



TECHNICKÁ UNIVERZITA V LIBERCI
Ekonomická fakulta



Replikace Business Intelligence řešení a její efektivita

Diplomová práce

Studijní program: N6209 – Systémové inženýrství a informatika

Studijní obor: 6209T021 – Manažerská informatika

Autor práce: **Bc. Michaela Polcarová**

Vedoucí práce: Ing. Athanasios Podaras, Ph.D.



Prohlášení

Byla jsem seznámena s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Ráda bych na tomto místě využila příležitosti a poděkovala vedoucímu mé diplomové práce, Ing. Athanasiovi Podarasovi Ph.D., za jeho trpělivost, čas a cenné rady v průběhu vypracování diplomové práce. Neméně vděčná jsem své rodině, přátelům a kolegům za jejich podporu po dobu mých studií.

Anotace

Diplomová práce se zaměřuje na problematiku implementace Business Intelligence řešení využívajícího centralizovaný datový sklad pomocí metody replikace. Práce popisuje vybrané nástroje, které jsou pro použití replikace stěžejní. Součástí diplomové práce je identifikace klíčových kroků, které zavedení Business Intelligence vyžaduje, a to pomocí metody WBS. Těmto krokům je dále přiřazena pracnost, která je vstupem pro následné zhodnocení popisované metody.

Klíčová slova: Business Intelligence, Datový sklad, Replikace, WBS.

Annotation

Thesis name: Replication of Business Intelligence solution and its efectivity

The Diploma thesis is focused on implementation of centralized data warehouse in order to achieve Business Intelligence solution using the method of replication. The work describes selected tools, that are crucial for application of replication solution. Part of the work is identification of key points, required for installation of Business Intelligence solution, by usage of WBS methodic. The estimated workload is assigned to these steps, which is an input for evaluation of replication method.

Keywords: Business Intelligence, Data warehouse, Replication, Work breakdown structure.

Obsah

SEZNAM OBRÁZKŮ:	9
SEZNAM TABULEK:	10
SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ:	11
1 ÚVOD	12
1.1 CÍLE PRÁCE A METODIKA.....	13
2 DATABÁZE	15
2.1 SQL	15
3 BUSINESS INTELLIGENCE	17
3.1 PŘÍKLADY CÍLŮ BUSINESS INTELLIGENCE	18
3.2 BENEFITY VYUŽÍVÁNÍ BI	19
3.3 NÁKLADY	20
3.4 RIZIKA	20
3.4.1 <i>Analýza dat</i>	22
4 DATOVÝ SKLAD	24
4.1 PRŮZKUMNÝ SKLAD	27
4.2 PŘÍPADY UŽITÍ	28
4.3 ARCHITEKTURA DATOVÉHO SKLADU	32
4.3.1 <i>Nezávislé datamarty</i>	33
4.3.2 <i>Sběrníková architektura</i>	33
4.3.3 <i>Hub-and-Spoke</i>	34
4.3.4 <i>Architektura centrálního úložiště</i>	34
4.3.5 <i>Federativní architektura</i>	36
4.3.6 <i>Porovnání architektur</i>	39
4.4 POUŽÍVANÝ SOFTWARE	40
4.4.1 <i>Teradata</i>	41
4.4.2 <i>Oracle</i>	41
4.4.3 <i>Amazon Web Services (AWS)</i>	41
4.4.4 <i>Cloudera</i>	42
4.4.5 <i>MarkLogic</i>	42
5 LOGICKÉ KOMPONENTY BUSINESS INTELLIGENCE	43
5.1 ZDROJOVÉ SYSTÉMY.....	43
5.2 KOMPONENTY DATOVÉ TRANSFORMACE	43
5.2.1 <i>Extract, Transform, Load – ETL</i>	43

5.2.2	<i>Enterprise application integration – EAI</i>	44
5.3	DATABÁZOVÉ KOMPONENTY	44
5.3.1	<i>Dočasné úložisko</i>	45
5.3.2	<i>Operativní úložisko</i>	45
5.3.3	<i>Datamarty</i>	45
5.4	ANALYTICKÉ KOMPONENTY	46
5.4.1	<i>On-Line Analytical Processing – OLAP</i>	46
6	WORK BREAKDOWN STRUCTURE	48
7	REPLIKACE CENTRALIZOVANÉHO ŘEŠENÍ	49
7.1	POPIS SITUACE	50
7.2	CENTRALIZOVANÉ ŘEŠENÍ V JEDNÉ ENTITĚ	51
7.2.1	<i>Vstupy</i>	51
7.2.2	<i>Výstupy</i>	53
7.2.3	<i>Modelová situace pro jednu entitu</i>	54
7.3	ESTIMACE NEBOLI ODHADOVANÁ PRACNOST PRO JEDNU ENTITU.....	58
7.4	ŘEŠENÍ VE VÍCE ENTITÁCH	67
7.4.1	<i>Výhody a nevýhody</i>	68
7.5	REPLIKACE ŘEŠENÍ DO VÍCE ENTIT.....	68
7.5.1	<i>Nástroje replikace</i>	69
7.5.2	<i>Proces replikace a její odhadovaná pracnost</i>	71
7.5.3	<i>Výhody a nevýhody</i>	79
8	VYHODNOCENÍ	80
8.1	NÁVRH NA BUDOUCÍ ROZVOJ.....	81
9	DISKUZE A ZÁVĚR	82
10	SEZNAM POUŽITÉ LITERATURY	83

Seznam obrázků:

Obrázek 1 - Kategorizace a logické umístění funkčních vrstev.....	14
Obrázek 2 - Definice tabulky	16
Obrázek 3 - Subjektová orientovanost	25
Obrázek 4 - Integrace neboli centralizace.....	26
Obrázek 5 - Stálost	27
Obrázek 6 - Časová rozlišitelnost	27
Obrázek 7 - Schéma datového toku s jednotlivými fázemi úpravy a zajištění dat. ...	29
Obrázek 8 - Schéma Hub and Spoke architektury.	34
Obrázek 9 - Schéma využívání federativních datových skladů	38
Obrázek 10 - Zastoupení architektury.....	40
Obrázek 11 - OLAP kostka.....	47
Obrázek 12 - Náklady na replikaci řešení	78
Obrázek 13 - Porovnání dvou přístupů v implementaci.....	80

Seznam tabulek:

Tabulka 1 - Klasické BI technologie.....	14
Tabulka 2 - Porovnání architektur.....	39
Tabulka 3 - Odhad pracnosti pro jednu entitu v jednotkách MD.....	65
Tabulka 4 - Parametrizační tabulka.....	70
Tabulka 5 - Odhad pracnosti v replikované entitě v jednotkách MD	76
Tabulka 6 - Náklady na replikaci řešení	78
Tabulka 7 - Porovnání dvou přístupů v implementaci	81

Seznam použitých zkratek a symbolů:

BI	Business Intelligence
CRM	Customer Relationship Management
DM	Datamart
DSA	Data staging area
DW	Data Warehouse
EAI	Enterprise Application Integration
EBIT	Earning before Interests and Taxes
ERP	Enterprise Resource Planning
ETL	Extract Transform Load
KPI	Key Performace Indicators
ODS	Operational Data Store
OLAP	Online Analytical Processing
RDF	Resource Description Framework
ROA	Return on Assets
SCM	Supply Chain Management
SQL	Structured Query Language
WBS	Work Breakdown Structure

1 Úvod

Charakteristikou této doby jsou informace. Data, interpretovaná v informace, transformovaná na znalosti, proudící k lidem skrze kanály, které byly vynalezeny v minulých letech. Mezi ty nejdůležitější řadíme televizi, mobilní telefony, ale zejména internet. Tak se stává, že jednou z nejcennějších výhod na trhu, je právě vlastnění a správné využití těchto informací.

S rychlým rozvojem informační gramotnosti jak široké veřejnosti, tak i podniků, se stává využití informačních systémů a služeb nedílnou součástí podnikových strategií a to nejen velkých korporátních společností, jako tomu bylo v minulosti, ale také malých a středních podniků.

Informace jsou hlavním tématem této diplomové práce, konkrétně jejich technologické zpracování a vyhodnocování – neboli problematika Business Intelligence v datových skladech. Vzhledem k rychlému vývoji informačních a komunikačních technologií, celkovému trendu společnosti integrovat tyto technologie do veškerých oblastí života a k rychle se měnícím potřebám podniků, se i chápání tohoto pojmu v definicích častokrát rozchází. S každou další interpretací získáváme nový rozměr vnímání jak Business Intelligence, tak i jejích komponent. V první části této diplomové práce se tedy zaměřím na samotný pojem Business Intelligence a na jeho různorodé interpretace. Vysvětlím, jaké jsou hlavní cíle a především výhody využití Business Intelligence v podniku.

K efektivnímu využívání dat a informací je nezbytné tato data shromažďovat a smysluplně je spravovat. Z tohoto důvodu se v další části své práce zaměřuji na problematiku datových skladů a jejich druhy architektury. Vysvětlím praktický přínos využití konkrétní architektury a v závěru kapitoly provedu srovnání a pokusím se vyjádřit výhody a nevýhody jednotlivých struktur datového skladu. Společnosti, které se rozhodnou využívat Business Intelligence v datových skladech, mají několik možností pro volbu vhodného softwaru, které provozují databáze. V další kapitole představím tyto používané softwary, a to přes Teradatu – tržního leadera na trhu, až po soukromé společnosti jako je například MarkLogic.

Získané teoretické poznatky o Business Intelligence a jejích komponent, datových skladech, architektuře a softwarech budu následně interpretovat na praktickém příkladu

společnosti v bankovní sféře s pobočkami po celém světě, která potřebuje spravovat informace nejen o svých interních klientech, ale také potřebuje spravovat konsolidovaná data napříč entitami. V první části práce popíši možnost řešení za předpokladu, že každá jednotlivá entita spravuje data sama. Následně se zaměřím na replikaci Business Intelligence řešení, tedy kopírování již funkčního modelu jedné entity, napříč ostatními entitami. Závěrem této diplomové práce je popsat výhody tohoto řešení a srovnání mezi prvním modelem, kdy si každá entita nechá vypracovat své centralizované řešení, a replikací Business Intelligence řešení.

1.1 Cíle práce a metodika

V této práci bude dodržována standardní metodika vypracování diplomové práce. Na základě obsahové analýzy zacílené na téma Business Intelligence, architektura Business Intelligence řešení a využití Business Intelligence. Díky obsahové analýze zkoumaného tématu budou navrženy nástroje a přístupy pro zvýšení efektivity implementace replikovaného řešení. Cílem diplomové práce je aplikovat replikované řešení na modelovou situaci pomocí metody WBS a vyhodnotit její efektivnost. Zhodnocení bude probíhat na základě porovnání odhadované pracnosti dvou přístupů, kdy prvním přístupem je implementace BI řešení každou entitou samostatně a druhým přístupem je právě aplikace replikovaného řešení.

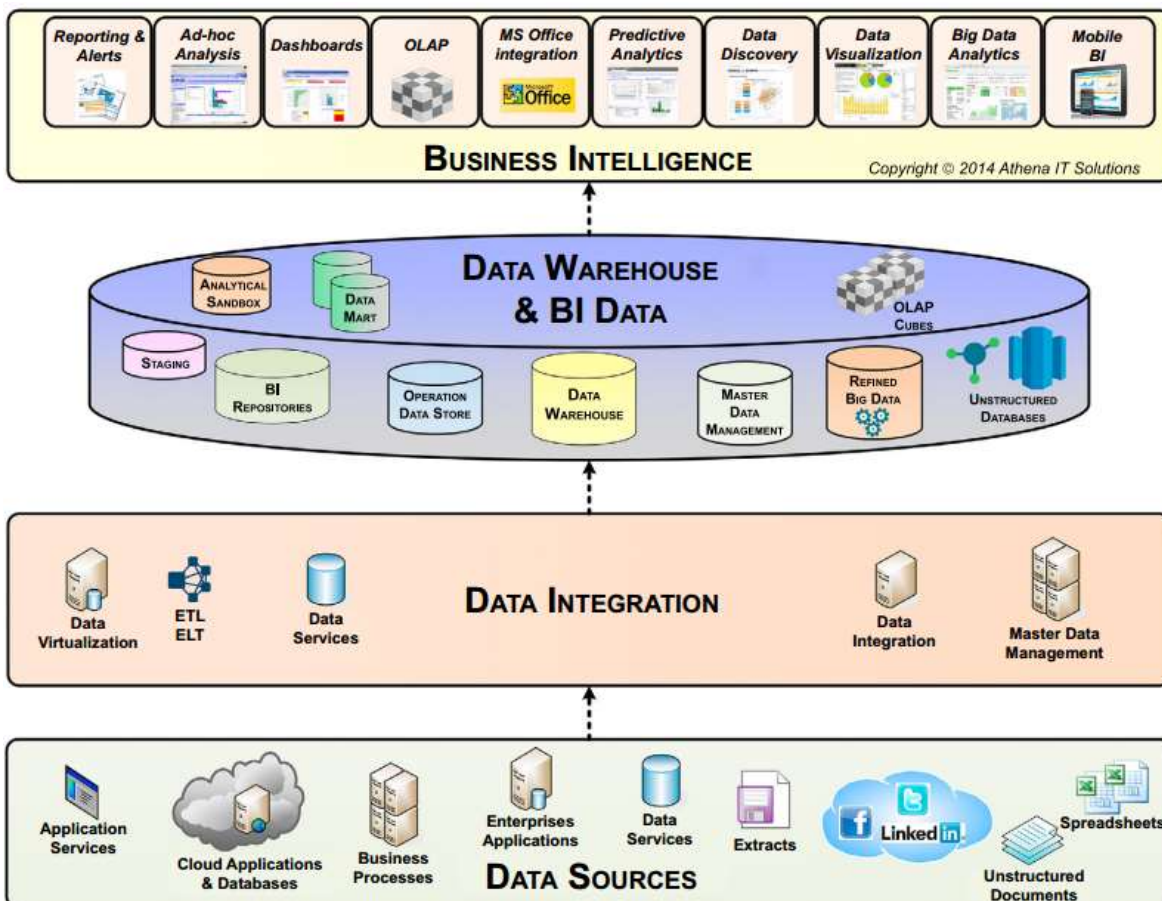
Práce bude následovat obecně přijímaná řešení replikace BI řešení v různých odvětvích lidské činnosti a nemá za cíl objevovat nová a neprozkoumaná řešení, které by samozřejmě samy o sobě vyžadovaly mnohem důkladnější a hloubkovou analýzu požadavků na data, výstupy, ale i architekturu z hardwarového a softwarového pohledu, analýzu nejnovějších trendů a rozbor současného lidského poznání na úrovni programovacích jazyků, cloudových řešení a teoretických konceptů, které slibují posun v paradigmatu tohoto tématu. Samozřejmě aby byla tato práce aktuální a pouze necitovala již známé, budou tyto nejnovější trendy zmíněny z pohledu klasických přístupů a ověřených řešení. Tím dostaneme synergii informací, které budou nejen odpovídat na naši prvotní otázku a budou tak naplňovat vytyčené cíle, ale zároveň předestřeme budoucnost BI jako takového.

Replikace BI řešení bude tedy sledovat standardní vypracování řešení, které můžeme vidět na Obrázek 1. Tyto standardní technologie rozpadnuté do funkčních vrstev jsou také shrnuté v tabulce 1 (Sherman, 2014).

Tabulka 1 - Klasické BI technologie

Funkční vrstva	Klasické technologie
Business intelligence	BI reportovací nástroj a BI reporting.
Datový sklad	Firemní datový sklad postavený na relační databázi.
Datová integrace	Nástroj ETL běžící v dávkové aktualizaci skladu přes noc.
Datové zdroje	ERP systémy běžící na tomto podnikovém datovém skladu.

Zdroj: Sherman, 2014



Obrázek 1 - Kategorizace a logické umístění funkčních vrstev

Zdroj: Sherman, 2014

2 Databáze

Pro potřeby této práce je nutné definovat pojmy databáze a objekty v ní. Termín databáze je dle Národní knihovny České republiky definován jako:

„Systém sloužící k modelování objektů a vztahů reálného světa (včetně abstraktních nebo fiktivních) prostřednictvím digitálních dat uspořádaných tak, aby se s nimi dalo efektivně manipulovat, tj. rychle vyhledat, načíst do paměti a provádět s nimi potřebné operace - zobrazení, přidání nových nebo aktualizace stávajících údajů, matematické výpočty, uspořádání do pohledů a sestav apod. Základními prvky databáze jsou data a program pro práci s nimi. Datový obsah tvoří množina jednotně strukturovaných dat uložených v paměti počítače nebo na záznamovém médiu, jež jsou navzájem v určitém vztahu a tvoří určitý celek z hlediska obsažených informací; data jsou přístupná výhradně pomocí speciálního programového vybavení - systému řízení báze dat.“(Nkp.cz, 2014)

Hlavním objektem databáze je pak tabulka, která zpravidla shromažďuje data o jednom druhu objektu reálného světa, slouží k uložení dat do databáze. (Muni.cz, 2015) Každá databázová tabulka dvourozměrná tabulka tvořená sloupci a řádky, přičemž sloupce reprezentují vlastnosti a jsou také nazývány atributy, zatímco řádky reprezentují samotné záznamy v tabulce. Atributy mají pevně stanovený význam i vlastnosti, tedy datové typy, a počet. Je tedy nemožné, aby záznamy existující v databázové tabulce obsahovaly různé informace s různými počty atributů. Pomocí použité databáze jsou vlastnosti tabulky omezeny, v případě jazyka SQL se jedná například o název položky, její datový typ, omezení délky jednotlivého atributu, či informace, zda je pole vyplněno povinně či volitelně. (Vse.cz, 2010).

2.1 SQL

SQL je zkrácený výraz převzatý s anglického Structured Query Language, což ve volném překladu znamená strukturovaný dotazovací jazyk. (Taylor, 2013) Jedná se o nejpoužívanější programovací jazyk speciálně navržený pro databáze, který umožňuje vytvářet databáze, definovat vlastnosti tabulek, přidávat do nich nová data, udržovat je a také k datům následně přistupovat, zobrazovat, či načítat pouze vybrané části dat. Tento programovací jazyk byl vyvinutý v sedmdesátých letech minulého století v IBM a v průběhu let se rozrostl a stal se standardem. Na následujícím obrázku můžeme vidět velmi

zjednodušenou definici tabulky vytvořenou pomocí SQL příkazu. Vidíme zde definici jména tabulky, všech jejích atributů společně s datovými typy a povinností. Z příkladu je patrné, že všechny položky tabulky Party_Address jsou povinné a pomocí primárního indexu jednoznačně identifikovatelné pomocí pole Party_Id.

```
CREATE SET TABLE Party_Address  
(  
  Party_Id CHAR(10) NOT NULL,  
  Country_Id_ISO CHAR(2) NOT NULL,  
  Postal_Code CHAR(6) NOT NULL,  
  Address_Street VARCHAR(80) NOT NULL,  
  Address_Town VARCHAR(40) NOT NULL)  
PRIMARY INDEX Party_Address_NUPI ( Party_Id )
```

Obrázek 2 - Definice tabulky

Zdroj: vlastní

3 Business Intelligence

Pokud hledáme definici pojmu Business Intelligence (BI), najdeme první odkaz již v roce 1958 (Luhn, 1958). Počínaje vymezením pojmů Intelligence jako „schopnost pochopit vzájemné vztahy předkládaných skutečností tak, aby vedly kroky k dosažení požadovaného cíle“ a Business jako „sbírku činností vykonávaných za jakýmkoli účelem, ať už jde o vědu, technologii, obchod, průmysl, právo, vládu, obranu atd.“ (Grossmann & Rinderle-Ma, 2015). Dohromady můžeme říci, že Business Intelligence je termín, označující celý komplex činností, úloh a technologií, které dnes stále častěji tvoří běžnou součást řízení podniků a jejich informačních systémů (Novotný et al., 2005). Můžeme také říci, že pod označením Business Intelligence si lze představit především výkonné analytické a vykazovací nástroje, které umožňují využít firemní data nejen k analýze již proběhlých jevů, ale také k predikcím budoucího vývoje (Khudhur, 2007). Technicky můžeme říci, že Business Intelligence je zastřešující termín, který se vztahuje ke znalostem, procesům, technologiím, aplikacím a postupům, které usnadňují podnikové rozhodování. Technologie Business Intelligence pracuje s použitými (historickými) daty v požadovaném kontextu a pomáhá přijímat podniková rozhodnutí pro budoucnost (Laberge, 2012). Nutno podotknout, že termín BI není zcela uspokojivě přeložitelný do českého jazyka, proto jak většina odborné veřejnosti, tak i toto pojednání zůstává u anglické verze.

Některé publikace a definice se rozcházejí v kategorizaci BI, respektive můžeme vidět dva přístupy k tomu, co vlastně BI je nebo co obsahuje. První z nich označuje BI jako jeden z mnoha nástrojů používaných pro podniková rozhodování. Zařazuje ho na stejnou úroveň jako datové sklady, data mining apod. Druhý názor pak přistupuje k BI jako k širokému komplexu všech nástrojů. Tato práce se pro své potřeby přiklání k druhému přístupu, tedy že BI není nezávislý nástroj, nýbrž je úzce spjat s ostatními.

Co ale tento pojem doopravdy znamená? Představme si BI jako nástroj, využívající historická data k učinění rozhodnutí ovlivňující budoucnost, a to na základě předem stanovených kritérií, určujících úspěch. BI umožňuje společnostem, konkrétně jejich řídicím pracovníkům a manažerům, se za použití spolehlivých dat informovaně rozhodovat o činnostech, procesech, strategických aktivitách atd. Obecně si pod BI můžeme představit využití aktivit, zdrojů či nástrojů k získání relevantního obrazu informací tak, aby nám efektivně posloužil v rozhodovacím procesu.

3.1 Příklady cílů Business Intelligence

Hlavním cílem Business Intelligence v podnikání je pomoci podnikovým manažerům, obchodním manažerům a dalším pracovníkům pracovat s lepšími a informovanějšími obchodními rozhodnutími. Společnosti také využívají technologii BI ke snižování nákladů, identifikaci nových obchodních příležitostí a zjišťování neefektivních podnikových procesů, které se dají zrekonstruovat.

Konkrétně můžeme tyto informace shrnout do několika bodů:

- Zjišťovat, kdo jsou nejlepší zákazníci společnosti, předpovídání množství peněz, které by byli ochotni vynaložit v příštích třech měsících potenciálním propojením s obchodními zástupci společnosti.
- Předvídat spotřebu energie v příštích dvou týdnech na základě průměrné poptávky po obdobných obdobích modulovaných různými povětrnostními podmínkami.
- Automaticky vyhodnotit globální ceny komodit a dynamicky vyvážit vysoké klientské investiční portfolia čistého kapitálu pro snížení rizika při zachování vysoké míry návratnosti.
- Průběžně kontrolovat příjmy a náklady současně s analýzou nejnákladnějších produktů společnosti.
- Monitoring klíčových aspektů, které mají vliv na spokojenost zákazníků pomocí agregace dat z tradičních informačních zdrojů a ze sociálních médií tak, aby bylo možné rychle a efektivně identifikovat a opravit problémy.
- Zajistit přesné kontroly, aby byly dodržovány regulatorní požadavky, předpisy, avšak také i udržet dobré veřejné mínění a zabránit v šíření špatné publicity.

