# Poslovne programske aplikacije i migracije baza podataka

**Špicar, Mario**

**Undergraduate thesis / Završni rad**

**2019**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:211:368980

*Rights / Prava:* Attribution 3.0 Unported/Imenovanje 3.0

*Download date / Datum preuzimanja:* **2024-03-28**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

*Repository / Repozitorij:*

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE**

**V A R A Ž D I N**

**Mario Špicar**

# POSLOVNE PROGRAMSKE APLIKACIJE I MIGRACIJE BAZA PODATAKA

**ZAVRŠNI RAD**

**Varaždin, 2019.**

**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE**

**V A R A Ž D I N**

**Mario Špicar**

**Matični broj: 44742/16-I**

**Studij: Primjena informacijske tehnologije u poslovanju**

# POSLOVNE PROGRAMSKE APLIKACIJE I MIGRACIJE BAZA PODATAKA

**ZAVRŠNI RAD**

**Mentor:**

Prof. dr. sc. Stjepan Vidačić

**Varaždin, studeni 2019.**

*Mario Špicar*

**Izjava o izvornosti**

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

## Sažetak

Baze i sustavi za upravljanje bazama podataka, kao temeljni resurs poslovnih programskih aplikacije kontinuirano se razvijaju, te se povremeno javlja potreba za nadogradnjom i migracijom baza podataka na višu razinu upravljačkog sustava, prilikom čega, ukoliko se postupak migracije ne provede na odgovarajući način, postoji mogućnost ugrožavanja stabilnosti rada pojedinih poslovnih programskih aplikacija ili poslovnog sustava u cjelini.

Razmatranjem predmetne problematike, sadržaj ovog rada usmjeren je u tri pravca. Prvi cilj je pojasniti predmetnu problematiku kroz odnose između poslovnih programskih aplikacija i baza podataka, te kroz okolnosti kod kojih dolazi do potrebe za nadogradnjama baza ili sustava za upravljanje bazama podataka. Drugi cilj je definirati metodički postupak kojeg bi se pri rješavanju predmetnog problema nadogradnji baza i sustava za upravljanje bazama podataka (npr. SQL Server) trebalo pridržavati i treći cilj, potvrđivanje ispravnosti metodičkog postupka na praktičnom primjeru nadogradnje baze podataka na višu razinu upravljačkog sustava Microsoft SQL Server.

Na temelju primjene metodike i rezultata uspješno provedene praktične nadogradnje baze podataka na višu razinu upravljačkog sustava Microsoft SQL Server, možemo konstatirati da su postavljeni ciljevi postignuti, te da je praktično potvrđena mogućnost kreiranja odgovarajuće metodike pomoću koje je moguće kontrolirati procese nadogradnje sustava i osigurati stabilnost poslovnih programskih aplikacija i poslovnog sustava u cjelini.



**Ključne riječi:** poslovne aplikacije, migracija podataka, baze podataka, MS SQL Server, informacijski sustav, nadogradnja sustava

# Sadržaj

Sadržaj ..... iii
1. Uvod ..... 1
2. Metode i tehnike rada ..... 3
3. Poslovne programske aplikacije i sustavi za upravljanje bazama podataka ..... 4
3.1. Okolnosti koje određuju odabir sustava za upravljanje bazama podataka ..... 8
3.1.1. Strateško promišljanje o arhitekturi informacijskog sustava ..... 9
3.1.2. Postojeće poslovne programske aplikacije i nabava novih ..... 10
3.1.3. Interni razvoj poslovnih programskih aplikacija ..... 10
3.1.4. Posjedovanje odgovarajućih vještina i znanja ..... 10
3.1.5. Trošak posjedovanja ..... 11
3.2. Utjecaj nadogradnje poslovnih programskih aplikacija na SUBP ..... 12
3.3. Utjecaj zamjene poslovnih programskih aplikacija na SUBP ..... 13
3.4. Utjecaj nadogradnje SUBP-a na poslovne programske aplikacije ..... 15
4. Microsoft SQL Server sustav za upravljanje bazama podataka ..... 16
4.1. Povijest razvoja ..... 16
4.2. Pregled verzija i međusobne kompatibilnosti ..... 21
4.3. Problemi nadogradnje baza podataka između različitih verzija MS SQL Server-a ..... 23
4.4. Metodologija nadogradnje MS SQL servera i korisničkih baza podataka ..... 26
4.5. Primjena metodologije na praktičnom primjeru ..... 31
5. Zaključak ..... 38
Popis literature ..... 39
Popis slika ..... 41
Popis tablica ..... 42

# 1. Uvod

Zahvaljujući sve većoj primjeni digitalnih tehnologija i inovacija u poslovanju, zadnjih nekoliko godina primjećuje se kontinuirano skraćivanja razdoblja u kojima poslovni sustavi moraju prilagođavati svoje poslovanje na novo nastale okolnosti.

U tim stalnim promjenama, koje nerijetko mijenjaju načine na koje radimo i poslujemo iz samog temelja, izgradnja kvalitetnog sustava upravljanja razvojem informacijskog sustava poduzeća nikada nije bila važnija nego što je to danas.

Dok s jedne strane primjena novih tehnologija donosi razne prednosti, poput mogućnosti razvoja potpuno novih proizvoda i usluga, veću kvalitetu proizvoda i pruženih usluga, veću ekonomičnost i efikasnost rada, veću prepoznatljivost i konkurentnost, s druge strane, takve promjene od poduzeća iziskuju i sve veća ulaganja koliko u samu tehnologiju toliko i u ljude, odnosno u izgradnju procesa upravljanja, razvoja, usklađivanja i održavanja vlastitih poslovnih aplikativnih i informatičko komunikacijski tehnologija (IKT).

Čak i u slučajevima kada poduzeća nemaju potrebu za čestim funkcionalnim nadogradnjama vlastitih programskih aplikativnih sustava, razni čimbenici iz okoline, poput digitalizacije lokalnih i upravnih tijela, te privatnih i poslovnih subjekata, zatim općeg rasta primjene internet tehnologija i računarstva u oblaku i sl. posredno utječu na potrebu za nadogradnjom sustava u funkcionalnom i sigurnosnom smislu.

Iako promjene obično zahvaćaju sve dijelove informacijskog sustava poduzeća u nekoj mjeri, poslovne aplikacije su ipak, kao osnovni dio informacijskog sustava poduzeća, a s njima u vezi i sustavi za upravljanje bazama podataka (*SUBP) (eng. Database Management System - DBMS*), kao ključna infrastrukturna komponenta poslovnih aplikacija, tim promjenama ipak najviše pogođeni.

Upravo zbog toga, poduzeća teže tome da pronađu načine i/ili konkretne metodike za nadogradnju kritičnih dijelova informacijskog sustava, kojima će osigurati da troškovi i resursi uključeni u te procese budu što niži, a da postupci promjene ili nadogradnje ključnih komponenti informacijskog sustava osiguravaju adekvatnu raspoloživost i sigurnost poslovnog sustava u cjelini.

Vodeći se tim razmatranjima i aktualnosti teme, cilj ovog rada je još jedan pokušaj u kreiranju smjernica i metodika čijom bi se primjenom poduzećima omogućilo da u postupcima nadogradnji i zamjene programskih poslovnih aplikacija, a posebno kod nadogradnji sustava za upravljanje bazama podataka postignu što bolje rezultate i smanje rizike od narušavanja stabilnosti vlastitog poslovnog sustava.

Zbog potrebe za ograničenjem opsega ovog rada u istom neće biti razmatrane metodike koje bi opisivale okolnosti i postupke zamjene postojećeg poslovnog programskog aplikacijskog sustava novim ili nadogradnje poslovnih programskih aplikacija, a što bi svakako mogla biti tema nekih budućih radova, već ćemo se isključivo fokusirati na problematiku nadogradnje baza podataka u slučajevima kada zbog raznih poslovnih i tehnoloških razloga dolazi do potrebe za nadogradnjom postojećeg SUBP-a na novu verziju, uz zadržavanje postojećih poslovnih programskih aplikacija i baza podataka.

Nakon uvoda i opisa metoda i tehnika rada, na početku treće cjeline opisat ćemo najčešće faktore koji utječu na odabir pojedinog sustava za upravljanje bazama podataka, a odmah u nastavku i kakvi su to sve odnosi između sustava za upravljanje bazama podataka i poslovnih programskih aplikacija tj. kako ti odnosi utječu na procese usklađenja i migracija baza podataka i SUBP-a.

U četvrtoj cjelini usredotočit ćemo se na sustav za upravljanje bazama podataka Microsoft SQL Server koji je jedan od vodećih relacijskih sustava za upravljanje bazama podataka. Na početku ćemo ukratko opisati povijest razvoja Microsoft SQL Server-a u zadnjih dvadesetak godina, a nakon toga i problematikom međusobne kompatibilnosti pojedinih verzija sustava i problemima koji se javljaju u postupcima nadogradnje korisničkih baza podataka na novije verzije SUBP-a.

U nastavku ćemo opisati, a kasnije i na konkretnom praktičnom primjeru pokazati metodiku optimalne nadogradnje SUBP-a, koja je nastala na temelju višegodišnjeg praktičnog iskustva autora na polju razvoja i održavanja poslovnih programskih aplikacija i baza podataka.

Na samom kraju, u petoj cjelini, kroz zaključak osvrnut ćemo na izloženu problematiku, na problemski zadatak, metodiku i postignute rezultate, a s obzirom na to što će rad detaljnije obuhvatiti samo dio problematike koja je navedena u uvodu, dodatno ćemo spomenuti i neke teme koje bi mogle biti predmet daljnjih istraživanja.

# 2. Metode i tehnike rada

Prilikom izrade ovog rada, korištene su metode analize stručnih i znanstvenih članaka i literature iz domene upravljanja poslovnim informacijskim sustavima i bazama podataka, te su posebno analizirani primjeri iz prakse kod kojih su rađene nadogradnje poslovnih aplikacija, IKT infrastrukture, sustava za upravljanje bazama podataka i samih baza podataka.

Uz stručnu literaturu, korišteni su i dodatni materijali nastali tijekom provođenja praktičnog testiranja nadogradnji baza podataka, a zaključci koji su izneseni, dobiveni su sintezom rezultata tih analiza i istraživanja, odnosno na temelju sinteze iskustava autora u domeni održavanja složenih informacijskih sustava.

Uz već navedene načine prikupljanja podataka, istraživačke aktivnosti provedene su i sekundarnim istraživanjem relevantnih domaćih i inozemnih članaka i studija, te javno dostupnih izvora sa web stranica navedenih u literaturi.

Za referenciranje i uređivanje svih izvora koji su iskorišteni u ovom radu, upotrijebljen je programski alat Zotero.

# 3. Poslovne programske aplikacije i sustavi za upravljanje bazama podataka

Na isti način kao što poslovne programske aplikacije tako i sustavi za upravljanje bazama podataka, tijekom vremena, prolaze kroz vlastite razvojne cikluse, te se kontinuirano popravljaju i nadograđuju.

Osim isporuka koje se odnose na otklanjanje identificiranih funkcionalnih ili sigurnosnih propusta i grešaka, novije verzije sustava za upravljanje bazama podataka (SUBP) praktički uvijek obuhvaćaju i uvode nove funkcionalnosti, a ponekad dolazi i do značajnih promjena ili potpunog ukidanja nekih postojećih standardnih funkcionalnosti. Prema tome ukoliko poslovna aplikacija ovisi o određenoj funkciji SUBP-a, koja je u novijoj verziji promijenjena ili ukinuta, bez pravovremene detekcije i rješavanja nastalog problema, stabilnost određene poslovne aplikacije, pa i cijelog informacijskog sustava nakon nadogradnje SUBP-a može biti narušena.

Bez obzira da li se radi o nadogradnji SUBP-a radi postizanja određene razine funkcionalnosti, kompatibilnosti, sigurnosti ili neke druge bitne značajke ili se radi o potrebi nadogradnje ili zamjene poslovne aplikacije koja ovisi o određenoj verziji baze podataka, kod svih ovih nabrojenih slučaja vrijedi jednako pravilo, a to je da se uz aktivnosti nadogradnje aplikativnih sustava ili sustava za upravljanje bazama podataka, mora voditi briga i o međusobnoj kompatibilnosti i funkcionalnosti, jer samo na taj način moguće je osigurati adekvatnu dostupnost, integritet i sigurnost poslovnog sustava.

U slučajevima kada poduzeće provodi zamjenu poslovne programske aplikacije tada u projektu uz sve ostale aktivnosti poput analize razlika, edukacije korisnika, kreiranje uputa i procedura za rad, pripreme IKT infrastrukture, parametrizacije i prilagodbe određenih funkcija novog sustava, testiranja sustava i sl. obavezno mora pripremiti i provesti i aktivnosti migracije podataka, bez čega realno nije ni moguć normalan nastavak poslovanja.

Kod gotovo svih slučajeva koji obuhvaćaju neki oblik migracije poslovnih programskih aplikacija i baze podataka, prvi korak je uvijek razmatranje optimalne strategije migracije, a tek nakon toga kreiranje projektnog plana, raspisivanje faza i aktivnosti samog projekta [1].

Odabir strategije migracije nije jednostavan, jer na istu utječu brojni parametri od kojih možemo nabrojati neke poput: potencijalnog troška migracije, mogućeg trajanja projekta, dostupnosti internih resursa, kompleksnosti izvornog i odredišnog sustava, kompleksnosti i broja vanjskih sustava s kojima sustav komunicira, te pitanja, da li je moguće osigurati razmjenu podataka u realnom vremenu ili ne, da li postoje neka interna ili eksterna ograničenja, kakvi su pravni rizici i načini regulacije odnosa između dobavljača i pružatelja usluga, količina

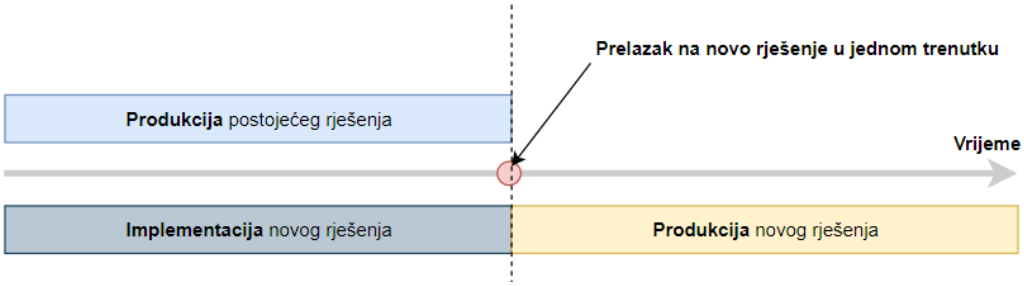
privremenih sučelja koje bi bilo potrebno razviti ukoliko bi se koristile određene metode migracije, koji bi bio trošak i rizik svake od mogućih varijanti itd.

U radu „*Method of data migration from one ERP system to another in real time*”, autora Vidačić, Pihir i Fabac [2], navode se tri najčešća oblika implementacije poslovnih programskih aplikacija i migracije podataka, a to su: „Big bang“ načelo, „Phased Rollout“ načelo i „Parallel Adoption“ načelo.

Kao što sam naziv sugerira, „Big bang“ načelo podrazumijeva cjeloviti prelazak na novi sustav sa točno definiranim trenutkom prelaska, „Phased Rollout“ podrazumijeva postupno uvođenje novih funkcionalnih modula sve dok se ne postigne cjelovita zamjena sustava, te „Parallel Adoption“ podrazumijeva uvođenje novog sustava kod kojeg se u određenom razdoblju paralelno koristi stari i novi sustav.

