenjen uporabi SUBP. Glede na podatke zasedenosti, ki se zbirajo, se ugotavlja, da je strežnik večino časa slabo izkoriščen, saj pri normalni obremenjenosti spletnega portala bazni strežnik večino časa čaka. Kljub temu pa strežnik mora biti tako zmogljiv, saj bi sicer ob povečani obremenitvi s strani aplikacije lahko prišlo do preobremenitve.

Zato sta moja predloga usmerjena v izboljšavo rešitve, in ne zamenjavo, saj sem z analizo prikazal, da je rešitev primerna za uporabo, obstajajo pa možnosti za izboljšavo izbrane rešitve.

Predlagam dva možna ukrepa:

- **Virtualizacija podatkovnega strežnika spletnega portala**

Z virtualizacijo strojne opreme bi lahko dosegli boljšo izkoriščenost obeh vozlišč, saj bi lahko poleg virtualnega strežnika, ki bi bil namenjen SUBP spletnega portala banke, strežnik uporabili še za druge aplikacije ali pa kot testni strežnik SUBP spletnega portala. Tako bi bil strežnik bolje izkoriščen. Če bi se izkazala potreba po več sistemskih virih za delovanje produkcijske baze, bi z nastavitvijo prioritet med virtualnimi okolji zagotavljali, da bi produkcijsko okolje vedno imelo zadosti sistemskih virov. Vse to bi pripomoglo k nižjemu strošku RC, saj bolj izkoriščena strojna oprema predstavlja prihranek tako pri hlajenju RC kot tudi porabi električne energije in velikosti RC. Z virtualizacijo bi tudi dosegli, da bi bilo okolje bolj neodvisno od fizične plasti, kar bi omogočalo boljše pogajalsko izhodišče pri nakupu novih strežnikov, ker bi odpadel problem navezanosti na enega dobavitelja. Virtualizirano okolje pa omogoča tudi lažjo pot v oblačno storitev, saj je z virtualizacijo že narejen prvi korak, to je abstrakcija fizične plasti.

- **Avtomatizacija rešitve DB2 HADR**

Vsaka prekinitev v delovanju ene od komponent sistema spletnega portala banke predstavlja razlog za nedelovanje storitve spletnega portala. Zato je namen uvedbe visoke razpoložljivosti na nivoju podatkovne baze, odprava kritične točke odpovedi. Le-to je bilo doseženo z uvedbo rešitve DB2 HADR, vendar ta še ne ponuja predvidljivega časa, ko je storitev ponovno na voljo ob težavah. Postopki preklopa med vozliščema so namreč ročni in odvisni od hitrosti in odzivnosti sistemskih administratorjev oziroma administratorjev

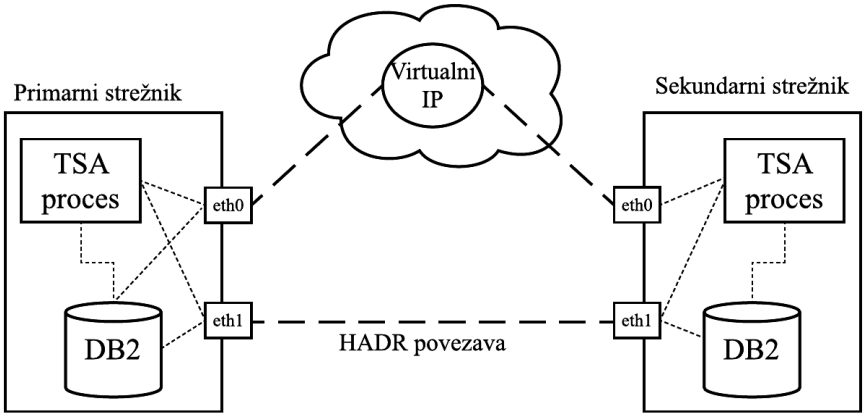
podatkovnih zbirk. Rešitev HADR ne ponuja možnosti avtomatskega odziva na prekinitev v delovanju primarne baze, tako da je v primeru težav potrebna ročna intervencija. Pri tem bi bilo treba razmisliti o uvedbi nove komponente, ki bi izvajala preklope med vozlišči avtomatsko, torej z uporabo programske opreme za gruče.

Rešitev, ki jo ponuja DB2, se imenuje TSA in je že del paketa. Glavne prednosti uporabe TSA pred drugimi namenskimi rešitvami za gruče so naslednje (Stedman & Lee, 2014, str. 3; Kapadia & Chiu, 2010):

- Neodvisen od operacijskega sistema, na katerem teče SUBP – seveda imajo rešitve, ki so neodvisne od operacijskega sistema, svoje slabosti. Glavna slabost je, da ne morejo izkoriščati možnosti operacijskega sistema, temveč uporabljajo samo osnovni nabor možnosti zaradi zagotavljanja kompatibilnosti z več operacijskimi sistemi. To pa je po drugi strani tudi velika prednost, saj lahko TSA konfiguracija ostane nedotaknjena, tudi če zamenjamo strežnik in operacijski sistem. Dodatno to predstavlja prednost za administratorje, ker je nabor ukazov enak ne glede na sistemsko platformo, na kateri teče SUBP v povezavi s TSA.
- Integrirana rešitev v DB2 SUBP - TSA se namesti že ob namestitvi SUBP DB2, tako da ni potrebna dodatna namestitev, ko se odločimo za njegovo uporabo. Prav tako sta DB2 in TSA interno povezana, tako da je avtomatizacija boljša in bolj predvidljiva. Integracija se kaže tudi pri nameščanju popravkov, saj se pri namestitvi DB2 popravkov (angl. *fixpack*) popravki naložijo tako na SUBP kot tudi na TSA, tako da se oba produkta nadgradita istočasno.
- Enostavnost uporabe – prednost za administratorje, ukazi so integrirani v nabor ukazov, ki jih pozna DB2 SUBP, tudi uporaba ukazov za upravljanje sistemske programske opreme za gruče se ne razlikuje od ukazov za upravljanje s SUBP. Tako se zmanjšata čas in strošek dodatnega izobraževanja DBA-jev, ne odpade pa v celoti, saj je kljub vsemu treba zagotoviti znanja o programski opremi TSA. Ni pa potrebe po znanju več variant programske opreme za gruče, saj je TSA možno uporabljati neodvisno od operacijskega sistema.
- Strošek licence – strošek licence za sistemsko programsko opremo je večinoma odvisen od zmogljivosti strojne opreme, kot recimo število procesorjev, število jeder procesorja itd. Strošek licenc je v večini primerov največji strošek, veliko večji kot strošek nakupa strojne opreme. Zato je pomembno pri uvedbi programske opreme razmisliti tudi o strošku licenc. V primeru TSA ni potrebnih dodatnih licenc za delovanje, saj je TSA že vsebovan v licenci SUBP DB2.

