



TECHNICKÁ UNIVERZITA V LIBERCI  
Ekonomická fakulta



# Tvorba FMEA databáze ve ŠKODA AUTO a. s.

## Bakalářská práce

*Studijní program:* B6209 – Systémové inženýrství a informatika

*Studijní obor:* 6209R021 – Manažerská informatika

*Autor práce:* **Aleš Zapadlo**

*Vedoucí práce:* Ing. Dana Nejedlová, Ph.D.





# Building FMEA database in ŠKODA AUTO a. s.

## Bachelor thesis

*Study programme:* B6209 – System Engineering and Informatics

*Study branch:* 6209R021 – Managerial Informatics

*Author:* **Aleš Zapadlo**

*Supervisor:* Ing. Dana Nejedlová, Ph.D.



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Aleš Zapadlo**  
Osobní číslo: **E13000054**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Manažerská informatika**  
Název tématu: **Tvorba FMEA databáze ve ŠKODA AUTO a. s.**  
Zadávací katedra: **Katedra informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

1. Databáze a jejich uplatnění v automobilovém průmyslu
2. Využití programového vybavení při tvorbě databází
3. Analýza současného stavu FMEA dokumentace ve ŠKODA AUTO a. s.
4. Návrh a tvorba FMEA databáze pomocí Microsoft Office
5. Zhodnocení výsledků tvorby FMEA databáze

Rozsah grafických prací:

Rozsah pracovní zprávy: **30 normostran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2. přeprac. a aktualiz. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2615-1.**

**DVOŘÁK, Drahoslav. Řízení projektů: Nejlepší praktiky s ukázkami v Microsoft Office. Brno: Computer Press, 2008. ISBN 978-80-251-1885-6.**

**ČSJ. VDA 4: Zajišťování kvality před sériovou výrobou. Kapitola: FMEA produktu, FMEA procesu. 2. přeprac. a aktual. Praha: ČSJ, 2006. ISBN 80-02-01682-3.**

**MANSFIELD, Richard. Mastering VBA for Microsoft Office 2007. 2nd ed. Indianapolis: Wiley Publishing, 2008. ISBN 978-0-470-27959-5.**

**Elektronická databáze článků ProQuest (knihovna.tul.cz).**

Vedoucí bakalářské práce:

**Ing. Dana Nejedlová, Ph.D.**

Katedra informatiky

Konzultant bakalářské práce:

**Ing. Mikuláš Koukolský**

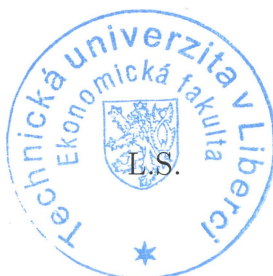
ŠKODA AUTO a. s.

Datum zadání bakalářské práce: **31. října 2015**

Termín odevzdání bakalářské práce: **31. května 2017**



doc. Ing. Miroslav Žižka, Ph.D.  
děkan



doc. Ing. Jan Skrbek, Dr.  
vedoucí katedry

V Liberci dne 31. října 2015

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou 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 bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom 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.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

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

Datum:

Podpis:

# Poděkování

Tímto bych chtěl poděkovat Ing. Daně Nejedlové PhD., za odborné rady a veškeré důležité informace při zpracování mé bakalářské práce a konzultantovi Ing. Mikuláši Koukolskému, který byl zároveň garantem mé roční řízené praxe ve společnosti ŠKODA AUTO a.s. Oběma bych chtěl poděkovat za odborné konzultace a zkušenosti při psaní bakalářských prací. V neposledním případě bych chtěl poděkovat za jejich trpělivost a ochotu při zodpovídání dotazů, které přispěly ke zdárnému dokončení práce.

Rád bych také poděkoval celému oddělení GQZ za výbornou spolupráci během řízené praxe ve ŠKODA AUTO a.s. v Mladé Boleslavi.

# Anotace

Tématem této bakalářské práce je tvorba databáze protokolů FMEA s využitím Visual Basic for Applications v Microsoft Excel ve ŠKODA AUTO a.s. Tato databáze poslouží ke zpřehlednění veškerých záznamů vytvořených ve společnosti a urychlí tak budoucí vyhledávání informací. Teoretická část je zaměřena na informace týkající se základních pojmů v oblasti databází. Dále se v teoretické části setkáme s neznámějšími systémy řízení správy dat, druhy databázových modelů a teorií v oblasti Visual Basic for Applications v Microsoft Excel. V praktické části se zabývám implementací vhodného databázového modelu spolu s volbou uživatelského rozhraní databáze. V závěrečné části práce je zhodnocena tvorba databáze a její prospěšnost pro budoucí projekty ve společnosti.

## **Klíčová slova:**

MS Office, FMEA, VBA, databáze

# Annotation

Building FMEA Database in ŠKODA AUTO a.s.