„**Big bang**“ kao načelo prelaska na novi sustav, koji je prikazan na slici 1. često se smatra najrizičnijim s obzirom na to što u slučaju javljanja neke veće greške često ne postoji jednostavan način za vraćanje na staro, migracija i postupci provjere migracije podataka složeni su, a prelazak pretpostavlja zaustavljanje informacijskog sustava poduzeća u kraćem razdoblju od početka do završetka aktivnosti migracije. Usprkos tome, kvalitetnom pripremom projekta od samog početka cjeloviti prelazak ipak donosi niz prednosti.

S obzirom da postoji usredotočenost na samo jedan cilj, ukupno trajanje uvođenja novog sustava ovom metodom obično je kraće u odnosu na druge metode, troškovi uvođenja su niži, pravne i regulatorne nesigurnosti zbog kraćeg trajanja također manje dolaze do izražaja, nema potrebe za dodatnom IKT infrastrukturom, a istovremeno je potrebno i manje dodatnog razvoja, pošto nema potrebe za razvojem dodatnih sučelja. Bitno je naglasiti da se time izbjegava i dodatna složenost u upravljanju informacijskim sustavom poduzeća, jer što je projekt duži sve više promjena izvan projekta i poslovnih potreba generiranih unutar poduzeća počinju negativno utjecati i na sam tijek projekta.



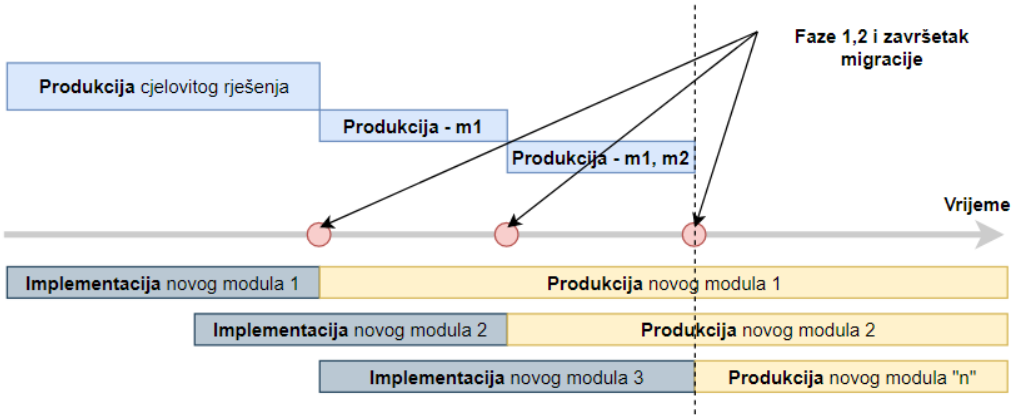

Slika 1. „Big bang“ strategija migracije i prelaska na novi sustav (izrada autora)

„**Phased rollout**“ kako je prikazano na slici 2. opisuje način prelaska sa starog sustava u fazama i to obično kod složenih sustava kod kojih se mogu izdvojiti funkcionalni podsustavi koji se mogu zamijeniti novim programskim podsustavima, čime se ujedno smanjuje koncentracija rizika u odnosu na cjelovitu zamjenu sustava.

U odnosu na cjeloviti prelazak, mogućnost modularne implementacije i uvođenja, smanjuje ukupne rizike implementacije, međutim za ovakav način uvođenja mora postojati ispunjen preduvjet, a to je adekvatna modularna struktura starog i novog sustava.

Ukoliko se radi o visoko integriranom sustavu prelazak u fazama bio bi nemoguć bez dodatnog razvoja i intenzivnog testiranja raznih sučelja čija svrha bi bila premostiti i povezati pojedine dijelove starog i novog sustava, čime bi umjesto smanjenja rizika i troškova došlo do povećanja složnosti sustava, a time bi se generirali upravo suprotni efekti od onih željenih.

Prema tome za uvođenje u fazama, najbolji su kandidati oni sustavi kod kojih je novi i stari sustav modularno izgrađen, te kada prilikom zamjene pojedinog modula/aplikacije ne dolazi do bitnog narušavanja vanjskih tokova podataka, odnosno kada već postoji mogućnost integracije putem standardnih sučelja.

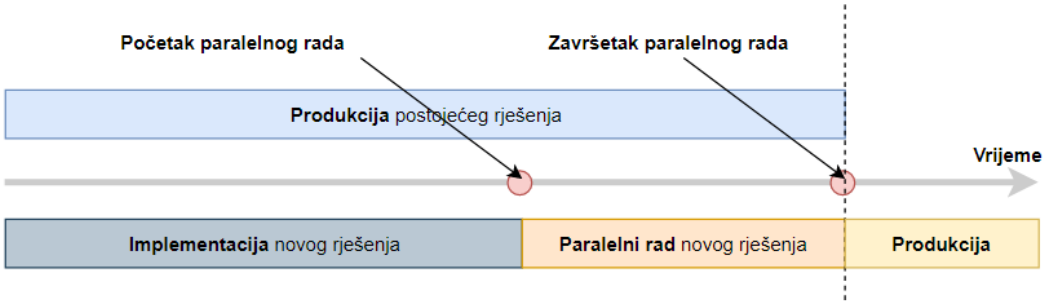



Slika 2. „Phased rollout“ strategija migracije i prelaska na novi sustav (izrada autora)

„**Parallel Adoption**“ kako je prikazano na slici 3. podrazumijeva uvođenje novog sustava na način, da se u određenom razdoblju omogućava paralelno korištenje starog i novog sustava, pa se tako korisnici mogu postepeno obučavati i prilagođavati na novi sustav, a u tom razdoblju mogu ga testirati i dodatno prilagođavati. Nakon završetka prilagodbe i provjere ispravnosti novog sustava, stari sustav se jednostavno gasi čime završava proces migracije.

Za uspješnu primjenu ove strategije izrazito je važno osigurati adekvatnu razmjenu podataka između oba sustava. Najčešći način prijenosa podataka je putem specijaliziranih

aplikacija ili skripti koje iz postojeće baze podataka podatke prenose u novu bazu podataka automatizmom [2], međutim moguće su i situacije, kada se radi o manjoj i ograničenoj količini podataka da se podaci prenose ručno u oba sustava, odmah ili sa kraćim vremenskim odmakom.




Slika 3. „Parallel adoption“ strategija migracije i prelaska na novi sustav (izrada autora)

Upravo ovaj tip strategije i metodika paralelne migracije između dva ERP sustava u realnom vremenu uspješno je upotrijebljena i detaljno opisana u radu autora Vidačić, Pihir i Fabac [2].

Iako se te tri strategije obično razmatraju u kontekstu zamjene poslovnih programskih aplikacija i sustava, primjenjive su i na projektima nadogradnje ili zamjene sustava za upravljanje bazama podataka koje mogu, ali i ne moraju istovremeno biti dio istog projekta.

U slučajevima kada se radi samo o nadogradnji sustava za upravljanje bazom podataka na višu verziju i ukoliko se na istom poslužitelju nalazi više baza podataka zasebnih poslovnih programskih aplikacija, a koje su međusobno kompatibilne sa novom verzijom SUBP-a, nadogradnja može biti obavljena u jednom koraku za sve baze podataka istovremeno, čime su tom jednom akcijom (promjenom) pogođene sve povezane poslovne programske aplikacije, ali isto tako moguće je, ukoliko postoji opravdan poslovni razlog, nadogradnju SUBP-a provesti postepeno provođenjem migracije pojedinih baza podataka na nove instance SUBP-a, te kao treća mogućnost, moguće je upotrijebiti i strategiju paralelne migracije uz uvođenje dodatnog (paralelnog) poslužitelja ili instance baze podataka pomoću koje se osigurava paralelni rad i testiranje ispravnosti rada poslovne programske aplikacije na novoj nadograđenoj bazi podataka odnosno sustavu za upravljanje bazama podataka.

Pošto svaka od opisanih strategija ima svoje prednosti i nedostatke, važno je da se odmah u samom početku razmatranja projekta detaljno analiziraju sve okolnosti, pozitivni i negativni učinci i da se tek tada donesu konačne odluke o daljnjem tijeku migracijskih aktivnosti.

Kako je već spomenuto, odluka o odabranoj strategiji migracije donosi se prilikom faze planiranja projekta, prilikom čega se razmatraju brojni parametri poput razmatranja trenutne i buduće arhitekture i međusobne povezanosti poslovnih programskih aplikacija, da li sve poslovne programske aplikacije omogućavaju nadogradnju na višu verziju SUBP sustava, da li postoji mogućnost migracije u fazama, koje je maksimalno, a koje očekivano trajanje migracije za pojedinu opciju, koje su mogućnosti zaštite podataka i konfiguracija poslužitelja za slučaj ukoliko migracija ne bude uspješna, o kojim se sustavima za upravljanje bazama podataka radi (npr. Microsoft SQL Server, Oracle, Infomix, DB2, MySQL, Postgres, itd.) i koje su im tehničke mogućnosti, kakve su politike licenciranja, kakve su mogućnosti povezivanja, kakvi su zahtjevi za hardverskim resursima i sl.

Zbog velikog broja čimbenika koji mogu utjecati na proces migracije, na isti se ne smije gledati kao na jednostavan tehnički posao koji je ograničen na administratore baze podataka, već se u proces migracije mora uključiti veći broj raznih stručnjaka i to, od krajnjih korisnika, poslovnih analitičara, projektanata, programera, sistemskih inženjera pa sve do administratora baza podataka i drugih stručnjaka.

Svaki pothvat nadogradnje ili zamjene poslovnih programskih aplikacija i/ili sustava za upravljanje bazama podataka mora biti adekvatno isplaniran i proveden, a način organizacije provođenja tih aktivnosti dakako da ovisi o složenosti procesa i sustava koji su u taj proces uključeni.

Prema tome, aktivnosti migracije mogu biti organizirane kao projekt i dio projektnog programa npr. zamjene integriranog poslovnog informacijskog sustava, zatim mogu biti organizirane u sklopu projekta zamjene određene poslovne programske aplikacije ili mogu biti organizirane kao potpuno samostalni projekt iniciran iz procesa redovnog održavanja informatičke infrastrukture.

## 3.1. Okolnosti koje određuju odabir sustava za upravljanje bazama podataka

Kako bi se osigurao optimalan rad, prema razvoju i održavanju informacijskog sustava mora se pristupiti planski i stručno, pa prema tome, promatrajući različite poslovne sustave, ovisno o razini postignute informatizacije i digitalizacije, u sklopu vlastite poslovne strategije ili paralelno, gotovo uvijek definira se i odgovarajuća IT strategija.

Koristi od definiranja IT strategije su višestruke i to ne samo kako bi se kreirali srednjoročni i/ili dugoročni planovi vezanih uz primjenu IKT-a, već i kako bi se preciznije definirale smjernice i okviri financijskog proračuna vezanog uz investicije i troškove, potrebni

ljudski i materijalni resursi, ciljevi vezani uz sigurnost sustava, te ključne tehnologije i standardi koje je u ostvarivanju poslovne strategije potrebno uspostaviti.

Posebno važna kategorija za svako poduzeće su podaci i informacije, odnosno načini na kojima se upravljanja njima. Pošto količina podataka kontinuirano raste, poduzeća za iste moraju osigurati adekvatnu dostupnost, integritet, sigurnost i trajnost i to kroz sve duža vremenska razdoblja, te iz toga možemo zaključiti da odabir i korištenje određene tehnologije za upravljanje bazama podataka, definitivno spada u kontekst strateških odluka poduzeća.

S obzirom na praktična iskustva autora u definiranju strategije razvoja informacijskog sustava za institucije financijskog sektora, u nastavku je izdvojeno nekoliko kriterija za koje autor smatra da u odnosnu na ostale faktore u većoj mjeri utječu na odabir i standardizaciju sustava za upravljanje bazama podataka, a to su:

- Strateško promišljanje o arhitekturi informacijskog sustava,
- Postojeće poslovne programske aplikacije i nabava novih,
- Interni razvoj poslovnih programskih aplikacija,
- Posjedovanje odgovarajućih vještina i znanja,
- Trošak posjedovanja (*eng. Total Cost of Ownership - TCO*).

### 3.1.1. Strateško promišljanje o arhitekturi informacijskog sustava

Prilikom strateškog promišljanja vezanog uz arhitekturu sustava, informacijski sustav promatramo u cjelini i pokušavamo detektirati koji od tih dijelova nisu optimalni, odnosno koji se mogu nadograditi ili zamijeniti novim, a da se pri tome postigne manja složenost ukupnog poslovnog programskog aplikativnog sustava i pripadajuće IKT infrastrukture.

Ukoliko razmatramo uvođenje novog SUBP-a, iz strateške perspektive postavljaju se pitanja, da li poduzeće već ima u upotrebi neki specifični SUBP ili više njih, na koji način se koriste pojedini SUBP-ovi, da li je moguće iskorištavanje postojećih resursa? Posebno je interesantno analizirati, da li postoji mogućnost optimizacije postojećeg stanja, da li je moguće putem projekata zamjene pojedinih aplikativnih rješenja ili migracija baza podataka smanjiti broj raznovrsnih SUBP-a na onu mjeru koja ne narušava funkcioniranje sustava, a postiže željene rezultate?

S obzirom na razvoj tehnologije i mogućnosti koje se u današnje vrijeme nude, a tu se posebno ističu sustavi temeljeni na računarstvu u oblaku i virtualizacija IKT infrastrukture, umjesto dugotrajnih projekata optimizacije postojećeg sustava, ponekad je znatno lakše u

potpunosti preći na novi model upravljanja IKT infrastrukturom i na taj način uz visoku fleksibilnost korištenja hardverske i softverske infrastrukture, ograničiti troškove samo na kapacitete koje poduzeće stvarno u tom trenutku treba.

### 3.1.2. Postojeće poslovne programske aplikacije i nabava novih

Kada poduzeće koristi i posjeduje određeni sustav za upravljanje bazom podataka, sasvim je logično da već kod odabira novih programskih rješenja analizira arhitekturu tih novih rješenja i pažljivo odabire ona rješenja koja ne narušavaju harmonizaciju postojećeg sustava.

Naravno, postoje izuzeci kod kojih zbog nekog razloga poduzeće ne može jednostavno odabrati sustav koji bi se uklopio u postojeću IKT arhitekturu, međutim takve slučajeve treba pokušavati izbjegavati, jer kratkoročni dobitci vezani uz rješavanja nekog specifičnog poslovnog zahtjeva mogu dugoročno rezultirati raznim neplaniranim situacijama i skrivenim troškovima razvoja, održavanja, licenciranja, integracije itd.

### 3.1.3.Interni razvoj poslovnih programskih aplikacija

U slučajevima kada se poslovne programske aplikacije razvijaju manje ili pretežito interno, preporučljivo je da se razvoj provodi na temelju unaprijed definiranih standarda i dogovorene arhitekture kod koje je SUBP samo jedna od tih ključnih standardiziranih komponenti.

Pri tome, standardizacijom je moguće postići brojne pozitivne efekte npr. poput boljeg iskorištavanja postojećih znanja organizacije, lakše povezivanje između pojedinih poslovnih programskih aplikativnih sustava, smanjenje operativnih rizika, smanjenje opsega sigurnosnih ranjivosti, smanjenje ukupnih troškova i sl.

Čak i u situacijama kada poduzeća koriste pretežito rješenja vanjskih dobavljača, posjedovanje internog razvoja, a time i određenih znanja i vještina vezanih uz određenu vrstu SUBP-a, predstavlja jedan od značajnih parametara prilikom definiranja IT strategije, standardizacije i odabira određenog SUBP-a.