Delovanje TSA v tipični HADR-postavitvi, enaka postavitev je tudi v primeru podatkovne baze spletnega portala, je predstavljeno na Sliki 11.

*Slika 11: Avtomatizacija DB2 HADR z uporabo TSA*




*Vir: Prirejeno po D. Kapadia & M. Chiu, Using DB2 High Availability Disaster Recovery with Tivoli Systems Automation and Reliable Scalable Cluster Technology, 2010, str. 1, slika 1.*

Na obeh strežnikih sta nameščena SUBP in programska oprema TSA za upravljanje strežniških gruč. Le-ta je odgovorna za nadzor nad sistemskimi viri, kot so mrežni vmesniki, bazne instance ter HADR-podatkovne baze. Aplikacija se na gručo povezuje prek virtualnega IP-naslova. Če imamo v strežniku možnost namestiti več mrežnih vmesnikov, lahko HADR-povezavo povežemo prek druge povezave (eth1 na Sliki 11), ko teče promet med podatkovno bazo in aplikacijo (na Sliki 11 je to eth0). S tem povečamo mrežno zmogljivost in minimiziramo vpliv HADR-povezave na odzivnost podatkovne baze. Če pride do težav v delovanju primarnega strežnika, TSA avtomatično preklopi HADR-podatkovno bazo na sekundarno vozlišče in zamenja povezavo virtualnega IP-naslova. Aplikacija v takem primeru ne zazna prekinitve, saj se sam preklop zgodi v manj kot minuti. TSA lahko nadzoruje več sistemskih virov, odločitev pa je v pristojnosti sistemskih administratorjev in DBA-jev ter je odvisna od zahtev. Tako lahko TSA izvede preklop ob odpovedi kateregakoli sistemskega vira, lahko pa izvede preklop samo ob težavah v delovanju celotnega strežnika in odpoved mrežne povezljivosti ne predstavlja razloga za preklop vozlišča.

## SKLEP

V magistrskem delu sem obravnaval področje visoke razpoložljivosti podatkovne baze spletnega portala banke kot načina za zagotavljanje neprekinjenega poslovanja v banki. V prvem delu sem obravnaval literaturo s področja neprekinjenega poslovanja ter zagotavljanja obnove po nesreči z vidika informacijskih sistemov. Obravnaval sem tudi zakonodajo s področja operativnih tveganj v banki, usmeril sem se v operativno tveganje zagotavljanja neprekinjenega poslovanja, ki je v bančništvu zakonodajno predpisano s strani regulatorja.

Zato banke morajo imeti in vzdrževati načrte neprekinjenega poslovanja, kar sem tudi raziskal s pregledom letnih poročil največjih bank, ki delujejo v Sloveniji.

V nadaljevanju sem se usmeril v analizo uvedbe visoke razpoložljivosti podatkovne baze, ki je prikazala, da obstaja velik razkorak med neprekinjenim poslovanjem in visoko razpoložljivostjo. Če je za obnovo delovanja po nesreči zadovoljiv tudi nekoliko daljši izpad storitev, zahtevan čas je opredeljen z BIA-analizo, je pri rednem poslovanju vsak izpad delovanja nezaželen.

S tem namenom se informacijski sistemi načrtujejo tako, da poleg obnove po nesreči zagotavljajo tudi visoko razpoložljivost. To pomeni, da so dostopni v primeru načrtovanih prekinitev, kot na primer ob nadgradnjah posameznih komponent sistema, ter seveda tudi med nenačrtovanimi prekinitvami. Tako lahko rečemo, da je visoka razpoložljivost nadgradnja načrtov neprekinjenega poslovanja, usmerja pa se praviloma na IT-storitve in njihovo razpoložljivost.

Z analizo možnih rešitev zagotavljanja visoke razpoložljivosti podatkovne baze sem preveril, ali je uvedena rešitev res ustrezna, predvsem pa sem z več vidikov poskušal analizirati, ali obstaja boljša rešitev od uvedene. Tako sem izvedel analizo s pomočjo metode tehtanih vsot, kjer so bili zahtevani kriteriji naslednji:

- hitrost obnove delovanja sistema po nesreči,
- kompleksnost uvedbe,
- kompleksnost vzdrževanja,
- kompleksnost nadzora,
- vpliv na delovanje aplikacije,
- strošek uvedbe rešitve.

S ponderiranjem posameznih kriterijev sem dosegel, da so imeli kriteriji različno vrednost v končni oceni. Na tem mestu bi omenil, da sta ponderiranje in ocena narejena na podlagi mojih preteklih izkušenj ter ugotovitev, ki sem jih pridobil ob pregledu literature. Zato ocenjujem, da se subjektivnosti ocene nisem mogel v celoti izogniti, kar je sicer pomanjkljivost uporabljene metode, vendar menim, da sem prav zaradi natančne opredelitve kriterijev oceno rešitve približal objektivnim dejstvom.