Všechny tyto scénáře sdílejí klíčové charakteristiky: výsledky analýzy dat naznačují akce pro konkrétní jednotlivé role, které mohou vést k obchodní výhodě. A to jsou jen některé příklady použití, které mohou být získávány pomocí BI (Loshin, 2012).

3.2 Benefity využívání BI

Hannula & Pirttimäki, 2003 provedli empirický výzkum 50 finských firem, aby zjistili, co BI představuje pro tyto společnosti a které výhody BI jsou pro ně nejdůležitější. Mnoho z těchto výhod bylo také identifikováno jinými autory (Negash, 2004; Taylor, 2007; Turban et al., 2010; Watson & Wixom, 2007 nebo Spruijt, 2014):

- Lepší a kvalitnější informace
- Lepší objevování hrozeb a příležitostí
- Růst znalostní základny
- Zlepšení sdílení informací
- Zvýšená efektivita
- Snadnější získávání a analýza informací
- Rychlejší rozhodování
- Úspora času
- Úspory nákladů

Tento seznam není vyčerpávající a lze najít další výhody popsané v jiných publikacích. Nicméně, většina výhod BI je společnostmi považována za nehmatatelné definice a pouze podporují podnikání jako takové (Spruijt, 2014).

Mnoho společností pak tyto klíčové informace sdružuje do indikátorů, kterými sledují zdraví firmy, případně další výkonnostní charakteristiky. Obecně se tyto indikátory označují jako klíčové ukazatele výkonnosti (KPI). KPI spojuje činnost podniku s cíli tím, že definuje měřitelné hodnoty úspěchu. KPI mohou odkazovat na některé aspekty výkonu obchodního procesu nebo na podnik jako celek. Lze rozlišovat mezi kvantitativními ukazateli prezentovanými jako čísla, praktickými ukazateli propojenými s procesy, směrovými ukazateli, říkajícími, zda se organizace zlepšuje či nikoli, akčními indikátory pro kontrolu zavedených změn nebo finančními ukazateli (Grossmann & Rinderle-Ma, 2015).

3.3 Náklady

Náklady na zavedení plnohodnotného BI poskytující všechny výhody zmíněné výše jsou samozřejmě vysoké. Tyto náklady se dají rozdělit do několika skupin (Negash, 2004; Watson & Ariyachandra, 2005; Spruijt, 2014):

- Hardwarové náklady: V závislosti na tom, co již existuje, je třeba zavést datový sklad speciálně pro BI. Pro podporu systému BI může být vyžadována aktualizace stávající infrastruktury.
- Náklady na software: Vedle nákladů samotného softwarového balíku BI může být zapotřebí další software, například předplatné různých zdrojů dat.
- Náklady na realizaci: Je třeba zavést systém, ale je třeba vzít v úvahu i náklady na údržbu a náklady na školení.
- Personální náklady: Nakonec bude zapotřebí zaměstnanců, kteří budou pracovat se systémem.

3.4 Rizika

Přestože BI má mnoho potenciálních výhod, jak je popsáno výše, je důležité také identifikovat rizika. Bohužel literatura je na toto téma velmi omezená. I když existuje řada článků, které se zabývají výhodami, jsou rizika podceňována. Existuje však jedno riziko, které se objevuje v řadě článků, kterým je kvalita dat (Strong et al. 1997; Spruijt, 2014).

Důvodem, proč je kvalita dat rizikem, je skutečnost, že řešení BI je pouze tak přesné a efektivní jako analyzované údaje, takže je nezbytná vysoká kvalita dat. Jinými slovy, kvalita činností a analýz je pouze tak dobrá jako kvalita základních údajů, na nichž jsou založeny. S nadsázkou lze pak říci, že nevzniká datový sklad, ale datová skládka. Je velmi důležité si tento fakt uvědomit při práci s informacemi generovanými nástroji BI. Spolehnout se na nepřesné, neúplné, nejasné a irelevantní informace pro rozhodování je potenciálně velmi nebezpečné; nic neohrožuje výkon a obchodní hodnotu datového skladu více, než nevhodná, nepochopená nebo ignorovaná kvalita dat (Spruijt, 2014).

Kvalita dat by proto měla být pečlivě vyhodnocena před použitím BI k podpoře rozhodování. Ztráta informací, nedostatečná informovanost (nejednoznačnost),

bezvýznamná data a nesprávná data byly identifikovány jako nejvíce pozorované problémy s daty (Wand & Wang, 1996). Odhaduje se, že více než polovina projektů BI selže kvůli problémům s kvalitou dat a že problémy s kvalitou dat zákazníků stojí podniky USA více než 600 miliard dolarů ročně (Isik et al., 2013; Spruijt, 2014).

Měřítkem kvality dat je jejich použitelnost. Tedy za vysoce kvalitní data považujeme takové údaje, které přesně slouží následnému použití uživateli. To znamená, že užitečnost a použitelnost jsou důležitými aspekty kvality (Strong et al., 1997). Můžeme říci, že kvalita dat závisí na jejich reálném využívání. Kvalita dat je pak relativní pojem, protože pro některé využití jsou data v dostatečné kvalitě, zatímco pro jiné využití jsou nekvalitní. Příkladem může být finanční analýza podniku ve srovnání s finančním auditem. Zatímco v první situaci může být požadována přesnost pouze v jednotkách tisíců korun, v druhém případě je požadována přesnost na halíř (Wand & Wang, 1996; Spruijt, 2014).

Dalším možným rizikem je fakt, že problémy s nekvalitními daty většinou není možné odhalit před testováním BI řešení, tj. těsně předtím, než je řešení nasazováno na produkci. Tato nešťastná překvapení se stávají, když se neprovede důkladná analýza vstupních dat a organizace se místo toho zaměřuje na vizuální stranu výstupních produktů BI, jako jsou dashboardy a vizualizace dat. Proto by demonstrace produktů vždy měly vždy obsahovat informaci, že podnikatelé je mohou používat pouze tehdy, pokud podkladová data přesně odrážejí podnikové procesy důležité pro rozhodování. Klíčové atributy datové kvality, které je třeba posoudit, jsou shrnuty do pravidla 5C (Sherman, 2014):

- Čisté (clean) – jsou data bez chyb?
- Konzistentní (consistent) – existuje mnoho překrývajících se zdrojů s nekonzistentními daty?
- Konformní (conformed) – může podnik analyzovat data v běžných a sdílitelných dimenzích?
- Aktuální (current) – jsou data aktualizována a dostupná v požadované frekvenci?
- Obsáhlá (comprehensive) – jsou data potřebná pro analýzu k dispozici v tuto chvíli?

Pro zjištění aktuálního stavu dat je nutná koordinace s odborníky na zdrojové systémy a provedení profilování dat. Model 5C dává dobrý vstupní rámec pro analýzu dat. Lze podle

něj určit rozsah integrace a pročištění dat, které bude následně BI projekt využívat tak, aby mohl poskytnout analytické údaje, které firma požaduje.

Největším rizikem pro použití BI řešení je především kvalita dat. Před využitím BI a získáním výstupu je nezbytné položit si následující otázku: „Je kvalita zdroje dat dostačující pro cíl, kterému slouží?“ Pokud zní odpověď ne, je třeba zvážit, zda bude výstup BI vůbec relevantní a zda splní očekávání. Pokud této otázce nebudeme věnovat dostatečnou pozornost, můžeme dojít k chybnému rozhodnutí, které může mít pro organizaci velice negativní následky.

3.4.1 Analýza dat

Za elementární vyhodnocení dat považujeme jejich profilování, popř. jiný – anglický – výraz „Data assessment“. Při něm se dozvídáme následující základní informace o datech:

- Vyplněnost jednotlivých atributů datové sady.
- Frekvenční analýza hodnot atributů.
- Typické a extrémní hodnoty.
- Doménová analýza.

Jako další úroveň hodnocení dat služba nabízí hodnocení datové kvality, kdy hodnotíme shodu s deklarovanými pravidly

- Konzistence atributů.
- Rozsah hodnot atributů.
- Referenční integrita.
- Kontrolní součty.

Součástí hodnocení datové kvality je i hodnocení shody/neshody s informačním standardem společnosti.

Nejkomplexnější úroveň představuje hodnocení informační kvality, kdy se datová kvalita posuzuje v kontextu použití dat. V tomto případě jsou výstupy individuální podle požadavku zadavatele a mohou jimi být například:

- Posouzení vhodnosti použití dat pro zamýšlený účel.
- Posouzení rizik plynoucích z použití dat.
- Návrh metrik pro měření kvality dat v kontextu jejich použití.

Vzhledem ke komplexnosti této úrovně hodnocení je potřeba získat od zadavatele podrobnou specifikaci požadavku a dohodnout s ním rozsah a formu výstupů, zpravidla formou analytické schůzky.

4 Datový sklad

K tomu, abychom mohli data efektivně využívat, měřit a hodnotit, je nutné je odněkud získat, uložit, podle potřeby zpracovat a také smysluplně prezentovat. Právě proto technologie BI obsahuje a využívá systém data warehouse, neboli datového skladu a nástroje nutné k zobrazování dat v kontextu. BI pouze data zpracovává, způsobem, kterým jim přiřazuje význam, interpretační a prezentační schopnost. Tato data tedy musí někde získat, a to v takové podobě, aby bylo možné je efektivně využít. Koncept technologie datového skladu byl zpočátku představen Devlinem a Murphym, 1988. Navrhovali konstrukci databáze pouze pro čtení, která uchovává historický datový údaj pro operační a nabídkové integrační nástroje pro uživatele, aby vyhledávali, co chtějí, pro podporu rozhodování a analýzu (Hwang et al., 2004).

Definicí, co je datový sklad je opět velké množství. Přímočará definice říká, že datový sklad je integrovaný subjektivě orientovaný, stálý a časově rozlišený souhrn dat, uspořádaný pro podporu potřeb managementu (Inmon, 2002). Více technická definice říká, že datový sklad je jediným logickým (ne nezbytně fyzickým) úložištěm pro transakční nebo provozní údaje společnosti. Datový sklad samotný data nevytváří, každý byte uvnitř skladu má svůj původ jinde ve společnosti (Scheps, 2008). Manažerská definice datového skladu pak může být, že datový sklad je systém, který umožňuje shromažďovat, organizovat, uchovávat a sdílet historická data. Zahrnuje „použitá“ data pocházející z provozních systémů, které data zachytávají a používají v kontextu své funkce (Laberge, 2012).

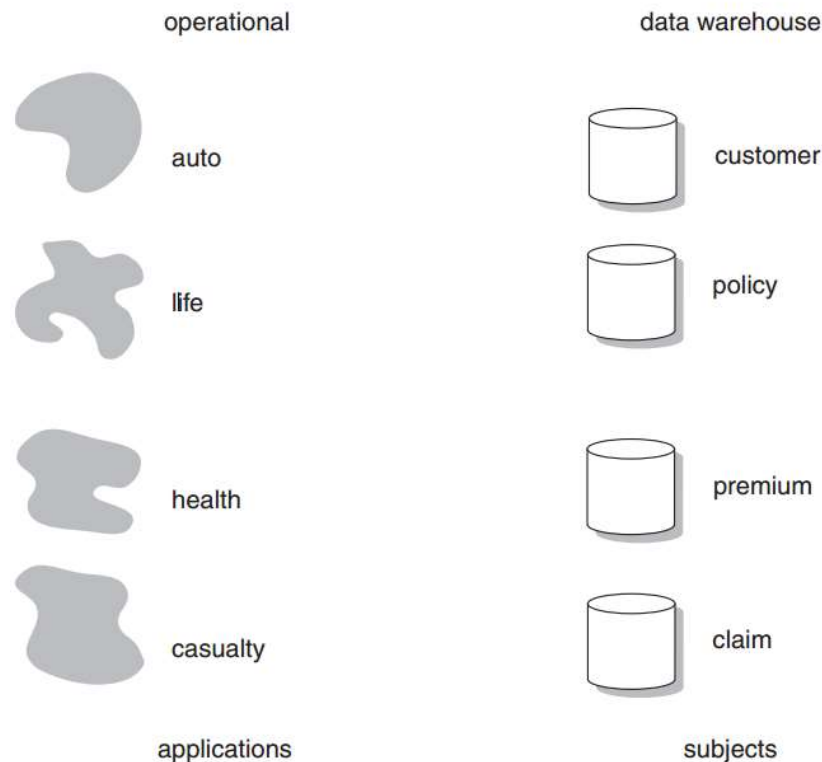
Samotný datový sklad musí splňovat následující náležitosti (Inmon, 2002):

- Subjektivá orientovanost (subject orientation)
- Integrace neboli centralizace (integration)
- Stálost (nonvolatility)
- Časová rozlišitelnost (time variance).

Subjektivá orientovanost datového skladu je zobrazena na obrázku 3. Klasické operativní systémy jsou organizovány kolem aplikací společnosti. Pro pojišťovací společnost mohou být žádosti o pojištění auta, zdraví, života nebo nehody. Hlavní předměty

pojišťovací společnosti mohou být zákazníci, pojistné a nároky. Pro výrobce mohou být hlavními oblastmi výrobek, objednávka, prodejce, kusovník a suroviny. V případě maloobchodu mohou být hlavními oblastmi produkt, SKU, prodej, prodejce a tak dále. Každý typ společnosti má svůj vlastní jedinečný soubor témat (Inmon, 2002).

Stručně můžeme říci, že data jsou ukládána podle jejich podstaty, nikoliv podle zdrojového systému, kde vznikla.

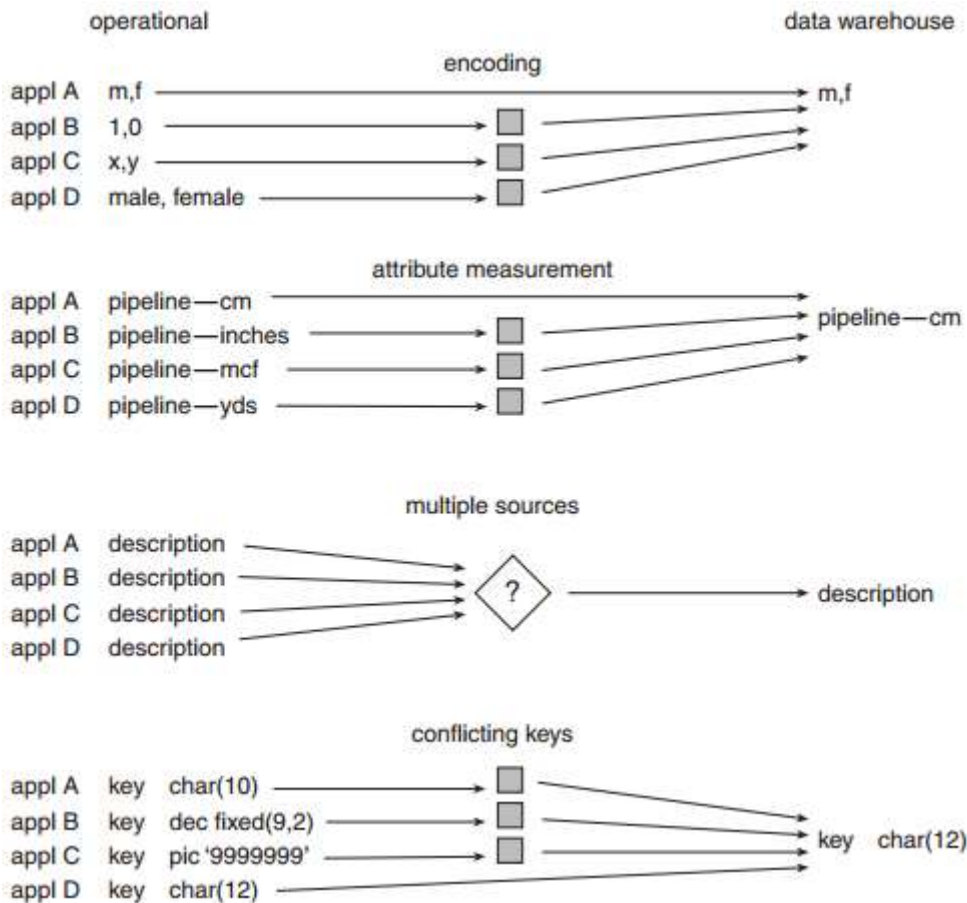


Obrázek 3 - Subjektová orientovanost

Zdroj: Inmon, 2002

Druhou charakteristickou vlastností datového skladu je to, že je integrován. Ze všech aspektů datového skladu je nejdůležitější právě integrace. Data jsou přenášena z více nesourodých zdrojů do datového skladu. Je tím míněno, že data jsou nahrána, zkonvertována, přeformátována, přeřazena a podobně. Výsledkem je, že data – jakmile se nachází v datovém skladu – mají jedno fyzické zobrazení. Obrázek 4 znázorňuje integraci, k níž dochází při přechodu dat z aplikačního operačního prostředí do datového skladu (Inmon, 2002).

Jinými slovy zde dochází k harmonizaci vstupů do standardizované podoby.

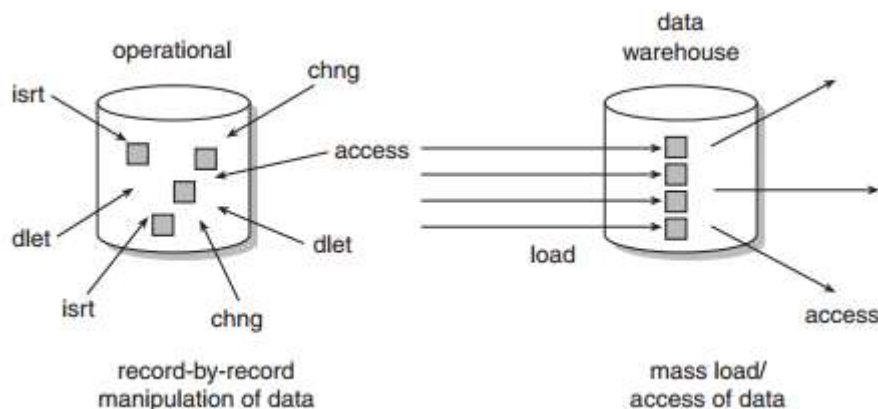


Obrázek 4 - Integrace neboli centralizace

Zdroj: Inmon, 2002

Třetím důležitým znakem datového skladu je, že je stálý. Obrázek 5 ukazuje stálost dat a ukazuje, že s jednotlivými záznamy z dat je manipulováno pouze jednou a najednou. Údaje datového skladu jsou načteny (obvykle najednou) a zpřístupněny, ale nejsou aktualizovány (ve všeobecném smyslu). Místo toho, když je jeden snímek dat do datového skladu načten a když dojde k následným změnám, je zapsán nový záznam snímku. Při tom dochází k uchování historie dat v datovém skladu (Inmon, 2002).

Pro shrnutí stálosti datového skladu, by se dalo říci, že v datových skladech jsou data pouze ukládána, tedy zde nevznikají, nezanikají a nemění se. Data jsou dále historizována a není možné je jakýmkoliv způsobem měnit.

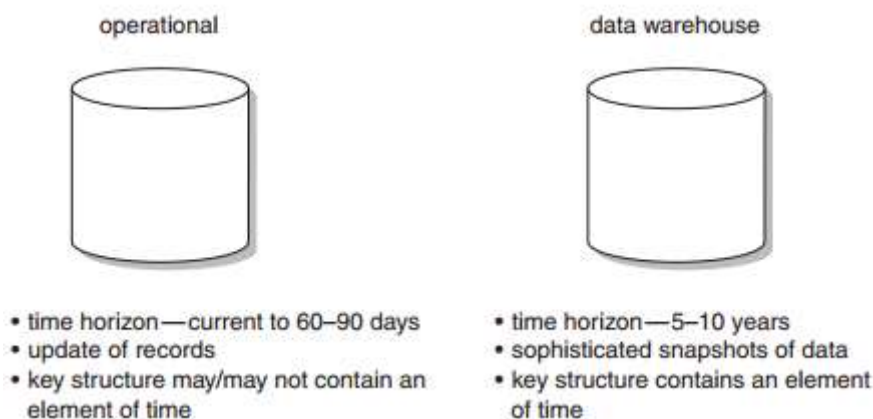


Obrázek 5 – Stálost

Zdroj: Inmon, 2002

Poslední charakteristickou vlastností datového skladu je časová rozlišitelnost. Tato vlastnost znamená, že každá jednotka informace je validní k určitému časovému okamžiku. V některých případech je tato informace rozlišena časovým razítkem. V jiných případech má záznam datum transakce. Ale v každém případě existuje určitá forma časového označení, aby se ukázal okamžik, k jakému datu je záznam validní. Obrázek 6 ukazuje, jak se může časový otisk zobrazit (Inmon, 2002).

Jinak řečeno, data jsou ve skladu historizována, čímž s sebou automaticky musí nést i časovou dimenzi, která je velice důležitá pro správnou interpretaci dat.