The topic of this bachelor's thesis is to create database of FMEA protocols using Visual Basic for Applications in Microsoft Excel in ŠKODA AUTO a.s. This database would streamline all records created in corporation and thus speed up all future searches for information. The theoretical part focuses on information related to the basic concepts of databases. Furthermore, the theoretical part deals with the most widely used database management systems along with the types of database models and with the theory of Visual Basic for Applications in Microsoft Excel. The practical part deals with the implementation of suitable database model, along with a choice of database interface. In the closing part the author evaluates created database and its usefulness for future projects in the company.

**Keywords:**

MS Office, FMEA, VBA, database



# Obsah

Obsah .....	7
Seznam ilustrací.....	9
Seznam tabulek.....	10
Seznam použitých zkratk .....	11
Úvod .....	13
1 Rešerše v oblasti databází a jejich využití v automobilovém průmyslu.....	15
2 Teorie tvorby databází .....	21
2.1 Co jsou data .....	21
2.2 Co je to databáze.....	22
2.3 Systém řízení báze dat .....	23
2.4 Databázový model .....	24
2.4.1 Druhy databázových modelů.....	24
3 Programové vybavení pro tvorbu databází.....	28
3.1 Příklady softwarových řešení systémů řízení báze dat .....	28
3.2 Microsoft Office .....	30
3.2.1 MS Word .....	30
3.2.2 MS Excel .....	31
3.2.3 VBA.....	31
3.3 Porovnání softwarových řešení SŘBD .....	32
4 Analýza současného stavu FMEA dokumentace ve ŠKODA AUTO a.s.....	33
4.1 Kvalita .....	33
4.2 FMEA.....	34
4.3 Dosavadní způsob uložení FMEA dokumentace .....	36
4.4 Tvorba FMEA dokumentace ve ŠKODA AUTO.....	36

4.5 RPZ.....	38
4.6 Zhodnocení současného stavu dokumentace.....	39
5 Tvorba FMEA databáze pomocí Microsoft Office.....	40
5.1 Výběr databáze .....	41
5.2 Řešení úloh pomocí VBA v Microsoft Excel.....	42
5.2.1 Unikátnost zápisu .....	44
5.2.2 Indexace obsahu a extrakce dat .....	44
5.2.3 Urychlení procedury .....	45
5.2.4 Sjednocení formátu.....	46
5.2.5 Jednotlivé přidávání souborů a složek.....	47
5.2.6 Link z databáze na protokoly.....	47
5.2.7 Oddělení protokolů .....	48
5.3 Řešení problémů .....	48
5.3.1 Workbooks.Open.....	49
5.3.2 Ukazatel průběhu.....	49
5.3.3 Opisování prázdných řádků .....	50
5.3.4 Výpis neprotokolových souborů.....	51
5.3.5 Změna metadat .....	51
5.3.6 Velikost databáze.....	51
6 Zhodnocení výsledků tvorby FMEA databáze ve ŠKODA AUTO a.s. ....	53
Závěr.....	55
Seznam literatury.....	57
Citace .....	57
Bibliografie.....	60

## Seznam ilustrací

Obr. 1: Vyplněný FMEA protokol .....	37
Obr. 2: Databáze FMEA vytvořená z dat protokolu z Obr. 1.....	42
Obr. 3: Vývojový diagram tvorby databáze v naprogramovaném prostředí Microsoft Excel ..	43

# Seznam tabulek

Tab. 1: Výhody a nevýhody SRBD .....	32
--------------------------------------	----

## Seznam použitých zkratek

API	Application Programming Interface - Termín ze softwarového inženýrství označující rozhraní pro programování aplikací.
CIA	Central Intelligence Agency – Zpravodajská služba U.S.A.
CTE	Common Table Expressions – Dočasný set výsledků definovaný použitím funkcí SELECT, INSERT, UPDATE, DELETE a CREATE VIEW.
ČSN	Československé státní normy vydávané Úřadem pro technickou normalizaci, metrologii a státní zkušebnictví.
FMEA	Failure Mode and Effects Analysis – Analytická metoda jejímž cílem je identifikovat slabá místa v procesu vzniku výrobku.
IIS	Internet Information Server – Webový server vytvořený společností Microsoft pro systém Windows.
ISO	International Organisation for Standardization – Mezinárodní federace sdružující národních standardizační organizace.
IoT	Internet of Things (Internet věcí) – Schopnost smart zařízení připojit se k ostatním zařízením využívajícím stejnou technologii za účel výměny dat a jejich zpracování.
OLE	Object Linking and Embedding – Je technologie vyvinutá společností Microsoft, která umožňuje vkládání a odkazování na data jiných dokumentů či objektů.
RZP	Risiko-Prioritätszahl – parametr určující závažnost dané vady, vykalkulovaný během jednání FMEA.
SQL	Structured Query Language – Strukturovaný dotazovací jazyk pro práci v relačních databázích.
SSD	Solid State Drive – Datové medium založené na technologii Flash paměti, bez mechanických součástí.
SŘBD	System řízení báze dat
V2V	Vehicle to Vehicle – Technologie umožňující bezdrátovou komunikaci mezi auty za účelem výměny dat a jejich zpracování.
VBA	Visual Basic for Applications – Programovací jazyk od společnosti