Tako iz analize zaključujem, da uvedena rešitev visoke razpoložljivosti tehnično ni najboljša možna rešitev. Predvsem pri kriteriju vpliva na visoko razpoložljivost aplikacije se izkaže, da obstajajo boljše rešitve. S tem namenom sem v nadaljevanju podal predloge za izboljšavo.

Pri predlogih za izboljšavo sem se predvsem usmeril v predloge za izboljšavo obstoječe rešitve, in ne za njeno zamenjavo. Tako predlagam, da se obstoječa rešitev torej ohrani, da

pa se izvedeta še dodatna ukrepa, ki bosta pripomogla k izboljšanju visoke razpoložljivosti podatkovne baze spletnega portala banke.

Omenjena predloga izboljšave rešitve sta:

- Uvedba virtualizacije podatkovnih strežnikov z namenom boljše izkoriščenosti strojne opreme in možnega finančnega prihranka v prihodnosti.
- Avtomatizacija DB2 HADR rešitve – kot je bilo v nalogi poudarjeno, je razlog ene tretjine prekinitev napovedana prekinitev, preostalo so nenapovedane prekinitve. Nenapovedane prekinitve se lahko zmanjšajo ali celo odpravijo z izvedbo avtomatizacije storitve visoke razpoložljivosti. S tem odpadejo prekinitve v delovanju, ki so posledica odziva skrbnikov na težave v delovanju podatkovne baze. Če rešitev ni avtomatizirana, je namreč od posameznega skrbnika odvisno, kako hitro se odzove na težave.

Z uvedbo predlaganih izboljšav bi rešitev pomenila višji nivo zagotavljanja visoke razpoložljivosti podatkovne baze in kot taka menim, da ustreza vsem zahtevam. Zato menim, da je rešitev primerna za podatkovno bazo spletnega portala banke, kakor tudi, da se rešitev visoke razpoložljivosti uporabi pri drugih postavitvah podatkovnih baz v banki.

# LITERATURA IN VIRI


1. Abanka d.d. (2016). *Letno poročilo 2015*. Ljubljana: Abanka d.d.
2. Acharya, A. P., Bal, T., Clark, C., Goering, A., Graesser, D., Littera, A., Manhaes M., Scott, G. (2016). *Establishing a secure hybrid cloud with the IBM PureApplication family*. Raleigh: IBM Redbooks.
3. Alcott, T. (2010). *The WebSphere Contrarian: High availability (again) versus continuous availability*. Najdeno 20. maja 2016 na spletnem naslovu http://www.ibm.com/developerworks/websphere/techjournal/1004_webcon/1004_webcon.html
4. Anjard, R. P. (1994). The Basics of Database Management Systems (DBMS). *Industrial Management & Data Systems*, *94*(5), 11–15.
5. Baklarz, G., & Zikopoulos, P. C. (2008). *DB2 9 for Linux, Unix, and Windows: DBA guide, reference and exam prep* (6th ed.). Upper Saddle River: IBM Press.
6. Banka Slovenije. (2015). *Sklep o upravljanju s tveganji in izvajanju procesa ocenjevanja ustreznega notranjega kapitala za banke in hranilnice.* Najdeno 5. junija 2016 na spletnem naslovu https://www.uradni-list.si/1/content?id=77137
7. Barnes, J. C. (2001). *A guide to business continuity planning*. Chichester: John Wiley.
8. Bartkowski, S., De Buitlear, C., Kalicki, A., & Loster, M. (2012). *High availability and disaster recovery options for DB2 on Linux, UNIX, and Windows*. Poughkeepsie: IBM Corporation.
9. *Basel 2 — ENISA.* (2016, september 6). Najdeno 9. junija 2016 na spletnem naslovu https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/business-process-integration/governance/frameworks/basel-2
10. Belton, V., & Stewart, T. J. (2002). *Multiple Criteria Decision Analysis an Integrated Approach*. Boston: Springer US.
11. British Standards Institution. (2012). *ISO 22301:2012(E): Social security -- Business continuity management systems -- Requirements*. Geneva: International organization for standardization.
12. Brooks, C., Bedernjak, M., & Juran, I. (2002). *Disaster recovery strategies with Tivoli Storage management*. San Jose: IBM Corporation.
13. Chernobai, A. S., Rachev, S. T., & Fabozzi, F. J. (2007). *Operational risk: a guide to Basel II capital requirements, models, and analysis*. Hoboken: Wiley.
14. Chong, R. F. (2008). *Understanding DB2: learning visually with examples* (2nd ed.). Upper Saddle River: IBM Press Pearson plc.
15. Cochrane, B. J., & McKnight, K. A. (2013, junij 20). *DB2 JSON capabilities, Part 1: Introduction to DB2 JSON*. Najdeno 20. junija 2016 na spletnem naslovu http://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson1/
16. Critchley, T. (2015). *High availability IT services*. Boca Raton: CRC Press/Auerbach.
17. De Haes, S., & Van Grembergen, W. (2015). *Enterprise governance of information technology: achieving alignment and value, featuring COBIT 5* (2nd ed). Cham: Springer.