Obrázek 6 - Časová rozlišitelnost

Zdroj: Inmon, 2002

4.1 Průzkumný sklad

Pro správné pochopení pojmu je také důležité říci, co datový sklad není. Datový sklad není místem, kde dochází k vysoce výkonným transakcím, kde dochází k rozsáhlému

statistickému zpracování nebo místo, kde je umístěno smíšené a komplexní pracovní zatížení. Pokud je zapotřebí provést složité statistické zpracování, je potřebný průzkumný sklad. Průzkumný sklad má mnoho podobností s datovým skladištěm, avšak existuje zde řada zřetelných rozdílů. Mezi nejvýznamnější rozdíly mezi datovým skladem a průzkumným skladem patří (Inmon & Linstedt, 2015):

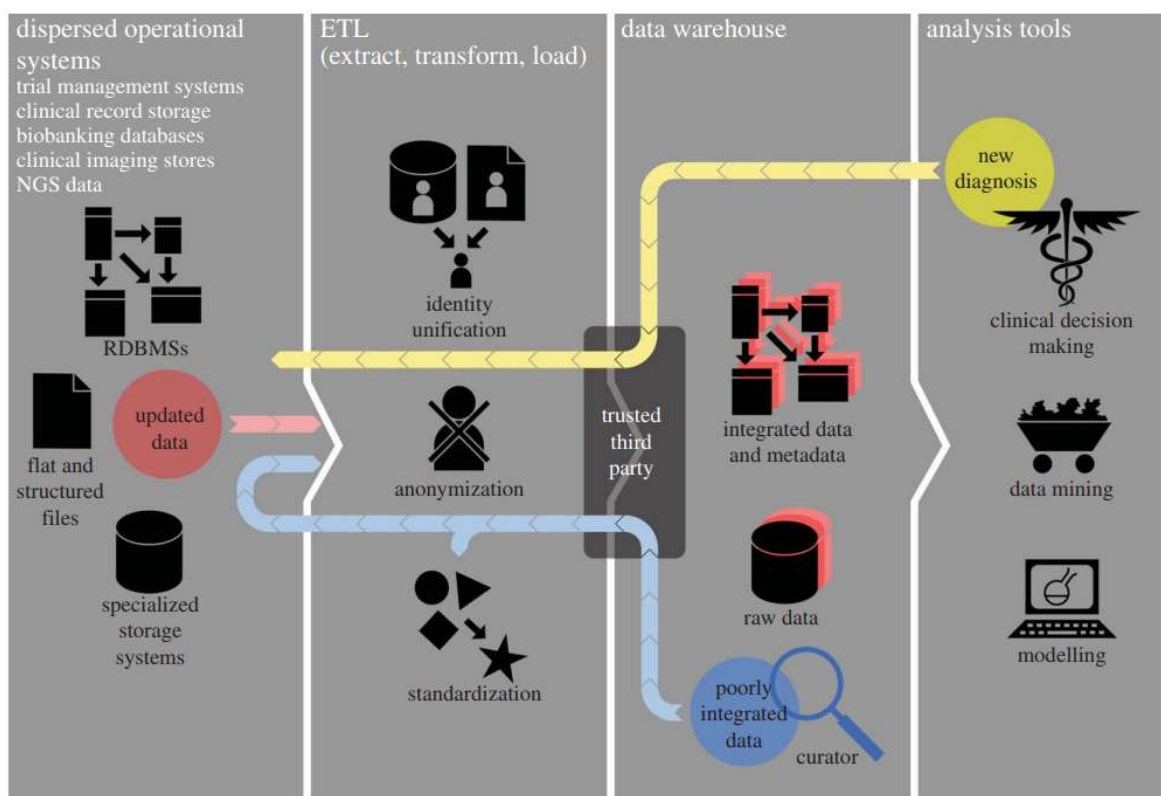
- Datový sklad je trvalá struktura, zatímco průzkumný sklad je postaven na základě projektu nebo podle potřeby.
- Datový sklad je vybudován tak, aby vyhovoval softwaru Business Intelligence (BI), zatímco průzkumný sklad je vybudován tak, aby vyhovoval softwaru pro statistickou analýzu.
- Datový sklad obsahuje data, která jsou vysoce normalizovaná, zatímco průzkumný sklad často obsahuje data, která jsou upravena pro vstupy do statistické analýzy, která bude provedena.
- Datový sklad obsahuje data z původního prostředí, zatímco průzkumný sklad obsahuje data z původního a externího prostředí. Ve skutečnosti datový sklad za normálních okolností neobsahuje mnoho (pokud nějaké) externích dat, zatímco průzkumný sklad obsahuje mnoho externích dat.

4.2 Případy užití

Vedle technického popisu datového skladu je důležitá i motivace, proč datový sklad s těmito požadavky vůbec používat. V následující kapitole bude představeno několik typických zástupců použití datových skladů.

Jako první příklad zde uvedeme lékařský projekt založený na kazuistikách a důkazech (EBMC²), který je zaměřen na analýzu procesů léčby rakoviny kůže. Projekt byl veden společným financováním mezi Medical University of Vienna, University of Vienna (katedrou dermatologie), Center of Medical Statistics, Informatics, and Intelligent Systems a výzkumnou skupinou Data Analytics and Computing, Knowledge Engineering a Workflow Systems and Technology. Cíle analýzy se týkají léčby pacientů i výkonů instituce (nemocnice) pomocí různých KPI (Grossmann & Rinderle-Ma, 2015).

Jako další lékařský projekt můžeme zmínit projekt p-medicine, který rozvíjí infrastrukturu spojující různé medicínsky relevantní prvky a ukládá je do systémů podpory klinických rozhodnutí. Jádrem tohoto projektu je shromažďování a standardizace údajů z klinických studií a systémů řízení pacientů za účelem meta analýzy, odvozování statistik, které se používají pro výpočetní modely a poskytování zdrojových dat pro podporu rozhodování. Takové údaje jsou obecně považovány za citlivé, přičemž předmětem je zdravotní stav jednotlivců, kteří nemusí chtít, aby byly informace sdíleny s jejich ošetřujícím lékařem, zejména pokud existuje možnost, že mohou být pacienti svým stavem přímo identifikováni. V důsledku citlivé povahy údajů musí být zavedena dostatečná metodika tak, aby byla získána důvěra pacientů a zároveň byly splněny právní normy. Proto, stejně jako omezení interoperability a standardizace různých zdrojů dat při budování datového skladu, tak i právní omezení hluboce ovlivňují architekturu klinických datových skladů. Ty lze obecně rozdělit na ochranu dat, zabezpečení dat a etické omezení (Jefferys et al., 2013). Aby byla zajištěna korektní správa dat splňující bezpečnostní kritéria, je nutné zajistit i vhodný datový tok. Příklad takové implementace je na obrázku Obrázek 7.



Obrázek 7 - Schéma datového toku s jednotlivými fázemi úpravy a zajištění dat.

Zdroj: Jefferys et al., 2013

Spojení medicíny a datových skladů v posledních několika letech poskytuje významné objevy na poli medicínské statistiky, o čemž vypovídá velké množství vědeckých prací publikovaných na toto téma – viz např.: VanderWeele et al., 2018; Garcelon et al., 2018; Dang et al., 2018; Marco-Ruiz et al., 2015 a další.

Druhým příkladem je Higher Education Processes (HEP) – vysokoškolský kurz na Fakultě informatiky, University of Vienna, kde byla pozorována data na základě využití výukové platformy CeWebs. Údaje se shromažďují ze čtyř odlišných služeb, tj. Fóra, podání, registrace a hodnocení kódu, na platformě zaměřené na vzdělávání CeWebs. K dispozici jsou záznamy z každoročně nabízených vysokoškolských kurzů po dobu 3 let (každoročně probíhající kurz). Celkově bylo 330 studentů a 18 511 akcí. V tomto případě použití byla data shromážděna ve formátu .csv a anonymizována.

Velice novátorský přístup použití datového skladu popisuje článek od Bouadi et al., 2017. Je zde probíráno navržení datového skladu pro ukládání a analýzu simulačních dat z prostorově distribuovaného agro–hydrologického modelu TNT2 (Topography–based Nitrogen Transfer and Transformations). Použití agro–hydrologických modelů umožňuje vědcům a zúčastněným stranám, aby reprezentovali, porozuměli a formulovali hypotézy o fungování agroenvironmentálních systémů a předpověděli jejich vývoj. Tyto modely vyhodnocované v datovém skladu generují simulace vlivů struktury krajiny, změny zemědělského systému a jejich prostorového uspořádání na kvalitu vody. Modely vytvářejí řadu mezivýsledků, které jsou dále spravovány, analyzovány a transformovány do použitelných informací (Bouadi et al., 2017).

Další velkou doménou použití datových skladů je problematika Data mining, neboli dolování dat. To lze charakterizovat jako sofistikované speciální metodiky k získávání předem neznámých, často skrytých a potenciálně užitečných informací z rozsáhlého databázového prostředí. Jedná se o speciální analýzy odvozené z obsahu dat, nejsou tedy předem definované uživatelem. Přínosem aplikace této techniky je objevování nových skutečností, které mohou přinést přidanou hodnotu zejména manažerům. Techniky mají zejména statistický podklad, ale mohou využívat také složité algoritmy či neuronové sítě. Příkladem využití dolování dat může být například analýza nákupního košíku, úvěrových rizik a pojistných podvodů, nebo také analýza rizika přechodu zákazníka ke konkurenci. Data mining využívá především tyto metody:

- Rozhodovací stromy – zobrazený pomocí prediktivního modelu, má strukturu stromu, kde jsou zdrojová data přiřazena do jednotlivých kategorií znázorněných jednotlivými uzly. Hlavní výhodou je její přehlednost a možnost interpretace.
- Neuronové sítě – vzorem umělé neuronové sítě jsou principy chování biologických struktur, jako například lidského mozku. Pomocí neuronových sítí a aplikací jejich algoritmů (předem stanovených i samoučících se) lze nacházet podobnosti v datech a tvorbu prediktivních modelů.
- Clustering – jedná se klasický nepřímý data mining, kdy uživatel nemá předem daná kritéria, a pouze doufá, že za použití nástroje odhalí související a užitečné informace. Pod clusteringem si můžeme představit shlukování dat do kategorií tak, aby si jednotky přiřazené jedné skupině byly charakteristikami více podobné než s ostatními skupinami. To nám umožňuje například objevovat různé segmenty v datech.

Jako poslední příklad můžeme uvést velice rozšířené použití datových skladů v bankovních institucích, které využívají pro správu svých dat a oddělení různé softwarové aplikace a nástroje – personální a účetní systémy, systémy pro správu poskytnutých půjček, klientské rozhraní pro internetové bankovníctví, online platby, zajištění technologického chodu poboček a jiné.

Z uvedených případů užití můžeme vyvodit několik závěrů. Uvedené společnosti produkují data z různých odvětví, oddělení, interní a externí informace. Všechny tyto nástroje zpracování informací využívají odlišnou terminologii, granularitu uložení dat, způsoby uložení neboli formát a mimo jiné také různý hardware k uložení. A právě díky návrhovým vlastnostem datových skladů můžeme data sdružovat a analyzovat, i když jde o fundamentálně jiné problematiky.

K dosažení správného rozhodnutí bude s největší pravděpodobností potřeba podnikové informace kombinovat. Vede nás to k potřebě data konsolidovat, agregovat a ukládat na jedno jediné místo. Tím se dostáváme k architektuře datového skladu.

4.3 Architektura datového skladu

Softwarová architektura je definována jako základní „struktura“ systému, zahrnující hlavní funkce systému, správu a distribuci dat, druh a styl jeho uživatelského rozhraní, platformu, na které běží a tak dále (Hohmann, L., 2003). Tato definice je konzistentní s dalšími popisy, např. od Basse et al., 2013, kteří definují softwarovou architekturu programu nebo výpočetního systému jako strukturu nebo struktury systému, které obsahují softwarové elementy, vlastnosti těchto elementů a vztahy mezi nimi.

Architektura je nezávislá na technologii a databázové platformě. Všechny typy relačních databází a on-line analytických zpracování dat (OLAP) mohou být plnohodnotně využity, pokud jsou navrženy v souladu s architekturou. Datové sklady se nevyhnutelně skládají z mnoha samostatných strojů s různými operačními systémy a systémy pro správu databází (DBMS). Jsou-li navrženy koherentně a pokud sdílejí jednotnou architekturu, výsledkem je sloučení do jednoho integrovaného celku (Kimball & Ross, 2002).

Přestože uvedené definice softwarové architektury jsou užitečné, jsou ze samotné podstaty problematiky natolik zjednodušující, že nejsou schopny vzít do úvahy komplexní záběr, kterým se architektura zabývá. Proto zde bude místo všeobjímající definice přistoupeno k problematice spíše z praktického úhlu pohledu.

Studie Meta Group zjistila, že výběr architektury je jedním z klíčových faktorů ovlivňující úspěšnost běhu datového skladu (Laney, 2000). I zpráva společnosti Gartner určila rozhodnutí o výběru architektury jako jednu z pěti problémových oblastí spojených s projekty datových skladů. Špatné rozhodnutí o výběru architektury může vést k problémům, jako je nedostatečná škálovatelnost, potíže s výkonem či ztráta základní vlastnosti a to „jediná verze pravdy“ (Strange, 2003). Ačkoli jsou datové sklady vybudovávány více než dvě desetiletí, výběr přístupu k budování není zcela jasně vymezen. Mimo jiné i proto se jednotlivá řešení potýkají se stále se opakujícími problémy při budování datového skladu a následného poskytování jednoznačných, přesných, integrovaných a včas reportovaných údajů. Proto se stále vede diskuze a vznikají rozporuplné názory, jakou architekturu je nejlepší využívat.

Vzhledem k důležitosti výběru architektury existuje překvapivě málo výzkumů na toto téma. Literatura má tendenci buď popsat architektury, poskytnout případové studie nebo prezentovat údaje průzkumu o popularitě různých možností. Svět se tedy spíše rozhoduje na

základě veřejné diskuze, kterou vedou dva světově uznávaní odborníci v oblasti datových skladů – Bill Inmon a Ralph Kimball – kteří stojí na opačných stranách. Bill Inmon obhájí Hub and Spoke architekturu (tj. centralizovaný datový sklad se závislými datamarty) nebo také Corporate Information Factory, či DW 2.0. Naopak Ralph Kimball obhájí sběrníkovou architekturu neboli Bus Architecture (Ariyachandra & Watson, 2006). V následujícím textu bude popsáno pět klasických architektur datového skladu, včetně dvou výše zmíněných.

4.3.1 Nezávislé datamarty

Za nejjednodušší formu architektury považujeme samostatné datové marty, jichž může existovat mnoho a navzájem nejsou nijak propojeny. Počátkem sedmdesátých let, kdy se začaly systémy pro podporu rozhodování vyvíjet, byly tyto systémy zásadně odlišné od toho, co známe dnes pod pojmem operační či transakční systémy. Ovšem i v nedaleké historii můžeme najít uplatnění pro tento způsob budování skladu a to sice na úrovni aplikací jako je Microsoft Excel či Access (Laberge, 2012).

Jedná se o aplikačně zaměřený přístup ke správě dat, jelikož úložiště byly navrhovány tak, aby vyhověly potřebám jedné či více aplikací. Podnikové oddělení zaměstnávalo své malé IT týmy, které načítaly data ze zdrojových systémů a spravovaly data způsobem vyhovující právě dané divizi, např. finanční, či marketingové. Díky tomu sice mohou splňovat lokální požadavky na data, ovšem neposkytují „jedinou verzi pravdy“ tolik nutnou pro data organizace. Datové trhy tak mezi sebou mají nekonzistentní definice dat a používají různé dimenze a metriky, které způsobují složitou analýzu dat skrze jednotlivé marty (Laberge, 2012).

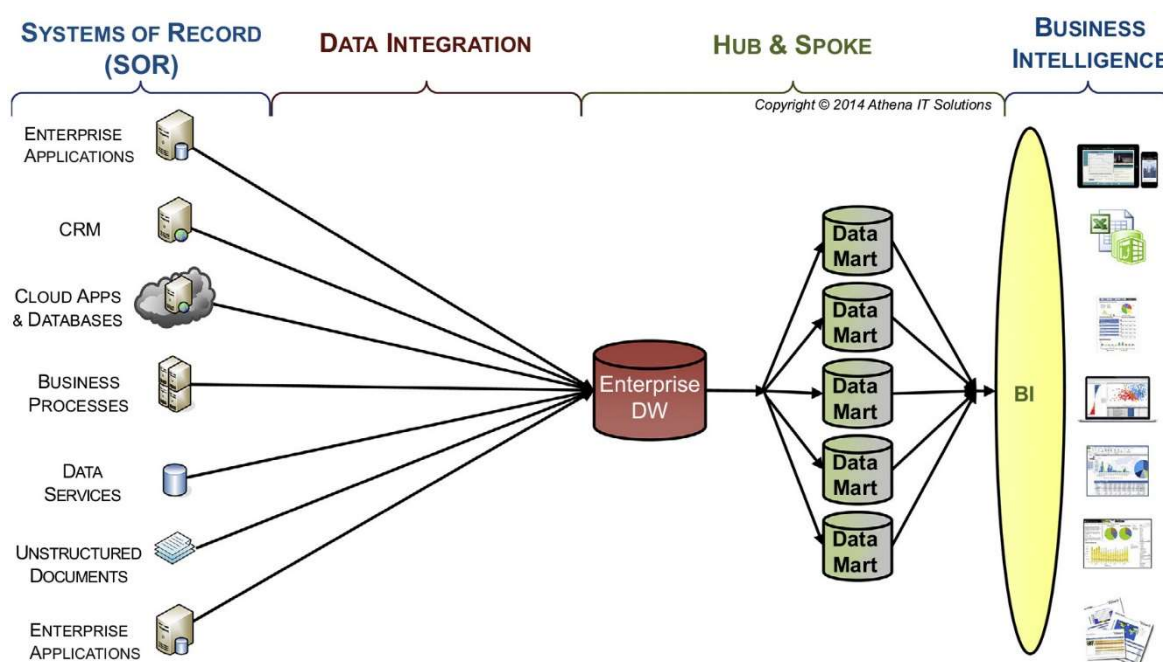
4.3.2 Sběrníková architektura

Sběrníková architektura poskytuje racionální přístup k správě podnikových dat v datových skladech. Během fáze návrhu architektury tým navrhuje globální standardizované dimenze a jednotnou interpretaci dat přes celý podnikový sklad. Tím se vytváří rámec pro architekturu dat. Poté je řešena implementace samostatných datových martů. Oddělené datové marty do sebe postupně zapadají a navzájem se podporují. V momentě, kdy dochází k interpretaci dat, je jednoduché spojit data z jednotlivých martů a dosáhnout tak ucelené informace podpořené více logickými celky.

Sběrníková architektura umožňuje manažerům datových skladů získat to nejlepší z obou světů. Mají architekturní rámec, který řídí celkový design, ale businessové problémy jsou rozděleny do datových martů, které mohou být implementovány v relativně krátkém čase. Samostatné vývojové týmy pro vývoj datamartů postupují dle architektury jádra a pracují poměrně nezávisle a paralelně. (Kimball & Ross, 2002).

4.3.3 Hub-and-Spoke

Architektonický přístup typu Hub-and-Spoke se postupně stával stále populárnějším, až se nakonec stal součástí architektonické osvědčené praxe. Hlavní rozdíl spočíval v tom, že Hub-and-Spoke postavil fyzický DW (datový rozbočovač) spíše než že se snažil dosáhnout virtuálního rozbočovače. Virtuální rozbočovač je jednoduchý návrh, ale v situacích reálného světa, tedy v implementačním prostředí se ukázal jako velmi složitý (Sherman, 2014).



Obrázek 8 - Schéma Hub and Spoke architektury.

Zdroj: Sherman, 2014

4.3.4 Architektura centrálního úložiště

Myšlenka centralizace datového skladu vychází z toho, že je vhodné shromáždit veškeré informace o datech, procesech, architektuře a použití z několika stávajících

datových skladů a sloučit je do jediného centrálního prostředí neboli systému. Tento projekt může mít logickou nebo fyzickou povahu.

Centralizovaný datový sklad je takovým konceptem, který obsahuje integrované údaje extrahované z více operačních systémů a sloučené s daty z externích informačních systémů. Jedná se o takovou implementaci, kdy jeden datový sklad slouží potřebám několika samostatných obchodních divizí. Tato architektura ukládá podniková data v atomární podobě normalizovaným způsobem a to při zachování plné časové historie. (Geekinterview.com, 2015)

Výhody architektury centralizovaného datového skladu (9Gauge.com, 2017):

- **Datová integrita: Jediný zdroj pravdy.** Jedním z hlavních účelů datových skladů je integrovat data pro potřeby analýzy výkonu, pochopení trendů a vytváření obchodních strategií. Díky umístění všech dat na jedno úložiště, za použití centralizovaného obchodního modelu, se zvyšuje důvěryhodnost dat, která nadále slouží pro přesnější analýzu. Tato přidaná hodnota vyplývá z důsledné standardizace dat pro všechny uživatele.
- **Úspora času a zlepšení efektivity:** Uživatelé mohou provádět více práce s přidanou hodnotou tím, že již nemusí velkou část svého času věnovat shromažďování, konsolidaci a kontrolou přesnosti dat z několika různých datových zdrojů. V mnoha případech jsou takto oddělená data v různých formátech, různé granularitě, což produkuje nutnost soustředit se na interpretaci analyzovaných dat.
- **Zabezpečení:** Centralizace poskytuje vysoký stupeň zabezpečení a kontroly nad datovým skladem.
- **Snadné řízení:** Centralizovaná databáze je jednodušeji spravována, jelikož operace vyžadující komplexní analýzu a dotazování již nevyžadují zvyšování složitosti sítě. Všichni uživatelé se navíc budou učit jedinému systému.
- **Centrální tým:** Pro potřeby analýzy a podpory konečných uživatelů není potřeba mít alokované analytiky v jednotlivých obchodních jednotkách. Díky centralizovanému skladu lze také aplikovat centralizovaný tým analytiků, kdy takový tým spadá většinou pod správu IT oddělení. Specialisté v tomto oddíle jsou pak schopni

rozumět datům napříč celým datovým skladem nehledě na businessové potřeby, dochází tak k synchronizaci veškerých informací v databázi bez nutnosti zaměření na konkrétní problematiku.

Nevýhody architektury centralizovaného datového skladu (9Gauge.com, 2017):

- **Technologická závislost:** Skutečnost, že jsou veškerá data uložena pouze na jednom místě, vede k nutnosti plánovaných výpadků při údržbě, ale i ke zvyšování rizika ztráty, či dočasné nedostupnosti při poruše takového skladiště. Toto riziko lze samozřejmě mitigovat dostatečným zálohovacím systémem, což ale nevyhnutelně vede ke zvyšování investice do celého řešení.
- **Náklady:** Protože jednotný datový sklad musí udržovat a denně procesovat tisíce záznamů, je nutné do takové investice vložit relativně vysoký kapitál. Dále se může ukázat, že případné snahy a požadavky o rozšíření datového skladu mohou být potenciálně velice nákladné.
- **Závislost na dodavateli:** Díky centralizovanému přístupu nezbyvá než se spolehnout na pouze jednoho dodavatele a tím se připravit o možnost se v budoucnu spolehnout na tržní ceny. Přejít k jinému dodavateli pak může být neúnosný náklad.

4.3.5 Federativní architektura

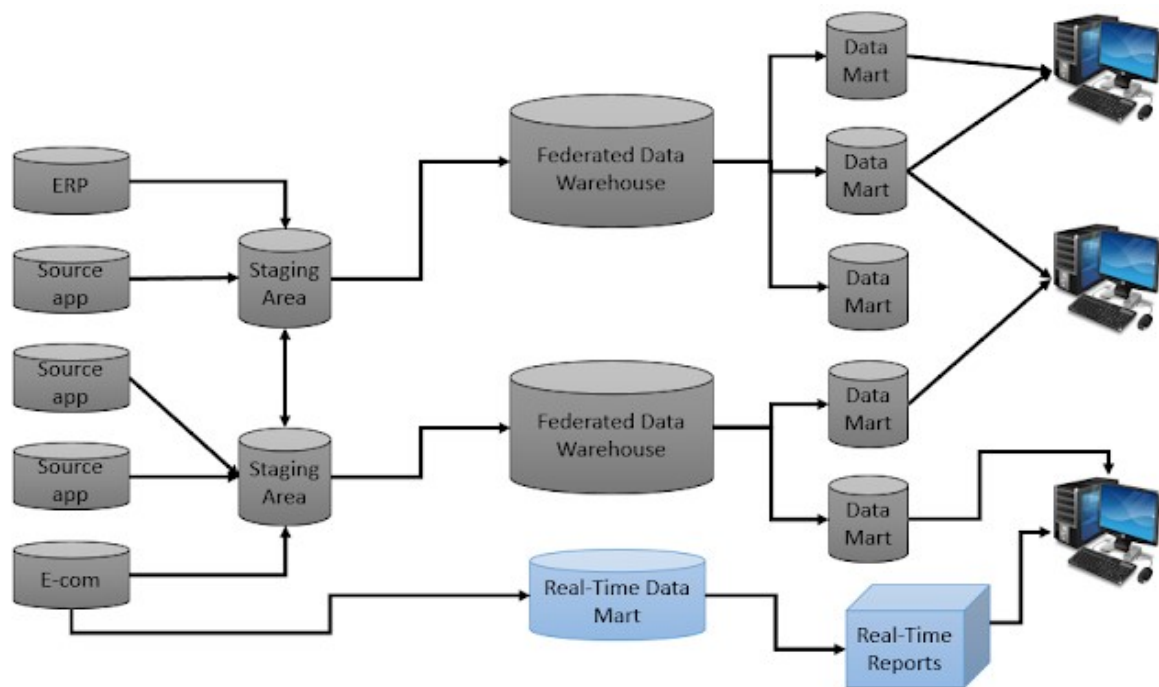
Jedná se o zvláštní architekturu multidatabázového systému, který transparentně mapuje více autonomních databázových systémů do jedné federativní sítě. Je to architektura používána k integraci heterogenních datových skladů a poskytuje jednotnou verzi pravdy v celé organizaci.