	Microsoft používaný v balíčku kancelářských nástrojů Microsoft Office.
WISIWYG	What You See Is What You Get – Teoretický model textových procesorů, který převádí podobu stránky, která se zobrazuje na obrazovce do přesné reálné podoby na papíře. Datová verze na obrazovce by tedy měla být totožná s tisknutou stránkou.
XML	Extensible Markup Language – Jazyk určený především pro výměnu dat mezi aplikacemi. Popisuje jen strukturu věcného obsahu.

# Úvod

Společnost ŠKODA AUTO a.s. je největším výrobcem automobilů v České republice a je zároveň jedním z největších výrobců automobilů na světě. V roce 2014 se značka ŠKODA dosažením pomyslné hranice jednoho milionu dodaných vozů posadila na přední místa světových výrobců automobilů. (Výroční zpráva, 2015) Sídlo společnosti je v Mladé Boleslavi, kde je také umístěn její největší závod. V rámci České republiky vlastní ještě závod ve Vrchlabí a Kvasinách. Od roku 2007 patří společnost pod koncern Volkswagen Group, kde si udržuje silnou pozici, zejména díky vysokému podílu na trhu a tržbám.

V současnosti dochází k bouřlivému vývoji moderních technologií a k jejich aplikaci do výroby, jenž zajistí levnější a kvalitnější produkci automobilů. Je však potřeba vývoj a integraci nových technologických řešení pečlivě kontrolovat a zajistit tak pokud možno stejnou, v lepším případě vyšší, kvalitu výstupů, aby se k zákazníkům dostávaly jen ty nejkvalitnější produkty s požadovanými vlastnostmi. K udržení kvality může společnost využívat široké spektrum metod kvality. Metody kvality mohou být kombinovány za předpokladu stanovení jasného systematického plánování již od vzniku výrobku. Jednou z možností, jak předvídat problémy ve výrobě a kontrolovat tak budoucí kvalitu výroby je analýza FMEA. Z důvodů stoupající tendence inovací vozů ŠKODA AUTO a vniku nových modelových řad, nastává nutnost provádět stále častější analýzy FMEA, čímž vzniká stále se zvyšující množství protokolů. Díky nepřebornému množství dokumentů je jakékoliv vyhledávání specifických znaků čím dál složitější a ztrácí se tak prospěch minulých analýz pro budoucí projekty.

Cílem této bakalářské práce je proto problematika vytvoření databáze se vstupními daty extrahovanými z velkého množství nesourodých dokumentů protokolů FMEA, jež byly zpracovány do ucelené formy, odpovídající standardům využívaným ve společnosti ŠKODA AUTO a.s. Autor bakalářské práce nejdříve provede rešerši v oblasti databází využitelných v automobilovém průmyslu, uvede teorii databází, zhodnotí dostupné systémy řízení bází dat a jasně stanoví výhody a nevýhody jednotlivých řešení. Dále se bude práce zabývat analýzou současného stavu FMEA dokumentace ve ŠKODA AUTO.

V praktické části práce bude popsána tvorba FMEA databáze pomocí programovacího jazyka Visual Basic for Applications v aplikaci Microsoft Excel. V závěru práce provede autor zhodnocení vytvořené databáze, její důsledky pro další vývoj analýz a dokumentací FMEA a prospěch databáze v budoucích inovacích značky ŠKODA AUTO a.s.



# 1 Rešerše v oblasti databází a jejich využití

## v automobilovém průmyslu

Data se v posledních letech stávají nedílnou součástí automobilového průmyslu a ovlivňují ho od základu. Automobilky shromažďují data o vývoji a kvalitě automobilů se záměrem je statisticky analyzovat a identifikovat tak zdroje chyb ve výrobě. Podstatným problémem, který se začíná projevovat, je zvyšující se množství dat a jejich nároky na systematické ukládání, třídění, analyzování a aplikaci výsledku analýz zvyšování kvality ve výrobě. Přitom tomu tak nebylo vždy.