18. Dimovski, V., & Penger, S. (2008). *Temelji managementa*. Harlow: Pearson Education.
19. Dines, R. (2011). The State of Disaster Recovery Preparedness. *Disaster recovery journal*, *24*(1), 12–22.
20. Erder, M., & Pureur, P. (2016). *Continuous architecture: sustainable architecture in an agile and cloud-centric world*. Amsterdam: Morgan Kaufmann/Elsevier.
21. European System of Central Banks. (2007). *ESCB definicije glavnih terminov s področja neprekinjenega poslovanja plačilnih sistemov in sistemov poravnave vrednostnih papirjev.* Najdeno 20. maja 2016 na spletnem naslovu https://www.bsi.si/library/includes/datoteka.asp?DatotekaId=3150
22. Evropska Centralna Banka. (2006). *Business continuity oversight Expectations for systemically important Payment systems (sips)*. Najdeno 8. junija 2016 na spletnem naslovu https://www.bsi.si/library/includes/datoteka.asp?DatotekaId=3668
23. Finn, A. (2013). *Windows server 2012 hyper-v installation and configuration guide*. Indianapolis: Sybex.
24. Forrester. (2013). *How Organizations Are Improving Business Resiliency With Continuous IT Availability.* Najdeno 15. maja 2016 na spletnem naslovu http://www.emc.com/collateral/analyst-report/forrester-improve-bus-resiliency-continuous-it-avail-ar.pdf
25. Gartner. (2015). *Magic Quadrant for Operational Database Management Systems.* Najdeno 13. maja 2016 na spletnem naslovu https://www.gartner.com/doc/reprints?id=1-2PMFPEN&ct=151013
26. Gartner. (2016). *Magic Quadrant Research Methodology.* Najdeno 14. maja 2016 na spletnem naslovu http://www.gartner.com/technology/research/methodologies/research_mq.jsp
27. Gregg, M. (2006). *CISSP*. Indianapolis: Que Certification.
28. Gregory, P. H. (2008). *IT disaster recovery planning for dummies*. Hoboken: Wiley.
29. Gril, M. (2003). *Varnost in tehnološka zaščita informacijskega sistema v banki* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
30. Hiles, A. (2002). *Enterprise risk assessment and business impact analysis: best practices*. Brookfield: Rothstein Associates.
31. Hrvatin, R. (2007). *e-Bančništvo - neprekinjeno delovanje s pomočjo funkcije HADR.* (diplomsko delo). Maribor: EPF - Ekonomsko-poslovna fakulteta.
32. IBM. (2011). *IBM InfoSphere Replication Server: SQL Replication Guide and Reference.* Armonk: IBM Corporation.
33. IBM. (2012). *DB2 Version 9.7 for Linux, UNIX, and Windows - Data Recovery and High Availability Guide and Reference.* Armonk: IBM Corporation.
34. IBM. (2016). IBM - *Why mainframe?* Najdeno 13. maja 2016 na spletnem naslovu http://www-03.ibm.com/systems/z/why-mainframe/
35. ISACA. (2011). *Societal security — Business continuity management systems — Requirements.* Geneva: International organization for standardization.
36. ISACA. (2012a). *COBIT 5: a business framework for the governance and management of enterprise IT*. Rolling Meadows: ISACA.

37. ISACA. (2012b). *COBIT 5: enabling processes*. Rolling Meadows: ISACA.
38. ISACA. (2012c). *ISACA Glossary of Terms English - Slovenian*. Najdeno 20. maja 2016 na spletnem naslovu http://www.isaca.si/dokumenti/ISACA_Glossary_Translation-SI.pdf
39. Kapadia, D., & Chiu, M. (2010). *Using DB2 High Availability Disaster Recovery with Tivoli Systems Automation and Reliable Scalable Cluster Technology [CT321]*. Najdeno 4. junija 2016 na spletnem naslovu http://www.ibm.com/developerworks/data/tutorials/dm-1009db2hadr/
40. Klarič, K. (2014). Analiza vpliva na poslovanje v praksi. *Konferenca Neprekinjeno poslovanje.* Najdeno 5. junija 2016 na spletnem naslovu https://www.palsit.com/gradiva-BCP-2014/gradivo/Kristijan_Klaric_NLB.pdf
41. Klarič, K. (2016). Testiranje načrtov neprekinjenega poslovanja. *Konferenca Neprekinjeno poslovanje 2016.* Najdeno 5. junija 2016 na spletnem naslovu https://www.palsit.com/gradiva-BCP-2016/gradivo/Testiranje%20Na%C4%8Drtov%20neprekinjenega%20poslovanja_Klari%C4%8D_NLB.pdf
42. Kosutic, D. (2016). *Disaster recovery site - What is the ideal distance?* Najdeno 24. maja 2016 na spletnem naslovu http://advisera.com/27001academy/knowledgebase/disaster-recovery-site-what-is-the-ideal-distance-from-primary-site/
43. Kramer, J. (2003). *The CISA prep guide: mastering the certified information systems auditor exam*. New York: Wiley.
44. Kuznetsov, S. D., & Poskonin, A. V. (2014). NoSQL data management systems. *Programming and Computer Software*, *40*(6), 323–332.
45. Laan, S. (2011). *IT infrastructure architecture: infrastructure building blocks and concepts*. b.k.: Lulu Press.
46. Lavrič, R. (2004). *Analiza replikacije na primeru portala NLB* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
47. Liu, L., & Özsu, M. T. (Ur.). (2009). *Encyclopedia of database systems*. New York: Springer.
48. Long, J. O. (2012). *ITIL 2011 at a glance*. New York: Springer.
49. Marcus, E., & Stern, H. (2003). *Blueprints for high availability* (2nd ed.). Indianapolis: Wiley Publishing.
50. Mattord, H. J., & Whitman, M. E. (2014). *Business continuity: state of the industry report*. Waltham: Elsevier.
51. McInnis, D., Zhuge, Y., Rockwood, J., & Causley, R. (2011). *Best Practices: DB2 High Availability Disaster Recovery.* Najdeno 2. junija 2016 na spletnem naslovu https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wc9a068d7f6a6_4434_aece_0d297ea80ab1/page/High%20Availability%20Disaster%20Recovery
52. Moeller, R. R. (2013). *Executive's guide to IT governance: improving systems processes with service management, CobiT, and ITIL*. Hoboken: Wiley.