### 3.1.4. Posjedovanje odgovarajućih vještina i znanja

U listu bitnijih parametara prilikom odabira SUBP-a, možemo uvrstiti i posjedovanje odgovarajućih znanja i vještina, pošto svako poduzeće mora voditi računa o tome da li posjeduje djelatnike s odgovarajućim znanjima, i vještinama i da li će morati zapošljavati nove ili educirati postojeće djelatnike.

Ukoliko poduzeće ne može osigurati adekvatnu podršku unutar poduzeća, tada je naravno jedina opcija ugovaranje podrške u vidu eksternalizacije tih aktivnosti od strane tvrtki

koje se baze razvojem i održavanjem informacijskih sustava i koje posjeduju odgovarajuće educirane i certificirane djelatnike za tehnologiju koju poduzeće koristi.

Eksternalizacija administracije baza podataka može izgledati kao veoma jednostavno rješenje, međutim u praksi to može postati veoma složen problem, posebno kada se na istom poslužitelju nalazi više baza podataka različitih dobavljača poslovnih programskih aplikacija.

U takvim i sličnim složenijim slučajevima kada je potrebna visoka razina sigurnosti i dostupnosti, preporučljivo je odvajanje poslova administracije i podrške i razvoja, čime se postiže da sve promjene na produkcijskim sustavima budu provedene tako da se poštuju parametri raspoloživosti i sigurnosti sustava.

Organizaciju i način održavanja pojedinih sustava ne možemo generalizirati, međutim možemo konstatirati, da organizacija održavanja prati složenost poslovnog sustava, pa prema tome složeniji poslovni sustavi koji zahtijevaju visoku raspoloživost i sigurnost poput financijskih institucija i drugih velikih poslovnih sustava, teže prema upravljanju iznutra, dok je kod manjih poslovnih sustava (koji nemaju vlastita znanja i vještine) upravljanje poslovnim programskim aplikacijama i bazama podataka prepušteno prvenstveno dobavljačima pojedinih poslovnih programskih aplikacija.

### 3.1.5. Trošak posjedovanja

Jedan od najbitnijih parametra, posebno za vlasnika i upravu svakog poduzeća, za upotrebu određene tehnologije su naravno troškovi uvođenja i korištenja.

Pri tome se posebno mora voditi računa o politici licenciranja, jer svaki od proizvođača sustava za upravljanjem bazama podataka na temelju vlastitih poslovnih politika definiraju opseg funkcionalnosti i ciljnu grupu korisnika, te na taj način prilagođavaju i definiraju svoje cijene. Posebno je potrebno voditi računa o tome da, bez obzira što za neke scenarije troškovi pojedine tehnologije mogu izgledati veoma niski, ukoliko poduzeće promijeni način korištenja sustava, ti troškovi mogu značajno rasti, pa čak i do te mjere da odabrana tehnologija u toj specifičnoj konfiguraciji bude neisplativa za poduzeće.

Zbog toga je potrebno uvijek voditi računa, koliko o trenutnim ulaganjima i troškovima licenciranja, edukacije i sl., toliko i o očekivanim ulaganjima i troškovima budućeg razdoblja.

Struktura tih troškova naravno ovisi o odabranom modelu implementacije SUBP-a, međutim općenito možemo ih podijeliti na tzv. „jednokratne troškove„ (kapitalne investicije) koje se obično amortiziraju tijekom određenog razdoblja korištenja i operativne troškove održavanja sustava.

Kako bi se dobila usporediva informacija o ukupnim troškovima pojedinog SUBP-a i izbjegli tzv. „skriveni“ troškovi, potrebno je prikupiti što više dostupnih informacija, a nakon toga sve identificirane troškove svesti na zajednički nazivnik, npr. na razinu mjesečnih troškova za unaprijed zadano razdoblje.

Pri tome, kapitalne investicije svode se na mjesečne troškove amortizacije zadanog razdoblja, dok se operativni i drugi troškovi prikazuju u dinamici u kojoj se pojavljuju.

U novije vrijeme, primjenom računarstva u oblaku, sve više su prisutni modeli najma IKT infrastrukture, kod kojih poduzeće nema potrebe ulagati u kapitalne investicije i brinuti se o raznim nepredvidivim situacijama poput neplaniranih troškove održavanja sustava, već takvi modeli poduzeću omogućavaju optimalno dimenzioniranje i plaćanje po načelu utrošenog, čime se uz veću fleksibilnost, preciznije planiranje operativnih troškova, mogu ostvariti i dodatne financijske uštede.

## 3.2. Utjecaj nadogradnje poslovnih programskih aplikacija na SUBP

Tijekom cijelog životnog ciklusa poslovnih programskih aplikacija, temeljem poslovnih, regulatornih, sigurnosnih i drugih vrsta zahtjeva kreiraju se zahtjevi za nadogradnjom, koji u određenoj mjeri obuhvaćaju baze i sustave za upravljanje bazama podataka.

Utjecaj tih nadogradnji, na baze podataka i sustave upravljanja bazama podataka uglavnom nisu značajne, osim u onim slučajevima kada se radi o implementaciji nekih sasvim novih funkcionalnosti SUBP-a koje zahtijevaju njegovu nadogradnju.

Razvojem poslovnih programskih aplikacija korištenjem modernih alata i upotrebe ORM okvira (*eng. Object Relational Mapping - ORM*) kojim se postiže direktno mapiranje entiteta baze podataka u programske klase, razvijene poslovne programske aplikacije postaju platformski neutralne u smislu određene vrste SUBP-a, pa je moguće korištenje gotovo bilo kojeg SUBP-a poput Oracle, MySQL, Microsoft SQL Server, Postgresql i drugih.

Razvoj programskih aplikacija putem ORM okvira pretpostavlja da je poslovna logika uglavnom implementirana na razini same aplikacije, pa se time izbjegava korištenje specifičnih funkcionalnosti određenog SUBP-a. U tim slučajevima umjesto korištenja ugrađenih funkcija, pohranjenih procedura, okidača, i drugih specifičnih funkcija baze podataka, gotovo sve operacije obavljaju se isključivo kreiranjem SQL upita direktno iz same aplikacije.

S druge strane, u slučajevima kada su potrebne najviše performanse i iskorištavanje najboljih karakteristika SUBP-a, dio poslovne logike koja je bliska bazi podataka smješta se u

samu bazu podataka, jer se time složene operacije nad podacima izvode na razini baze podataka, bez potrebe nepotrebnog prijenosa podataka između serverskih i klijentskih aplikacija. Međutim, takav pristup pridonosi većoj ovisnosti aplikativnog sustava i određenog sustava za upravljanje bazom podataka.

## 3.3. Utjecaj zamjene poslovnih programskih aplikacija na SUBP

U slučajevima kada ulaganje u postojeće poslovne programske aplikacije više ne može osigurati adekvatnu podršku poslovnim procesima ili u slučajevima kada poduzeće zbog promjene strategije mora značajnije mijenjati programsku podršku, pristupa se organizaciji projekta zamjene poslovnih programskih aplikacija.

Iako je razmatranje kriterija, mogućih pristupa i metodike odabira nove poslovne programske aplikacije odnosno poput ERP sustava (*eng. Enterprise Resource Planning - ERP*), CRM sustava (*eng. Customer Relationship Management - CRM*) ili nekog drugog sustava, van opsega ovog rada, završetkom faze odabira dolazi se do faze pokretanja projekta implementacije novog sustava, a time i problematike migracija baza podataka.

Prilikom implementacije sasvim novog rješenja, uz uvođenje novih funkcionalnosti, svako novo rješenje mora omogućiti podršku za trenutne poslovne zahtjeve i proizvode, tako da je već u fazi odabira budućeg rješenja veoma važno obratiti pozornost na sve tehničke preduvjete, tj. obratiti pozornost na razlike između postojećeg i novog sustava, kako na razini modela podataka, što je bitno kasnije za definiranje strategije migracije, tako i na razini procesa i funkcija, što je bitno za definiranje opsega potrebnih dorada novog sustava.

Od trenutka odabira do početka korištenja novog sustava, projekt obično polazi kroz niz faza, od kojih su u nastavku izdvojene sljedeće:

a) iniciranje projekta sa pripremom „knjige“ projekta,

b) odabir metode migracije podataka,

c) priprema IKT infrastrukture,

d) instalacija novog rješenja na okoline (npr. „test“, „migracija“, „produkcija“),

e) edukacija ključnih korisnika i parametrizacija novog rješenja,

f) analiza razlika funkcionalnosti u odnosu prema stvarnim potrebama i iniciranje zahtjeva za nadogradnjom,

g) kreiranje poslovnih pravila za migraciju podataka,

h) razvoj pomoćnih sustava i provođenje migracije podataka,

i) edukacija korisnika i korisničko testiranje,

j) kreiranje novih procedura i uputa za rad,

k) priprema za prelazak u produkciju,

l) prelazak u produkciju,

m) post-produkcijske aktivnosti stabilizacije sustava,

n) analiza („lessons-learned“) i zatvaranje projekta.

Kao što se može primijetiti iz nabrojenih grupa aktivnosti, pokretanje i provođenje zamjene poslovnih programskih aplikacija je sve samo ne jednostavan poduhvat, te su tijekom projekta mogući različiti scenariji.

Promatrajući utjecaje zamjene poslovnog programskog sustava u odnosu na sustav za upravljanje bazom podataka, mogu se javiti dva značajna problema i to:

a) problem kompatibilnosti između stare i nove baze podataka i postojećeg SUBP-a na razini sustava za upravljanje bazama podataka, npr. različiti proizvođači i proizvodi poput Microsoft SQL Server-a, MySQL-a, Oracle-a, Postgresql-a itd. i

b) problem logičke kompatibilnosti npr. korištenje različitih poslovnih modela i dizajna baza podataka ovisno o funkcionalnostima poslovne programske aplikacije.

Ukoliko bismo pokušali razmotriti još neke varijante utjecaja zamjene poslovnih programskih aplikacija (PPA) na SUBP, tada bismo mogli identificirati i sljedeće slučajeve:

a) nova PPA je kompatibilna sa postojećim SUBP, nova baza podataka može se implementirati na postojeći SUBP ili višu verziju istog SUBP-a, postoji mogućnost direktne nadogradnje postojeće baze podataka,

b) nova PPA je djelomično kompatibilna sa postojećim SUBP, potrebno je usklađenje baze podataka na trenutnu verziju SUBP-a,

c) kada za implementaciju nove PPA-e poduzeće mora implementirati novi SUBP različite verzije postojećeg ili nekog drugog proizvođača.

    a. varijanta u kojoj je potrebno nadograditi bazu podataka na novu verziju postojećeg SUBP-a,

    b. kada ne postoji mogućnost nadogradnje i preostaje samo migracija podataka iz postojećeg SUBP-a u novu bazu podataka i novi SUBP.

## 3.4. Utjecaj nadogradnje SUBP-a na poslovne programske aplikacije

Osim slučajeva kada postoji potreba za nadogradnjom ili zamjenom poslovnih programskih aplikacija, pojavljuju se i slučajevi kada postoji potreba nadogradnje sustava za upravljanje bazom podataka.

Ti slučajevi ovise o dinamici isporuke novih verzija sustava za upravljanje bazama podataka od strane proizvođača softvera tj. o dinamici izdavanja „zakrpa“ kojima se rješavaju određeni sigurnosni propusti i druge vrste grešaka softvera.

Za razliku od nadogradnji sigurnosnih propusta i detektiranih grešaka operacijskih sustava, a koje se najčešće provode kao dio standardnog procesa upravljanja zakrpama pod nadzorom sistemskih inženjera, nadogradnje sustava za upravljanje bazama podataka moraju biti provedene sa znatno većom pažnjom, jer iste mogu značajno utjecati na stabilnost poslovnih aplikacija.

Naravno, primjena automatskih sustava za nadogradnju operacijskih sustava mogu negativno utjecati na rad poslovnih programskih aplikacija, međutim rizik u takvim slučajevima uglavnom je niži u usporedbi sa rizicima nadogradnje SUBP-a.

Proces upravljanja promjenama, vodeći se najboljim praksama, poput ITIL-a (*eng. Information Technology Infrastructure Library – ITIL*), preporučuje uspostavu odgovarajućeg kontrolnog - testnog okruženja pomoću kojeg se sve promjene prvo testiraju, a tek nakon toga primjenjuju na produkcijsku okolinu, međutim za sustave upravljanja bazama podataka to nije dovoljno.

Analizirajući različite slučajeve u kojima dolazi do značajnijih problema, a povezani su sa nadogradnjama programskih aplikacija ili baza podataka, najozbiljniji problem predstavljaju slučajevi kada poduzeće dugoročno ne ulaže u razvoj poslovnih programskih aplikacija i ne prati trenutno stanje tehnologije, te se izlaže situaciji u kojoj raskorak između podržanih tehnologija poslovnih aplikacija i aktualnih verzija SUBP-a sve više raste.

U takvim slučajevima, kada nove verzije SUBP-a ne podržavaju rad tehnološki zaostalih i starijih poslovnih programskih aplikacija ili baza podataka, poduzeća se suočavaju sa izborom pokretanja projekta nadogradnje ili njihove zamjene.

Upravo je dio te problematike obrađen u sljedećim poglavljima ovog rada.

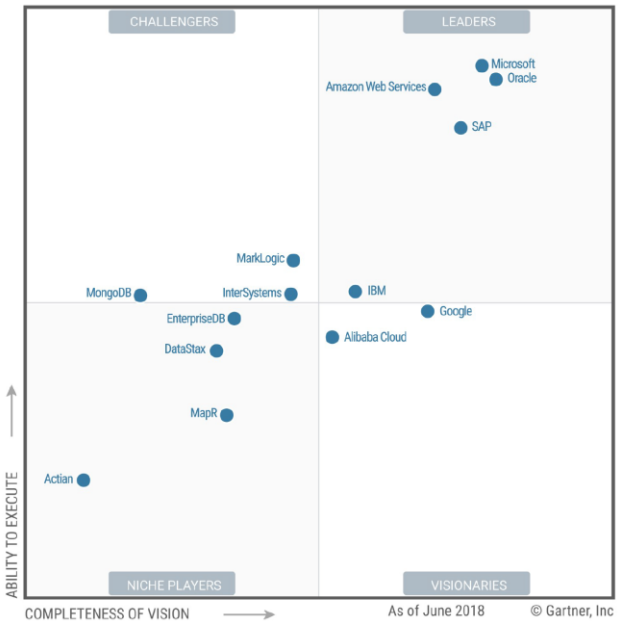
# 4. Microsoft SQL Server sustav za upravljanje bazama podataka

U prethodnoj cjelini opisane su situacije i problemi koji se mogu javiti prilikom upravljanja razvojem poslovnih programskih aplikacija i baza podataka, koji su univerzalni i neutralni iz pozicije sustava za upravljanje bazama podataka, a u nastavku ćemo se pozabaviti sa jednim od najčešćih i najkorištenijih sustava za upravljanje bazama podataka, a to je Microsoft SQL Server.

## 4.1. Povijest razvoja

Od samih početaka razvoja Microsoft SQL Server verzije 6.0 koja je obuhvaćala tek neke osnovne funkcionalnosti i kasnijih verzija 7.0 i 2000 koje su označile prekretnicu u razvoju proizvoda, Microsoft SQL Server danas, u verziji 2017 na platformama Windows, Linux i Cloud (Azure), uz podršku za cijeli niz najnaprednijih tehnologija postao je proizvod sa karakteristikama koje se traže prilikom najzahtjevnijih poslovnih primjena.

Prema Gartneru, usporedbe relacijskih i ne-relacijskih sustava za upravljanje bazama podataka (slika 4), Microsoft SQL Server je vodeći sustav za upravljanje bazama podataka prema mogućnostima izvršenja, a uz Oracle i prema obuhvatu i potpunosti vizije.