Jaké jsou důvody k implementaci federativní architektury? V dnešní době již není neobvyklé, že má společnost více poboček v různých regionech, ať už v rámci jednoho národa, či po celém světě. V takto velkých společnostech, kde se podnikání rozšiřuje na mnoho geografických celků, či kde jsou divize oddělovány organizační hranicí, se nabízí vybudovat několik datových skladů na úrovni jednoho vývojového týmu spravující jednu obchodní oblast na úrovni regionu, pro potřeby její analýzy a operaci s daty (Fernando, 2017).

V tomto případě je nutné mít pro každý sklad administrátora, který zajistí překonání překážek týkající se kompatibility, či výkonu. Kritickým problémem při komunikaci několika vzdálených zařízení je pak bezpečnost. Zabezpečení takové sítě může být značně finančně náročné a společnosti musí vynaložit více prostředků na vhodná technologická opatření.

Neméně obvyklou situací pak je snaha postavit Business Intelligence nad již existujícími heterogenními datovými sklady, které již mohly existovat před touto potřebou, například jako důsledek fúzí a akvizic. V takovém případě není efektivní opustit již vybudovaný datový sklad a vytvořit zcela nový tak, aby podnik dosáhl unifikované podoby dat, mnohem jednodušší je zavést federativní architekturu a integrovat tak datové sklady s cílem zachovat celistvost a spolehlivost dat bez přílišných dodatečných investic (Fernando, 2017).

Vzhledem k tomu, že základní databázové systémy zůstávají autonomní, federativní databázový systém pojednává o sloučení několika nesourodých databází, tedy žádná skutečná integrace dat neexistuje. Jedná se pouze o federovanou nebo také virtuální databázi, která umožňuje uživatelům a klientům ukládat a načítat data z několika nesouvisících databází jediným dotazem. Za tímto účelem musí být tato architektura schopna rozložit iniciální dotaz na několik poddotazů, a to i pomocí jiných programovacích jazyků (ZenTut.com, 2012).



Obrázek 9 - Schéma využívání federativních datových skladů

Zdroj: Fernando, 2017

K dosažení co nejúspěšnější implementace federativního datového skladu, by měla organizace dodržet několik bodů. Jednotlivé datové sklady by měly sdílet shodné dimenze, které jsou definovány ve společném obchodním modelu. Použití tohoto unifikovaného obchodního modelu vede ke sjednocení obchodních významů, datových struktur, či identických údajů tak, aby byla co nejvíce podpořena datová integrita mezi jednotlivými datovými sklady. Mezi další potřeby pro aplikaci federativního přístupu patří nutnost využívání stejného typu databáze, stejné hardwarové platformy, zálohování a archivace. Dále by měl být v rámci federativního datového skladu použit jediný nástroj ETL, který využívá jednotnou společnou databázi metadat za účelem snižování rizika konfliktu mezi různými datovými definicemi. Pro zajištění zabezpečení by měly všechny datové sklady využívat jeden bezpečnostní model. Výhody federativní architektury jsou především (van Leuken, 2012):

- Jednoduchost implementace: Federativní architektura se nesnaží přestavět původní datové sklady do jednoho, což často způsobuje neúspěch, naopak integruje všechny starší datové sklady a související BI řešení do nového systémů, zcela poskytujícího analytické schopnosti v rámci celé společnosti.

- Doba implementace: Díky výše zmíněným přístupům, je doba implementace tohoto systému je doba implementace znatelně kratší oproti vybudování nového centralizovaného řešení.
- Decentralizované zdroje a kontrola
- Paralelní vývoj
- Nezávislost

Naopak můžeme najít nevýhody takovéto architektury. Hlavními nedostatky mohou být např. (van Leuken, 2012):

- Požadavky na architekturu
- Požadavky na vývojové týmy

4.3.6 Porovnání architektur

Srovnání jednotlivých architektu se dá udělat různými způsoby. Jedním z přístupů mohou být celkové náklady, rozšiřitelnost, údržba apod. Článek od Ariyachandra & Watson, 2006 navrhli způsob srovnání pomocí čtyř obecných metrik: Informační kvalita, systémová kvalita, individuální dopady a organizační dopady. Výsledky jsou v tabulce 2.

Tabulka 2 - Porovnání architektur

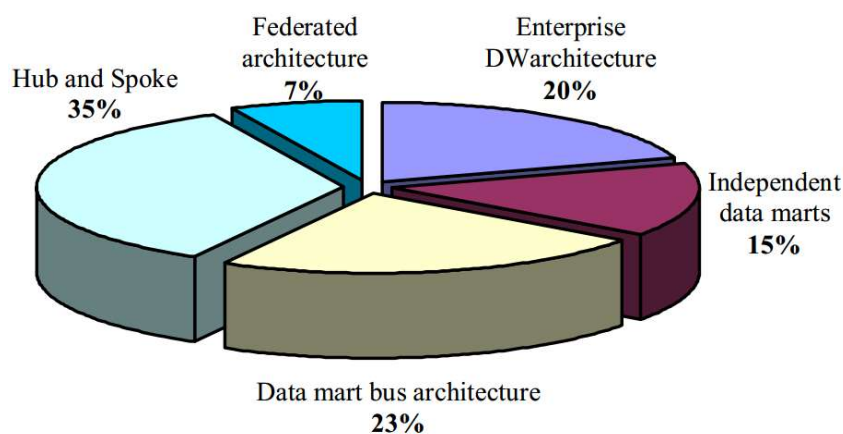
	Nezávislé datamarty	Sběrníková	Hub and Spoke	Centralizovaná	Federativní
Informační kvalita	4,42	5,16	5,35	5,23	4,73
Systémová kvalita	4,59	5,60	5,56	5,41	4,69
Individuální dopady	5,08	5,80	5,62	5,64	5,15
Organizační dopady	4,66	5,34	5,24	5,30	4,77
Průměr	4,69	5,48	5,44	5,40	4,84

Zdroj: Ariyachandra & Watson, 2006

Otázky používaly sedmibodovou stupnici, přičemž vyšší skóre naznačovalo úspěšnější architekturu. Hub-and-spoke, sběrníková a centralizovaná architektura získaly v průměru podobné výsledky v měřených metrikách. Nezávislé datamarty vykazovaly nejnižší hodnotu

u všech sledovaných metrik. Toto zjištění potvrzuje všeobecnou znalost, že nezávislé datamarty jsou nevhodné architektonické řešení (Ariyachandra & Watson, 2006).

Naopak co se týče využívání jednotlivých architektur, nalézáme zajímavé informace. Dle článku Alsqour et al., 2012. využití jednotlivých architektur v šedesáti zkoumaných společnostech v Polsku je rozdělené jinak, než bychom očekávali dle srovnávací analýzy z článku uvedeného v předchozím odstavci – viz Obrázek 10. Důvodem, proč je oproti výsledkům využíváno větší množství nezávislých datamartů je historické, jelikož nezávislé datamarty jsou levnější variantou architektury datových skladů. Naopak federativní architektura je ze své logiky méně využívána, jelikož jde o nákladné a často specifické využití, které je inherentně zvoleno jen v úzkém rozsahu případů.



Obrázek 10 - Zastoupení architektur

Zdroj: Alsqour et al., 2012

4.4 Používaný software

Software pro datové sklady provozuje databáze, které tvoří datový sklad společnosti. Software datového skladu nahrává data do stávající databáze a spouští dotazy, které vybírají datové soubory pro následnou analýzu.

Datový sklad funguje odděleně od databáze, která spouští každý den transformace dat společnosti a má uchovávat historická data z mnoha různých zdrojů, zatímco transakční databáze zapisuje nově získané informace do definované struktury skladu. Datový sklad zahrnuje všemožné typy dat přenášených z různých typů softwaru, jako je CRM, účetní software nebo ERP software.

Kvůli složitosti ukládání do datového skladu musí být používaný software vysoce propracovaný, schopen obhospodařovat velké množství dat a musí být schopen rozlišit a analyzovat data z nejrůznějších zdrojů (TechnologyAdvice, 2018).

4.4.1 Teradata

Teradata je tržním leaderem v oblasti datového skladu, více než 30 let. Jedním z klíčových vlastností je, že všechny funkce databáze (skenování tabulek, skenování indexů, připojení, třídění, vkládání, mazání, aktualizace, načtení a všechny nástroje) jsou prováděny paralelně po celou dobu. Další specialitou je skenování tabulky. Jednou z hlavních funkcí Teradaty je technika nazvaná synchronní skenování, které umožňuje skenovat požadavky, které jsou již v procesu. Maximální souběžnost je dosažena optimálním využitím každého skenování. Teradata udržuje dostatečně podrobný profil řízených dat, které efektivně prohledává skenováním pouze omezeného úložiště, kde mohou být nalezeny výsledky dotazu (McKnight, 2014; Walker, 2018).

4.4.2 Oracle

Oracle je již po desetiletí tak zažitá platforma, že její název je v podstatě synonymem relačních databázích a datového skladování. Databáze Oracle 12c je průmyslovou normou pro vysoce výkonné škálovatelné, optimalizované datové sklady. Výhody Oracle jsou např. služba Oracle Change Data Capture (CDC), která zjednodušuje proces identifikace změněných dat od poslední extrakce. Změny lze identifikovat buď synchronně, pokud jde o transakci, pomocí mechanismu založeného na spouštění, nebo asynchronním vyvedením archivovaných protokolů. Navíc heterogenní přenosné tabulkové prostory poskytují účinný mechanismus pro přesun velkého množství dat mezi databázemi Oracle na různých hardwarových platformách. Externí tabulky umožňují transformaci dat tak, jak jsou, ať už jsou načítány nebo vykládány z databáze. Unikátní funkcionalitou je automatická správa sdílené paměťové oblasti (SGA), která eliminuje potřebu určení optimálního přidělení paměti pro každou komponentu (Hobbs et al., 2005; Walker, 2018).

4.4.3 Amazon Web Services (AWS)

Posun paradigmatu v oblasti ukládání dat a skladování do cloudu v posledních několika letech byl významný a Amazon je tržním leaderem tohoto přístupu. Amazon nabízí

celou paletu nástrojů pro ukládání dat a zdrojů, které doplňují jeho platformu cloudových služeb. Existuje například Amazon Redshift, rychlé, plně spravovatelné řešení pro datové úložiště v cloudu. AWS Data Pipeline, webová služba určená pro přenos dat mezi stávajícími AWS datovými službami. Elastic MapReduce, která poskytuje snadno spravované řešení Hadoop na platformě služeb AWS (Radford, 2014; Walker, 2018).

4.4.4 Cloudera

Společnost Cloudera se v posledních letech stala významným poskytovatelem korporátního řešení pro ukládání a zpracování dat na bázi technologie Hadoop. Společnost Cloudera nabízí Enterprise Data Hub (EDH) pro svou řadu provozních datových skladů nebo datových skladů. EDH se zaměřuje se na dávkové zpracování, interaktivní SQL, podnikové vyhledávání a pokročilou analýzu - společně s robustním zabezpečením, řízením, ochranou dat a řízením. Datový sklad Cloudera je založen na open-source softwaru Hadoop. Organizace nabízí řadu různých balíčků služeb založených na Hadoopu, včetně Cloudera Express a Cloudera Enterprise (Walker, 2018).

4.4.5 MarkLogic

MarkLogic je soukromá softwarová firma založená v Silicon Valley. Byla založena v roce 2001 a nabízí podnikovou databázovou platformu NoSQL. Použití NoSQL a dalších alternativních forem skladování způsobuje další posun v paradigmatu datových skladů. MarkLogic je velice inovativní firma, která ve svém řešení má mnoho různých platform. Využívá SPARQL (sémantický dotazovací jazyk pro platformu RDF), pro poskytnutí bohatšího a hlubšího pohledu na data způsobem, který je v relačních modelech dosažitelný znatelně složitějším způsobem. Začlenění technologií založených na sémantických jazycích společně s cloud technologiemi a Hadoopem představuje další úroveň inovací, která udržuje datové sklady škálovatelné a přizpůsobitelné (Walker, 2018).

5 Logické komponenty Business Intelligence

Hlavní komponenty Business Intelligence se dají rozdělit na několik funkčních celků, lišících se používaným softwarem, hardwarem, náročností obsluhy a samozřejmě cíli.

Abychom mohli vůbec vyhodnocovat data, musíme je odněkud získat a upravit tak, aby byla použitelná v datovém skladu. Tím se zabývají zdrojové systémy. Následně se pomocí komponent datové transformace výchozí data připraví do žádané struktury. Po úpravě dat se ukládají do databázových komponent, ať už jen dočasně, tak především trvale do historizačních tabulek. Nakonec jsou tyto tabulky využívány jako zdroj informací pro analytické komponenty a reporting.

5.1 Zdrojové systémy

Zdrojové neboli operační, transakční či produkční systémy podniku sice nejsou součástí BI, ale jsou jeho primárním a často jediným zdrojem dat a jsou tedy pro fungování BI kriticky důležité. Jsou to systémy, které slouží k ukládání a zpracovávání podnikových transakcí a to v reálném čase a nejsou určeny k analytickým funkcím.

V podnicích lze nalézt mnoho druhů zdrojových systémů podporujících různá oddělení podniku, jako například ERP, SCM či CRM systémy. Tyto systémy se liší nejen svým určením, tedy obsahem, ale také použitou technologií, byly zaváděny v různých časových horizontech, ukládají se na různých hardwarových úložištích a to vede k jejich nekonzistenci. Díky tomu může být proces jejich získávání a integrace velice náročný, bereme-li v potaz objem a strukturu dat, jako i jejich formu.

5.2 Komponenty datové transformace

Data získané ze zdrojových systémů je potřeba přenést do datového skladu, který si ovšem, díky své specializované struktuře, může žádat data připravit do požadované formy, očistit je. K tomu využíváme komponenty datové transformace.

5.2.1 Extract, Transform, Load – ETL

Prvním a zároveň nejvýznamnějším krokem celého procesu BI je tzv. ETL – Extract, Transform, Load – také známým jako datová pumpa. ETL nástroj slouží zejména pro přenos

dat mezi dvěma a více systémy. Funguje na základě dávkového režimu, kdy jsou data získávána v určitých časových intervalech (denních, měsíčních).

Krok extrakce pokrývá získání dat ze zdrojového systému a zpřístupňuje jej pro další zpracování. Hlavním cílem fáze je získat všechny potřebné údaje ze zdrojového systému s co nejmenšími prostředky. Tento krok by měl být navržen tak, aby neovlivňoval nepříznivě zdrojový systém pokud jde o výkonnost, dobu odezvy nebo jakýkoli druh zamykání (DII, 2011).

Krok transformace používá soubor transformačních pravidel pro úpravu dat ze zdroje na požadované informace. To zahrnuje konverzi všech měřených dat na stejnou dimenzi pomocí stejných jednotek, aby se mohly později agregovat. Transformační krok také vyžaduje spojení dat z několika zdrojů, třídění, odvození nových vypočtených hodnot a použití sofistikovaných validačních pravidel. Tato data upravená a očištěná jsou vhodná pro potřeby dotazování a analýzy (DII, 2011).

Fáze loadu přemístí transformovaná data do trvalé cílové databáze. Jakmile je proces loadu dokončen, samotný proces ETL končí. Záleží pak na konkrétní situaci v organizaci, jak často je ETL spouštěno a datový sklad se tak průběžně aktualizoval s nejnovějšími údaji.

5.2.2 Enterprise application integration – EAI

Dalším z nástrojů datové transformace je EAI, jejímž úkolem je integrace primárních systémů podniku (často neschopných spolu navzájem komunikovat). Nástroj je využíván pro zjednodušení a automatizaci podnikových procesů a to bez nutnosti zásahu do fungování, či struktury již existujících aplikací. Hlavním rozdílem oproti ETL je schopnost pracovat v reálném čase.

5.3 Databázové komponenty

Při studiu databází a datových skladů je také vhodné popsat jaké jsou jejich komponenty. Tím je myšleno jaké nástroje jsou jimi využívány. Mezi databázové patří dočasné úložiště, operativní úložiště a datové marty.

5.3.1 Dočasné úložisko

Dočasné úložisko dat, anglicky pak Data Staging Area (DSA), má za hlavní úkol rychlou a efektivní extrakci netransformovaných dat ze zdrojových systémů. Použití toho nástroje je úzce spjata s ETL. Jde o nepovinnou komponentu k jejíž využití vedou dva důvody. Prvním jsou zatížené transakční systémy vyžadující přenášet data s minimálním dopadem na výkonnost. Druhým důvodem pak jsou systémy, u kterých je data nutné konvertovat do požadovaného formátu. Data v dočasném úložišti nejsou agregována, jsou nekonzistentní, nenesou s sebou časovou dimenzi, jedná se pouze o aktuální data, která se s dalším přenosem smažou a nahradí novým snímkem. Zvážíme-li uvedené charakteristiky uložení dat v DSA, dojdeme k závěru, že na tuto komponentu není možné aplikovat analytické nástroje, ani prezentační vrstvu.

5.3.2 Operativní úložisko

Operativní úložisko dat, anglicky Operational Data Store (ODS), se řadí mezi další komponenty vrstvy ukládání dat, jejíž existence v BI řešení není povinná. Hlavním přínosem ODS je podpora analytických procesů koncových uživatelů, na rozdíl od DSA, kam uživatelé nemají přístup. Cílem je uživatelům zpřístupnit data pro analýzu a to s minimální dobou odezvy po jejich zpracování. Data jsou stejně jako v DSA bez historie a jedná se pouze o aktuální data, rozdílem ovšem je jejich konsolidovanost, konzistentnost a subjektivní orientovanost.

5.3.3 Datamarty

Z pohledu architektury a charakteristik jsou datamarty (DM), velice podobné datovým skladům. Rozdílem je, že DM jsou problémově orientované, jsou vystavěné pro potřeby určitého obchodního procesu a jsou obvykle určeny omezenému okruhu uživatelů, zabývajícím se konkrétní podnikovou problematikou. Na design každého datového martu jsou aplikovány velice specifické požadavky, každé datové tržiště musí být definováno dimenzemi a v rámci jednoho datového skladu jsou všechna tržiště vybudována na základě shodných a odpovídajících dimenzí a faktech.

5.4 Analytické komponenty

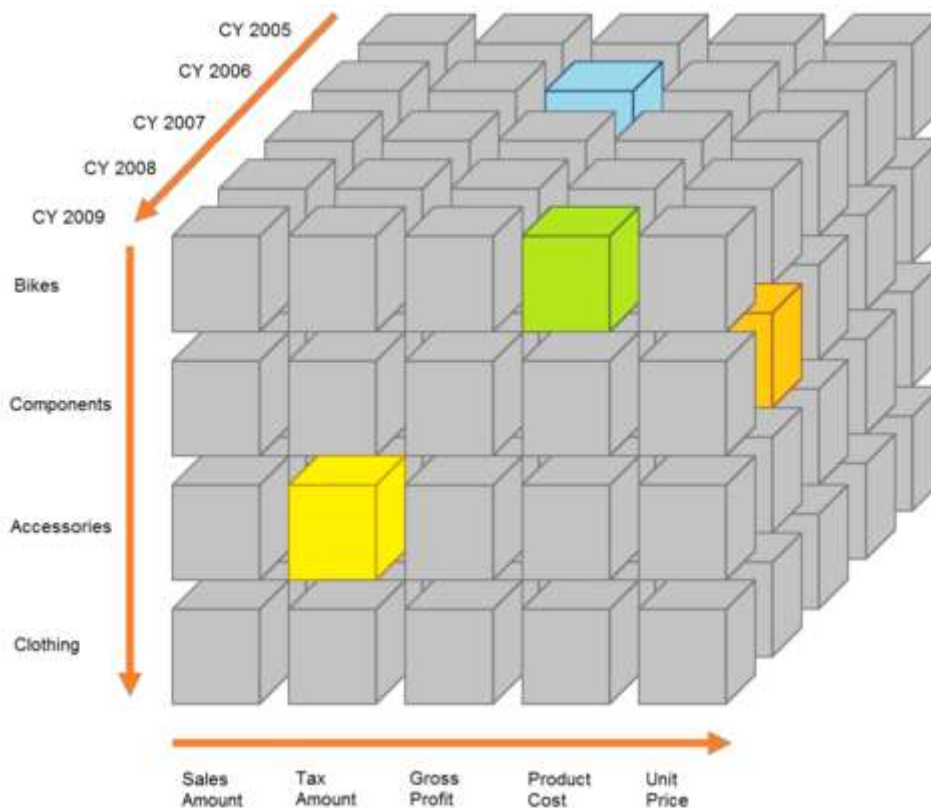
Za analytické komponenty považujeme ty nástroje, které používá přímo uživatel a které mu slouží k vytváření podnikových reportů a analýz vedoucí k podpoře konkurenceschopnosti a prosperitě. Neodmyslitelnou charakteristikou těchto nástrojů je jejich přívětivé uživatelské rozhraní.

Teprve až tyto analytické nástroje reálně dodávají výstupy z datového skladu a jsou využívány pro reporting. Dále mohou reporty sloužit pro auditní účely. Z časového hlediska můžeme reporting rozdělit na dva přístupy:

- Pravidelný reporting – SQL dotazy jsou spouštěny v pravidelných intervalech, zpravidla na denní či měsíční bázi,
- Ad-Hoc reporting – jednorázové dotazy formulované uživatelem, vznikající z aktuálních potřeb podniku, nevážící se k pravidelnému časovému horizontu,

5.4.1 On-Line Analytical Processing – OLAP

On-Line Analytical Processing je technologie založená na zobrazování dat z datového skladu za pomoci OLAP kostek. Mluvíme tedy o multidimenzionálním databázovém systému dat, kdy jsou každé OLAP kostce přiřazeny konkrétní dimenze, množiny sledovaných hodnot, jako je čas, segmentace či produkty. Úkolem takové kostky je umožnit uživateli nahlížet na data z různých úhlů pohledu, poskytovat možnost online data procházet, přímo provádět analýzy. Z pohledu časového začazení jde o pravidelný reporting. (Vaisman, 2014)



Obrázek 11 - OLAP kostka

Zdroj: Galaktikastoft.com, (2017)

- MOLAP = Multidimensional OLAP je klasickou formou užití OLAP a často jsou tato dva termíny navzájem zaměňovány. Pomocí MOLAP jsou data uložena v multidimenzionálních, neboli binárních kostkách.
- ROLAP = Relational OLAP je považována za efektivnější z pohledu práce s velkými objemy dat, zejména s modely dimenzí vyznačující se vysokou kardinalitou. Jedná se o technologii, která multidimenzionalitu řeší pomocí relačních databází, které musí být vystavěny právě pro tento účel.
- HOLAP = Hybrid OLAP využívá hlavní přednosti dvou předchozích technologií a to sice tak, že detailní data jsou uložena v relační databázi a agregované složky pak v binárních OLAP kostkách.
- DOLAP = Desktop OLAP není tak široce používána jako výše zmíněné. Je založena na extrakci potřebných dat pro konkrétní analýzu tak, že si stáhne pouze podmnožinu kostky a nad těmito osekávanými daty pak uživatel provádí analýzu. Hlavní výhodou je možnost uložení na lokální disk, čímž odpadá nutnost být připojen k serveru.