53. Naujoks, U. (2002). Business Continuity Planning (BCP) in a Globalised Bank. V M. J. Wieczorek, U. Naujoks, & B. Bartlett (Ur.), *Business Continuity* (str. 99–117). Berlin: Springer Berlin Heidelberg.
54. NLB d.d. (2016). *Letno poročilo NLB Skupine 2015*. Ljubljana: NLB d.d.
55. Nova KBM d.d. (2016). *Letno poročilo 2015 skupine Nove KBM in Nove KBM d.d. Vrednote, ki sežejo dlje.* Maribor: Nova KBM d.d.
56. Pogačar, M. (2010, maj). Rešitev iz prakse: Uvedba Hyper-V in Microsoft System Center Virtual Machine Manager v NLB d.d. *NT konferenca 2010.* Najdeno 8. junija 2016 na spletnem naslovu https://www.ntk.si/urnik/arhiv/2010/resitev_iz_prakse_uvedba_hyper_v/804
57. Portnoy, M. (2012). *Virtualization essentials*. Indianapolis: John Wiley & Sons, Inc.
58. Protiviti. (2013). *Guide to Business Continuity Management Frequently Asked Questions.* Najdeno 20. maja 2016 na spletnem naslovu https://www.protiviti.com/en-US/Documents/Resource-Guides/Guide-to-BCM-Third-Edition-Protiviti.pdf
59. Savage, M. (2002). Business continuity planning. *Work Study*, *51*(5), 254–261.
60. Scheepers, M. J. (2014). Virtualization and containerization of application infrastructure: A comparison. *University of Twente, Enschede*. Najdeno 10. maja 2016 na spletnem naslovu http://referaat.cs.utwente.nl/conference/21/paper/7449/virtualization-andcontainerization-of-application-infrastructure-a-comparison.pdf
61. Schmidt, K. (2006). *High availability and disaster recovery concepts, design, implementation*. Berlin: Springer.
62. Scott, D. (2005). *Survey Confirms There Are Many Effective Disaster Recovery Strategies.* Najdeno 10. maja 2016 na spletnem naslovu https://www.gartner.com/doc/476877/survey-confirms-effective-disaster-recovery
63. Sklep o plačilnih sistemih. *Uradni list RS* št. 73/09 in 5/11.
64. Sklep o ureditvi notranjega upravljanja, upravljalnem organu in procesu ocenjevanja ustreznega notranjega kapitala za banke in hranilnice. *Uradni list RS* št. 73/15.
65. Snedaker, S., & Rima, C. (2014). *Business continuity and disaster recovery planning for IT professionals* (2nd ed.). Amsterdam: Syngress.
66. Spremić, M., Bajgorić, N., & Turulja, L. (2013). Implementation of it Governance Standards and Business Continuity Management in Transition Economies: The Case of Banking Sector in Croatia and Bosnia-Herzegovina. *Economic Research-Ekonomska Istraživanja*, *26*(1), 183–202.
67. Stedman, P., & Lee, P. (2014). *Automating HADR on DB2 10.1 for Linux, UNIX and Windows Failover Solution Using Tivoli System Automation for Multiplatforms.* Armonk: IBM Corporation.
68. Svata, V. (2013). System View of Business Continuity Management. *Journal of Systems Integration*, *4*(2), 19.
69. Šavli, U. (2008). *Razvoj informacijskega sistema EnKa za spremljanje sheme ugodnosti* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
70. Triantaphyllou, E. (2000). *Multi-criteria Decision Making Methods: A Comparative Study*. Boston: Springer US.

71. Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. b.k.: James Turnbull.
72. Unicredit Banka Slovenija d.d. (2016). *Razumemo vaš vsak dan. Razvijamo 360° rešitve. 2015 Letno poročilo / Annual Report*. Ljubljana: Unicredit Banka Slovenija d.d.
73. Wang, Y., Li, Z., & Xu, J. (2007). Protecting and Recovering Database Systems Continuously. *Advances in data and web management: Joint 9th Asia-Pacific Web Conference, APWeb 2007, and 8th International Conference on Web-Age Information Management, WAIM 2007* (str. 765-776). Huang Shan: Advances in Data and Web Management.
74. Warrick, C. (2003). *Seven Tiers of Disaster Recovery Solutions.* Najdeno 3. aprila 2016 na spletnem naslovu http://www.redbooks.ibm.com/abstracts/tips0340.html
75. Watters, J. (2014). *Disaster recovery, crisis response, and business continuity a management desk reference*. New York: Apress.
76. White, B., De Leon, C. A., Hoogerbrug, E., Palacio, E., Pinto, F., Sannerud, B., … Yang, J. J. (2016). *IBM z13 and IBM z13s technical introduction*. Poughkeepsie: IBM Redbooks
77. Zikopoulos, P., & Astorino, S. (2011, 18. avgust). *Licensing distributed DB2 9.7 servers in a high availability (HA) environment. IBM.* Najdeno 4. junija 2016 na spletnem naslovu http://www.ibm.com/developerworks/data/library/techarticle/dm-0909db2halicensing/dm-0909db2halicensing-pdf.pdf
78. Zakon o bančništvu. *Uradni list RS* št. 25/15.
79. Žnidar, B. (2006). *Vloga informatike pri zagotavljanju neprekinjenega poslovanja v slovenskih podjetjih* (magistrsko delo). Ljubljana: Ekonomska fakulteta.