Slika 4. Gartnerov kvadrant operacijskih sustava za upravljanje bazama podataka [3]

Istovremeno, prema Gartneru (slika 5), Microsoft SQL Server je zadnjih dvanaest godina vodeća platforma za podatkovnu analizu i sustave poslovne inteligencije (*eng. Business Intelligence – BI*).

Iako Microsoft SQL Server u početcima nije smatran sustavom za upravljanje bazama podataka koji bi bio pogodan za ozbiljne poslovne primjene u transakcijski orijentiranim sustavima, gdje se očekivala visoka dostupnost i veća mogućnost iskorištenja hardverskih mogućnosti koje su tada uglavnom pružali samo sustavi koji su bili pogonjeni Unix operacijskim sustavima, poput IBM DB2, Oracle, Informix i sl., Microsoft SQL Server je zbog svoje jednostavnosti bio ipak veoma često korišten kao sustav za izgradnju sustava skladišta podataka i sustava poslovne inteligencije.




Slika 5. Gartnerov kvadrant platformi za analizu podataka i poslovnu inteligenciju [4]

Poput konkurentskih sustava i Microsoft SQL Server je također isporučivan u nekoliko različitih izdanja, sa različitim mogućnostima, ograničenjima i cijenom.

Iako je kroz vrijeme dolazilo do promjena u nazivima i namjenama pojedinih izdanja, trenutno su aktualna izdanja Microsoft SQL Servera 2019 (tablica 1.), „Enterprise“, „Standard“, „Developer“, „Express“ i zasebno „Web“ izdanje koje je namijenjeno za upotrebu u „hosting“ Internet web okruženjima [5].

Svako od izdanja namijenjeno je za određene poslovne scenarije, te su prema tome prilagođene funkcionalnosti, ograničenja i modeli licenciranja koji su prikazani na slici 6. u nastavku [5].

Tablica 1. Modeli licenciranja SQL Server 2019 [5]

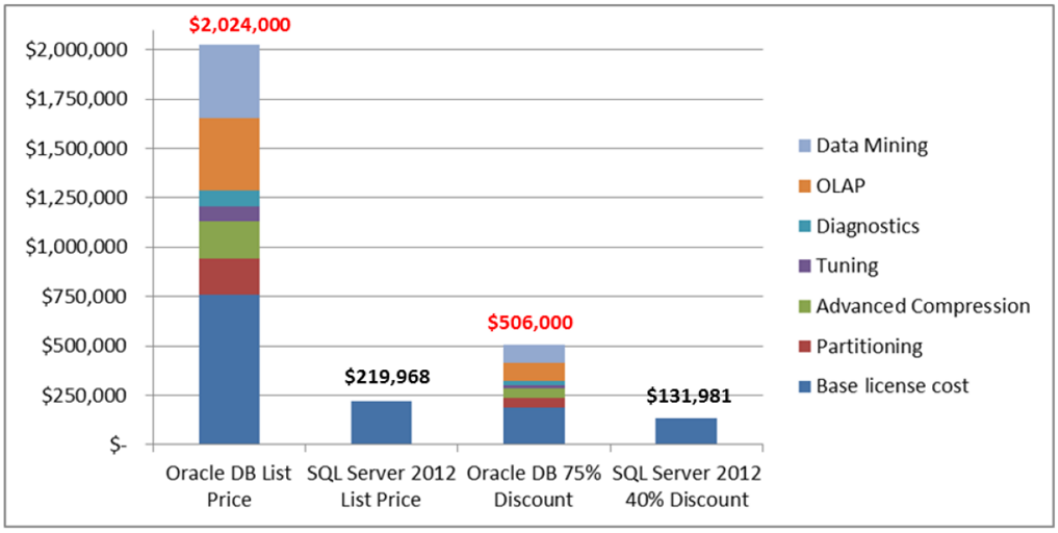
| SQL Server 2019 | Opcija načina licenciranja | |
|---|---|---|
| | Server + klijentska licenca (CAL) | Jezgra procesora (per core) |
| Enterprise | ● | ● |
| Standard | ● | ● |
| Developer | Besplatno | |
| Express | Besplatno | |

Pojašnjenja vezana uz način licenciranja pojedinih izdanja Microsoft SQL Servera mogu se naći na stranicama proizvođača [6], dok Vidačić i Špicar [7] navode da „u postupku odabira verzije i modela licenciranja SQL Servera vrlo je važno unaprijed provesti analize i pravilno procijeniti očekivana opterećenja sustava, koje su to nužne funkcionalnosti sustava za upravljanje bazom podataka, da li postoji potreba za visokom dostupnosti (eng. High Availability - HA), da li će se sustav koristiti u scenarijima kod kojih nije moguće jednoznačno utvrditi broj korisnika (npr. aplikacije dostupne sa Interneta), zatim da li se očekuje instalacija na operacijskom sustavu unutar fizičkog ili virtualnog računala (npr. da li postoji potreba odvajanja testne, pred-produkcijske i produkcijske okoline), te da li postoji potreba za izgradnjom fizički odvojene (sekundarne) lokacije sa aktivnom ili pasivnom replikacijom podataka radi omogućavanja brzog oporavka od katastrofe i dr.“.

U usporedbama između najjačih verzija proizvoda Microsoft SQL Server i Oracle-a, ukoliko ne ulazimo u neke specifične funkcionalnosti pojedinog proizvoda, vidljivo je da je ukupni trošak korištenja Microsoft SQL Server, za isti set grupa funkcionalnosti, značajnije niži, međutim ti troškovi mogu biti i još niži ukoliko se pravilno vodi računa o trenutnim i budućim potrebama poslovanja odabirom određene verzije, izdanja (verzije), modela licenciranja i odgovarajućih kapaciteta.

Na slikama (6,7) prikazana je usporedba između najjačih izdanja Microsoft SQL Servera i Oracle sustava za upravljanje bazama podataka, pa iako se usporedba ne temelji na recentnijim istraživanjima, međusobni odnos cijena nije se značajno mijenjao tijekom zadnjih nekoliko godina.

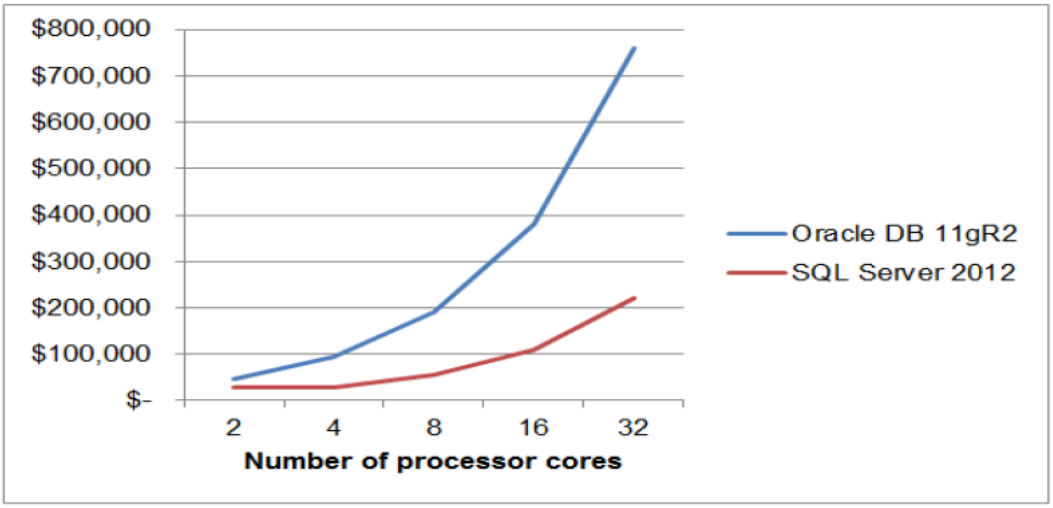
Na slici 6. Sve funkcionalnosti koje su kod Oracle-a navedene zasebno, uključene su u SQL Server 2012 izdanje (Enterprise) s kojim je usporedba i provedena.




Slika 6. Usporedba cijene Oracle 11gR2 i Microsoft SQL Server 2012 [8]

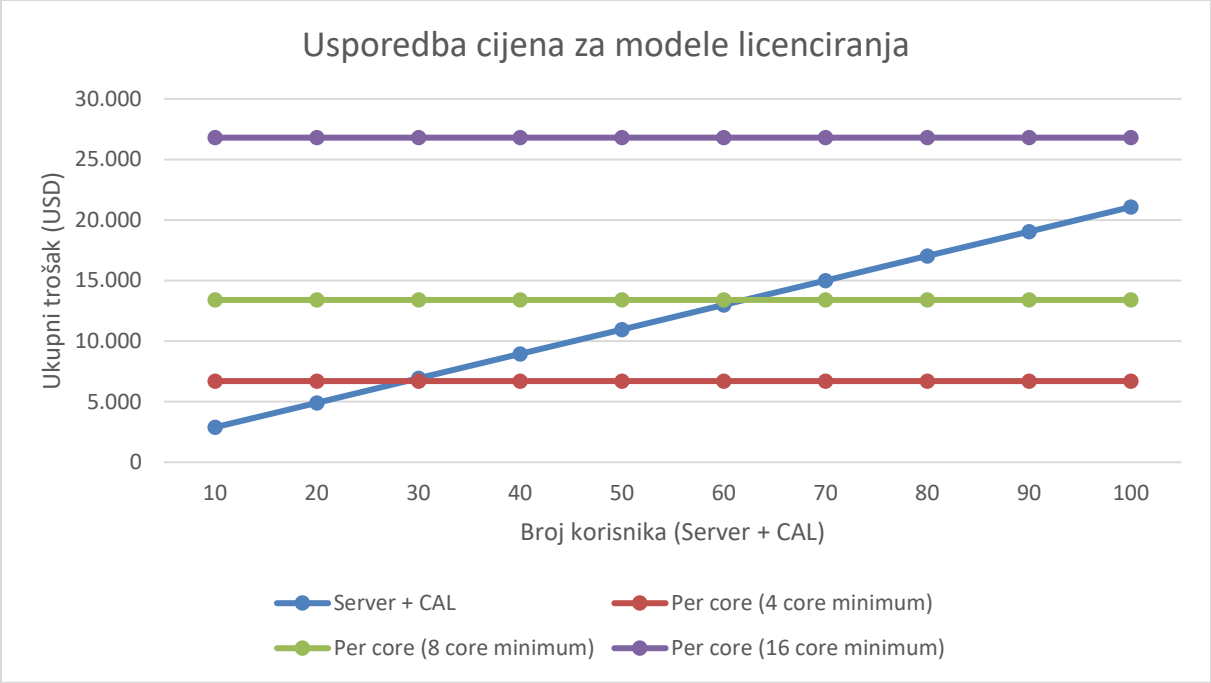
U nastavku na slici 7. prikazan je odnos cijena  između Oracle i Microsoft SQL server proizvoda u odnosu na broj korištenih (licenciranih) procesorskih jezgri, tako da se na veoma jednostavan način može doći do zaključka koji od proizvoda je financijski prihvatljiviji u određenom scenariju.

Na sličan način Microsoft SQL Server moguće je usporediti i sa drugim sustavima za upravljanje bazama podataka, međutim te usporedbe nisu u fokusu ovog rada.




Slika 7. Troškovi Oracle i Microsoft SQL Servera prema broju procesorskih jezgri [8]

Vidačić i Špicar [7] u svom istraživanju za Microsoft SQL Server 2017 Standard izdanje dolaze do zanimljive usporedbe ukupnih troškova u odnosu na model licenciranja i broj korisnika koje sustav treba podržati, pa se tako na slici 8. može primijetiti da, osim u slučajevima kada se radi o primjeni SQL Server baze podataka na Internetu, gdje nije moguće jednoznačno odrediti broj korisnika, korištenjem modela licenciranja po korisniku (poslužiteljske licence + korisničke pristupne licence) moguće je postići značajne uštede bez potrebe za ograničavanjem broja poslužitelja ili smanjivanjem performansi sustava.




Slika 8. Kretanje cijena MS SQL Servera 2017 ovisno o modelima licenciranja [7]

Za razliku od „Enterprise“ i „Standard“ izdanja, koje su prvenstveno namijenjene ozbiljnim poslovnim primjenama, postoje i besplatna izdanja i to:

a) „Express“ izdanje je prvenstveno namijenjeno za manje zahtjevne primjene poput razvoja jednostavnih desktop ili web aplikacija kod kojih nametnuta ograničenja na broj procesorskih jezgri, veličinu radne memorije i kapaciteta baze podataka ne predstavljaju poseban problem. („Express“ izdanje moguće je nadograditi na višu razinu, odnosno „Standard“ izdanje, bez većih problema).

b) „Developer“ izdanje koje je ograničeno samo licenčno, s obzirom da ga je moguće koristiti samo u svrhu razvoja i testiranja poslovnih programskih aplikacija i baza podataka. (Po svojim mogućnostima jednak je „Enterprise“ izdanju).

## 4.2. Pregled verzija i međusobne kompatibilnosti

Od 1996 godine do danas izdano je jedanaest osnovnih verzija SQL Server-a, od kojih svaka u nekoliko izdanja. Na temelju dostupnih podataka, u tablici 2. za svaku verziju naveden je broj ažuriranja koji uključuje nadogradnje sustava u obliku pojedinačnih, kumulativnih i servisnih nadogradnji (eng. Service Pack).

Tablica 2. Popis verzija i ažuriranja Microsoft SQL Server-a na dan 09.11.2019.

| Redni Broj | Naziv verzije SQL Servera | Razdoblje ažuriranja | Broj godina ažuriranja | Broj ažuriranja |
|---|---|---|---|---|
| 1. | SQL Server 2019 | 2019 | 1 | 1 |
| 2. | SQL Server 2017 | 2016 – 2018 | 2 | 27 |
| 3. | SQL Server 2016 | 2015 – 2018 | 3 | 69 |
| 4. | SQL Server 2014 | 2013 – 2018 | 5 | 90 |
| 5. | SQL Server 2012 | 2010 – 2018 | 8 | 91 |
| 6. | SQL Server 2008 R2 | 2009 – 2019 | 10 | 71 |
| 7. | SQL Server 2008 | 2007 – 2018 | 11 | 88 |
| 8. | SQL Server 2005 | 2005 – 2012 | 7 | 246 |
| 9. | SQL Server 2000 | 2000 – 2012 | 12 | 330 |
| 10. | SQL Server 7.0 | 1998 – 2012 | 14 | 47 |
| 11. | SQL Server 6.5 | 1996 – 2005 | 9 | 21 |

(Izvor: Prema [9])

Posebna vrsta nadogradnji do verzije SQL Server 2016 su servisni paketi, koji su poput kumulativnih nadogradnji sadržavali sve promjene do trenutka njihovog izdavanja. Obično bi se izdavali do najviše pet puta za određenu verziju SQL servera te su prolazili kroz intenzivne regresijske testove.

Od verzije SQL Server 2017, dinamika ažuriranja je promijenjena, te su servisni paketi u potpunosti izbačeni iz rasporeda, a umjesto njih sada istu razinu testiranja prolaze kumulativne nadogradnje, tako da proizvođač sada preporučuje njihovo proaktivnu instalaciju.

Postupcima ažuriranja SQL Servera, primjenom paketa za brzu nadogradnju (*eng. Quick Fix Engineering – QFE*), sigurnosnu nadogradnju (*eng. Common Vulnerabilities and Exposures – CVE*), kumulativnu nadogradnju (*eng. Cumulative Update – CU*) i sl. potrebno je pristupiti veoma oprezno.