6 Work breakdown structure

Work breakdown structure (WBS – někdy též v češtině jako „Hierarchická struktura činností“) je proces dělení celkové práce na projektu do více strukturovanějších úkolů, které jsou hierarchicky uspořádány. Úroveň podrobností by měla představovat celkový rozsah projektu, přičemž úkoly by měly být zvládnutelné (Norman et al., 2008; Project management institute, 2008). Systém WBS je typicky navržen metodou shora dolů. Horní úrovně WBS jsou rozloženy do logických seskupení práce, následované další úrovní dolů a tak dále. Tak lze naplánovat složku nejnižší úrovně WBS a její náklady lze odhadnout, sledovat a kontrolovat. Na obr. 1 je znázorněna konstrukce rozložení práce projektu tunelu metra jako příklad (Siami-Irdemoosa et al., 2015).

Existuje mnoho různých metod, které lze použít k vytvoření WBS. Zatímco existuje obecná shoda, že WBS je základním manažerským prvkem, na kterém jsou založeny mnohé procesy řízení projektů, existuje překvapivě malá shoda ohledně nejlepší metody pro vytvoření WBS (Norman et al., 2008). Jednou z hlavních otázek v tomto ohledu je, jak lze optimální WBS identifikovat ze všech možných struktur. Ústav řízení projektů stanovil, že "kvalitní WBS je WBS zkonstruovaný tak, aby splňoval všechny požadavky na jeho použití v projektu". Při použití tohoto principu kvality je optimální WBS v komplexním podzemním projektu vysoce kvalitní struktura práce, kde konkrétní obsah a typ prvků WBS vhodně řeší plný soubor potřeb projektu. Příklady kvalitních charakteristik WBS v komplexním podzemním projektu jsou následující (Siami-Irdemoosa et al., 2015):

- Obsahuje specifické typy komponent WBS potřebné pro komplexní podzemní projekt.
- Poskytuje "dostatečné" detaily pro sdělování rozsahu komplexního podzemního projektu.
- Dosáhne "dostatečné" úrovně rozkladu pro efektivní komplexní řízení podzemního projektu.

7 Replikace centralizovaného řešení

Tato práce pojednává zejména o replikaci Business Intelligence řešení. Replikaci můžeme chápat uložení stejných dat na několika různých úložných zařízeních, nebo také stejnou výpočetní úlohu opakovanou v čase. Pro potřeby této práce je ale potřeba tuto terminologii chápat v souvislostech tohoto pojednání. Replikace je zobecnění jednoho funkčního řešení do konzistentního a zpětně kompatibilního funkčního celku schopného obsluhovat více podobných řešení.

Vezmeme-li v potaz např. nadnárodní společnosti, mající za úkol společný reporting, bude taková organizace potřebovat data konsolidovat nejen na regionální či divizní úrovni. Bude nutné taková data získávat na úrovni mateřské společnosti. Díky tomu se společnost stane schopnou plnit jak regulatorní, tak i interní potřeby shromažďování a analýzy dat. Jako takovou organizaci si můžeme představit obchodní banku či automobilový koncern, podnikající na území více států, či vlastníci několik dceřiných poboček. Zde se budeme orientovat zejména na takové typy společností, kde mají pobočky stejný předmět podnikání.

K pokrytí výše popsaného vznikne pravděpodobně jakési centralizované řešení. Jak ale této centralizace dosáhnout? Jedním ze způsobů je právě replikace Business Intelligence řešení, mluvíme zejména o aplikaci již vyvinutého řešení na ostatní pobočky mateřské společnosti. Zjednodušeně řečeno se jedná o převzetí řešení tak jak již jednou bylo analyzováno, vyvinuto a nasazeno, a jeho využití v pobočkách a dceřiných společnostech za použití dodatečných úprav. Replikaci rozumíme kopírování obchodního i logického modelu jedné entity tak, aby došlo k zajištění konzistence dat napříč entitami. Díky tomuto přístupu společnost získává jistotu o přesnosti, užitečnosti a hodnotě reportovaných dat, tedy že transformace konkrétních veličin v entitě A odpovídá transformaci téže veličiny v entitě B. Získanou přidanou hodnotou je snadná interpretace hodnot napříč celou společností. Pokud by totiž každá odnož mateřské společnosti chápala určité obchodní pojmy různými způsoby, nebylo by možné takto získaná data efektivně agregovat a interpretovat je v informace, výsledné reporty a následně i rozhodování managementu na nich založené by bylo podloženo zkreslenými daty.

Jako příklad vezmeme pojem „aktivní smlouva“, který je různě chápan v různých obchodních divizích společnosti. Z marketingového pohledu je smlouva aktivní, pokud nedošlo k jejímu uzavření či vypršení. Z hlediska finančního jakožto i riskového je ale

smlouva aktivní, pokud se na účtu k ní vztaženému nachází nenulový zůstatek. Jinak řečeno, pracovníci finančního oddělení potřebují pracovat i s takovými smlouvami, které jsou sice oficiálně uzavřené, nicméně se k nim vztahují určité finanční zůstatky, nebo jsou prováděny transakce, které musejí být příslušně zaúčtovány. Na druhou stranu zaměstnanci marketingového oddělení budou potřebovat pracovat s informací o zániku daného obchodu tak, aby například byli schopni vyhodnotit otevření a uzavření počtu smluv za daný kvartál. Při replikaci daného řešení tedy nestačí pouze sladit architekturu, ale také obchodní model a očekávání různě specializovaných odvětví.

O BI řešení poboček neboli entit se zpravidla stará jeden vývojový tým, který se skládá z business analytiků či konzultantů a vývojářů. Takto centralizovaný tým je schopen udržovat řešení co nejvíce v souladu s regulatorními i ostatními metrikami nadřazené společnosti, ale také úzce spolupracuje s konkrétními lokálními uživateli, což vede k vyrovnaní požadavků ze strany zákazníka, architektury i konečného uživatele.

7.1 Popis situace

Jako objekt zkoumání této práce byla vybrána bankovní sféra. Banky a bankovníctví samo o sobě jsou rozsáhlým komplexem pokrývajícím jak prodej specifického řešení klientům, tak ale i regulatorní složky podnikání a nutnost sledovat finanční toky ve společnosti více než v jakémkoliv jiném odvětví. Vzhledem ke stále větší globalizaci řešení, expanzi firem do okolních států i kontinentů, ale i stále se zvyšující soutěživosti na poli konkurence, je na banky vyvíjen neustálý tlak jak ze strany klientů, tak ze strany tzv. regulátorů, kteří působí na území státu, kontinentu, či územního celku.

Pro účely práce tedy vezměme v potaz bankovní společnost mající dceřiné pobočky po celém světě. Banka tedy potřebuje zkoumat nejen interní chování společnosti, ale také celkové chování napříč všemi pobočkami, které vlastní a které nepřímo ovlivňují také samotnou mateřskou společnost. Mateřská společnost se snaží a je nucena monitorovat své dceřiné pobočky, stejně tak dceřiné společnosti sledují své lokální pobočky. Tato nutnost vychází z potřeb zvyšování obrátu, snižování rizika či z regulatorních důvodů.

Vycházíme tedy z bodu, kdy každá jedna entita již má aplikovaný dostatečně automatizovaný transakční systém, na vyšších či nižších úrovních již produkuje jisté reporty a výstupy. Ovšem v každé jednotlivé entitě se zároveň projevují jisté odlišnosti, které není jednoduché na úrovni skupiny konsolidovat a interpretovat, ať už je to způsob implementace

řešení, či přístup k datům jako takovým. Výsledkem tedy bude, že vlastníci společnost nedosáhne jedné verze pravdy, není schopna lokální data a reporty využít pro své účely a tím vzniká potřeba data řídit už na úrovni těchto poboček.

Tato potřeba s sebou přináší jedno z řešení, čímž je implementace BI. Přenecháním implementace jednotlivým entitám ovšem povede k tomu, že každá z nich vybere různé řešení, s různými obchodními modely, různou architekturou a nemožností propojit systémy ačkoliv jsou si operativně velice blízké. Subjektem podnikání každé z bank je totiž stejný produkt, stejná sada obchodních dovedností, dokonce stejná sada bankovních specifik, které je potřeba sledovat. Jak již vychází z přechozích kapitol, nejlepším řešením pro implementaci BI pro více entit, kdy je nutnost sledovat jednotnou verzi pravdy je centralizované řešení.

7.2 Centralizované řešení v jedné entitě

Při implementaci centralizovaného BI řešení pro jednu entitu je nutné se soustředit na dostupné vstupy a požadované výstupy. Pomocí architektury centrálního datového skladu dosáhneme již na úrovni entity jednotné verze pravdy a konsolidace dat mezi jednotlivými odvětvími.

7.2.1 Vstupy

Jako primární zdroj dat využijeme světově rozšířený standard bankovního řešení Amplitude od společnosti Sopra. Amplitude, což je plnohodnotný, modulární a integrovatelný core bankovní systém, který je ve světě využíváný cca 200 bankami v 50 zemích. (Soprabanking.com, 2017). Díky modularitě je možné Amplitude rozšířit o libovolné businessové informace. Amplitude zpracovává denní bankovní transakce, aktualizace účtů a dalších finančních záznamů napříč všemi pobočkami banky. Systém v zásadě zahrnuje možnosti vkladů, půjček a zpracování úvěrů s rozhraním k systémům hlavních knih. Mezi klíčové služby banky patří vytváření nových účtů, obsluha úvěrů, zpracování vkladu a výběrů a řízení vztahů se zákazníky. Pomocí core systému jsou zaměstnanci na kterékoliv pobočce schopni přijmout nového zákazníka a založit k němu odpovídající dokumentaci, mají přístup ke všem zákazníkům společnosti a mohou se tak odkudkoliv starat o jeho portfolio. Jednoduše řečeno jsou veškeré transakce provedené zaměstnancem, či klientem, prováděny, spravovány a monitorovány právě tímto systémem.

Kromě základního zdrojového systému Amplitude pak můžeme mít ještě zdroje doplňující, popř. externí. Mezi tyto zdroje řadíme zejména dva systémy: Moduly a externí kalkulátory. Moduly jsou rozšíření jak Amplitude, tak i systémy třetích stran, které se párují s core.

Modul třetích stran může být např. leasing, nebo ProvIAS. Historicky se stává, že správa leasingových dat je spravována jinými a méně sofistikovanými nástroji, jelikož jde z logiky věci o méně náročné řešení, které může fungovat samo o sobě bez core bankovního řešení. Spojení core a leasingu potom vyžaduje další business analýzu, aby spojený produkt fungoval jak po technické stránce, tak i po businessové.

Co se týče druhé skupiny externích zdrojů dat – externích kalkulátorů – tam je situace potenciálně mnohem složitější, jelikož zasahuje i do architektury datových toků. Jako příklad lze uvést vyhodnocování zdravotní klientů pomocí nástrojů státní centrální banky. (Soprabanking.com, 2017) Takto navržený externí kalkulátor nemůže být ze své podstaty implementován přímo v samotné komerční bance, ale vstupní data musí být zaslána ze skladu ven do centrální banky. V následném kroku centrální banka zašle kalkulovaný výstup zpět do skladu. Pokud bychom finanční zdraví klientů vyhodnocovali na denní bázi, ETL by v tomto případě muselo běžet dvakrát – před zasláním vstupu a následně po obdržení výstupu. V této situaci by samozřejmě také došlo k prodloužení doby, kvůli čekání na zaslání extraktu. Při použití externích kalkulátorů je také třeba zvážit, jak se zachovat v případě, že požadovaný extrakt nepřijde. V takovém případě je třeba nastavit pravidla a postupy, protože například při regulačních záležitostech by se komerční banka mohla dostat do potíží.

Obecně je přidávání nových datových zdrojů náročný úkol a jeho zapracování může trvat řádově i několik měsíců.

Pro naši situaci využijeme Sopra Amplitude jako core bankovní systém, ve kterém se nacházejí data o nejdůležitějších oblastech podnikání. A to sice informace o zákazníkovi, o jeho účtech, otevřených produktech a provedených transakcích. Před samotnou kalkulací nákladů je nutné provést analýzu těchto zdrojových dat spolu s požadavky na nové reporty, výstupy, které mají být dodány jak společnosti jako takové tak regulačním institucím.

7.2.2 Výstupy

Požadavkem naší společnosti je umožnění vygenerování čtyř reportů, které budou nadále sloužit k podporování rozhodování společnosti, ale i k pokrytí regulačních územních požadavků.

Prvním z těchto reportů je profitabilita banky, kterou budeme měřit podle tradičního ukazatele a sice ROA (return on assets), samozřejmě nejen na úrovni samotné banky, ale také na úrovni všech poboček a jednotlivých nabízených produktů. K tomu, abychom toho mohli dosáhnout, je nutné získat ze zdrojového systému veškerá aktiva a pasiva, informace o produktech nabízených bankou a o hierarchii jednotlivých poboček.

$$ROA = \frac{EBIT}{Aktiva}$$

Druhým reportem je hlášení o prodeji skrze firmu. Jedná se o přehled stavu prodejních aktivit v rámci společnosti, ukazuje trendy, ke kterým dochází v objemu prodeje v čase, ale také ukazuje výkonost prodejních manažerů. Takové zprávy pomáhají řídicím pracovníkům rozhodovat se o nových tržních příležitostech, či dalších krocích za účelem zlepšení celkového výsledku. Pro sestavení podobné zprávy budeme potřebovat informace o otevření nové smlouvy, neboli o prodeji, o tom kdo prodej uskutečnil a komu byl prodej poskytnut.

Třetím reportem, který je významný pro celkové zhodnocování zdraví a úspěšnosti jednotlivých poboček pro analýzu celkové úspěšnosti a produkceschopnosti je sledování performance. Tento důležitý report vystihuje základní charakteristiky poboček popř. jednotlivých celků a vyhodnocuje, jak daná je pobočka efektivní a efektní. Podle tohoto reportu můžeme porovnávat jednotlivé pobočky mezi sebou, odhalovat rozdíly a zároveň mitigovat případná rizika, které mohou během působení vznikat. Další neméně důležitou přidanou hodnotou tohoto reportu je analýza jednotlivých pracovníků na pobočkách. Pomocí těchto výstupů můžeme zavádět tzv. balance scorecard, která vytváří vazbu mezi strategií a operativními činnostmi s důrazem na měření výkonu. Tímto reportem nepřímo naplňujeme jeden z hlavních cílů BI - práce s lepšími a informovanějšími rozhodnutími a zjišťování neefektivních podnikových procesů.

Posledním a neméně důležitým reportem – risk, neboli kreditní riziko. Jedná se o takové riziko, které vyplývá se skutečností, že dlužník jen velmi nepravděpodobně bude

schopen dosáhnout svých závazků a splatit je v plné výši, či pokud dlužník selhal ve splácení po dobu 90 dní. Banka je povinna sledovat rizikovost svých zákazníků předtím, než jim poskytne úvěr, ale i v průběhu tohoto vztahu, proto je pro ni právě riskový report velice důležitý.

7.2.3 Modelová situace pro jednu entitu

Před samotným počátkem stavění datového skladu je vždy nutné provést prvotní analýzu a vyčíslit náklady spojené s realizací. Toho dosáhneme pomocí párování dat vstupů s požadavky na řešení. Víme, že je použita architektura centrálního datového skladu, dále máme identifikované obchodní požadavky a také vstupní zdroje.

TECHNICKÉ VRSTVY

Hlavním předpokladem této práce je fakt, že součástí dodávky modelového řešení není dodání technického podkladu, tedy databázových serverů, či softwaru. Pro naše potřeby jsou tyto zahrnuty jako fixní náklady, které jsou v gesci zákazníka.

LOAD ZDROJOVÝCH DAT

Prvním krokem k realizaci datového skladu je vytvoření loadu zdrojových dat do databáze. Protože jsme identifikovali hlavní zdrojový systém, jako první se zaměříme na něj.

1. Základní zdrojový systém

Tento systém obsahuje desítky zdrojových tabulek, je nutné se zaměřit pouze na ty, které nám pomohou k docílení požadovaných výstupů. Pro tyto účely jsme pomocí analýzy vybrali 20 základních tabulek, které obsahují nejdůležitější oblasti podnikání, z nichž ty nejdůležitější jsou popsány níže. Mimo jiné je nutné zkoumat velikost objektů a komplexnost dat, která obsahují. Zkoumáním jejich struktury a datových typů dojdeme k poznání, jak obtížné bude data integrovat do systému. Pro potřeby diverzifikace modelového řešení jsme stanovili komplexnost dat jako pět jednoduchých tabulek, tzn. tabulka mající maximálně deset atributů, osm středně obtížných tabulek majících deset až třicet atributů a sedm velmi obtížných tabulek, které jsou definovány třiceti a více atributy.

- 1.1. Zákazník – tabulky obsahující data o klientech jako je jeho jméno, adresa, identifikační číslo (číslo občanského průkazu, číslo cestovního pasu atd.), národnost či domácí pobočka, která klienta obsluhuje.
- 1.2. Účet – veškeré účty, které jsou vždy ve vztahu ke klientovi a mohou mít různou podstatu, od běžného účtu, přes kontokorent až k účtu mezinárodnímu. V této oblasti nalezneme atributy jako je číslo účtu, datum založení, měna, ve které je účet veden, pobočka, na které byl účet založen, účet hlavní knihy, na kterém jsou vedeny veškeré pohyby, produkt, ke kterému se vztahuje, a samozřejmě také obnos vyjádřený v jednotkách měny, které účet obsahuje, ať už kladný, či záporný.
- 1.3. Transakce – zde se jedná o jedinou tabulku, která udržuje veškeré pohyby, které byly na účtech provedeny, sleduje, zda byly provedeny klientem, či automatickým příkazem, na jakou částku byly založeny, ke kterému účtu se transakce vztahuje a kdy byly zaúčtovány.
- 1.4. Management data neboli smlouvy – veškeré produkty, které jsou bankou vytvořeny musí být podloženy smlouvou. Tyto smlouvy označujeme jako management data, kde je sledováno, kdy byla smlouva založena, na jaký obnos, kdy se očekává případné splacení dlužné částky a o jaký typ smlouvy se jedná.
- 1.5. Spolu s klíčovými informacemi také základní bankovní systém poskytuje tzv. referenční data, jako je například hierarchie produktů, účty hlavní knihy a jejich struktura, segmentace klientů nebo také parametrizace dat, kupříkladu pravidla o životním cyklu smlouvy.

Zdrojový systém je do datového skladu zpravidla přenášen jako druh textového souboru. V tomto případě se jedná o .csv soubor, který je nutný nahrát do databáze. To zrealizuje loadovací workflow, neboli datová pumpa. Díky data assessmentu jsme zjistili, že zdrojová data obsahují pouze informace v datovém typu text, číslo, či datum. To značně ulehčí jak load tak i uložení dostupných dat.

2. Externí datové zdroje

Mezi externí datové zdroje zahrnujeme takové zdroje, které přichází z vnějšku společnosti jako takové. Jedná se většinou o data podpůrné, referenční.

INTEGRACE DAT DO JÁDRA

Dalším krokem je vytvoření jednotného datového jádra, které obsahuje a integruje veškerá data. Protože respektujeme centralizovanou architekturu, je nutné vytvořit dvě vrstvy, jednu, která data přejímá a transformuje, a druhou, která data historizuje. Dalším krokem pak bude navrhnout model, který obsahuje veškeré informace o chování datového skladu, stejně jako o obchodních pravidlech, podle kterých se data zpracovávají. Pro modelové řešení jsme rozvrhli integrační tabulky podle jejich obtížnosti na tabulky velice složité, které jsou Party, Account a Agreement. Tyto tabulky mají velmi složitá transformační pravidla. Jako středně těžké jsme identifikovali tabulky Party_Address a Party_Rating, která mají jednodušší transformace dat a v poslední skupině jednoduchých tabulek se nachází referenční data Product, Segmentation a Chapter, která jsou zpravidla přejímána beze změny.

1. Tabulka Party – Libovolná skupina, která je v oblasti zájmu banky. Jde o fyzické i právní osoby, organizace, popř. zájmové skupiny.
 - 1.1. Party_Address – tabulka obsahuje data týkající se adresy klientů
 - 1.2. Party_Rating – tabulka obsahuje informace o interním i externím ohodnocení klientů
2. Tabulka Account – Tabulka obsahuje klientské účty vedené v bance v současnosti i v minulosti. Klientský účet je výkazem pohledávek nebo závazků mezi bankou a klientem založených na smlouvě o vedení účtu. Obsahuje také tak zvané vnitřní aplikační účty a analytické účty hlavní knihy. Účet je realizací produktu. Analytické účty obsahují zůstatky na účtech, objemy kreditních a debetních transakcí za aktuální měsíc a průměrné zůstatky za uplynulý měsíc v hlavní knize. Na účtech banka zaznamenává veškeré transakce realizované mezi zákazníkem a bankou, díky čemuž je schopna monitorovat finanční situaci zákazníka na jednotlivých účtech. Účty rozdělujeme na kladné neboli kreditní a záporné neboli debetní. Dále účty, které zákazník využívá za účelem vkladu peněz, označujeme za depozitní, a ty, které zákazník využije pro výběr peněz, tedy k půjčení, nazýváme jako účty úvěrové.
3. Tabulka Agreement – Tabulka obsahuje všechny dohody s klienty, např. všechny výskyty vztahu mezi klientem a bankou, kde klient je (nebo byl) vlastníkem produktu.

Dohoda vytváří instanci produktu. Tabulka obsahuje různé typy dohod: Vkladové účty, Úvěrové účty, Ztracené úvěrové účty, Přímé bankovníctví, Platební karty, Diskontované účty, Sběrné poukázky, Pojištění, Investiční bankovníctví, Měnové portfoliové účty. Informace v tabulce je odvozena ze základních tabulek pro určitý typ obchodu. Každá dohoda je určena svým unikátním identifikátorem (Agreement_Id) a má pouze jednoho vlastníka (Party_Id) a představuje pouze jeden produkt.