# PRILOGE

**KAZALO PRILOG**

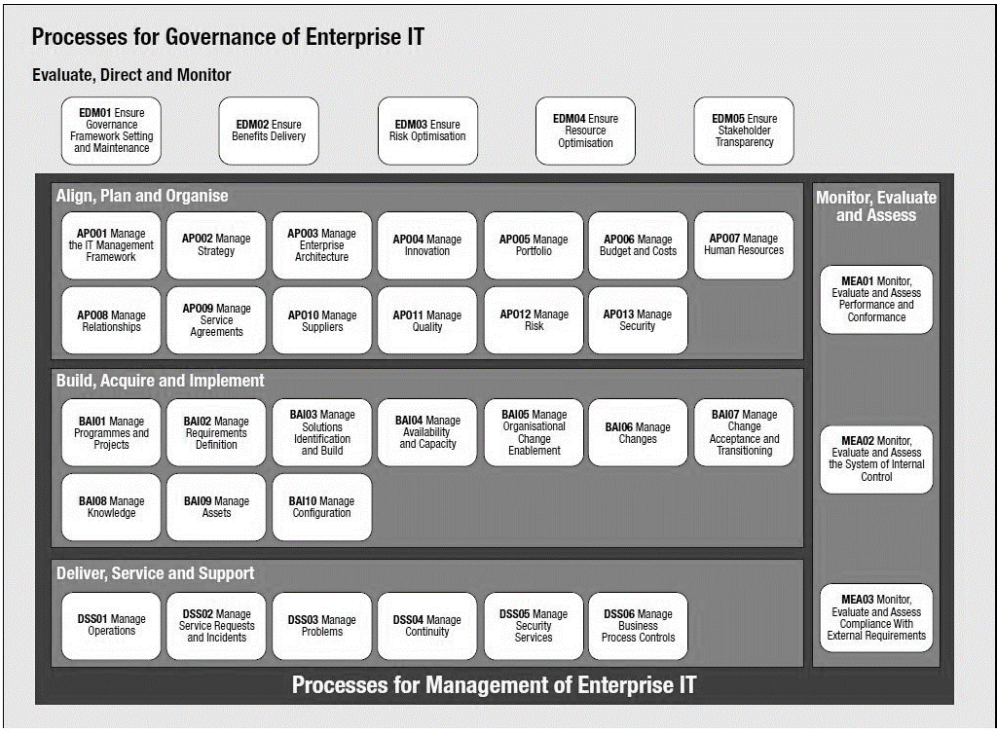
**Priloga 1: Terminološki seznam in seznam kratic**

ACID – Atomicity, Consistency, Isolation, Durability – Atomarnost, konsistenčnost, neodvisnost in trajnost transakcije
ACR - *Automatic client reorute* – Avtomatski preklop klientov
AFR – Annualized failure rate – Letna verjetnost odpovedi
AWS – Amazon Web Services – Oblačne storitve podjetja Amazon
Basel II – Priporočilo za oblikovanje politik v zvezi z obvladovanjem tveganj v bankah
BCM- Business Continuity Management - Upravljanje neprekinjenega poslovanja
BCP – Business Continuity Plan – Načrt neprekinjenega poslovanja
BIA – Business Impact Analysis – Analiza vpliva na poslovanje
BS – Banka Slovenije
COBIT – Conrol Objectives for Information and Related Technologies – Standard kontrolnih ciljev za informacijsko in sorodno tehnologijo
DB2 HADR – DB2 High Availability and Disaster Recovery – Rešitev DB2 SUBP za zagotavljanje visoke razpoložljivosti in obnove po nesreči.
DBA – Database administrator – Upravljavec baz podatkov
DMZ – Demilitarized zone – Demilitarizirano območje
DNS – Domain name system – Sistem domenskih imen
DR – Disaster recovery – Obnova po nesreči, obnova po katastrofi
DRP – Disaster recovery plan – Načrt obnove po nesreči
ECB – European Central Bank – Evropska centralna banka
ESCB – European System of Central Banks – Evropski sistem centralnih bank
HA – High Availability – Visoka razpoložljivost
IP – Internet Protocol – Internetni protokol
ISACA – Kratica, ki je včasih pomenila Information Systems Audit and Control Association
ISO – International Organization for Standardization – Mednarodna organizacija za standardizacijo
IT – Information technology – Informacijska tehnologija
ITIL – IT Infrastructure library – Zbirka napotkov za upravljanje in uvajanje IT-storitev
JSON – JavaScript Objet Notation – Enostaven format za izmenjavo podatkov
LAN – Local area network – Lokalno omrežje
LOB – Large object – Veliki objekti v podatkovni bazi
Mainframe – Centralni računalnik
MCDA – Multiple – Criteria decision analysis - Večkriterijsko odločanje
MTBF – Mean time between failures – Povprečni čas med odpovedmi
NoSQL – Non SQL, Non relational – Tip podatkovne baze, ki ni enak relacijski bazi
ODBC – Open Database Connectivity – Odprta podatkovna povezljivost
OLTP – On-line transaction processing – Transakcijski sistemi
PDCA - Plan-Do-Chek-Act – Demingov krog nenehnega izboljševanja
RC – Računalniški center
RPO – Recovery Point Objective – Ciljna točka obnove podatkov

RS – Republika Slovenija
RTO – Recovery Time Objective – Ciljni čas vzpostavitve delovanja IT-sistema
SAN – Storage Area Network – Omrežje za shranjevanje podatkov
SLA – Service Level Agreement – Nivo zagotavljanja storitev
SUBP – Sistem za upravljanje baz podatkov
SWOT – Strenghts, Weaknesses, Opportunities and Threats – Analiza prednosti, slabosti in priložnosti, nevarnosti
TCO – Total Cost of Ownership – Skupni stroški lastništva
TSA – Tivoli System Automation – Rešitev za avtomatizacijo strežniških gruč
VLAN – Virtual local area network – Virtualno lokalno omrežje
XML – Extensible markup language – Razširljivi označevalni jezik