Dříve se během výroby automatickými stroji na lince pracovalo zejména se vstupními daty. Jednalo se především o data parametrů komponentů, která byla zadávána do robotů na hale. Data, určená k archivaci se skladovala v papírové formě v kartotékách a hromadila se stále se zvyšujícím tempem. Do dnešní doby zůstala povinnost archivovat karty vozů ještě patnáct let od výroby vozu. Tyto karty jsou samozřejmě pořád z papíru. Digitalizace této karty je teprve v plenkách a fungují pouze zkušební pilotní projekty paralelně se současným způsobem papírové archivace. Na sběru dat v automobilovém průmyslu se výrazně podílí také řídicí jednotky vozidla. Řídicí jednotka je zařízení, které má na starosti správu a řízení všech elektronických systémů v automobilu, jako jsou například brzdové systémy, vstřikování paliva, senzory apod. Ve ŠKODA AUTO se poprvé objevila řídicí jednotka ve voze Škoda Felicia. Do té doby byly všechny funkce vozidla zajištěny mechanicky (tažná lana, klouby, kladky) nebo jedno-účelovými elektrickými systémy (spínače, relé). (Pernet, 2016)

Zatímco ostatní průmyslové i neprůmyslové obory již započaly fázi digitalizace a několik let úspěšně zpracovávají velké objemy dat, konzervativní odvětví jako je to automobilové stojí před podobnou výzvou. Dnes již objemy dat v zpracovaných automobilkách dosahují velikosti řádově stovek terabajtů. Ze statistik provedených výrobcí automobilů vyplývá, že objem uložených a setříděných dat průmyslovými výrobcí automobilů v roce 2013 dosahoval až 480 TB. Takové množství dat průměrně dokázala shromáždit jediná automobilka za jeden rok svého fungování. Data jsou většinou ve formátu čistého textu,

avšak ne výhradně. Jedná se především o data extrahovaná z řídicích jednotek automobilů a data získaná z nově se vyvíjejících přístrojů, které podporují technologii IoT a sbírají tak obrovská kvanta sensorových dat. Tento objem však bude narůstat. (Luckow, 2015) Lze předpokládat, že meziroční nárůst objemu dat bude dosahovat až 11,1 PB. (Dorsey, 2013) Toto tvrzení se opírá o fakt, že během roku 2018 jen samotné BMW bude mít vyrobeno zhruba 10 000 000 vozů se schopností konektivity, které budou vysílat zpět svému výrobcí 1 TB dat za jediný den. (Luckow, 2015) To je zapříčiněno vysokým nárůstem senzorů a věcí se schopností konektivity. Jedna z takových technologií je například V2V. Vehicle to vehicle je technologie, která zajišťuje komunikaci mezi jednotlivými vozy a umožňuje tak efektivní fungování například navigačních nebo komunikačních aplikací a v blízké budoucnosti se předpokládá využití této technologie pro autonomní řízení. Řídicí jednotka vozidla je schopna sama o sobě generovat několik gigabajtů dat během jedné vteřiny. Pouze nepatrná část z tohoto objemu je však vhodná pro další využití. I tak ale narůstající objemy dat stále více tlačí na efektivnější ukládání a zpracování relevantních dat. V současnosti je toto ukládání řešeno pomocí relačních databází se systémy řízení bází dat typu Oracle, MySQL, PostgreSQL, SQLServer. Avšak očekává se daleko důslednější propojení do komplexních databázových systémů, kde jednotlivé databázové systémy budou propojeny k účelu výměny relevantních dat a jejich efektivnějšímu ukládání. (Luckow, 2015)

Dalším řešením jak efektivně uchovávat a především zpracovávat nasbíraná data v tzv. „data lake“ je Hadoop Framework. Hadoop byl navržen jako open-source implementace abstraktního programovacího modelu MapReduce společnosti Google. Big data společností obsahují miliony záznamů o vysokém datovém objemu, a proto jsou často uchovávány na vysokokapacitních serverech. Z toho důvodu je uživatel, který bude provádět analýzu takových dat, dost omezen. Společnost buď nemá prostředky na vybudování dostatečně velkého lokálního datového úložiště, nebo je omezena datovým přenosem, který je nutný ke čtení záznamů. Implementace Hadoop řeší obě situace. Místo toho, aby se Big data přenášela na zařízení vybavené programem pro zpracování, přenesou se program na server, kde jsou umístěna. Hadoop je zatím velmi komplikovaný, ale v poslední době zažívá obrovský rozvoj. (Luckow, 2015)