4. Tabulky referencí

4.1. Product – odpovídá typu obchodu, jedná se o jednoznačný kód označující druh obchodní smlouvy

4.2. Segmentace – jedná se o zařazení všech klientů do předem definovaných skupin, neboli segmentů a využívá se například pro cílení marketingových nabídek. Jednou z takových segmentací může být rozřazení zákazníka podle velikosti a vlastností jeho podnikání na malé a střední podniky, korporace nebo například banky.

4.3. Chapter – referenční data k hlavní účetní knize, kdy chapter je analytický účet, na který je účtováno. Hlavní využití je ve finančním oddělení.

DATA V MARTOVÉ VRSTVĚ

Pro správnou businessovou interpretaci dat je důležité vystavět datové marty nad unifikovaným jádrem tak, aby nesly správnou obchodní informaci, kterou požaduje dané oddělení. Na základě cílů, kterých chceme dosáhnout jsme identifikovali tři hlavní obchodní oddělení, které bude vytvářet reporty, a pro které je vhodné postavit vlastní mart. Jak již bylo nadefinováno, mezi naše výstupy patří čtyři reporty, bude tedy nutné pro každý z nich připravit čtyři výstupní pohledy, které samy o sobě budou klasifikovány jako velice obtížné.

1. Oddělení financí

Tento úsek má na starosti celou řadu činností, které se vážou k tvorbě a vyhodnocování bankovních strategií. Z hlediska financí má na starosti nejen vnitřní hospodaření, ale také například daně, hodnocení výnosů a nastavení úrokových sazeb. V našem modelovém řešení bylo jedním z výstupů postavení reportu pro potřeby profitability. Report profitability bude sloužit zejména tomuto oddělení, které na jeho základě bude dále vyvíjet a spravovat model ziskovosti produktů a služeb.

2. Oddělení prodeje

Úsek marketingu neboli prodeje obstarává pro jednotlivé segmenty klientů přípravu marketingových a komunikačních strategií a řízení produktů. Díky novému řešení reportu prodeje se budou moci pracovníci tohoto oddělení lépe rozhodovat a budoucím vývoji a tvorbě produktů, která jsou následně bankou nabízeny klientům dle jejich zařazení do jednotlivých segmentů.

3. Oddělení sledování rizikovosti klienta

Hlavním úkolem oddílu sledování rizik je dohlížet na rizika vyplývající z činností banky a to například na úvěrová a tržní. Jeho práce spočívá zejména ve vytváření metodik řízení rizik a měření úvěrových a tržních rizik. Na základě naší dodávky tak podpoříme nejen tyto aktivity, ale také proces posuzování úvěrového rizika u nových produktu, či také schvalování kreditních limitů pro největší klienty.

7.3 Estimace neboli odhadovaná pracnost pro jednu entitu

Zejména při vypracování nabídky společnosti, která poptává dodávku Business Intelligence v jakékoliv podobě, je nejdůležitějším bodem při rozhodování odhadovaná pracnost vyjádřená v MD (zkrácené z anglického Man Day, neboli ekvivalent k vyjádření pracovní doby jednoho člověka pro interpretaci kolik práce je nutno k provedení určitého úkonu). Pro přípravu odhadování pracnosti využijeme metody Work breakdown structure, která je popsána v kapitole 6.

Jako první je tedy nutné rozvrhnout co je cílem projektu, jaké výstupy jsou očekávány a co přesně má být dodáno zákazníkovi a to na nejvyšší úrovni. Je zapotřebí soustředit se na hlavní části našeho projektu. Mezi hlavní části naší dodávky tedy patří:

- Vytvoření datového modelu
- Vytvoření dokumentace řešení
- Vytvoření loadovacího workflow
- Vytvoření materializované vrstvy, jak transakční tak historické, Vytvoření agregovaných dat jednotlivých oddělení pomocí datových tržišť

Druhým bodem pak je rozpad těchto cílů na dílčí úlohy, které budou pokryty členy týmu. Jedná se o proces vytvoření detailnějšího pohledu na celý proces, kdy je důležité položit si dotazy jako například, co vše je potřeba podniknout, aby bylo možné tento cíl splnit, jaké jiné úlohy jsou s touto spjaty a tedy jsou na sobě závislé, jaké jsou konkrétní požadavky na tuto úlohu? V souvislosti s dodávkou datového skladu byly identifikovány následující body 1 – 16 vypsane níže.

Sepsáním všech těchto podúloh jsme si nejen připravili stabilní podklad pro přípravu naceňování, ale také jsme si krok po kroku navrhli životní cyklus projektu. Posledním a nejvíce důležitým bodem je tedy samotné nacenění. Pro vlastní nacenění použijeme mechanismus, kdy mezi proměnné kalkulace patří počet odhadovaných objektů a jejich náročnost.

1. Revize stávajícího, lokálního řešení, a organizace workshopů se zaměstnanci banky

Ať už je tým analytiků sebezkušenější, vždy je nutné dostat se do kontaktu s lidmi, kteří mají klíčové znalosti o fungování celé společnosti, kteří s daty reálně na denní bázi pracují, a tak jim nejlépe rozumí, mají povědomí o veškerém chování i o chybách, které se v systému vyskytují. Naším úkolem je pochopení jejich potřeby využití systému co nejlépe. Při procházení stávajícího řešení můžeme jak získat přehled o tom, jak celý proces doteď fungoval, ale i představu o místech, která se musí vylepšit, kde je prostor pro využití našich zásahů a oprav.

Odhadovaná pracnost = 5MD, obsahuje zejména načtení dokumentace, porozumění dané problematice, konzultace se zaměstnanci. Tuto aktivitu vykonává analytik, či tým analytiků, kteří dle vlastního uvážení plánují pravidelná setkání s lidmi z oboru, dokud dokonale nerozumí dané problematice. Očekáváme, že členové týmu si mimo jiné také načtou dostupnou literaturu, či již mají zkušenosti z jiných projektů.

2. Získání datových zdrojů

Tento bod je klíčový pro pokračování projektu. Analýza ani plánování dodávek nemůže pokračovat aniž bychom měli dostupná data, která budeme pro naše řešení využívat. Je tedy nutné zejména na tuto část klást značný důraz při jednání se zákazníkem a tyto vstupy získat včas.

Odhadovaná pracnost = 1MD, zahrnuje komunikaci s dodavatelem datového zdroje, domluva potřebných dat a parametrů. Ze zkušeností víme, že ne vždy jsou dodané soubory v požadovaném formátu, či kvalitě. Počítáme tedy s tím, že bude potřeba datový soubor revidovat a pravděpodobně několikrát vrátit dodavateli s vysvětlením problematiky.

3. Data assessment datového zdroje

Dalším krokem bude data assessment datového zdroje, kdy analytik, či tým analytiků prochází daný soubor a to jak z technického, tak datového a logického úhlu pohledu. Jedná se zejména o kontrolu ve smyslu, zda je soubor nepoškozený, v očekávaném formátu, které bude možné zpracovat námi dodávanou technologií, zda jsou veškeré atributy vyplněné, zda jsou vyplněné daty, která podle daného datového typu, či dokonce zda data odpovídají stanovené definici hodnoty, kterou mají reprezentovat.

Odhadovaná pracnost = 10,5MD, při velikosti scope 20 objektů tzn. 0,5 MD na zkoumaný objekt a 0,5MD pracnosti na technický load dat do databáze. Odhadujeme, že po načtení dat do databáze je analytik schopný projít data tabulku po tabulce a zjistit, zda hodnoty odpovídají očekávání, jak jsou data kvalitní a zda je možné s nimi pracovat. Zároveň také analytik získává přehled o stavu, má možnost zkoumat propojení mezi daty a nacházet vzájemné souvislosti. Je to první fáze datové analýzy, pomocí níž jsme schopni začít s vlastním návrhem řešení.

4. Navržení datového modelu

Důležitým bodem analýzy je sestavení datového modelu, tedy modelu, který zdůrazňuje datové toky uvnitř datového skladu. Tento model slouží jak vstup pro vývoj, tak jako dokumentace k řešení.

Odhadovaná pracnost = 24,275MD, při různé komplexnosti objektů. V tabulce 3 můžeme tento úkol najít pod číslem 4a a 4b. Záměrně je tato aktivita rozdělena do dvou bodů zejména pro různou pracnost obou oblastí. První oblastí je tzv. stage, kde neprobíhají žádné transformace, data se pouze načítají a ukládají. Tady je tedy pracnost relativně nižší. Počítáme, že na jednoduchý objekt potřebuje analytik 0,125 MD, na středně obtížný pak 0,25 MD a pro složitý objekt 0,5 MD. Modelování zahrnuje jak popis chování daného objektu, tak i výpis jeho atributů, datových typů a vytvoření primárního identifikátoru. Druhou oblastí pak jsou samotné transformační procesy, kde dochází k různě složitým

výpočtům, transformacím očišťování dat od nedokonalostí a chyb. Zde analytik provádějící modelování spotřebuje 0,3 MD na jednoduchý objekt, 0,75 MD na středně složitý objekt a 1,5 MD na velice komplikovaný objekt.

5. Sepsání funkční specifikace

Iniciálním bodem funkční specifikace je dopodrobna popsání požadavek zákazníka. Dále je pokračováno v popisu navrhované řešení a to na úrovni objektů, neboli tabulek v databázi, skriptů, které je potřeba k obsluze datového skladu. Specifikace jako taková ovšem obsahuje i detail jdoucí až na jednotlivé atributy objektů, jejich datové typy a pravidla podle kterých jsou transformovány, neboli pravidla, která určují jak jsou prováděny výpočty a veškeré manipulace s daty, ať už mluvíme o datové konverzi, či hlídání datové kvality.

Odhadovaná pracnost = 18 MD. Jak je již popsáno výše, jedná se o zevrubnou dokumentaci řešení, která má i do budoucna sloužit jako průvodce projektem. Je nutné, aby všechny role, které při řešení zapojujeme byly schopné takovému dokumentu porozumět, musí být tedy sepsány do detailu, ale zároveň nesmí obsahovat příliš technických obrátů. Taková specifikace se většinou pohybuje v rozsahu několika desítek stran, kdy zevrubně vysvětluje jak obchodní model, tak i model datových toků a veškeré technické parametry skriptů a objektů. Dále také může upozorňovat na určité nesrovnalosti, které vznikly v průběhu vývoje ať už nedokonalostí dat, či zjištěním, které by mohlo vést ke zlepšení v dalších iteracích. Odhad byl vytvořen jako násobek počtu zahrnutých objektů * 0,5 MD.

6. Vytvoření datového workflow

Nejdůležitějším vstupem pro tvorbu loadovacího workflow je datový model. Tento model obsahuje zdrojové a cílové tabulky a vztahy mezi nimi, tzv. mapping. Pomocí softwarového nástroje pro vytvoření workflow, např. Informatica, se pak tvoří workflow jako takové. V programu Informatica jsou definovány datové typy, přesnost údajů, informace o zdrojovém souboru jako takové, tzn. oddělovač sloupců, definice tzv. hlavičky neboli prvního řádku souboru. Probíhá zde také první transformace, očištění dat jako takových. Na základě informací z datového modelu je nadefinované, jak se které atributy mají chovat. Pokud tedy neodpovídají tomuto očekávání, může se buď upravit datový model, nebo je tato situace nechtěná a taková chybná data se odlévají do errorových souborů, které slouží k hlášení a uchování vadných dat. Po provedení těchto kroků je nutné řešení

zkontrolovat, tedy provést zkušební load, který zpravidla probíhá nad prázdným souborem a slouží k odhalení případných nesrovnalostí očekávání s realitou.

Odhadovaná pracnost = 3,8 MD při tvorbě jednoho workflow, který zpracovává zdrojová data a tři transformačních workflow.

7. Sepsání technické specifikace pro potřeby datového workflow

Datové workflow má několik náležitostí, které je potřeba zdokumentovat pro budoucí potřeby. Mezi ně patří například typ datového souboru, který bude zpracováván, četnost zasílání dokumentu atd.

Odhadovaná pracnost = 6 MD pro jeden zdrojový systém a tři transakční fáze, kdy na základě expertních odhadů víme, že loadovací workflow pro zdrojový systém je technicky i časově náročnější, proto zabere vývojáři 1,5 MD na jeho tvorbu. Oproti tomu transakční fáze, které zpravidla obsahují pořadí spouštěných skriptů a jiné náležitosti nevytíží vývojáře natolik a proto odhadujeme jeho pracnost na 1 MD/fázi.

8. Příprava zadání pro vývojáře

Pokud jsou již veškeré kroky výše provedeny, analytický tým dále pokračuje s vytvořením zadání pro vývojáře. Vývojovému týmu jako vstup slouží dokumentace sepsána výše, ale i nadále potřebuje analytické konzultace, pro lepší pochopení.

Odhadovaná pracnost = 21 MD, vzhledem k faktu, že u dokumentace řešení není záhodno jít do technických detailů, je potřeba, aby analytik byl neustále v kontaktu s vývojářem, a zejména aby připravil zadání na úrovni atributů a jednotlivých pravidel. Odhady se odvíjejí od komplexnosti jednotlivých pravidel, kdy každý objekt si vyžaduje jinou náročnost provedení. Pokud se jedná o jednoduchý objekt, který má jednoduchá pravidla odhadujeme pracnost na 0,75 MD na objekt, při střední obtížnosti na 1,5 MD na objekt a při velké komplexnosti objektu na 2,25 MD.

9. Vývoj

S pomocí všech vstupů probíhá konkrétní vývoj řešení, který zahrnuje vytvoření struktur, vytvoření plnicích skriptů. Vývojář takto vytváří pomocí SQL kódu jak struktury

tabulek, tak i SQL kódy, které naplní data. Jedná se o kód, který musí být bezchybný a proto i vývojář aplikuje na svoji práci určité testy, viz další kapitola.

Odhadovaná pracnost = 14 MD, zde předpokládáme, že zadání od analytika bylo natolik přesné, že je vývojář schopný psát kód aniž by se od něj vyžadovalo pravidla přepisovat, či je kontrolovat, proto odhadujeme, že samotné napsání kódu do formální podoby zabere na jednotku méně času, než analytikovi sepsání zadání. Pro vývojáře tedy platí následující nacenění: 0,5 MD na jednoduchý kód, 1 MD na kód mírně složitý a 1,5 MD na kód velice složitý.

10. Testy vývojáře

Jak již bylo řečeno, i vývojář si svoji práci testuje. Takové testy považujeme za unit testy, kdy pracovník vývoje obsáhne také část, kterou analytik málokdy do své analýzy zahrne a to zejména performance skriptů, jejich optimalizace a v neposlední řadě také udržitelnost a čitelnost vlastního kódu.

Odhadovaná pracnost = 11 MD, ačkoliv samotné psaní kódu jak je popsáno výše je považováno za nižší objem jednotek než při zadání do vývoje, vývoj jako takový se navyšuje o kontroly, které provádí sám vývojář. Tyto kontroly nezahrnují faktické testování, nýbrž formální náležitosti a náležitosti potřebné pro bezchybnou funkcionalitu na požadované platformě. Vývojáři bude kontrola a ladění jednoduchého skriptu trvat 0,375 MD, středně těžkého 0,75 MD a 1,2 MD pro složitý skript.

11. Příprava funkčních testů

Správné otestování dodávky je jednou z nejdůležitějších fází projektu. Pro jeho snadný průběh spolupracuje tým testerů s týmem analytiků a za pochopení celého mechanismu sepisují testovací případy, pomocí nichž jsou pak schopni ověřit funkčnost řešení.

Odhadovaná pracnost = 16 MD, jelikož pro naši modelovou situaci je analytik tatáž osoba jako tester, předpokládáme, že již osoba má povědomí o tom, jak mají jednotlivé objekty fungovat. Není tedy třeba nového zaučení, čímž se ušetří jak čas, tak i náklady. Nicméně je nutné pečlivě připravit testovací scénáře jak pro kontrolu správnosti návrhu nebo vývoje, tak i pro otestování veškerých alternativních scénářů, které mohou při ostrém provozu nastat. Odhady odvozujeme od počtu objektů, které se v řešení nachází násobené konstantním číslem 0,25 MD na objekt.

12. Testování interními pracovníky

Tým testerů je zodpovědný za odchytení všech možných chyb a nedostatků jak analýzy tak skriptů. Je jejich úkolem zevrubně prověřit chování řešení a jejich cílem je udělit souhlas k dalším fázím projektu.

Odhadovaná pracnost = 15,8 MD. Exekuce vlastních testů je navazující aktivitou po přípravě testů, poté co jsou sepsány funkční testy, je nutné, aby členové týmu poctivě procházely jednotlivé funkčnosti a hlásili chyby. Na tuto aktivitu jsme vyhradili 25% ze součtu práce analytika na návrhu a práce vývojáře na vývoji.

13. Testování s konečnými uživateli a jejich podpora

Protože bude řešení využíváno konečnými uživateli, je jejich názor stěžejní pro akceptaci dodávky. Zde jsou zapojeni reální koncoví uživatelé, kteří mají své očekávání a tato uplatňují při validaci všech částí řešení, od samotného chování až po design jednotlivých reportů.

Odhadovaná pracnost = 9,5 MD, konečné slovo o tom, zda systém pracuje dle očekávání ovšem bude mít koncový uživatel, ten je tedy také vyzván k akceptaci daného řešení. Očekáváme, že podpora uživatelů, jakožto seznámení jich s technickými podklady, vysvětlení jednotlivých funkcionalit a odpovídání na jednotlivé nálezy, bude pokryta procentuálním vyjádřením dodaného množství práce a to ve výši 15 %.

14. Příprava řešení k nasazení

Pakliže je zákazník spokojený, je nutné připravit celou sadu skriptů pro nasazení na produkci, tzv. připravení nasazovacího balíku.

Odhadovaná pracnost = 2 MD, to jakým způsobem se řešení připravuje pro produkci, je v každém prostředí odlišné, nicméně předpokládáme, že našemu týmu vývojářů příprava zabere 2 MD.

15. Sepsání průvodky nasazení

Spolu s nasazovacím balíkem je nutné také připravit průvodku, kde je detailně popsáno, jak je vhodné postupovat, při uvádění řešení na produkci. Protože se často jedná o iteraci dodávky, je nutné vystihnout veškeré závislosti na ostatních modulech.

Odhadovaná pracnost = 1 MD, protože je nutné celou dodávku provozovat, a to předpokládáme bude obsluhovat jiný tým lidí, než ten, který řešení vyvinul, je zapotřebí sepsat takovou dokumentaci, která bude obsahovat technické aspekty dodávky.

16. Nasazení řešení

V závěrečné fázi pak dochází k vlastnímu nasazení, které zpravidla probíhá vlastním útvarem banky, za pomoci týmu, který je za dodávku zodpovědný.

Odhadovaná pracnost = 6,3 MD, v závislosti na velikosti řešení jsme připraveni poskytnout podporu pro tým provozovatelů, který bude obsahovat asistenci při nasazení, ale i kontroly dat těsně po nasazení. Tato část dodávky je sice posledním bodem, nicméně neméně důležitým, vzhledem k tomu, že i zde se mohou objevit chyby a dříve neobjevené anomálie. Pracnost je vyjádřena 10% s celkové dodávané hodnoty.

Tabulka 3 - Odhad pracnosti pro jednu entitu v jednotkách MD

Oblast	Úloha	Popis	Počet objektů	Analytik	Vývojář	Celkem
Vstupy	1. Revize stávajícího řešení			5,0		5,0
	2. Získání datových zdrojů			1,0		1,0
Data	3. Data assessment zdrojových dat	Technický load	1	0,5	1,0	1,5
		Zkoumání dat	20	10,0		10,0
Workflow	4a. Příprava modelů pro zdrojová data	Jednoduchá	5	0,6		0,6
		Střední náročnost	8	2,0		2,0
		Vysoká náročnost	7	3,5		3,5
	6. Tvorba loadovacího workflow	Core systém	1		1,5	1,5
		Transformační fáze	3		2,3	2,3
	7. Testy a tvorba dokumentace pro workflow	Core systém	1		1,5	1,5
Transformační fáze		3		3,0	3,0	
Transformace dat	4b. Příprava modelů	Jednoduchá	3	0,9		0,9
		Střední náročnost	2	1,5		1,5
		Vysoká náročnost	7	15,8		15,8
	8. - 9. Vývoj skriptů	Jednoduchá	3	2,3	1,5	3,8
		Střední náročnost	2	3,0	2,0	5,0
		Vysoká náročnost	7	15,8	10,5	26,3
10. Test vývojáře	Jednoduchá	3		1,1	1,1	
	Střední náročnost	2		1,5	1,5	
	Vysoká náročnost	7		8,4	8,4	
Dokumentace	5. Sepsání funkční specifikace		36	18,0		18,0
Testy	11. Příprava testovacích scénářů		32	16,0		16,0

	12. Vlastní testy = 25%			11,3	4,4	15,8
	13. Podpora uživatelského testování = 15%			6,8	2,7	9,5
Balíky	14. Příprava balíků = 2MD		1		2,0	2,0
	15. Příprava průvodky = 1MD		1		1,0	1,0
	16. Podpora nasazení = 10%			4,5	1,8	6,3
Celkem MD				112,4	46,2	164,6

Zdroj: vlastní

Jak již je patrné z tabulky 3, celkový odhad pracnosti pro vybudování datového skladu je 164,6 člověkodne. Nacenení proběhlo na základě expertních odhadů čerpajících ze zkušeností při práci na podobně stavěných projektech. Při nacenení bereme v potaz dvě hlavní role v celém projektu a to sice analytik a vývojář. Mezi kompetence analytika patří rozumět obchodním datům, které plánujeme zpracovávat, znalosti Business Intelligence a architektury datového skladu. Předpokládáme, že má minimálně základní znalost jak SQL, tak i testování a proto je u nás v tomto případě analytik zároveň také tester. Naproti tomu vývojář musí mít excelentní znalost již zmiňovaného jazyka SQL, musí být schopný porozumět zadání a přesně jej vykonat. Mimo jiné očekáváme také vlastní invenci v podobě hlášení nesrovnalostí, optimalizace skriptu a finalizace formální podoby. Tyto dvě role spolu musí být neustále v úzkém kontaktu po dobu celé implementace, jejich spolupráce je klíčovou, jelikož se navzájem doplňují o informace ze své oblasti působení.

7.4 Řešení ve více entitách

Výše popsané řešení se zaměřuje na jednu entitu. Naší modelovou situací je ovšem mateřská společnost vlastníci deset dceřiných poboček, které samostatně obchodují na vlastních lokálních trzích s velice podobným portfoliem produktů, ovšem přístup k reportingu má každá jiný. Nehledě na vyspělost územích celků, kde se pobočky nacházejí, totiž dochází k odlišnostem v systémech, předpokladem je, že ani jedna z poboček nemá zavedené centralizované řešení Business Intelligence.

Vezmeme-li tedy v potaz, že by si každá entita zvlášť nechala vybudovat centralizované řešení odpovídající odhadům popsaným v kapitole 7.3, nejen, že by došlo k přímému násobení odhadované pracnosti počtem entit, ale také by se navýšila pracnost vlastníci společnosti data sbírat a kontrolovat. Vycházíme totiž z předpokladu, že by každá entita přistupovala k implementaci svým způsobem, najala vždy jiného dodavatele a použila vlastní obchodní model. Což zcela jistě není nesprávné, ale docela to neslouží potřebám mateřské společnosti. Ta naopak vyžaduje striktní dodržování jedné verze pravdy, kvalitní dodávky reportingu, které jsou postavené na stejných základech a tedy se spolu dají konsolidovat a porovnávat, v neposlední řadě ale také požaduje dodávky v daných časových termínech, což by ruční kontrola velice omezovala.