## Priloga 2: COBIT 5 procesi

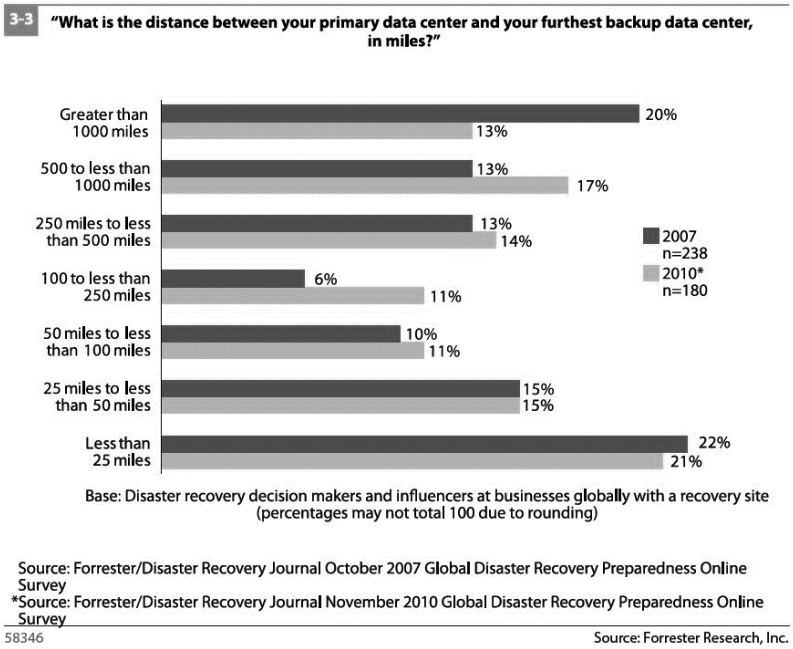
*Slika 1: Cobit 5 procesi*




*Vir: ISACA, COBIT 5: a business framework for the governance and management of enterprise IT, 2012a, str. 33, slika 16.*

## Priloga 3: Graf oddaljenosti med RC v miljah

*Slika 2: Oddaljenost med RC*




*Vir: R. Dines, The State of Disaster Recovery Preparedness, 2011, str. 16, slika 3-3.*

## Priloga 4: Dvajset ključnih načel pri zagotavljanju visoke razpoložljivosti sistemov

Marcus & Stern (2003, str. 75–104):

1. KISS princip
2. En problem, ena rešitev
3. Izraba zunanjih virov
4. Poenostavi konfiguracijo
5. Uporaba preverjene strojne opreme
6. Uporaba preverjene programske opreme
7. Dizajn, ki upošteva rast
8. Učenje iz zgodovine
9. Razdelitev okolij
10. Testiranje vsega
11. Planiranje
12. Uporaba SLA
13. Dokumentiranje
14. Uporaba principov nadzora nad spremembami
15. Ne pozabi na zmogljivost
16. Konsolidacija strežnikov
17. Varnost
18. Odprava kritične točke odpovedi
19. Vedno preveri vse
20. Ne varčuj po nepotrebnem

## Priloga 5: Magični kvadrant za operativne SUBP

*Slika3: Magični kvadrant za operativne SUBP*


*Vir: Gartner, Magic Quadrant for Operational Database Management Systems, 2015, slika 1.*

## Priloga 6: HADR-konfiguracijske nastavitve

Primarna baza:

```
HADR_LOCAL_HOST          host1
HADR_LOCAL_SVC          port1
HADR_REMOTE_HOST         host2
HADR_REMOTE_SVC          port2
HADR_REMOTE_INST         dbinst1
HADR_TIMEOUT            120
HADR_SYNCMODE            NEARSYNC
HADR_PEER_WINDOW         120
```

Sekundarna - standby baza:

```
HADR_LOCAL_HOST          host2
HADR_LOCAL_SVC          port2
HADR_REMOTE_HOST         host1
HADR_REMOTE_SVC          port1
HADR_REMOTE_INST         dbinst1
HADR_TIMEOUT            120
HADR_SYNCMODE            NEARSYNC
HADR_PEER_WINDOW         120
```

**Priloga 7: Nadzorni ukazi in pregled statusa HADR-povezave**

- ukaz db2pd –hadr

Ukaz nam pokaže status povezave, v tem primeru na primarni lokaciji (Role=Primary), iz rezultata se vidi status HADR-povezave (State=Peer), tip sinhronizacije in status povezave med bazama (ConnectStatus=Connected).

```
HADR Information:
Role    State          SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Primary Peer           Nearsync 0                1201

ConnectStatus ConnectTime                    Timeout
Connected     Wed May 25 09:07:36 2016 (1464160056) 120

LocalHost                    LocalService
SERVR1                       XXXX1

RemoteHost                   RemoteService      RemoteInstance
SERVR2                       XXXX2              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0017503.LOG 7133       0x000000BB1A39D549

StandByFile  StandByPg  StandByLSN
S0017503.LOG 7133       0x000000BB1A39D237
```

Enako je na sekundarni lokaciji:

```
HADR Information:
Role    State          SyncMode   HeartBeatsMissed   LogGapRunAvg (bytes)
Standby Peer           Nearsync 0                479

ConnectStatus ConnectTime                    Timeout
Connected     Wed May 25 09:07:36 2016 (1464160056) 120

LocalHost                    LocalService
SERVR2                       XXXX2

RemoteHost                   RemoteService      RemoteInstance
SERVR1                       XXXX1              db2inst1

PrimaryFile  PrimaryPg  PrimaryLSN
S0017503.LOG 7134       0x000000BB1A39E045

StandByFile  StandByPg  StandByLSN         StandByRcvBufUsed
S0017503.LOG 7133       0x000000BB1A39DFC5 0%
```