Veliki broj administratora nadogradnje vezane uz Microsoft proizvode primjenjuje automatski koristeći neki od sustava za automatsku primjenu nadogradnji ili npr. sustav za ažuriranje operacijskog sustava Windows (*eng. „Windows Server Update Services – WSUS*),

pa iako je općenito korištenje automatiziranih sustava preporučljivo i podrazumijeva se, u slučaju nadogradnji sustava za upravljanje bazama podataka, ukoliko se postupci nadogradnje prepuste automatici, to može dovesti do ozbiljnih problema.

Kod nadogradnji SQL Servera, a tako i drugih sustava, veoma je bitno osigurati testno okruženje na kojem je moguće provesti testiranje nadogradnji i tek nakon uspješnog testiranja i provjere ponašanja sustava u realnim uvjetima pustiti nadogradnju na produkcijsko okruženje.

Ukoliko se radi o sigurnosnim nadogradnjama, a posebno u slučajevima kada se radi o sustavu koji je izložen Internet-u ili sličnom okruženju, bitno je što žurnije provesti procjenu rizika, a tek nakon toga donijeti konačnu odluku o načinu primjene.

Kao što je prikazano na tablici 3. svaka verzija SQL Servera ima unaprijed definiranu određenu razinu kompatibilnosti (*eng. Compatibility Level*), ali isto tako podržava i određene razine kompatibilnosti nižih verzija. Postavljanjem parametra kompatibilnosti na razini baze podataka od strane korisnika ili automatski prilikom kreiranja baze podataka, SQL Serveru se na razini baze podataka ograničava set funkcionalnosti koje može koristiti, pa je tako na jednoj instanci poslužitelja omogućeno korištenje baza podataka starijih i novijih verzija SQL Server sustava.

Iako razine kompatibilnosti omogućavaju korištenje baza podataka kreiranih na prethodnim verzijama SQL Servera, osim razine kompatibilnosti potrebno je ipak voditi računa i o drugim faktorima poput opcija, zastarjelih i nepodržanih značajki sustava i sl. koje ukoliko se ne tretiraju na ispravan način mogu utjecati na nemogućnost korištenje baze podataka ili pak utjecati na performanse sustava [10].

Tablica 3. Podržane razine kompatibilnosti SQL Server-a

| Redni Broj | Naziv verzije SQL Servera | Oznaka verzije | Razina kompatibilnosti | Podržane razine kompatibilnosti |
|---|---|---|---|---|
| 1. | SQL Server 2019 | 15 | 150 | 150,140,130,120,110,100 |
| 2. | SQL Server 2017 | 14 | 140 | 140,130,120,110,100 |
| 3. | SQL Server 2016 | 13 | 130 | 130,120,110,100 |
| 4. | SQL Server 2014 | 12 | 120 | 120,110,100 |
| 5. | SQL Server 2012 | 11 | 110 | 110,100,90 |
| 6. | SQL Server 2008 R2 | 10.5 | 100 | 100,90,80 |
| 7. | SQL Server 2008 | 10 | 100 | 100,90,80 |
| 8. | SQL Server 2005 | 9 | 90 | 90,80 |
| 9. | SQL Server 2000 | 8 | 89 | 80 |

(Izvor: Prema [9])

## 4.3. Problemi nadogradnje baza podataka između različitih verzija MS SQL Server-a

Problemi koji se javljaju kod nadogradnje Microsoft SQL Server-a i pripadajućih baza podataka, mogu ovisiti o različitim faktorima, koje je autor na temelju primjera iz prakse svrstao u sljedeće grupe:

a) Hardverski i softverski zahtjevi,

b) Dostupnost sustava (tijekom nadogradnje),

c) Razlika u verzijama SQL Servera,

d) Kompatibilnost postojećih baza podataka,

e) Opseg korištenja ostalih servisa iz paketa Microsoft SQL Servera,

f) Sigurnosni zahtjevi.

Osnovni preduvjet za nadogradnju SQL Servera je osiguranje minimalnih hardverskih i softverskih uvjeta poslužitelja.

Ukoliko se radi o produkcijskom okruženju visoke dostupnosti (*eng. High Availability – HA*), kod kojeg više poslužitelja pruža istu uslugu, nadogradnja je moguće provesti na jednom od poslužitelja u grupi, čime ne dolazi do narušavanja rada poslovnih programskih aplikacija.

S druge strane, ukoliko se radi o okruženju koje nije konfigurirano za visoku dostupnost, tada nadogradnja hardvera ili određenih softverskih komponenti može dovesti do situacije u kojoj tijekom te nadogradnje pristup bazama podataka nije moguć.

Kvalitetnim planiranjem, kreiranjem i testiranjem plana nadogradnje, čak i kod sustava koji nisu dizajnirani za visoku dostupnost moguće je postići da vrijeme uskraćivanja pristupa bazi bude minimalno, a ukoliko postoji definirano razdoblje u kojem se obavljaju poslovi održavanja sustava, tada je migraciju moguće obaviti bez ikakvog utjecaja na poslovanje i na dogovorenu dostupnost sustava.

Posebni problemi javljaju se u slučaju kada između trenutne produkcijske verzije i verzije SQL Servera na koji se vrši migracija postoji više „među-verzija“, jer u tim slučajevima direktna nadogradnja SQL Servera često nije moguća, pa je za migraciju potrebno koristiti više različitih instanci ili ukoliko to nije moguće, više fizičkih ili virtualnih poslužitelja.

U takvim slučajevima javlja se problem razine kompatibilnosti baze podataka, a problem je moguće riješiti tako, da se baza podataka prvo prebacuje na najvišu moguću verziju koja podržava postojeću razinu kompatibilnosti baze podataka, zatim se pokreće podizanje

razine kompatibilnosti na maksimalnu vrijednost i tek nakon toga se ista prebacuje na ciljanu verziju SQL Servera.

Prema podacima iz tablice 3. možemo vidjeti da izravna nadogradnja baze podataka sa razine kompatibilnosti 80, koja odgovara verziji SQL Server 2000, nije moguća na verziju SQL Servera 2012.

Da bi se takva migracija mogla provesti, nužni preduvjet je nadogradnja baze podataka na razinu kompatibilnosti 90 tj. SQL Server 2005, a koja je minimalno potrebna razina kompatibilnosti koja podržava nadogradnju na SQL Server 2012.

Trenutnu razinu kompatibilnosti za sve baze na SQL Server-u moguće je provjeriti pokretanjem sljedećeg SQL upita:

```
SELECT name,compatibility_level
FROM   sys.databases;
```

Ukoliko je potrebno izvršiti promjenu razine kompatibilnosti za neku od baza podataka prije ili nakon migracije, to je moguće izvršenjem SQL naredbi u nastavku, gdje je potrebno varijable NAZIV_BAZE i RAZINA_KOMPATIBILNOSTI zamijeniti sa stvarnim podacima o nazivu baze podataka i brojčanoj oznaci razine kompatibilnosti:

```
ALTER DATABASE NAZIV_BAZE
SET COMPATIBILITY_LEVEL = RAZINA_KOMPATIBILNOSTI;
```

Osim spomenutih problema vezanih uz hardverske i softverske preduvjete i osiguranja adekvatne dostupnosti sustava, kako je već pojašnjavano, najčešći problemi kod migracije baza podataka vezani su uz probleme kompatibilnosti između različitih SQL Server poslužitelja i baza podataka koje sudjeluju u nadogradnji ili migraciji koje možemo svrstati u četiri osnovne grupe [11]:

a) Zastarjele značajke sustava,
b) Nepodržane značajke sustava,
c) Promjene konfiguracije i postavki,
d) Promjene u ponašanju sustava.

Osim problema razine kompatibilnosti koja spada u grupu značajki sustava pod oznakom a) i b), mogu se javiti i drugi problemi koji se npr. odnose na zastarjele i nepodržane značajke sustava, promjene u konfiguraciji i/ili ponašanju sustava. Ti problemi ne moraju se nužno javiti tijekom postupka same nadogradnje, već je moguće da baza podataka nakon postupka nadogradnje neće u potpunosti funkcionirati na ispravan način [10][12].

Pošto se neki od problema mogu javiti tek nakon provedenog postupka nadogradnje, a mogu biti uzrokovani primjenom nepodržanih SQL naredbi, SQL hintova, nepodržanih tipova

podataka, sistemskih pohranjenih procedura, serverskih varijabli i sl. analiza na utvrđivanju tih nedostataka i kompatibilnosti mora se provesti već u fazi planiranja nadogradnje i migracije na novu verziju SQL Servera [13][14].

Osim korištenja specijaliziranih alata, analizu mogućnosti nadogradnje postojeće verzije SQL Servera i pripadajućih baza podataka moguće je provesti i na jednostavan način, a to je pretvaranjem baze podataka u izvorni kod (SQL skriptu) i pokretanjem skripte na ciljnoj verziji SQL Servera. Pri tome, ukoliko baza podataka sadrži bilo koju od nepodržanih značajki, kreiranje baze i objekata baze podataka pratit će poruke o grešci.

Dobra praksa prilikom korištenja SQL Servera je kontinuirano provođenje analize zastarjelih značajki sustava koji će biti nepodržani u budućim verzijama SQL servera, a koje je moguće dobiti pokretanjem odgovarajuće SQL naredbe :

```
SELECT *
FROM   sys.dm_os_performance_counters
WHERE  counter_name = 'Usage'
   AND cntr_value > 0
```

Kontinuiranim praćenjem zastarjelih i nepodržanih značajki, usklađivanjem i certifikacijom baza podataka prema određenim razinama kompatibilnosti, postiže se brža i jednostavnija migracije između uzastopnih verzija Microsoft SQL Servera.

Ukoliko migracija između SQL Servera obuhvaća veću razliku u verzijama, tada se za postupak migracije, uz analizu dostupne dokumentacije proizvođača dodatno preporuča korištenje već spomenutih alata i čarobnjaka za analizu migracije i/ili korištenje SQL alata za skriptiranje izvorne baze podataka i kreiranja nove uz migraciju podataka.

Nakon uspješno provedene migracije podataka, kako bi se sustav doveo u ispravno stanje važno je sprovesti neke od aktivnosti održavanja baze podataka i to najmanje:

a) Ažurirati korisnike i prava pristupa (*eng. orphaned users and access rights*),
b) Ažurirati indeks-e i statistike,
c) Provjeriti postavke specifičnih postavki sustava i baze podataka,
d) Ažurirati planove održavanja (*eng. maintenance plans and jobs*),
e) Provjeriti funkcioniranje povezanih servisa.

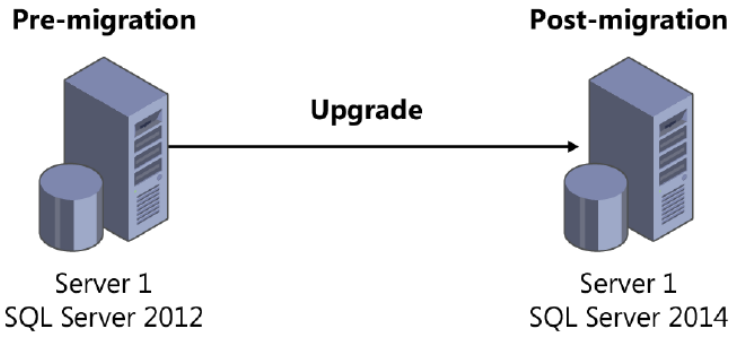
Osim aktivnosti na razini sustava za upravljanje bazama podataka, veoma je bitno provesti i dodatna korisnička testiranja uz pomoć poslovnih programskih aplikacija, jer se samo tako mogu u potpunosti potvrditi adekvatne performanse i funkcionalnost SQL Server poslužitelja i baza podataka.

## 4.4. Metodologija nadogradnje MS SQL servera i korisničkih baza podataka

Svaki postupak nadogradnje sustava predstavlja ozbiljan informatički poduhvat, za koji je potrebno osigurati vrijeme, materijalne i ljudske resurse i adekvatne planove. Iako se svakom postupku nadogradnje i migracije baza podataka može pristupiti kao zasebnom projektu, upotrebom odgovarajućih metodika za migraciju može se značajno skratiti trajanje projekta, može se poboljšati kvaliteta i uspješnost cijelog pothvata.

Prema R. Mistry i S. Misner [15][16][17], te vodeći se preporukama proizvođača, nadogradnju SQL Servera i baza podataka moguće je izvesti na temelju dvije opće strategije:

a) „**In-Place**“ strategija nadogradnje, koja obuhvaća instalaciju novije verzije SQL servera preko postojeće instalacije SQL Servera, kao što je prikazano na slici 9.,



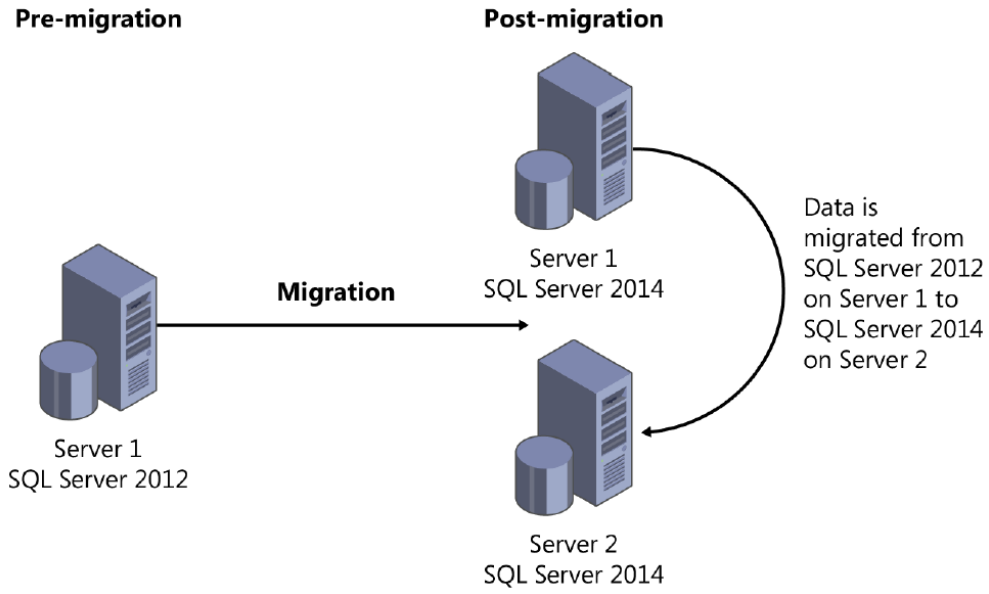

Slika 9. „In-Place“ strategija migracije [17]

Pozitivna stana ovakve migracije je to što nisu potrebni dodatni hardverski zahtjevi, konfiguracija poslužitelja i konfiguracija klijentskih aplikacija ostaje ista, baze podataka prolaze automatsku nadogradnju na minimalnu razinu podržane kompatibilnosti (i dalje je potrebno provesti prethodnu analizu i otkloniti značajke sustava koje više neće biti podržane), nema potrebe za kreiranjem kopija baza podataka, čime se postiže znatno brža nadogradnja.

Negativna strana ovakve migracije je to što ne postoji mogućnost parcijalne nadogradnje određenih baza podataka, što je tijekom cijelog procesa nadogradnje sustava nedostupan (potrebno je planirati vrijeme nedostupnosti), a u slučaju bilo kakve greške koja bi dovela do prekida nadogradnje može sustav dovesti u stanje kod kojeg bi vraćanje unazad moglo biti veoma kompliciranо i dugotrajno.

Zato se u ovakvim scenarijima preporučuje provođenje probne nadogradnje i migracije sustava na testnom okruženju, a prije same nadogradnje i zaštitu konfiguracije sustava i podataka poslužitelja.

b) „**Side-By-Side**“ nadogradnja kao što je prikazano na slici 10., obuhvaća migraciju kod koje se nadogradnja SQL Server sustava na višu verziju obavlja na način da se uz postojeći poslužitelj paralelno uspostavi novi poslužitelj (slučaj s dva poslužitelja) odnosno da se na postojećem poslužitelju (uz zadovoljavanje minimalnih uvjeta) uspostavi nova instanca SQL Servera (slučaj s jednim poslužiteljem).