Spolu s daty z běžného provozu rostou také požadavky na ukládání dat před výrobou a během výroby vozidla. Jedná se o data z analýz před vypuštěním automobilu do sériové výroby, ale také během výroby samotné. S postupující digitalizací průmyslu jsou zaváděny do výroby nové přístroje, schopné vysílat data během produkce automobilu. Příkladem transferu dat během produkce je nová statistická aplikace chy-Stat od společnosti Q-Stat ve ŠKODA AUTO a.s., která monitoruje kvalitu produktů on-line během výroby a poskytuje tak okamžitou odezvu kvalitářům společnosti. Tato data jsou poté ukládána do relačních databází pro pozdější využití na poli kvality automobilu. Před tím, než však dojde k výrobě, musí se jednotlivé aspekty vozů analyzovat a zkoumat, zda nemůže dojít k chybám během výroby, kterým lze předejít. Tato analýza se nazývá FMEA. Více o tomto tématu bude zmíněno v kapitole FMEA. Analýzu FMEA dnes již provádí každá automobilka a ukládá protokoly, vzniklé během jednání analytiků společností pro pozdější využití v dalších modelech automobilů. Toto jednání se koná u každého náběhu nového vozidla do výroby. Témata na FMEA jsou vybrána především dle předešlých zkušeností z minulých modelů a průzkumů trhu. Takových jednání a protokolů, které na nich vznikají, je potom velké množství, jelikož se nároky na kvalitu neustále zvětšují a také zcela logicky roste počet minulých modelů. Když pak mluvíme o celé historii automobilky, jedná se o tisíce protokolů FMEA. Hledání v nich, analyzování dat a aplikace výsledků analýz může být pak poněkud komplikovaná. Z toho důvodu je vyvíjen tlak na jejich převedení do přehledné databáze.

Aby databáze FMEA práci skutečně ulehčovala, je nutné, aby bylo splněno několik požadavků. Databáze musí především automatizovat formální stránku celého protokolu a jeho ukládání. Jestliže databáze zvládne automaticky provádět úkony, které jinak musí lidé opakovat stále dokola, může se tím poté soustředit na analýzu částí protokolů. Dále je nutné zajistit, aby již vypracovaný protokol do databáze nemusel pracovník zadávat ručně. Přípravený protokol musí být jednoduše importovatelný do databáze, jelikož by v opačném případě zvyšoval časové nároky na údržbu dané databáze. Dalším nárokem na databázi je fakt, že databáze musí být konzistentní a úplná. Musí obsahovat záznam o každém funkčním dílu či komponentu. Zároveň musí obsahovat vyhodnocení veškerých příčin a následků zadaných analytikem. Musí být schopna třídit informace na stejné úrovni hierarchie dílu a zobrazovat je tak i uživateli. Musí být realizována tak, aby na ní mohl

pracovat více než jeden uživatel - v nejlepším případě celý tým FMEA. Zároveň musí umožňovat analytikovi zobrazit kontrolu plnění FMEA. (Kukkal, 1993)

Vytvoření databáze FMEA má spoustu výhod. Krom již zmíněného ušetření času oproštěním analýzy od repetitivních úkolů a snadnějšího dohledávání informací, umožňuje například smysluplné třídění dat. Jelikož protokoly, kterých bude několik tisíc, budou obsahovat některé společné výrazy, je pak možné určité komponenty podle nich vyhledat. Lze tak zobrazit data relevantní pro jeden určitý funkční díl za celou historii společnosti. Dalším nesporným kladem je ta skutečnost, že již samotné vytvoření databáze nutí uživatele k používání jednotné terminologie. Dosáhne se tak daleko vyšší konzistence databáze. Mezi další pozitiva se řadí samozřejmě schopnost databáze provádět kontrolu plnění FMEA. Jestliže je zadán termín, do kterého je nutné nápravné opatření, na kterém se dohodl tým na jednání vykonat, bude později potřeba zkontrolovat status těch opatření, která ještě vykonána nebyla. Tato funkce je jednou z nejdůležitějších a značně tím pomůže analytikům, jelikož tak nemusí procházet každý protokol zvlášť a nechá si od databázového systému vystavit report o plnění. Výhodou je také vytvoření prostředí, které může prohlížet libovolný počet uživatelů. Zjednoduší se tak transfer dat mezi odděleními a pomůže tak vyhledávat mezi příbuznými tématy podobných projektů ostatních oddělení. (Kukkal, 1993)

Způsobů umístění protokolů FMEA do ucelené databáze je skutečně mnoho. V současné době, kdy je snaha o digitalizaci veškerých výstupů, je hlavní prioritou uchovat tyto digitální verze dokumentů pro pozdější využití tak, aby byly později snadno dohledatelné. Proto databáze nejen v automobilovém průmyslu čeká obrovský rozmach. Databáze FMEA je již součástí každé větší společnosti a programů či databázových systémů, které s tímto druhem dat pracují je nepřehledné množství.