- ukaz db2 get snapshot for db XXXX

```
HADR Status
  Role                = Primary
  State               = Peer
  Synchronization mode   = Nearsync
  Connection status      = Connected, 25.05.2016 09:07:36.292607
  Heartbeats missed      = 0
  Local host          = SERVR1
```

```
  Local service          = XXXX1
  Remote host            = SERVR2
  Remote service         = XXXX2
  Remote instance        = db2inst1
  timeout(seconds)       = 120
  Primary log position(file, page, LSN) = S0017503.LOG, 7139, 000000BB1A3A3A62
  Standby log position(file, page, LSN) = S0017503.LOG, 7139, 000000BB1A3A37F8
  Log gap running average(bytes) = 765

HADR Status
  Role                 = Standby
  State                = Peer
  Synchronization mode   = Nearsync
  Connection status      = Connected, 25.05.2016 09:07:36.292659
  Heartbeats missed      = 0
  Local host             = SERVR2
  Local service          = XXXX2
  Remote host            = SERVR1
  Remote service         = XXXX1
  Remote instance        = db2inst1
  timeout(seconds)       = 120
  Primary log position(file, page, LSN) = S0017503.LOG, 7138, 000000BB1A3A2B73
  Standby log position(file, page, LSN) = S0017503.LOG, 7138, 000000BB1A3A29AB
  Log gap running average(bytes) = 564
```

- ukaz db2top

```
            Primary                      Standby
  ---------- ------------------------------ ------------------------------
host       SERVR1                         SERVR2
  Service    XXXX1                          XXXX2
  Instance   DB2INST1                       db2inst1
  Logfile    S0017503.LOG                   S0017503.LOG
   Log PAGE   7140                          7140
   Log LSN    1A3A4BF7                       1A3A4999
```

## Priloga 8: Izsek skripte za nadzor DB2 HADR

```
# HADR Status – izpis s db2pd komando
      db2pd -db $1 -HADR > $LOG

# izpis statusa iz log fileja
      ROLE1=$(cat $LOG |grep "Primary "| head -c -n 1)"S"
      ROLE=$(echo $ROLE1 |head -c -n1)

#glede na status HADR se izvedejo nadaljnje akcije
      if [ $ROLE = "P" ];
            then
# preveri se delovanje hadr
        STATE=$(cat $LOGDB | grep "Primary "| head -c -n 21|sed 's/Primary //')…
…
…
# preveri status hadr-a
      if [ $STATE != "Peer" ]; then
  # poslji po mailu obvestilo da HADR ne dela
      echo "$DATUM: HADR ne dela  $STATE
…
```

## Priloga 9: Skupine ocen po posameznem kriteriju ocenjevanja

### K1 - Hitrost obnove delovanja sistema ob nesrečah

| Ocena | Opis ocene |
|---|---|
| 1 | več kot teden |
| 2 | več kot dan |
| 3 | v 1 dnevu |
| 4 | od 2 do 4 ure |
| 5 | manj kot 2 uri |

### K2 - Kompleksnost uvedbe rešitve

| Ocena | Opis ocene |
|---|---|
| 1 | Kompleksna, postavitev ni možna brez zunanje pomoči |
| 2 | Kompleksna, zahteva izobraževanje DBA in sistemskih administratorjev ter pomoč pri uvedbi s strani zunanjih svetovalcev ter natančen načrt izvedbe na sistemskem in baznem nivoju |
| 3 | Sprejemljiva, zahteva natančno načrtovanje rešitve, sodelovanje med sistemskimi administratorji in DBA-ji zaželeno |
| 4 | Lahka, lahko jo izvede izkušen sistemski administrator ali DBA |
| 5 | Lahka, izvede jo lahko sistemski administrator ali DBA z manj izkušnjami |

### K3 – Kompleksnost vzdrževanja rešitve

| Ocena | Opis ocene |
|---|---|
| 1 | Spremembe na bazi (DDL) vplivajo na delovanje rešitve in zahtevajo poseg DBA, poseg je kompleksen, prekinitev v delovanju |
| 2 | Spremembe na bazi (DDL) vplivajo na delovanje rešitve in zahtevajo poseg DBA, poseg je manj kompleksen, prekinitev v delovanju |
| 3 | Spremembe na bazi (DDL) vplivajo na delovanje rešitve in zahtevajo poseg DBA, poseg ni zahteven, prekinitev v delovanju |
| 4 | Spremembe na bazi (DDL) ne vplivajo na delovanje rešitve in ne zahtevajo posega DBA, nekatere spremembe niso podprte v celoti |
| 5 | Spremembe na bazi (DDL) ne vplivajo na delovanje rešitve in ne zahtevajo posega DBA |

### K4 – Kompleksnost nadzora rešitve

| Ocena | Opis ocene |
|---|---|
| 1 | Potreben stalen nadzor, ni uporabnih orodij |
| 2 | Potreben stalen nadzor, orodja so na voljo |
| 3 | Potreben občasni nadzor, možna avtomatizacija |
| 4 | Potreben občasni nadzor, skripte za avtomatizacijo že vključene pri namestitvi rešitve |
| 5 | Preverjanje alarmov, rešitev je v celoti avtomatizirana ali pa je rešitev aktivna na obeh vozliščih |

## K5 – Vpliv na HA-aplikacije ob preklopu vozlišč podatkovne baze

| Ocena | Opis ocene |
|---|---|
| 1 | Preklop vozlišč pomeni prekinitev v delovanju aplikacije za več kot 1 uro |
| 2 | Preklop vozlišč pomeni prekinitev v delovanju aplikacije za več kot 5 minut in manj kot 1 uro |
| 3 | Preklop vozlišč pomeni prekinitev v delovanju aplikacije za več kot 1 minuto, manj kot 5 minut |
| 4 | Preklop vozlišč pomeni prekinitev v delovanju aplikacije za največ 1 minuto |
| 5 | Ni prekinitve |

## K6 – Strošek uvedbe rešitve

| Ocena | Opis ocene |
|---|---|
| 1 | Velika investicija > 100.000 EUR, investicija v strojno, programsko opremo, najem zunanjih storitev |
| 2 | Investicija od 50.000 do 100.000 EUR |
| 3 | Investicija od 25.000 do 50.0000 EUR |
| 4 | Investicija do 25.000 EUR |
| 5 | Ni potrebne dodatne investicije |