Slika 10. „Side-By-Side“ strategija migracije [17]

Pozitivna strana ovakve strategije je u tome što postoji daleko veća kontrola nad procesom migracije, postoji mogućnost postepene migracije korisničkih baza podataka, tako da se utjecaj na poslovne programske aplikacije može minimizirati.

Također, produkcijska instanca može biti dostupna cijelo vrijeme migracije, omogućeno je paralelna nadogradnja i testiranje migrirane baze podataka, a u slučaju bilo kakvih nepredviđenih okolnosti, moguće je zaustaviti proces migracije bez utjecaja na produkcijske baze podataka i nastaviti kada se problem otkloni.

Osnovni nedostaci ovakve strategije su u tome što su potrebni dodatni hardverski i softverski resursi (fizički ili virtualni poslužitelj), a ukoliko se radi o slučaju sa migracijom na

novi poslužitelj potrebne su dodatne intervencije na konfiguraciji klijentskih aplikacija za pristup novom poslužitelju, te dodatni diskovni prostor za smještaj novih migriranih baza podataka.

Na temelju razmatranja i iskustava autora na provođenju migracija baza podataka i nadogradnji SQL Server poslužitelja, te dostupnoj literaturi i dokumentaciji proizvođača, u nastavku na slici 11. grafički je prikazana predložena metodika migracije SQL Server baze podataka.

U prvom koraku potrebno je proučiti konfiguraciju trenutne baze, operacijski sustav te minimalne uvjete koji su potrebni za migraciju na željenu verziju sustava, te na temelju svih podataka o okolini definirati strategiju migracije [18].

Nakon toga pripremaju se okoline za testiranje migracije i produkciju. Ukoliko je odabrana strategija koja obuhvaća novi poslužitelj, u toj fazi provodi se instalacija i konfiguracija ciljnog sustava na fizičkom ili virtualnom okruženju, a ukoliko se radi o „In-Place“ strategiji tada se pripremaju preduvjeti za instalaciju nove instance SQL Servera na postojećem poslužitelju. Najvažnije je osigurati adekvatnu okolinu na kojoj će biti moguće provesti odgovarajuće simulacije i pripremiti konačan plan migracije.

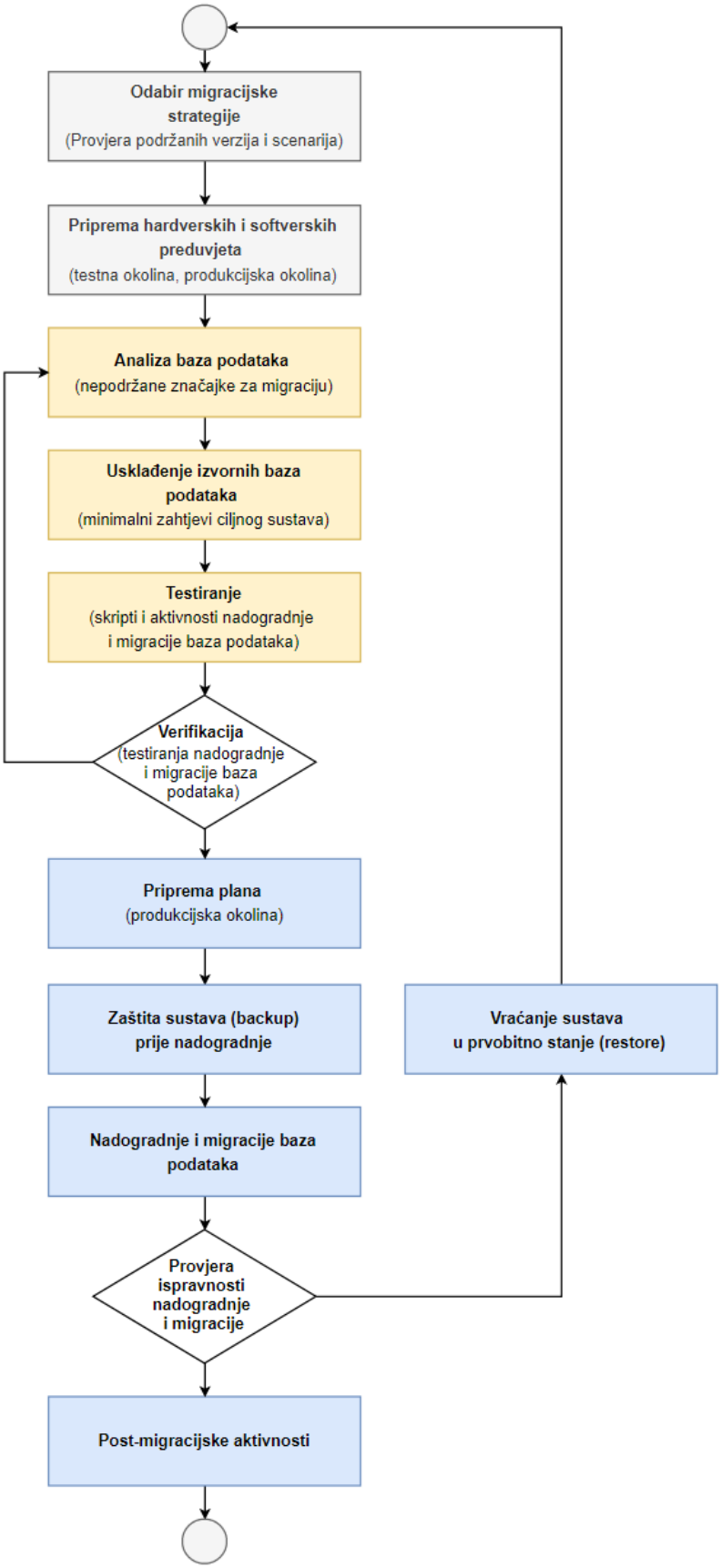
Treći korak obuhvaća analizu nepodržanih značajki SQL Server sustava za sve verzije između trenutne verzije i ciljne verzije uključujući napomene vezane uz promjene u konfiguracijama i moguće utjecaje na performanse sustava [12].

Između ostalog, preporuča se instalacija čarobnjaka za analizu migracije objekata baze podataka (*eng. Data Migration Assistant – DMA*) pomoću kojeg se mogu detektirati problemi koji bi mogli utjecati na uspješnost migracije i kasnije ispravno funkcioniranje baze podataka.

Osim korištenja čarobnjaka moguće je provesti skriptiranje cijele baze podataka i drugih povezanih objekata i skripti poput planova za održavanje i sl. te testiranje provesti pokretanjem na novoj SQL Server verziji uz praćenje poruka o greškama. Osim tih načina, preporuča se provjera `sys.dm_os_performance_counters` tablice iz koje se mogu dobiti podaci o zastarjelim značajkama koje se trenutno koriste, a koje bi mogle prouzročiti probleme prilikom migracije

Sve te informacije važne su za sljedeću fazu u kojoj se provode potrebna usklađenja objekata baze podataka (tipova polja tablica baza podataka, pohranjenih procedura, funkcija, okidača, pogleda itd.).

Ukoliko postoji mogućnost, usklađenja se mogu provoditi i prije same migracije, kroz redovno održavanje sustava, čime se umanjuje koncentracija rizika, omogućuje se opsežnije korisničko testiranje u stvarnim (produkcijskim) uvjetima i smanjuje opseg promjena koje bi se morale provesti tijekom same migracije.




Slika 11. Metodika nadogradnje SQL Servera i baza podataka (Izrada autora)

Peti korak odnosi se na testiranje pojedinih skripti i koraka s ciljem simuliranja stvarnih produkcijskih uvjeta kakvi će biti prilikom migracije, kako bi se prikupili potrebni podaci na temelju kojih se provode dodatna podešavanja plana migracije.

Nakon uspješno provedenih testiranja, pristupa se izradi konačnog plana migracije, a ukoliko još uvijek neki od rezultata nisu zadovoljavajući, poput npr. trajanja migracije i slično, još uvijek je moguće vratiti se i pronaći odgovarajuća rješenja.

U fazi kreiranja plana migracije potrebno je formalizirati aktivnosti i odgovornosti pojedinih sudionika, komunikacijski plan, način odlučivanja, popis korištenih alata, SQL skripti i drugih informacija pomoću kojih se kontrolira tijek migracije.

Konkretna migracija započinje provođenjem adekvatne zaštite softverskih komponenti, konfiguracije SQL Servera i korisničkih podataka, čime se osigurava da u slučaju nepredvidivih situacija postoji mogućnost vraćanja u vrijeme prije početka migracije.

Nakon provedene zaštite sustava i podataka, pokreću se aktivnosti koje mogu obuhvatiti instalaciju ili nadogradnju SQL Server-a (u „Side-By-Side“ scenariju taj dio se provodi unaprijed kao jedna od preduvjeta), zatim kreiranje baze podataka i punjenje podataka putem migracijskih skripti ili pomoću procedure za zaštitu i vraćanje baze podataka ili na način direktnog kopiranja baze podataka pomoću metoda „detach“ i „attach“, a sve ovisno o odabranom scenariju.

Nakon provedene nadogradnje i migracije baze podataka, kako bi se mogla donijeti odluka o nastavku produkcijskog rada na novom sustavu potrebno je provesti testiranje od strane korisnika *(eng. User Acceptance Test – UAT*), odnosno ukoliko migracija nije uspješno obavljena ili su identificirani problemi koji bi mogli utjecati na stabilnost poslovnih programskih aplikacija, pokreće se procedura za vraćanje podataka i sustava u vrijeme prije početka migracije (*eng. Rollback*).

Procedura za vraćanje nije nužna u onim situacijama kada još nije došlo do aktiviranja novog poslužitelja ili instance npr. u scenariju „Side-By-Side“, dok je u scenariju „In-Place“, zbog brisanja postojeće instance i instalacije novog softvera i modificiranja baza podataka nužna.

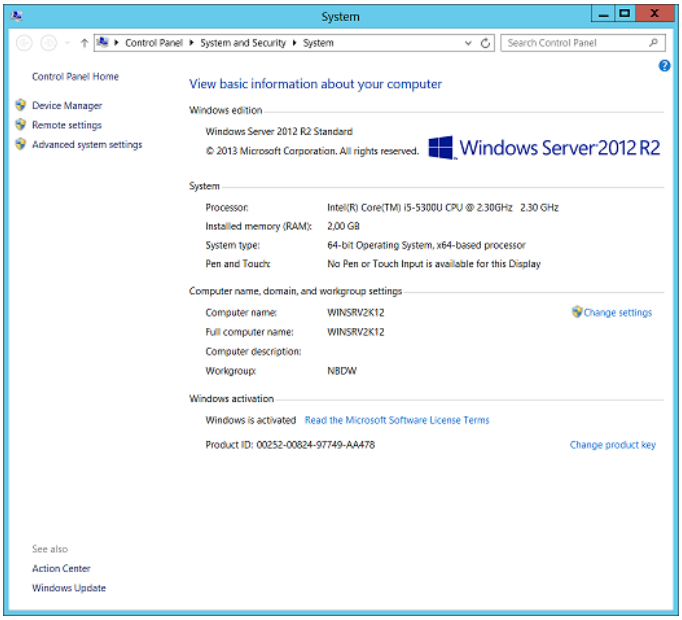
Nakon provedene migracije, obično se provode dodatne aktivnosti kojima je zadaća dodatno poboljšanje performansi i stabilnosti SQL Server sustava, a u koje spada ponovno kreiranje svih indeksa tablica baza podataka, ponovno prevođenje pohranjenih procedura, kreiranje novih statistika i raznih drugih sličnih podešavanja sustava, a koji se sukcesivno provode na temelju kontinuiranog praćenja rada sustava.

## 4.5. Primjena metodologije na praktičnom primjeru

Kako bismo mogli dokazati funkcionalnost predložene metodologije, kreiran je scenarij migracije baze podataka sa sustava SQL Server 2008R2, na kojem se nalazi korisnička baza podataka „mscondb“ razine kompatibilnosti 80 (SQL Server 2000), na razinu SQL Servera verzije 2017.

Prateći opisanu metodologiju, u prvom koraku razmatranjem mogućih strategija nadogradnje sustava i baza podataka, odabrana je „Side-By-Side“ metoda (strategija) migracije, kod koje je pripremljena dodatna instanca fizičkog ili virtualnog poslužitelja (osnovne postavke poslužitelja prikazane su na slici 12.).

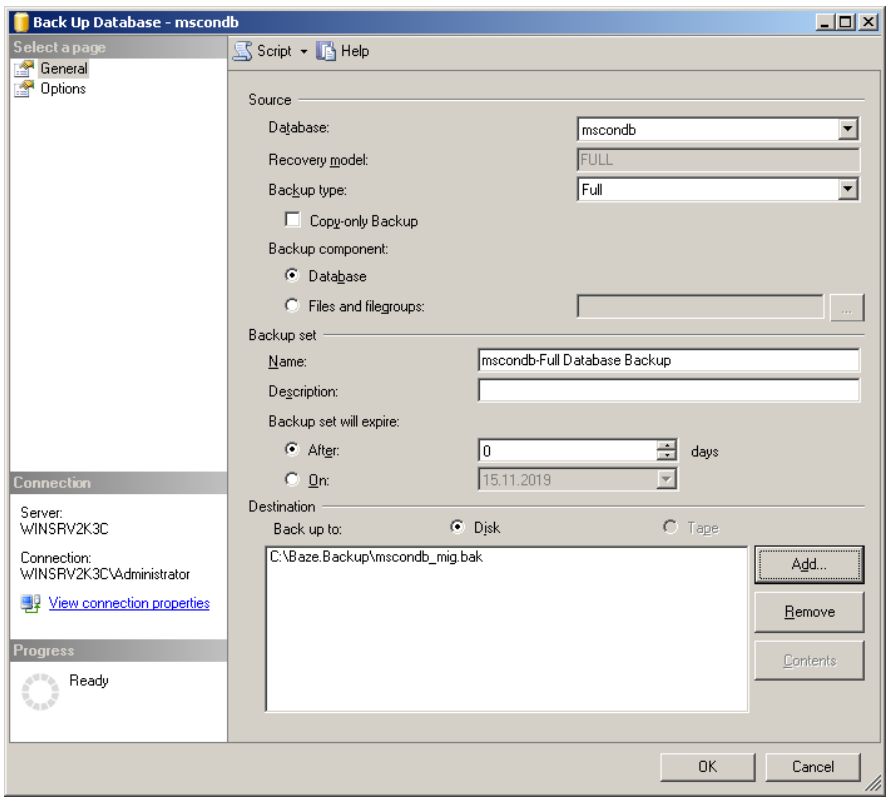
U drugom koraku koji obuhvaća pripremu svih preduvjeta, uz postojeću okolinu sa izvornom bazom podataka pripremljena je i buduća produkcijska okolina koja će u prvoj fazi služiti za testiranje i pripremu migracije, a nakon uspješno provedenih testiranja koristit će se kao nova produkcijska okolina. Nakon pripreme okoline i podešavanja SQL Server-a 2017 za analizu izvorne baze podataka (pošto ćemo se koristiti službenim alatom) potrebno je instalirati „*Microsoft Data Migration Assistant*“ softver.