Například software SCIO™ od společnosti Plato používá většina automobilek sdružených pod koncernem Volkswagen. Ten našel uplatnění zejména v oblastech plánování komponentů a výroby. Dalším programem, který našel své využití především ve vývoji v koncernu Volkswagen je IQ od společnosti APIS. Oba softwary poskytují rozšířenou analytickou podporu při vypracovávání FMEA protokolů společně se zobrazením historie,

porovnáváním analýz, sledováním indexů, grafickým zobrazením analýzy, databáze všech protokolů a uživatelsky přívětivé prostředí. Program tohoto typu se v podstatě stará o celý workflow FMEA včetně všech analýz, upozornění na končící termín plnění apod. Jediné co, je po uživateli požadováno, jsou vstupní data. Programy od těchto a jim podobných společností jsou velmi sofistikované a jednoduché na používání. Je to z toho důvodu, že se jejich výrobci zabývají pouze touto problematikou a dokáží tak vyvinout daleko dokonalejší nástroj, než společnosti, které se programovým řešením zabývají obecně. V ostatních odděleních koncernu není nutné pro tvorbu a analýzu FMEA protokolů používat drahé automatické programy, jelikož počet protokolů není tak vysoký, aby se zakoupení licence na software vyplatilo. Proto se také často k zaznamenání protokolů a jejich správě v koncernu používá Microsoft Excel. (Arbeitskreis VW, 2015)

Další způsob realizace databáze FMEA je pomocí webového rozhraní. Již v roce 1999 se experimentovalo s tímto rozhraním. Bylo vytvořeno zejména kvůli nedostupnosti programového vybavení v asijských oblastech, kvůli odlišnému kódování a zejména pak proto, že vydané softwary byly dostupné v licencích pro izolovaný počítač a nesplňovaly jednotlivé specifikace asijských závodů. Systém se skládá ze tří částí. FMEA klient, FMEA databáze a FMEA webový server. Tento princip spočívá v tom, že počítač s přístupem na internet vlastníci FMEA klienta se připojí k webovému serveru. Odtud si počítač stáhne interface. Uživatel poté do klienta zadá všechny potřebné specifikace, které se buď vyhodnotí přímo na lokální síti, nebo se posílají zpět na server k vyhodnocení. Výsledek celé procedury je pak odeslán webovým serverem na databázové úložiště FMEA. Toto úložiště využívá databázovou architekturu Microsoft SQL Server. (Huang, 1999)

Na stejném principu se objevila i inovativní řešení využívající stejný princip, který je jen obohacen o novější technologie. Například komunikace klienta se serverem je uskutečněna pomocí IIS od společnosti Microsoft. K vytvoření uživatelsky přátelského rozhraní byla použita technologie ASP.NET, díky které je k dispozici velká řada běžných tlačítek, text boxů a funkcí, na které jsou uživatelé na webových stránkách zvyklí. Pokročilá logika funkcí byla dosažena pomocí .NET tříd. Celý systém byl místo tří částí rozdělen na čtyři vrstvy: Prezentační vrstva, obsahující uživatelské rozhraní, Business Objektová vrstva, obsahující veškerou logiku programu, vrstva přístupu dat a databázová vrstva. Databáze

zde opět využívá SQL architekturu. Jednalo se však jen o modernizaci již existujícího systému. (Awad, 2012)

Na internetu najdeme i veřejně přístupné databáze, které spravuje Americké centrum pro analýzu spolehlivosti (American Reliability Analysis Center). Tato databáze obsahuje data z mnoha oborů a každá společnost se může do programu zapojit ať už jen nahlížením do zveřejněných analýz nebo věnováním vlastních. Některé společnosti se však nemohou spolehnout na veřejně dostupné databáze FMEA nebo na běžné softwarové vybavení dostupné na trhu. To může být zapříčiněno vyššími požadavky na bezpečnost dat nebo tak specifickými procesy výroby produktů, že běžně dostupné vybavení jednoduše na vypracování analýzy nestačí. Příkladem může být čínská AVIC (Aviation Industry Corporation of China), státem vlastněná společnost pro letectví a obranu, která si nechala vypracovat systém databáze FMEA na míru. Jednalo se o velice složitý a sofistikovaný systém databází, obsahující dva hlavní sektory: software databáze, rozdělená na tři subdatabáze, a hardware databáze, rozdělená na čtyři subdatabáze. CEPREI Laboratory je další společností, které byla databáze FMEA „ušita na míru“. Databáze byla v tabulkové formě a obsahovala data z testovacích typů, fenoménů, analýz zátěže, nápravných opatření apod. Mezi další příklady institucí, které musí provádět své analýzy pomocí specializovaných programových řešení, jsou například vojenské instituce. Například Čínská lidová osvobozená armáda si pro tento účel pozvala experty, aby shromáždili relevantní data pro analýzy vybavení obrněných vozidel. Výsledkem byla databáze FMEA, která hrála důležitou roli při následných analýzách spolehlivosti vozidel. (Chen, 2014)

## 2 Teorie tvorby databází

### 2.1 Co jsou data

Pojem data je odvozeninou od latinského slova Datum označující danou a neměnnou skutečnost. To, že lze data měnit tedy neznamená, že měníme skutečnost (realitu), ale spíše se jedná o zcela jinou skutečnost, kterou chceme nějakým způsobem zaznamenat. V elektronickém světě počítačů lze data nazvat textový, číselný, grafický, zvukový či úplně odlišný vjem zaznamenaný na paměťové médium a zpracovávaný elektronickou platformou (počítač, čtečka, přehrávač atp.).