Z popisu modelové situace víme, že mateřská společnost vlastní jedenáct dceřiných firem na jiných územních celcích, s jinou kulturou a s jinou technologickou vyspělostí. Za předpokladu, že řešení pro jednotlivou entitu vychází na 164,6 MD, pak jednoduchým násobením těchto odhadů s počtem entit dojdeme na číslo 1 810,6 MD. Zde ovšem odhadované náklady nekončí. Víme, že mateřská společnost je nucena reportovat regulátorům jako jsou kupříkladu centrální banky a proto má jasnou představu o výstupu. Pokud ale dostane jedenáct hlášení z jedenácti různých řešení, je více než pravděpodobné, že bude muset každé z hlášení projít tým specialistů a data schválit, či poslat k opravě. To je nejen časově, ale i finančně náročné. Taková situace není dlouhodobě udržitelná, vzhledem k faktu, že takové konsolidace by musely probíhat minimálně na měsíční bázi.

Jak tedy takovou situaci řešit? Jak dosáhnout jediné verze pravdy ve všech subspečnostech a zároveň dostát termínům a zajistit kvalitní výstupy? Odpovědí na tyto otázky je řízená replikace Business Intelligence řešení, díky které je možné nejen snížit

plánované výdaje na zavedení řešení, ale také výrazně navýšit kvalitu výstupů a snížit množství manuálních kontrol.

7.4.1 Výhody a nevýhody

Jako každé řešení, i toto s sebou nese určité výhody a nevýhody. Mezi hlavní výhody patří zejména autonomnost jednotlivých entit, tzn. že v případě potřeby změny je pobočka schopna zavést řešení v podstatě okamžitě pomocí vlastního vývojového týmu a vlastních prostředků. Další výhodou jsou nižší zaváděcí náklady a celková nutnost kooperace. V podstatě se jedná o systém, kdy každá entita samostatně rozhoduje o svém vlastním řešení, má vlastní tým specialistů, kteří takovému řešení zcela rozumí a tedy je jednoduché jej spravovat.

Na druhé straně zde ale máme také řadu nevýhod. Prvotní nevýhodou, jak již bylo zmíněno, je určitá přirozená rozlišnost přístupů všech entit, které zároveň se svou autonomností do řešení přinášejí řadu nesrovnalostí a odchylek. Tyto rozdílnosti jsou pak nutné konsolidovat na úrovni mateřské společnosti, kde tím vzniká nutnost zapojit další tým specialistů, kteří budou rozumět nejen potřebnému cíli, ale také všem implementačním technikám a obchodním modelům jednotlivých poboček. Zejména díky tomu, že nejsou pobočky nuceny mezi sebou spolupracovat a porovnávat navzájem získané výstupy, může dojít k nedodržení jednoho základního principu datových skladů a sice pravidlo jednotné pravdy.

7.5 Replikace řešení do více entit

Primárním cílem naší společnosti je uspokojit potřeby zákazníka. Tyto potřeby je třeba monitorovat na úrovni dceřiných poboček, což lze provádět několika možnými způsoby. Díky tomu, že pomocí Business Intelligence lze realizovat celou řadu výhod (viz kapitola 3.2 – Benefity využívání BI), rozhodla jsem se ve své práci využít právě tento systém. Ovšem je také třeba zvážit, že pomocí centralizovaného řešení sice můžeme dosáhnout výše uvedených benefitů, na druhou stranu ovšem nejsme schopni ovlivnit způsob, jakým je systém vytvořený, zda se chová podle našich představ, či zda se v budoucnu bude rozvíjet stejně ve všech entitách. Právě díky této skutečnosti vzniká další potřeba mateřské společnosti a tou je konsolidace dat na úrovni nad entitami. Této konsolidace se nám podaří dosáhnout právě pomocí replikace.

7.5.1 Nástroje replikace

K tomu, abychom mohli řešení efektivně replikovat, neboli kopírovat, budeme využívat nástroje, které převedou manuální práci na automatickou. Tím se proces nejen několikanásobně urychlí, ale také dochází ke snižování rizika lidské chyby. Zároveň je ovšem nutné počítat s tím, že zavedení těchto nástrojů jednorázově navýší náklady spojené s replikací.

7.5.1.1 Velocity nástroj

Velocity je nástroj, který využívá tzn. Template language. Z angličtiny převzato, template vyjadřuje šablonu, language jazyk. Jednoduše řečeno se jedná o takový programovací jazyk, který pracuje na základě šablon. Primárním prostředkem tedy je vytvořit šablonu, strukturovaný soubor, do kterého poté uživatel, neboli vývojář vkládá vlastní parametry a vlastní kódy v souladu s již zmíněnou šablonou, a vlastní skripty jsou pomocí nástroje jednoduše vygenerovány. (Apache.org, 2016)

V kontextu tohoto řešení se tedy jedná o vytvoření určitého hlavního skriptu, který bude obsahovat veškeré nutné náležitosti jako je například hlavička skriptu, přihlašovací prostředky a nebo také určení kódovací formátu, jako je UTF8. Tomuto hlavnímu skriptu říkáme šablona, nebo také master skript.

Dalším krokem je definice proměnných, zástupných symbolů, které nazýváme parametry. Tyto parametry jsou uloženy v knihovně jazyka a mohou představovat v našem případě konstanty jako je územní měna, rozlišovací kód pobočky, či dokonce jsme pomocí nich schopni rozlišit celé úseky kódu tak, kdy jedna podmínka kódu může být platná pro jednu pobočku, ale není již platná pro pobočku druhou. V takovém případě lze využít parametrů, které jsou jednoduše vloženy do hlavní šablony. Vytváření parametrů je jasně definováno, každý parametr musí mít specifický identifikátor, kdy jméno parametru je omezeno pouze na znaky abecedy bez diakritiky, pomlčku a podtržítka. Dále je nutné parametr uvést prefixem pomocí znaku dolar (\$) a ohraničeno složenými závorkami viz příklad: `#{branchCode}`. Je dobrým zvykem nazývat parametry v anglickém jazyce. Níže je zobrazen příklad takového parametru pro označení pobočky, anglicky branch.

Předpokladem funkčnosti tedy je, že vývojář vytvoří hlavní skript, nadefinuje veškeré parametry pro pobočky a skripty jednotlivých entit jednoduše pomocí Velocity vygeneruje.

K tomu, aby bylo možné jednotlivé parametry využívat, je potřeba vytvořit technický podklad, kterým je myšlena databázová tabulka, obsahující seznam generovaných skriptů, jejich parametrů a hodnot, které obsahují. Následující tabulka ukazuje, jak má správně vypadat nadefinovaná tabulka obsahující hodnoty daného parametru. Příklad uvedený v tabulce 4 znázorňuje využití parametru pro kód měny. Jak je vidět šablona je stále stejná, v prvním řádku bude skript vygenerovaný pro první pobočku z kódem měny USD, pro druhou pobočku s kódem měny CZK a pro všechny ostatní pobočky, což symbolizuje znak otazníku, s kódem měny EUR. Všechny výskyty ve vygenerovaných skriptech, kde je vložen kód parametru budou nahrazeny zadanou hodnotou.

Tabulka 4 - Parametrizační tabulka

Template_Name	Branch_Code	Parameter_Name	Parameter_Value
Template_01	Branch_01	currencyCode	USD
Template_01	Branch_02	currencyCode	CZK
Template_01	?	currencyCode	EUR

Zdroj: vlastní

Využití Velocity nástroje tedy znamená udržovat veškerý nutný vývoj na jednom místě, je lehce ovladatelný a spravovatelný. Pokud víme, že je nutné provést změnu podmínky, která je platná pro všechny entity, není nutné změnit n skriptů, je potřeba pouze upravit hlavní šablonu, případně parametry a všechny skripty znovu vygenerovat. Díky tomu jsou snižovány jak náklady na vývoj, tak i razantně klesá chybovost způsobená manuální úpravou. Nelze ovšem pominout určitou počáteční investici, kterou zavedení této metodiky obnáší. Protože se jedná o zmapované prostředí již nějakou dobu využívané ve většině oblastech programování, předpokládáme, že startovní náklady budou 50 MD.

7.5.1.2 Data model management

Jak jsme již popisovali v předešlých kapitolách, modelování je nedílnou součástí dodávky datového skladu. Je o ovšem manuální proces, uvažujeme-li tedy vytvoření řešení pro jednu pilotní a deset replikovaných entit, znamenalo by to zapracovat modely celkem jedenáctkrát. To s sebou nese jak větší pracnost, tak také větší riziko manuální chyby.

Využijeme tedy stejný princip jako u Velocity nástroje, tedy templating language. Hlavní myšlenka spočívá ve vytvoření jednoho primárního modelu, který bude sloužit jako šablona a stěžejní zdroj informací pro všechny ostatní entity. V základě tedy využijeme model primární entity, který přetvoříme na jakousi šablonu, kde veškeré odlišnosti všech poboček budou reprezentovány parametry. Tyto parametry budou samy o sobě nadefinovány v databázové vrstvě stejným způsobem, jak již bylo vysvětleno v předchozí kapitole.

Zavedením šablony primárního modelu dosáhneme maximální konzistence všech odnoží modelů, mitigace rizika manuální chyby a vygenerováním modelu pomocí paramaterizace se nám podaří získat všech jedenáct datových modelů automatizovaným způsobem. Důležité je ovšem zmínit, že prvotní proces zavádění toho procesu do implementace je náročný, vzhledem k nutnosti odchycení všech nesrovnalostí, nutnosti harmonizace řešení a potřeby definovat veškeré proměnné. Proto také musíme počítat s náklady, které s sebou implementace nástroje data model maintenance nese. Tyto náklady odhaduje fixní položkou na 40 MD, což v podstatě znamená práci dvou zaměstnanců po dobu jednoho měsíce.

7.5.2 Proces replikace a její odhadovaná pracnost

Jak vlastně replikace probíhá si vysvětlíme krok po kroku na aktivitách, které jsme si předestřeli pomocí WBS. Největším přínosem samotné replikace je neduplikování stejné manuální práce, udržování řešení v harmonizovaném stavu. Protože jsme již prošli veškerými body v pilotní entitě, budou se od toho náležitě odvíjet jak kroky v replikované entitě, tak i odhadovaná pracnost. Stěžejním předpokladem replikace, jak bylo definováno, je fakt, že o všechny entity se stará tentýž tým, který je tím pádem schopný aplikovat své poznatky z pilotní entity na všechny ostatní pobočky.

1. Revize stávajícího, lokálního řešení, a organizace workshopů se zaměstnanci banky

Již jsme zanalyzovali stávající řešení v pilotní entitě, a protože všechny námi zkoumané entity sdílejí stejná zdrojová data, není již tento krok v replikované entitě tak náročný. Z pilotní entity si již s sebou neseme stěžejní poznatky, které budeme uplatňovat na entitu replikovanou, tzn. Plánem je ověřit si podobnost, respektive rozdílnost chování. Předpokládáme, že dceřiné pobočky se chovají víceméně stejně, mají velice podobný obchodní model a naším úkolem tedy je zjistit veškeré odlišnosti a nesrovnalosti. Toho dostáhneme právě porovnáním dokumentací a schůzkami se zaměstnanci replikované banky.

Odhadovaná pracnost = vyjádřená procentuálně je 20%, tedy 1MD pro jednu pobočku. Cílem této aktivity již není pochopit celkové chování, naopak je to pouze porovnání již zmapovaného řešení a evidence veškerých rozdílů. Díky tomu již nepotřebujeme stejný čas ani stejný materiál ať už lidský, či materiální.

2. Získání datových zdrojů

Pakliže entity nevyužívají stejnou správu pro zdrojová data, zde nám ani replikace nepomůže ke snížení požadovaných nákladů. Je nutné si data vyžádat od každé entity zvlášť.

Odhadovaná pracnost je tedy stále 1MD, víme totiž, že nestačí prohlédnout pouze jeden zdrojový soubor a očekávat, že i soubory pro ostatní entity budou ve stejné kvalitě. Naopak musíme vynaložit stejné úsilí pro každé replikované řešení.

3. Data assessment datového zdroje

Na základě zjištění, které jsou dostupné z pilotní entity již máme souhrn všech nedostatků, které se v zdrojových datech nacházejí, víme tedy přesněji, na co se zaměřit, kde hledat a jak data opravit, či jak k nim přistupovat. Díky tomu jsme schopni snížit odhadovanou pracnost na polovinu pro každou entitu.

Odhadovaná pracnost = 5,5MD, při nezměněné velikosti scope 20 objektů tzn. 0,25 MD na zkoumaný objekt a 0,5MD pracnosti na technický load dat do databáze. Veškerá tato snížení již klasicky vychází s přenositelností informací mezi jednotlivými entitami. Analytik, či tým analytiků, již tuto aktivitu jednou prováděl, tím pádem si je vědom všech úskalí, ví jak s daty pracovat, ví jaký je očekávaný stav a jaké rozdíly jsou zlomové při budoucí analýze.

4. Navržení datového modelu

Největší úspora nákladů je zde realizována díky aplikaci nově integrovaného nástroje, popsaného v kapitole 7.5.1.2. Pokud by tato aktivita probíhala odděleně ve dvou separátních vývojových týmech, náklady by se přímo násobily, vzhledem k tomu, že veškeré manuální kroky musí být provedeny právě tolikrát, o kolika entitách mluvíme. Protože ale řešení replikujeme a zároveň využíváme tohoto nástroje, jsme schopni celý modelovací proces projít pouze jednou a to s navýšením nákladů na parametrizaci výstupu. Tzn. ačkoliv jsou

entity velice podobné, nejsou zcela identické, je tedy nutné do modelu zanést také veškeré odlišnosti.

Odhadovaná pracnost = 12,1 MD, tedy o polovinu méně než bylo kalkulováno pro pilotní entitu. Tento fakt vychází právě z použití již výše zmiňovaného nástroje.

5. Sepsání funkční specifikace

Protože naše řešení neobsahuje každou entitu zvlášť, nýbrž je seskupuje do jednoho celku, není potřeba sepsat právě tolik dokumentace, kolik je vyvíjených datových skladů. Je pouze nutné zachytit všechny rozdílnosti a odlišnosti od pilotní entity. Náročnost takové aktivity tedy pro replikovanou pobočku klesá.

Odhadovaná pracnost = 3,6 MD pro replikovanou entitu, což je právě 20% z původního odhadu pracnosti. Je to dáno právě velice sníženou pracností, kdy více než polovina práce již byla odvedena na pilotní entitě.

6. Vytvoření datového workflow

Samotné vytvoření datového toku je operace, obsahující několik po sobě jdoucích kroků, které je v případě potřeby nutné opakovat tak dlouho, dokud load neproběhne úspěšně a nejsou hlášeny žádné chyby. Protože již toto odladění proběhlo na pilotní entitě, očekáváme, že replikace workflow již nebude tak náročná, respektive samotné vytvoření toku je již zmapováno a je pouze potřeba provést tyto kroky znovu za pomoci vědomostí získaných z první entity.

Odhadovaná pracnost = 1,9 MD.

7. Sepsání technické specifikace pro potřeby datového workflow

S narůstajícím počtem vyvíjených řešení se nemění ani náležitosti datového workflow, ani přístup k samotnému vývoji. Proto díky tomu, že jsme již jednou takový problém řešili, se nám podaří snížit pracnost této úlohy na polovinu.

Odhadovaná pracnost = 2,25 MD, což odpovídá právě 50% původní estimace.

8. Příprava zadání pro vývojáře

Ani kroky vedoucí k přípravě zadání pro vývojáře se s počtem entit nemění, víme, že pobočky jsou si více podobné než jsou rozdílné, a proto náš tým expertních analytiků ví přesně, jak má zadání vypadat, kde jsou slabá místa a na co se zaměřit. Je nutné specificky vyzdvihnout rozdílnosti, což má největší podíl na pracnosti uvedené níže.

Odhadovaná pracnost = 5,3 MD, obohatit zadání o další entity zabere analytikům pouze jednu čtvrtinu původního času. Je to dáno právě tím, že zadání je již vytvořené, musí se pouze upravit tak, aby odpovídalo potřebám nové pobočky.

9. Vývoj

Dalším z aktivit, která využije replikačního nástroje je právě vývoj. Právě zde bylo identifikováno největší množství manuální práce a také největší riziko lidských chyb, pokud by se postupovalo způsobem více vývojových týmu. Zkráceně řečeno, pokud by deset vývojářů psalo deset skriptů, pracnost se bude násobit, zatímco při použití Velocity tool nástroje tak, jak je popsán v kapitole 7.5.1.1 - Velocity, bude pouze jeden vývojář pracovat na jednotném skriptu, který bude šablonou pro veškeré pobočky. Jednoduchým vygenerováním vlastních skriptů pomocí nástroje tím ušetříme až 75% času a nákladů. Netřeba vyzdvihnout také značné omezení rizika chyby.

Odhadovaná pracnost = 3,5 MD, zapracování parametrů do definované podoby a následné doladění odlišností do skriptů již samo o sobě není tak náročné jako vlastní vytváření nových skriptů. Díky tomu jsme schopni skripty pro replikovanou entitu zapracovat v rozsahu 25% s původního odhadu.

10. Testy vývojáře

Díky napravení všech chyb již na úrovni šablony ani samotný proces testování již není tak náročný jako u první vyvinuté entity. Odhadujeme, že snížení nákladů bude možné až do výše 75%.

Odhadovaná pracnost = 2,8 MD, formální kontrola vygenerovaného skriptu již není tak složitá vzhledem k unifikované podobě, které bylo dosaženo díky Velocity toolu.

11. Příprava funkčních testů

Protože dva stěžejní body celé dodávky již jsou obsluhovány automaticky, předpokládáme, že většina případných chyb se bude vyskytovat na všech entitách. Vedlo by to k závěru, že test replikované entity se dá zvládnout ve stejném rozsahu jaký byl stanoven pro pilotní entitu a dodatečné náklady by tedy byly nulové. Tento závěr by byl ovšem mylný, protože z jich předchozí analýzy víme, že entity se od sebe navzájem liší a ačkoliv rozdíly nemusí být výrazné, dopad na výstupní data při nedodržení těchto odlišností by mohl být značný. Je tedy nutné testovací scénáře přizpůsobit každé entitě zvlášť. Již ze slova „přizpůsobit“ je jasně patrné, že se takové testy nepřipravují znovu od začátku, nýbrž se pouze upravují již existující. Tento fakt vede ke zdatelné úspoře.

Odhadovaná pracnost = 7,875 MD. K číslu jsme došli pomocí procentuálního vynásobení původní pracnosti, odhaduje totiž, že obohacení testů o odlišnosti jednotlivých dceřiných bank týmu analytiků zabere pouze polovinu původního času.

12. Testování interními pracovníky

Celková práce odvedená na replikované entitě je několikrát nižší než na entitě pilotní. Bohužel ale je nutné data testovat poctivě pro každou zemi zvlášť. Není možné vykonat testy na jedné entitě a prohlásit tak veškeré ostatní řešení za správná. U této aktivity tedy zůstáváme na stejném odhadu pracnosti jako u první entity.

Odhadovaná pracnost = 15,75 MD.

13. Testování s konečnými uživateli a jejich podpora

Ani zde nejsme dostatečně připraveni na snižování odhadovaných nákladů. Předpokládáme, že v každé entitě se nacházejí vlastní koncoví uživatelé o které se bude potřeba starat. Znamená to tedy navýšení komplexnosti podpory. I v tomto případě tedy námi odhadovaná pracnost zůstává nezměněná.

Odhadovaná pracnost = 9,45 MD.

14. Příprava řešení k nasazení

Každé řešení bude spravováno na vlastní databázovém podkladu, tudíž je nutné vytvořit zvláštní nasazovací balíček. Protože jsme ale již tuto aktivitu jednou prováděli, odhadujeme sníženou pracnost o polovinu.

Odhadovaná pracnost = 1 MD.

15. Sepsání průvodky nasazení

Stejně jako jsme byli schopni snížit náklady na sepsání veškerých ostatních dokumentací, i tady se můžeme spolehnout na první verzi dokumentací sepsaných v první entitě inspirovat se z nich a obohatit je o dané odlišnosti.

Odhadovaná pracnost = 0,5 MD

16. Nasazení řešení

Stejně jako podpora koncových uživatelů i podpora nasazení se nedá globálně snížit. Je nutné podpořit každé jedno řešení stejně, ačkoliv jsou si tolik podobné. Právě díky těmto důvodům odhadovaná pracnost zůstává konstantní i pro replikovanou entitu.

Odhadovaná pracnost = 6,3 MD.