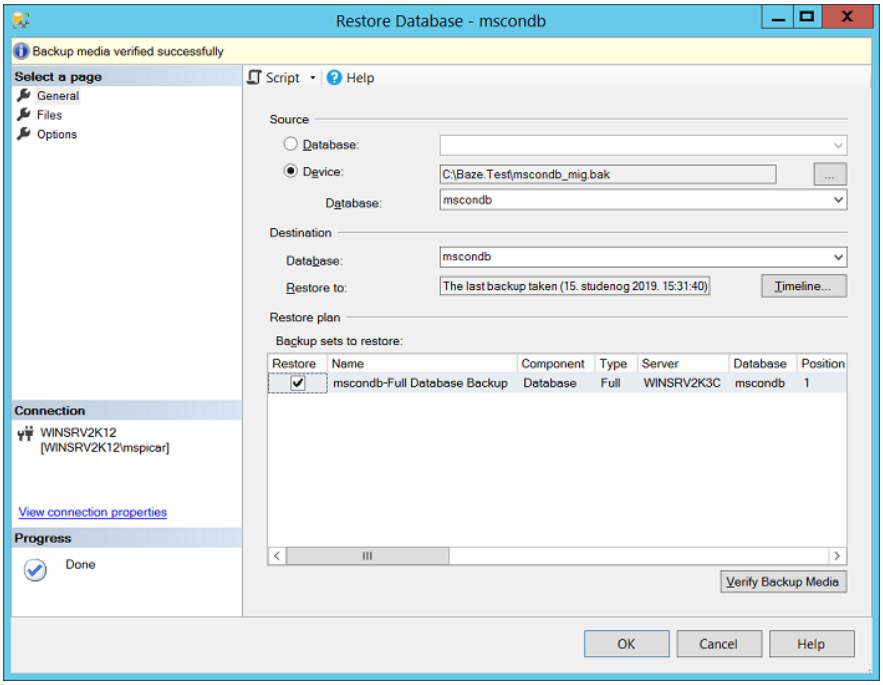

Slika 12. Nova produkcijska okolina za SQL Server 2017

Nakon zaštite baze podataka, prikazano na slici 13., zatim kopiranja i restauriranja baze podataka na „testnoj“ okolini, prikazano na slici 14., uz provjeru podataka iz službene dokumentacije pokrećemo analizu baze podataka i dokumentiramo sve promjene koje je

potrebno provesti kako bi se baza podataka uskladila do minimalne razine potrebne za migraciju na traženu verziju SQL Servera.



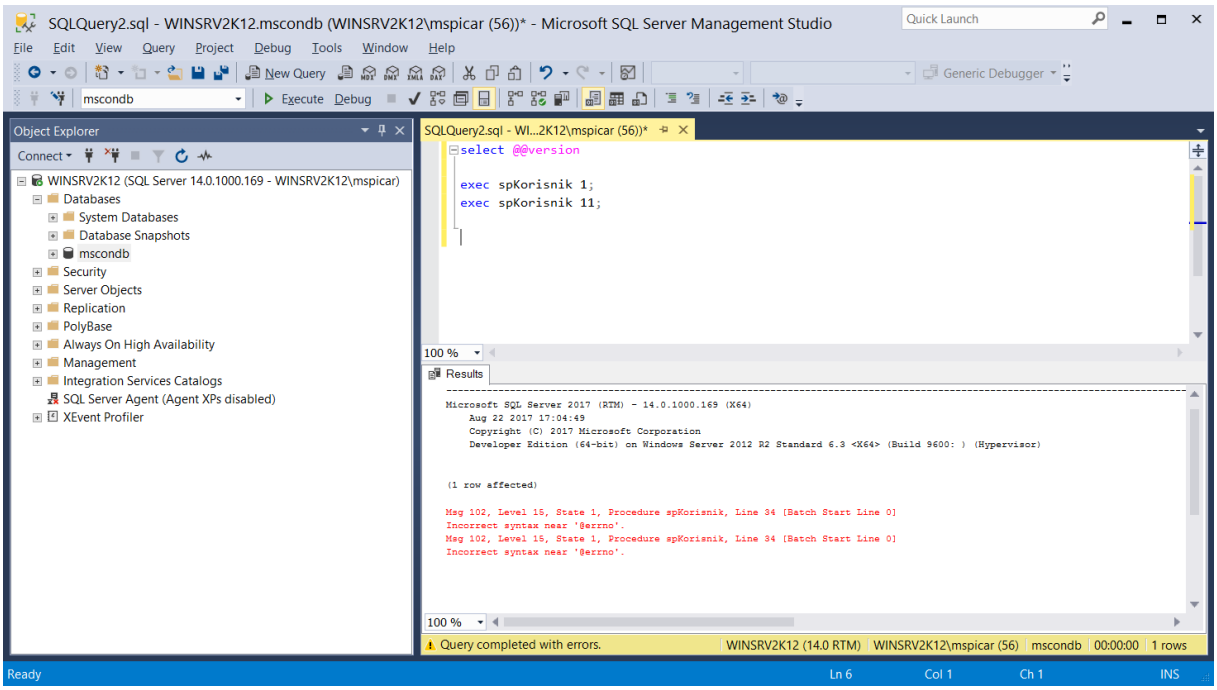

Slika 13. Zaštita (eng. backup) baze podataka na SQL Server-u 2008R2




Slika 14. Vraćanje (eng. restore) baze podataka za SQL Server 2017 okolini

Pošto je izvorna baza podataka verzije SQL Server-u 2008R2, kao način migracije na verziju 2017 omogućeno je korištenje „backup-restore“ metode (slike 13,14). Važno je napomenuti da će se prilikom postupka restauriranja podataka na SQL Server 2017 razina trenutne kompatibilnosti baze sa 80 (SQL 2000) automatski podići na razinu 100 (SQL 2008) koja je ujedno i minimalno podržana razina kompatibilnosti baze podataka na verziji SQL Server 2017.

Iako prilikom postupka restauracije baze podataka automatski dolazi i do promjene razine kompatibilnosti iste, time posao migracije nije završen, jer dalje postoji mogućnost da baza podataka sadrži određene značajke koje više nisu podržane (slika 15.).




Slika 15. Vidljive greške prilikom pokretanja određenih procedura nakon restauracije

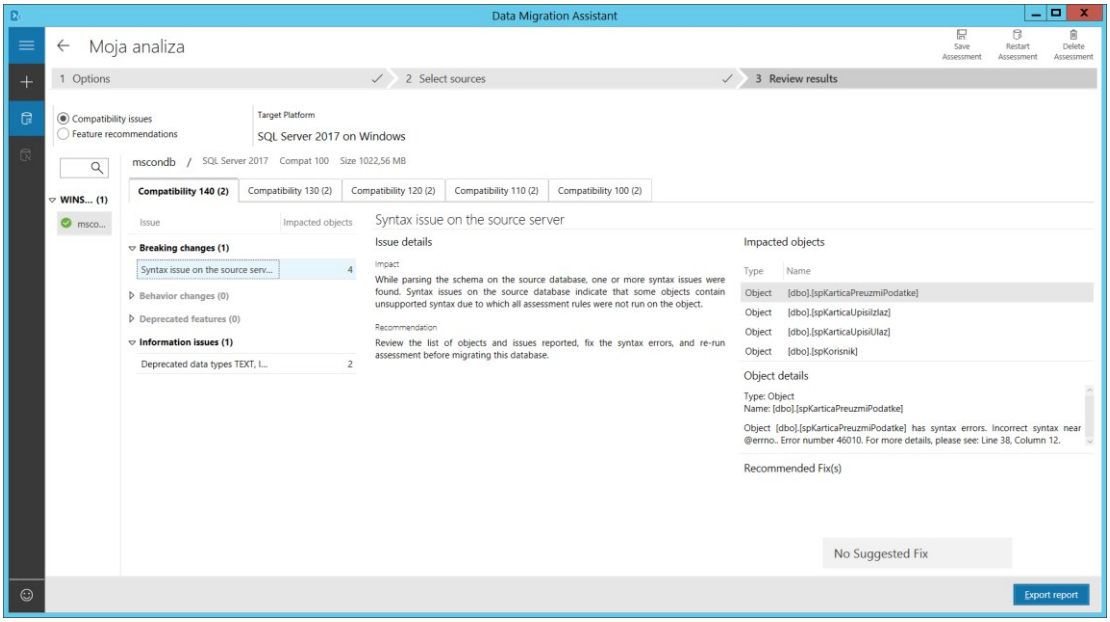
Kada se baza podataka prenosi na način „backup-restore“ struktura baze podataka i izvorni kod u pohranjenim procedurama, funkcijama, okidačima i drugim objektima ne mijenja se automatski, te se greške i nepodržane funkcionalnosti primjećuju tek u trenutku kada se aktiviraju i koriste (slika 15.).

Takvo ponašanje može biti veoma problematično, jer se greške mogu javiti tek prilikom prve upotrebe određene funkcionalnosti poslovne programske aplikacije.

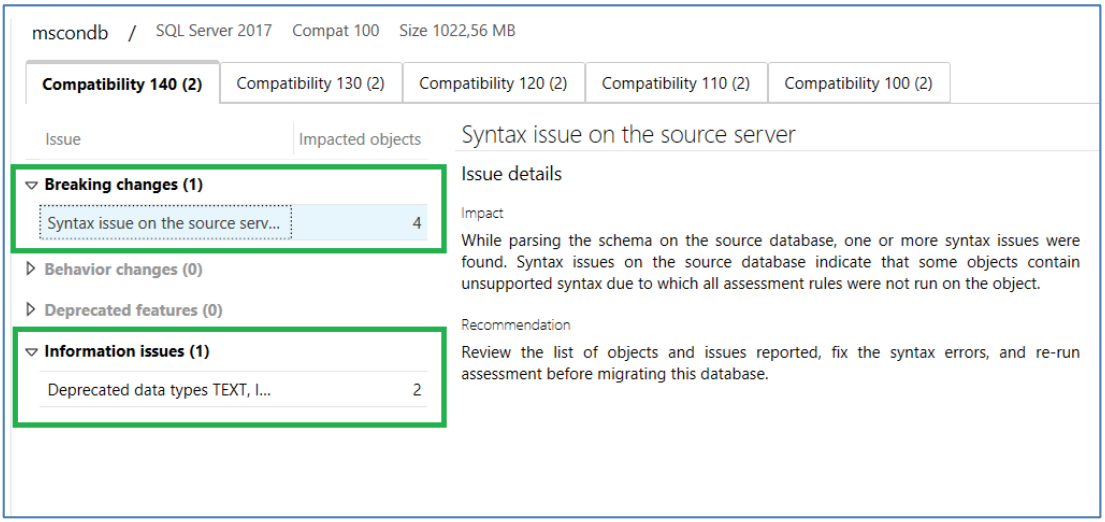
S druge strane, ukoliko se baza podataka između sustava prenosi skriptiranjem, što je obično slučaj kod baza podataka manje veličine, greške se mogu veoma lako uočiti odmah priliko izvršavanja pojedine skripte, u trenutku kreiranja nove baze podataka.

Pokretanjem alata za analizu baze podataka, kako je prikazano na slici 16., moguće je uočiti nepodržane ili zastarjele značajke baze podataka, što je detaljno prikazano na slici 17.

„Data Migration Assistant“ ne mora biti instaliran na poslužitelju, već se analizu može pokrenuta sa bilo koje umrežene radne stanice koja ima pristup do SQL Server poslužitelja.

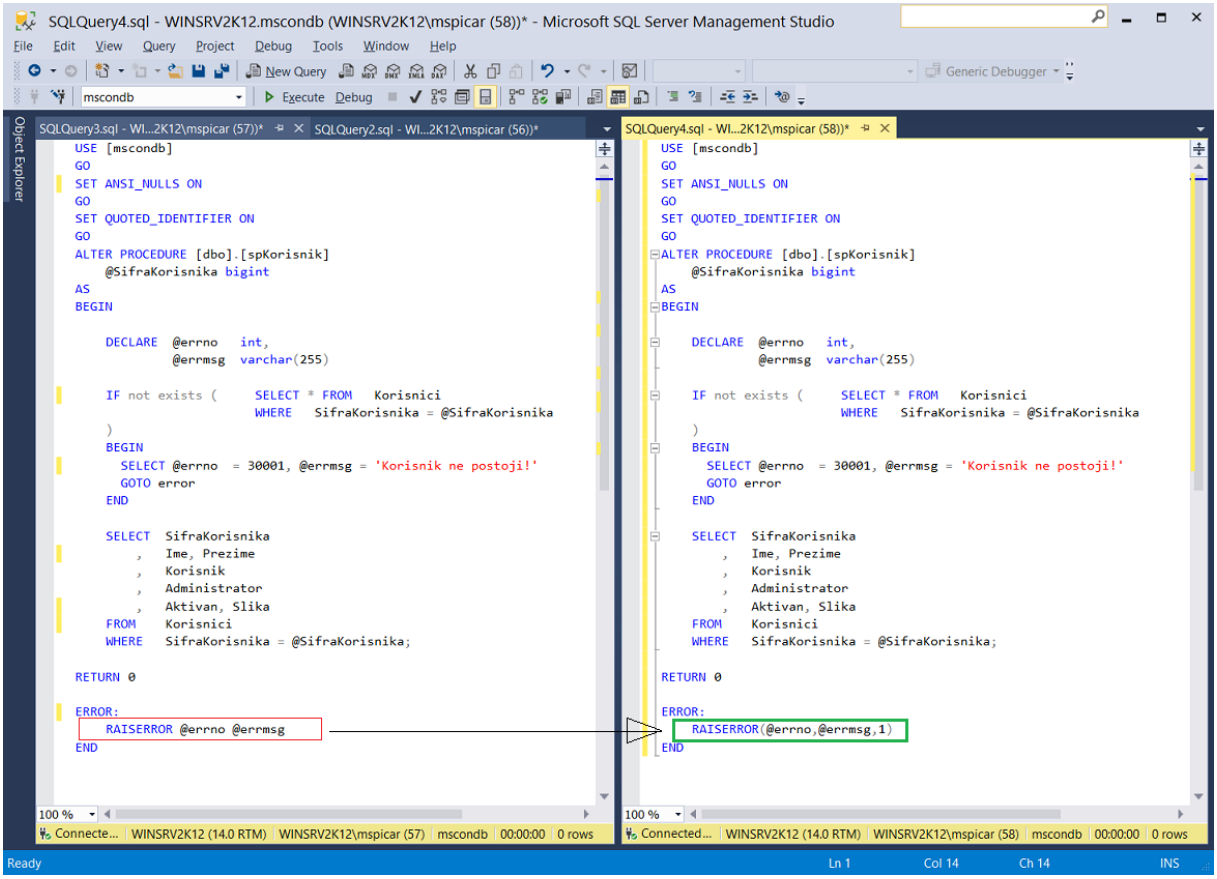



Slika 16. Alat za analizu baze podataka „*Data Migration Assistant*“

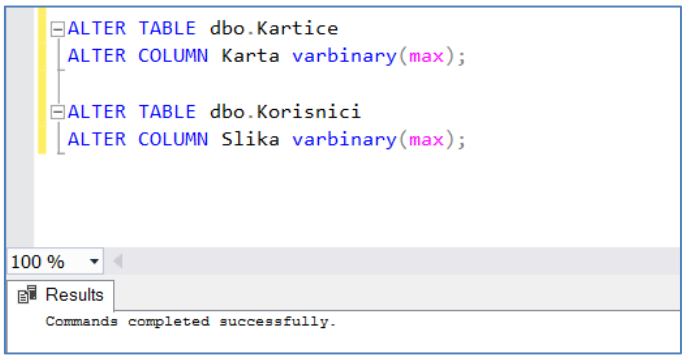



Slika 17. Detaljni prikaz nepodržanih i zastarjelih značajki od strane DMA

Alat za analizu migracije detektirao je četiri (4) nepodržane sintakse kod korištenja naredbe RAISERROR, koje su ispravljene na način kako je prikazano na slici 18., te dvije (2) reference korištenja zastarjelog tipa podataka IMAGE, koje su ispravljene pokretanjem niza SQL naredbi (slika 19.).