Data lze rozdělit na strukturovaná a nestrukturovaná. Strukturovaná data se vyznačují tím, že informace získané z media lze rozřadit dále na datové elementy. Ty rozlišují, zda jsou data například textový řetězec, znak, reálné a celé číslo, logická pravdivostní hodnota, datum nebo čas. Strukturovaná data tak zachycují čistě fakta, atributy, objekty apod. Pomocí datových elementů se pak zachytí strukturovaná hierarchie při tvorbě databáze. Například pole – záznam – relace – databáze. Data lze díky hierarchii lépe třídít a vybírat pouze ta, která mají pro řešení určitého problému v informatice nějaký význam. Naproti tomu nestrukturovaná data se dají vyjádřit jako nepřetržitý tok dat, který je ukončen až při úplném načtení. Může se jednat například o videozáznamy či zvukové soubory. Patří sem však i textové soubory. Ačkoliv text v souboru (MS Word, Notepad.exe atp.) by se dal chápat jako strukturovaná data, je nutno mít na paměti, že data dokumentu, ve kterém je text uložen, obsahují nejen text samotný, ale i formát, v jakém je text psaný, spolu s jazykovou lokalizací (např. Středoevropské písmo) a strukturou celého dokumentu. (Sklenák, 2001)

Data tak reprezentují určitou vlastnost, kterou však bez dalšího popisu, dat či kontextu buď nelze blíže specifikovat a tudíž i pochopit, nebo jde o pojem velmi užívaný a výskyt (například v databázích) by mohl být duplicitní. Vyhledávání by tak bylo značně nepřesné. Příkladem může být například číslo 9211012705 označující identifikační číslo pojištěnce,

které samo o sobě nic určitého neznamena nebo T100, které může znamenat jak motocykl značky Triumph nebo dvouhlavňový ruský tank Kirov. S další specifikací dostávají data větší smysl a i užitek. Mějme například větu: „Průměrná cena skleničky limonády ve městě Jablonec nad Nisou je 27,- Kč. Celou větu tak můžeme rozložit do strukturované podoby dat, které usnadní jejich pozdější vyhledání. Určíme atributy produkt, průměrná cena a město. Sklenička limonády je hodnotou atributu produktu, 27,- Kč je hodnotou atributu průměrná cena a Jablonec nad Nisou je hodnotou atributu město. Při vyhledávání je pak snazší vyhledávat nejdříve mezi atributy a určit, co je pro výsledky mého hledání dat důležité, a poté lze jednoduše vytrítit nadbytečná data a zrychlit tak celý proces.

## 2.2 Co je to databáze

Již od počátku existence dat a informací se rovněž objevuje problém s ukládáním dat, která nejsou v nejbližší budoucnosti potřeba, ale jsou důležitou informací pro pozdější použití. Problém je v umístění a strukturalizaci takových dat, aby po uložení jsme je následně bezpečně našli a začali s nimi pracovat, případně upravili některé jeho části a znovu uložili na stejné místo.

Pokud si běžný uživatel představí pojem databáze, intuitivně se mu vybaví „místo“ kam se ukládají v tištěné či digitální podobě informace. Tato představa není nesprávná, ale může být poněkud zavádějící. Například (Tyrychtr, 2015) tvrdí, že se jedná o 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. Tato hypotéza je sice zdařilejší, nikoliv však dostatečná, protože nerozlišuje mezi pojmy databáze a databázový systém.

Avšak nejpřesnější definici stanovil úřad pro technickou normalizaci spolu s normou ČSN ISO/IEC 2382-17 (ČSN ISO/IEC 2382-17, 1999, s 9), kde píše že „databáze je vymezena jakou soubor souvisejících dat postačujících pro daný účel nebo pro daný



system zpracování dat“. Je tedy jasné, že pojem databáze má mnoho výkladů a každý autor si jej vykládá po svém.

Nicméně se dá spolehlivě říci, že pojem databáze znamená soubor logicky uspořádaných dat, sloužících k určitému účelu, ať už se jedná o čtení či modifikaci stávajících údajů. Každý záznam v databázi musí být unikátní, aby bylo zajištěno co nejmenší duplicity. To lze zajistit přiřazením jedinečného identifikačního čísla (klíče), které jasně definuje daný záznam v databázi, protože se jako jediný atribut v databázi nesmí opakovat. (Žid, 1998)

Pro správnou funkci databáze je však nezbytné zvolit odpovídající Systém řízení báze dat, který zajišťuje jakési rozhraní mezi uživatelem a samotnou databází. Toto je popsáno v následující kapitole.