Tabulka 5 - Odhad pracnosti v replikované entitě v jednotkách MD

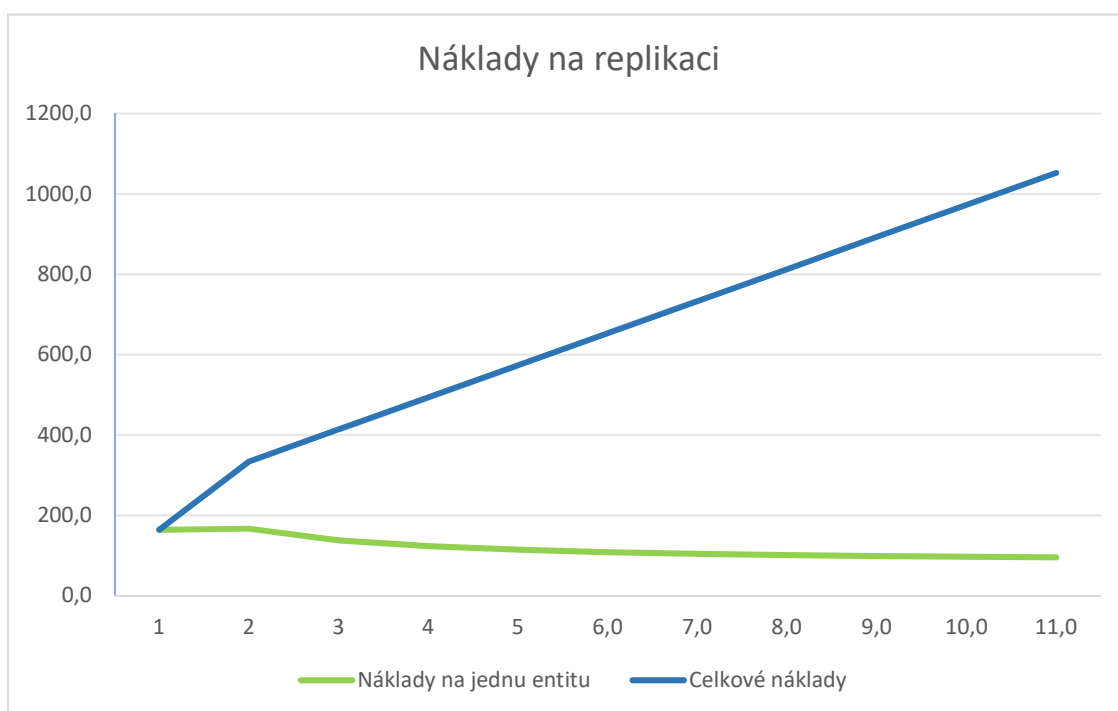
Oblast	Úloha	Popis	Počet objektů	Analytik	Vývojář	Celkem
Vstupy	1. Revize stávajícího řešení			1,0		1,0
	2. Získání datových zdrojů			1,0		1,0
Data	3. Data assessment zdrojových dat	Technický load	1	0,0	0,5	0,5
		Zkoumání dat	20	5,0	0,0	5,0
Workflow	4a. Příprava modelů pro zdrojová data	Jednoduchá	5	0,3	0,0	0,3
		Střední náročnost	8	1,0	0,0	1,0
		Vysoká náročnost	7	1,8	0,0	1,8
	6. Tvorba loadovacího workflow	Core systém	1	0,0	0,8	0,8
		Transformační fáze	3	0,0	1,1	1,1
	7. Testy a tvorba dokumentace pro workflow	Core systém	1	0,0	0,8	0,8
Transformační fáze		3	0,0	1,5	1,5	
Transformace dat	4b. Příprava modelů	Jednoduchá	3	0,5	0,0	0,5
		Střední náročnost	2	0,8	0,0	0,8

	8. - 9. Vývoj skriptů	Vysoká náročnost	7	7,9	0,0	7,9	
		Jednoduchá	3	0,6	0,4	0,9	
		Střední náročnost	2	0,8	0,5	1,3	
		Vysoká náročnost	7	3,9	2,6	6,6	
	10. Test vývojáře	Jednoduchá	3	0,0	0,3	0,3	
		Střední náročnost	2	0,0	0,4	0,4	
		Vysoká náročnost	7	0,0	2,1	2,1	
Dokumentace	5. Sepsání funkční specifikace		36	3,6		3,6	
Testy	11. Příprava testovacích scénářů		32		7,9	7,9	
	12. Vlastní testy = 25%				11,3	4,4	15,8
	13. Podpora uživatelského testování = 15%				6,8	2,7	9,5
Balíky	14. Příprava balíků = 2MD		1			1,0	1,0
	15. Příprava průvodky = 1MD		1			0,5	0,5
	16. Podpora nasazení = 10%				4,5	1,8	6,3
Celkem MD				58,5	21,3	79,8	

Zdroj: vlastní

Jak je vidět z tabulky 5, která je odrazem provedených estimací, při použití replikovaného řešení pro jednu entitu, jsou samotné náklady na implementaci Business Intelligence řešení za použití centralizovaného datového skladu 79,8 MD, což je pouze 49% z odhadované pracovní síly pro pilotní zemi. Pro objektivnost je ale potřeba k estimaci přičíst také náklady na integraci replikačních nástrojů popsaných v kapitole 7.5.1, což je v součtu 90 MD. To nás vede ke zjištění, že samotné náklady na realizaci řešení jsou nižší, jsou ovšem také navýšeny o jednorázové fixní náklady. Celkové náklady tedy dosahují výše 169,8 MD,

což je ve výsledku více, než pokud by implementované řešení proběhlo v každé entitě zvlášť.



Obrázek 12 - Náklady na replikaci řešení

Zdroj: vlastní

Tabulka 6 - Náklady na replikaci řešení

Počet entit	Náklady na entitu	Náklady na řešení
1	164,6	164,6
2	167,2	334,3
3	138,0	414,1
4	123,5	493,8
5	114,7	573,6
6,0	108,9	653,3
7,0	104,7	733,1
8,0	101,6	812,9
9,0	99,2	892,6
10,0	97,2	972,4
11,0	95,6	1052,1

Zdroj: vlastní

Na základě dat získaných expertním odhadem pracností byla sestavena tabulka 6, nad kterou je také graficky znázorněn průběh nákladů pomocí obrázku 12, kde na svislé ose jsou vypsány odhadované estimace, na vodorovné ose pak počet entit. Z průběhu obou funkcí si můžeme všimnout, že ačkoliv celkové náklady rostou, náklady rozpočtené na jednu entitu jsou pro jednu a dvě entity téměř stejné, nicméně pomocí replikace řešení nedosáhneme

u druhé entity snížení, ale naopak zvýšení o 2,6 MD. Od třetí entity ovšem už náklady na entitu klesají a tím se právě projevuje efektivnost daného řešení.

7.5.3 Výhody a nevýhody

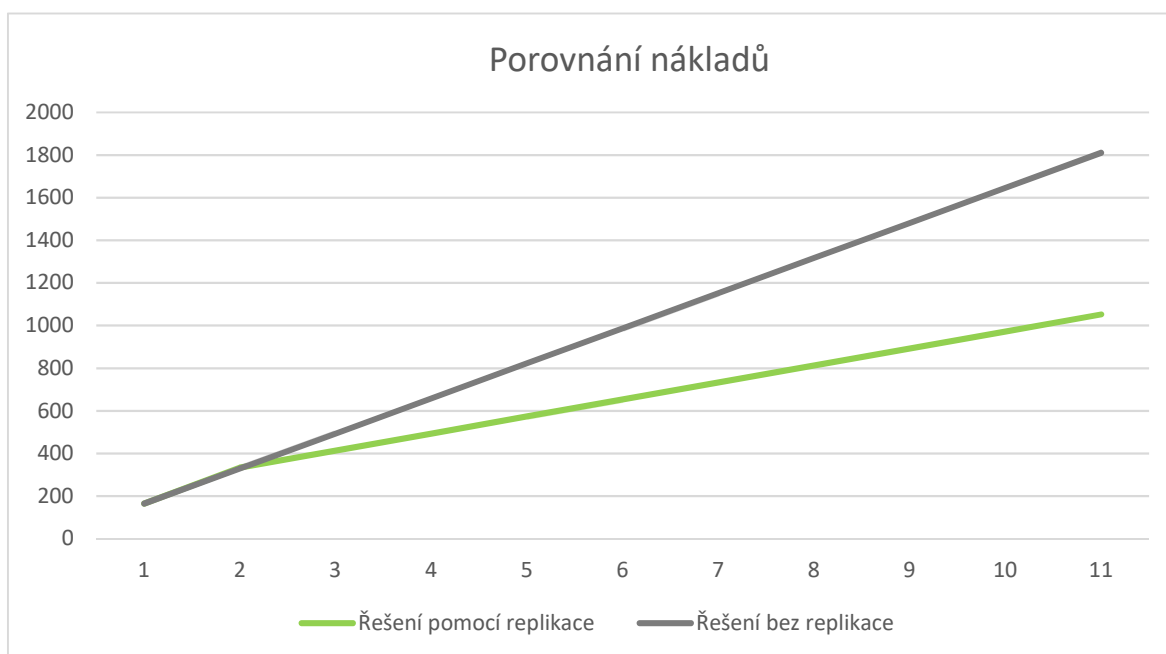
Přístup samotné replikace s sebou nese několik významných skutečností, které pozitivně ovlivňují celkové zhodnocení použité metody. Mezi tyto body se řadí zejména udržování jedné verze pravdy, totiž pokud je obchodní model harmonizován napříč entitami, a jsou známy a evidovány veškeré odchylky od standardního řešení, je řešení velice dobře udržovatelné a spravovatelné. Další značnou výhodou je informovanost mateřské banky, která skrze replikované řešení může sbírat, implementovat a řídit veškerá pravidla, která ji přímo ovlivňují. Kromě snížení nákladů tedy značně snížíme riziko rozdílnosti například regulačních výstupů.

I tento systém má samozřejmě svá úskalí a tím jsou mimo jiné také náklady na zavedení a jakási nutnost vyjednat s pobočkami body spolupráce. Vzhledem k územním, kulturním a jiným zvyklostem tedy nemusí být vždy jednoduché nastavit podmínky jednotného řešení, kdy v podstatě pobočka jako taková ztratí vlastní řízení daného řešení, ačkoliv to neznamena, že přijde o vliv na něj. Jako nevýhodu také můžeme vnímat potřebu velkého zapojení ze strany mateřské instituce, která je tímto nucena zpracovávat jednotlivé požadavky a implementovat je na úrovni konsolidovaného řešení, což může znamenat nemožnost nasazení nového řešení, pokud každá jedna entita na to není dostatečně připravena ať už technicky, či jinak.

8 Vyhodnocení

Abych mohla správně posoudit a zhodnotit, kdy je vhodné replikovat řešení, je třeba posuzovat dvě nezávislé proměnné – počet entit a způsob implementace řešení Business Intelligence. Jedním ze způsobů je tedy implementace bez použití replikačního přístupu, druhým způsobem je pak zavedení replikace.

Následující graf znázorňuje vývoj nákladů při dvou různých přístupech a beroucí v potaz různé počty entit. Na svislé ose jsou znázorněny odhadnuté náklady v jednotkách MD, na vodorovné ose pak právě počet entit. Jak je z obrázku 13, který je založený na datech z tabulky 7, patrné, počáteční pozice je pro oba přístupy stejná. Ať už je preferována varianta bez replikace, či varianta s replikací, první, neboli pilotní entita se náklady nijak neliší. Co ovšem vidíme u druhé entity, navýšení nákladů na integraci replikačních nástrojů způsobilo, že při využití replikačního přístupu, jsou náklady o 5,1 MD vyšší než při jeho nevyužití. Efektivnost se projevuje tedy až od třetí entity, kdy celkové ušetřené náklady na implementaci jsou celých 79,7 MD. Stejně postupujeme porovnáváním přibývajících entit jak je vidět v tabulce 7 sloupci Rozdíl. Konečně se dostáváme až k celému obsahu řešení a to implementace řešení do jedenácti vzájemně nezávislých řešení, kde celková úspora při využití replikovaného řešení činí 758,5 MD, což by pokrylo implementaci dalších devíti entit.



Obrázek 13 - Porovnání dvou přístupů v implementaci

Zdroj: vlastní

Tabulka 7 - Porovnání dvou přístupů v implementaci

Počet entit	Řešení pomocí replikace (MD)	Řešení bez replikace (MD)	Rozdíl (MD)
1	164,6	164,6	0
2	334,3	329,2	-5,1
3	414,1	493,8	79,7
4	493,8	658,4	164,6
5	573,6	823,0	249,4
6	653,3	987,6	334,3
7	733,1	1152,2	419,1
8	812,9	1316,8	503,9
9	892,6	1481,4	588,8
10	972,4	1646,0	673,6
11	1052,1	1810,6	758,5

Zdroj: vlastní

8.1 Návrh na budoucí rozvoj

Jak je patrné z analýzy provedené metodou WBS, některé z identifikovaných kroků jsou i nadále prováděny manuálně, tedy ručně. Vidíme tedy, že je zde dále prostor pro zlepšení, která identifikují zejména v oblastech technického řešení. Pokud bychom zavedli jednotnou infrastrukturu pro všechny obsažené entity, snížili bychom náklady na zavedení jednotlivých databázových komponent, a to zejména z toho důvodu, že by jednotlivé komponenty bylo možné mezi pobočkami sdílet.

Prostor pro budoucí rozvoj je také zavedení jednotné prezentační vrstvy, která touto prací není plně pokryta. Řešení předpokládá pouze zavedení výstupních pohledů, které umožní uživatelům provádět analytické dotazování do dat a vytvářet samotné reporty. Dalším krokem by tedy bylo zavedení softwarových nástrojů pro automatizaci těchto aktivit.

9 Diskuze a závěr

Business Intelligence ve svém širokém pojetí neslouží pouze pro podporu manažerského rozhodování nebo oporu při zavádění nových produktů, využívá se také jako zdroj pro regulatorní reporting. O to více je kladen důraz na kvalitu dat, přesnou interpretaci a jasnou konsolidaci zdrojových systémů, jakožto i jednotlivých oddělení samotné společnosti.

Modelové řešení popsané v této práci vystihuje potřeby mateřské bankovní instituce dosáhnout těchto cílů, tedy získávat data konsolidovaná napříč všemi vlastněnými společnostmi, a to jak v požadované kvalitě, tak i v požadovaném čase. Tento úkol se jeví jako velice těžký za předpokladu, že každá dceřiná firma tato data poskytuje na bázi vlastních obchodních modelů a vlastní interpretace. Právě na tuto oblast se zaměřuje praktická část diplomové práce, kde je vysvětlena problematika zavádění Business Intelligence řešení, založeného na centralizovaném datovém skladu, pomocí WBS metody.

Diplomová práce pojednává o postupu zavádění řešení do jedné entity a zohledňuje odhadnutou pracnost v jednotkách MD. V další části je rozebráno, jaké nástroje je nutné integrovat pro potřeby replikačního řešení a jak se pomocí nich a pomocí replikační logiky liší provedené kroky a odhadovaná pracnost pro replikovanou entitu. V závěru práce je poté vyhodnoceno, zda je replikované řešení efektivní, tedy zda dosahuje nižších nákladů než řešení, které nebylo vyvinuto metodou replikace. Díky znázorněným výhodám a nevýhodám každého z použitých přístupů, může tato práce sloužit jako podpora pro takovou společnost, která se právě rozhoduje, jak spravovat svá řešení v jednotlivých pobočkách. Ačkoliv se názory na tuto problematiku mohou lišit, dle mého názoru je vhodnější variantou právě použití řešení replikovaného, a to zejména z toho důvodu, že lépe slouží potřebám mateřské společnosti, lze jej snadněji udržovat a minimálně z pohledu nákladového je výhodnější.

Přínosem této práce je identifikace replikačních nástrojů pro potřeby implementace několika navzájem podobných řešení Business Intelligence. Výstupy této práce dokazují, že při použití replikační logiky je možné snížit náklady, mohla by tedy sloužit jako souhrn metod a postupů, jak takového snížení nákladů dosáhnout.

10 Seznam použité literatury

9Gauge.com. (2017). Is a centralized data system right for your organization?. 9Gauge Partners, LLC. [online]. Dostupné z: <https://9gauge.com/centralized-.distributed-data-system/>. [Cit. 15.7.2018].

ALSQOUR, M., MATOUK, K. & OWOC, M. L. (2012). A survey of data warehouse architectures — Preliminary results. Federated Conference on Computer Science and Information Systems, FedCSIS 2012, s. 1121-1126.

Apache.org. (2016). The Apache Software Foundation. [online]. Dostupné z: <http://velocity.apache.org/>. [cit. 3.9.2018]

ARIYACHANDRA, T., WATSON, H. J. 2006. Which Data Warehouse Architecture Is Most Successful?. Business Intelligence Journal. Secured Assets Yield Corporation Ltd.

BASS, L., CLEMENTS, P. & KAZMAN, R. (2013). Software architecture in practice. Upper Saddle River, NJ: Addison-Wesley.

BOUADI, T., CORDIER, M.-O., MOREAU, P., QUINIOU, R., SALMON-MONVIOLA, J., & GASCUEL-ODOUX, C. (2017). A data warehouse to explore multidimensional simulated data from a spatially distributed agro-hydrological model to improve catchment nitrogen management. Environmental Modelling & Software, 97, s. 229–242.

DANG, C., PHUONG, T., BEDDAG, M., VEGA, A., & DENIS, C. (2018). A data model for clinical legal medicine practice and the development of a dedicated software for both practitioners and researchers. Journal of Forensic and Legal Medicine, 57, s. 12–18.

DEVLIN, B. A., & MURPHY, P. T. (1988). An architecture for a business and information system. IBM Systems Journal, 27(1), 60–80. <https://doi.org/10.1147/sj.271.0060>

FERNANDO, L. (2017). Federated Data Warehouse Architecture. All About Data Warehousing, Data Mining & BI. [online]. Dostupné z: <http://blogsodatawarehousing.blogspot.com/2017/02/federated-data-warehouse-architecture.html>. [cit. 15.7.2018].

Galaktikastoft.com. (2017). OLAP glossary.[online]. Dostupné z: <https://galaktikastoft.com/blog/olap-glossary.html>. [cit. 8.8.2018]

GARCELON, N., NEURAZ, A., SALOMON, R., FAOUR, H., BENOIT, V., et al. (2018). A clinician friendly data warehouse oriented toward narrative reports: Dr. Warehouse. Journal of Biomedical Informatics, 80, 52–63.

- Geekinterview.com. (2015). Centralized data warehouse. [online]. Dostupné z: <http://www.learn.geekinterview.com/data-warehouse/data-types/centralized-data-warehouse.html>. [cit. 5.6.2018]
- GROSSMANN, W. & RINDERLE-Ma, S. (2015). *Fundamentals of business intelligence*. Heidelberg: Springer.
- HOBBS, L., HILLSON, S., LAWANDE, S., & SMITH, P. (2005). *Oracle 10g Data Warehousing* (s. 16-17). Elsevier.
- HOHMANN, L. (2003). *Beyond software architecture : creating and sustaining winning solutions*. Boston: Addison-Wesley.
- HWANG, H.-G., KU, C.-Y., YEN, D. C., & CHENG, C.-C. (2004). Critical factors influencing the adoption of data warehouse technology: a study of the banking industry in Taiwan. *Decision Support Systems*, 37(1), 1–21. [https://doi.org/10.1016/s0167-9236\(02\)00191-4](https://doi.org/10.1016/s0167-9236(02)00191-4)
- INMON, W. (2002). *Building the data warehouse*. New York: J. Wiley.
- INMON, W. H., & LINSTEDT, D. (2015). What a Data Warehouse is Not. In *Data Architecture: a Primer for the Data Scientist* (s. 127–132). Elsevier.
- JEFFERYS, B. R., NKWANKWO, I., NERI, E., CHANG, D. C. W., SHAMARDIN, L., HANOLD, S., ... COVENEY, P. (2013). Navigating legal constraints in clinical data warehousing: a case study in personalized medicine. *Interface Focus*, 3(2), 20120088–20120088.
- KHUNDHUR, P. (2007). Business intelligence: Je třeba přemýšlet. *Computer World*. [online]. Dostupné z: <http://computerworld.cz/whitepapers/business-intelligence-je-treba-premyslet-2095>. [cit. 15.7.2018]
- KIMBALL, R. & ROSS, M. (2002). *The data warehouse toolkit: the complete guide to dimensional modeling* 2nd Ed. New York: Wiley.
- LABERGE, R. (2012). *Datové sklady: agilní metody a business intelligence*. Brno: Computer Press.
- LANEY, D. (2000). Data warehouse factors to address for success. *HP Professional*, 14(5) 21-22.
- LOSHIN, D. (2012). *Business intelligence: the savvy manager's guide*. Waltham, MA: Morgan Kaufmann.
- LUHN, H. P. (1958). A Business Intelligence System. *IBM Journal of Research and Development*, 2(4), 314–319.
- MARCO-RUIZ, L., MONER, D., MALDONADO, J. A., KOLSTRUP, N., & BELIKA, J. G. (2015). Archetype-based data warehouse environment to enable the reuse of electronic health record data. *International Journal of Medical Informatics*, 84(9), 702–714.

- MCKNIGHT, W. (2014). Data Warehouses and Appliances. In Information Management (pp. 52–66). Elsevier.
- Muni.cz. (2015). Databázové systémy. Pedagogická fakulta Masarykovy univerzity. [online]. Dostupné z: http://www.ped.muni.cz/wtech/03_studium/cvt4/databaze.pdf. [cit. 20.8.2018]
- Nkp.cz (2014). Databáze národní knihovny. NK ČR. [online]. Dostupné z: https://aleph.nkp.cz/F/?func=direct&doc_number=000000089&local_base=KTD. [cit. 18.8.2018]
- NORMAN, E., BROTHERTON, S. & FRIED, R. (2008). Work breakdown structures : the foundation for project management excellence. Hoboken, N.J. Newtown Square, Pa: John Wiley & Sons Project Management Institute.
- NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1094-3.
- PARMENTER, D. (2007). Key performance indicators: developing, implementing, and using winning KPIs. Hoboken, N.J: John Wiley & Sons.
- Project management institute. (2008). A guide to the project management body of knowledge (PMBOK Guide). Newtown Square.
- RADFORD, C. (2014). Challenges and solutions protecting data within Amazon Web Services. Network Security, 2014(6), 5–8.
- SANTOSO, L. W., & YULIYA. (2017). Data Warehouse with Big Data Technology for Higher Education. Procedia Computer Science, 124, 93–99.
- SCHEPS, S. (2008). Business intelligence for dummies. Hoboken, N.J: Wiley.
- SHERMAN, R. (2014). Business intelligence guidebook: from data integration to analytics. Amsterdam: Elsevier.
- SIAMI-IRDEMOOSA, E., DINDARLOO, S. R., & SHARIFZADEH, M. (2015). Work breakdown structure (WBS) development for underground construction. Automation in Construction, 58, s. 85–94.
- Soprabanking.com. (2017). IFRS9. [online]. Dostupné z: <https://www.soprabanking.com/our-offer/solutions/finance-risk-and-compliance-16/component/ifrs9-11>. [cit. 15.8.2018]
- Soprabanking.com. (2017). Integrated core banking solution. [online]. Dostupné z: <https://www.soprabanking.com/our-offer/solutions/digital-wallet-23/component/sopra-banking-amplitude-22>. [cit. 15.8.2018]
- STRANGE, K.F. (2003). Making BI strategic: The key issues, LE-19-4691, Gartner Group.

TAYLOR, Allen G. SQL for dummies. 8th edition. Hoboken, New Jersey: John Wiley, [2013]. --For dummies. ISBN 978-1-118-60796-1.

TechnologyAdvice.com. (2018). What is Data Warehousing Software? [online]. Dostupné z: <https://technologyadvice.com/data-warehousing/>. [cit. 15.7.2018]

VAISMAN, Alejandro a Esteban ZIMÁNYI. Data warehouse systems: design and implementation. Heidelberg: Springer, 2014. Data-centric systems and applications. ISBN 978-3-642-54654-9.

VAN LEUKEN, S. (2012). Federated-, Decentralized – and Centralized DataWarehouse. The Sventor.com. [online]. Dostupné z: <http://www.svenvanleuken.com/federated-decentralized-and-centralized-datawarehouse/>. [cit. 15.7.2018].

VANDERWELEE, J., POLLACK, T., OAKES, D. J., Smyrniotis, C., Illuri, V., Vellanki, P., ... Wallia, A. (2018). Validation of data from electronic data warehouse in diabetic ketoacidosis: Caution is needed. *Journal of Diabetes and Its Complications*, 32(7), 650–654.

Vse.cz. (2010). Základy relačních databází. Vysoká škola ekonomická v Praze. [online]. Dostupné z: <https://gml.vse.cz/data/oppa-webdesign/zaklady-db.html>. [cit. 15.8.2018]

WALKER, J. (2018). Top 5 data warehouses on the market today. [online]. Dostupné z: <http://www.monitis.com/blog/top-5-data-warehouses-on-the-market-today/> [cit. 14.7.2018].

ZenTut.com. (2012). Federated Data Warehouse Architecture. [online]. Dostupné z: <http://www.zentut.com/data-warehouse/federated-data-warehouse-architecture/>. [cit. 15.7.2018].