Slika 18. Pregled pohranjene procedure i način ispravka nepodržane sintakse



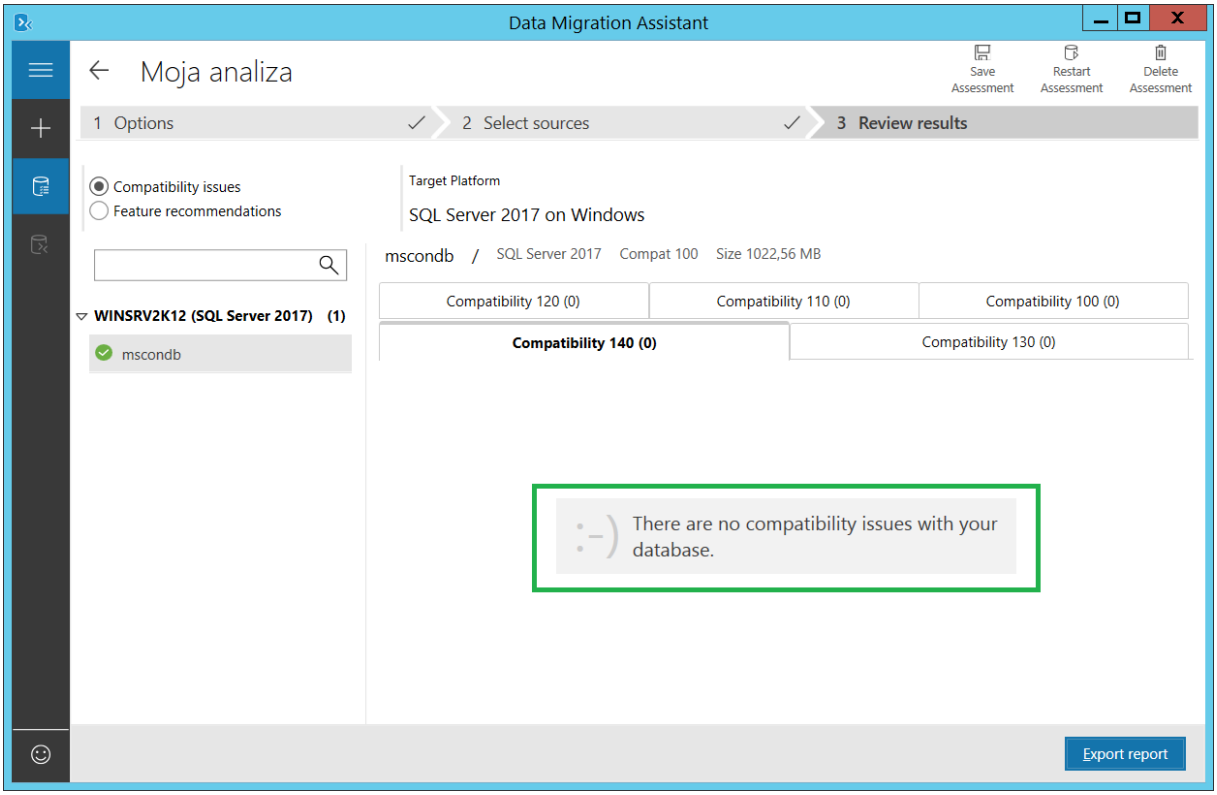

Slika 19. Promjena zastarjelog tipa podataka prema preporuci DMA

Bitno je razlučiti to da u slučaju korištenja nepodržanih značajki sustava dolazi do prekida ili ozbiljnog narušavanja rada sustava, dok kod većine ostalih slučajeva poput zastarjelih značajki i sl. sustav uglavnom funkcionira normalno, osim što se takve greške evidentiraju u zapisima SQL Servera, te se preporučuje ispravljanje i prestanak korištenja prilikom razvoja novih rješenja, pošto u nekim višim verzijama sustava one mogu prestati raditi.

Nakon detekcije svih grešaka i nekompatibilnosti, pripremaju se i testiraju skripte za ispravak procedura i baze podataka, a koje će se koristiti i kod konačne migracije.

Verifikacija skripti i funkcionalnosti „nadograđene“ baze podataka može se izvesti ponovnim puštanjem alata za analizu (slika 20), a često se izvodi i uz pomoć poslovnih programskih aplikacija i provođenjem korisničkog testiranja (*eng. User Acceptance Test - UAT*).

Završetkom i potvrdom svih ispravaka, izrađuje se konačni migracijski plan koji u sebi sadrži sve naučene lekcije i posebnosti uočene prilikom testiranja, a sa ciljem veće kontrole cijelog postupka.




Slika 20. Provjera kompatibilnosti i ispravnosti baze podataka nakon ispravaka

Kako je već naznačeno, za slučaj nepredviđenih okolnosti, prije pokretanja migracije na produkcijskoj okolini potrebno je izraditi plan povratka u stanje prije migracije, što se konkretno postiže ugrađenim procedurama zaštite i restauracije baze podataka.

Nakon izvršenja aktivnosti iz plana migracije, te preusmjeravanja korisničkih aplikacija na novi poslužitelj, poslovne programske aplikacije spremne su za produkcijsko korištenje, međutim prije početka korištenja nove migrirane baze podataka, preporučuje se pokretanje osvježavanja statistika i indeksa, te aktiviranje privremeno suspendiranih postupaka redovnog održavanja baze podataka.

Općenito, svrha postupaka održavanja baze podataka je, da baza podataka uvijek bude u najboljem stanju i da funkcionira unutar zadanih parametara.

Ti parametri mogu biti, na primjer, postotak dozvoljene interne fragmentacije podataka (*eng. Internal Fragmentation*), dozvoljeni postotak fragmentacije indeksa (*eng. Index Fragmentation*) ili nešto treće.

Provođenjem interne i eksterne defragmentacije, gdje pod internom defragmentacijom mislimo na bazu podataka, a pod eksternom defragmentacijom na diskovni pod-sustav, postiže se manji broj ulazno-izlaznih operacija što direktno doprinosi bržem učitavanju podataka i izvršavanju SQL upita.

Osim toga, veće performanse sustava mogu se postići i dodatnim pokretanjem postupaka kreiranja „statistika“, sadržavaju podatke o strukturama pojedinih indeksa i podatke o često korištenim poljima i tablicama, te trajanju pojedinih metoda za pristup tim podacima i indeksima.

Pomoću tih podataka pod-sustavu za optimizaciju upita unutar SUBP-a, kod pokretanja novog SQL upita, omogućeno je da uspoređuje i valorizira različite dostupne strategije i na temelju tih usporedbi izrađuje konačni optimiran izvršni plan (*eng. Execution Plan*) koji se tada sprema i koristi za izvršavanje SQL upita.

Pošto je pod-sustav za optimizaciju SQL upita ovisan o verziji SQL Server sustava, a nakon provedene migracije baze podataka na novu verziju SQL Server-a, statistike kreirane na prethodnoj verziji SQL Servera ne moraju sadržavati sve one potrebne podatke koji su potrebni za optimalno odlučivanje na trenutnoj verziji SQL Servera, osvježavanje svih statistika trebalo bi se svakako uključiti u projektni plan i biti provedeno u sklopu „post-migracijskih“ aktivnosti projekta migracije baza podataka.

# 5. Zaključak

Pošto oduvijek postoji potreba za pronalaženjem boljih metoda i metodika kojima će se postizati niži troškovi razvoja, održavanja, veća raspoloživost, stabilnost i sigurnost poslovnog sustava, promatrajući poslovne programske aplikacije i s njima povezane sustave za upravljanje bazama podataka, slična potreba postoji i za postupak nadogradnje korisničkih baza podataka i sustava za upravljanje bazama podataka.

Kako bi se postigla cjelovitost rada, u prvom dijelu naglasak je stavljen na okolnosti i odnose između poslovnih programskih aplikacija i baza podataka, zatim u drugom dijelu na okolnosti odabira sustava za upravljanje bazama podataka, te na kraju na metodički postupak i njegovu praktičnu primjenu.

Opisana problemska metodika temeljena je na iskustvu autora, studiji slučajeva iz prakse, te analize stručnih i znanstvenih članaka i literature.

Sama metodika obuhvaća jedanaest ključnih točaka, a to su: odabir strategije, priprema migracijske okoline, analizu baza podataka, usklađenje baza podataka, testiranje postupaka migracije, verifikacija, priprema migracijskog plana, zaštita sustava, provođenje nadogradnje (migracije), provjera ispravnosti migracije i post-migracijske aktivnosti.

Faze metodike slijednog su karaktera, međutim ovisno o uspješnosti pojedinih faza i rezultata testiranja postoji mogućnost ponovnog ponavljanja određenih faza.

U praktičnom dijelu, predmetna metodika upotrijebljena je za nadogradnju baze podataka „mscondb“ programskog rješenja „Recepcija“ iz Microsoft SQL Server verzije 2008R2 u verziju Microsoft SQL Server 2017, koja je uspješno provedena.

Prilikom izvođenja praktičnog dijela korišten je sljedeći licencirani softver: VMWare Workstation 15 Pro, Microsoft Windows Server 2003 i 2012, Microsoft SQL Server 2008 R2 i 2017, Microsoft SQL Management Studio 17.8.1 i Microsoft Data Migration Assistant 5.0.

Iako je ukratko spomenuto u početnim razmatranjima, tijekom rada nisu detaljno opisivane okolnosti i/ili postupci zamjene postojećeg poslovnog programskog aplikacijskog sustava novim, odnosno metodike nadogradnje poslovnih programskih aplikacija, međutim ta tema bi mogla biti veoma aktualna i predmet nekih budućih istraživanja i radova.

# Popis literature


[1] K. Kamal and A. Gazal Preet, “Choosing an Appropriate ERP Implementation Strategy,” *IOSR Journal of Engineering*, vol. 2(3), p. 5, Mar. 2012.

[2] S. Vidačić, I. Pihir, and R. Fabac, “Method of data migration from one ERP system to another in real time,” *Proceedings of 21st Central European Conference on Information and Intelligent Systems*, pp. 501–507, Sep. 2010.

[3] M. Adrian, D. Feinberg, and N. Heudecker, “Magic Quadrant for Operational Database Management Systems,” Gartner, Analysis, Oct. 2018.

[4] C. Howson, J. Richardson, R. Sallam, and A. Kronz, “Magic Quadrant for Analytics and Business Intelligence Platforms,” Gartner, Analysis, Nov. 2019.

[5] Microsoft Corporation, “SQL Server 2019 Licensing Data Sheet,” *SQL Server 2019 Licensing Data Sheet*, 11-Sep-2019. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing. [Accessed: 09-Nov-2019].

[6] Microsoft Corporation, “SQL Server 2019 Licensing Guide,” *SQL Server 2019 Licensing Guide*, 11-Sep-2019. [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing. [Accessed: 09-Nov-2019].

[7] S. Vidačić and M. Špicar, “Utjecaj razvoja sustava MS SQL Server na stabilnost poslovnih aplikacija i baza podataka,” *3rd International Scientific and Professional Conference (CRODMA 2018)*, pp. 147–157, Dec. 2018.

[8] Nikosan, “SQL Server 2012 Licensing Value vs. Oracle Database,” *SQL Server 2012 Licensing Value vs. Oracle Database*, 25-May-2012. [Online]. Available: https://blogs.msdn.microsoft.com/nikosan/2012/05/25/sql-server-2012-licensing-value-vs-oracle-database/. [Accessed: 02-Nov-2019].

[9] “Microsoft SQL Server Versions List,” *Microsoft SQL Server Versions List*. .

[10] P. Barthi, “Microsoft SQL Server database compatibility levels,” *Technical and Product News and Insights from Rackspace*, 19-Jun-2019. .

[11] Microsoft Corporation, “SQL Server Backward Compatibility,” *SQL Server Backward Compatibility*, 13-Aug-2012. [Online]. Available: https://docs.microsoft.com/en-us/sql/getting-started/sql-server-backward-compatibility. [Accessed: 11-Feb-2019].

[12] G. Berry, “The Importance of Database Compatibility Level in SQL Server,” *Semi-random musings about SQL Server performance*, 14-Jan-2019. .

[13] Microsoft Corporation, “Deprecated Database Engine Features in SQL Server 2008 R2,” *Deprecated Database Engine Features in SQL Server 2008 R2*, 20-Jul-2015. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms143729(v=sql.105). [Accessed: 11-Feb-2019].

[14] Microsoft Corporation, “Discontinued Database Engine Functionality in SQL Server 2008 R2,” *Discontinued Database Engine Functionality in SQL Server 2008 R2*, 20-Jul-2015. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms144262(v=sql.105). [Accessed: 11-Feb-2019].

[15] R. Mistry and S. Misner, *Introducing Microsoft SQL Server 2008 R2*. Redmond, Washington: Microsoft Press, 2010.

[16] R. Mistry and S. Misner, *Introducing Microsoft SQL Server 2012*. Redmond, Washington: Microsoft Press, 2014.

[17] R. Mistry and S. Misner, *Introducing Microsoft SQL Server 2014*. Redmond, Washington: Microsoft Press, 2014.

[18] Microsoft Corporation, “Supported Version and Edition Upgrades,” *Supported Version and Edition Upgrades*, 27-Jun-2016. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2012/ms143729(v=sql.110). [Accessed: 11-Feb-2019].

# Popis slika

Slika 1. „Big bang“ strategija migracije i prelaska na novi sustav (izrada autora) ..... 5
Slika 2. „Phased rollout“ strategija migracije i prelaska na novi sustav (izrada autora)..... 6
Slika 3. „Parallel adoption“ strategija migracije i prelaska na novi sustav (izrada autora)..... 7
Slika 4. Gartnerov kvadrant operacijskih sustava za upravljanje bazama podataka [3].....16
Slika 5. Gartnerov kvadrant platformi za analizu podataka i poslovnu inteligenciju [4] .....17
Slika 6. Usporedba cijene Oracle 11gR2 i Microsoft SQL Server 2012 [8].....19
Slika 7. Troškovi Oracle i Microsoft SQL Servera prema broju procesorskih jezgri [8] .....19
Slika 8. Kretanje cijena MS SQL Servera 2017 ovisno o modelima licenciranja [7].....20
Slika 9. „In-Place“ strategija migracije [17].....26
Slika 10. „Side-By-Side“ strategija migracije [17] .....27
Slika 11. Metodika nadogradnje SQL Servera i baza podataka (Izrada autora) .....29
Slika 12. Nova produkcijska okolina za SQL Server 2017 .....31
Slika 13. Zaštita (eng. backup) baze podataka na SQL Server-u 2008R2 .....32
Slika 14. Vraćanje (eng. restore) baze podataka za SQL Server 2017 okolini .....32
Slika 15. Vidljive greške prilikom pokretanja određenih procedura nakon restauracije.....33
Slika 16. Alat za analizu baze podataka „*Data Migration Assistant*“.....34
Slika 17. Detaljni prikaz nepodržanih i zastarjelih značajki od strane DMA.....34
Slika 18. Pregled pohranjene procedure i način ispravka nepodržane sintakse.....35
Slika 19. Promjena zastarjelog tipa podataka prema preporuci DMA.....35
Slika 20. Provjera kompatibilnosti i ispravnosti baze podataka nakon ispravaka.....36

# Popis tablica

Tablica 1. Modeli licenciranja SQL Server 2019 [5] .....18
Tablica 2. Popis verzija i ažuriranja Microsoft SQL Server-a na dan 09.11.2019. .....21
Tablica 3. Podržane razine kompatibilnosti SQL Server-a .....22