## 2.3 Systém řízení báze dat

Systém řízení báze dat je programové vybavení určené pro práci s bází dat. Mylně se SŘBD nazývá databázovým systémem, avšak to jsou dvě odlišné věci. Dalo by se říci, že databázový systém je tvořen právě bází dat spolu se Systémem řízení báze dat. Nemůže tedy jít o dvě stejné věci, neboť jedna je nedílnou součástí druhé.

Jedná se v podstatě o rozhraní mezi daty uloženými v databázi a uživatelem, který zadává systému příkazy. Základním účelem systému je pak vytvoření struktury databáze a její následná možnost modifikace. Kromě dnes už samozřejmých funkcí systému jako je mazání, vkládání či úprava stávajících záznamů, musí umět také pokročilejší funkce. Například podporu pro jeden nebo vícero datových modelů, podporu vyššího jazyka pro manipulaci s daty a jejich definici, správu klíčů nebo kontrolu autenticity a přístupů jednotlivých uživatelů do systému. (Dařena, 2005)

Systém řízení báze dat je právě tou proměnnou, kterou může uživatel zvolit při tvorbě databázového systému. Data jsou dána, tedy databáze jako taková zůstává neměnná. Na uživateli pak záleží, jaký SŘBD vybere jako ten nejvhodnější pro danou situaci a s tím

samozřejmě i databázový model. Konkrétní příklady softwarových řešení systémů řízení báze dat jsou zmíněny v kapitole 3.1.

## 2.4 Databázový model

Databázový model je model dat, který určuje logické uspořádání databáze a říká tím, jakým způsobem jsou data v bázi uložena, organizována a manipulována. Databázový model je vybrán na základě skladby, druhu a obsahu dat. Zároveň se databáze musí přizpůsobit uživatelským požadavkům na funkce, rychlost vykonávání instrukcí, spolehlivost, udržovatelnost, škálovatelnost a cenu za vyhotovení. Každý Systém řízení báze dat používá některý z databázových modelů, ale není výjimkou, že SŘBD nabízí variant hned několik podle preferencí uživatele. Model ale není jen způsob organizace dat v databázi. Určuje také, jaké operace může uživatel v SŘBD provést. Některé funkce, které najdeme u databázových modelů, nejsou shodné s těmi z ostatních modelů. Kupříkladu relační model databáze nabízí funkci join a select, kterou v hierarchickém modelu budeme hledat jen marně. (Kominácká, 2007)

### 2.4.1 Druhy databázových modelů

S počátkem rozvoje komerčních počítačů pro širokou veřejnost si společnosti začaly velmi rychle uvědomovat potenciál, který skýtala možnost ukládání dat, která byla potřeba pro periodické výpočty na terminálech. Jeden z takových příkladů jsou měsíční výpočty výplatní pásy. Před tím, než se objevily komerčně navržené databázové modely, byla data uložena volně jako takzvaná Flat-File. Jedná se v podstatě o první databázový model, který byl dvoudimenzionální a obsahoval jedinečná data, uložená na stále stejném místě. Tato data byla často neměnná a sloužila pro sekvenční čtení. Na zápis se v té době používaly magnetické pásy. Při nutnosti zápisu nebo modifikaci stávajícího údaje musel být celý obsah databáze přepsán na druhý kotouč, zatímco na první se znovu sekvenčně zapisovala již modifikovaná data s pomocí druhého kotouče, jež obsahoval originál. Celý proces

hledání, čtení a zápisu tak vyžadoval, aby byl soubor dat přečten od začátku až po hledaný údaj. Dnes se již kotouč nepoužívá, ale metoda je pořád stejná a proto je tento model z hlediska jakékoliv manipulace nejpomalejší. Výhody tohoto modelu jsou naprostá jednoduchost a pochopitelnost jednotlivých zapsaných dat a jejich struktury. Největší výhodou je však skutečnost, že lze databázi vytvořit i do obyčejné tabulky a jednoduše řadit a filtrovat zápisy. Na tento způsob ukládání dat není potřeba speciální aplikační software ani jazyk, jako je tomu často u ostatních modelů, jelikož díky své jednoduchosti na vyhledávání, modifikaci, zápis či čtení lze použít zcela obyčejné a často používané softwarové vybavení. Navíc mizí nároky na znalosti jakéhokoliv programovacího jazyka či znalostí o databázích. Nevýhoda je nízká úroveň zabezpečení z důvodů, jež jsou zmíněny výše. Dále vysoká možnost duplicity jak celých záznamů, tak jednotlivých polí, jež u některých záznamů nemusí být žádoucí a neschopnost přidat jakýkoliv údaj bez nutnosti doplnit ten samý údaj u všech dalších záznamů, což může být problém například v případě přidání e-mailové adresy u databáze obsahující několik tisíc zaměstnanců. (Burlison, 1999)

Hierarchický model se poprvé objevil v padesátých letech devatenáctého století, kdy ho představila společnost International Business Machines dnes známá pod zkratkou IBM. Jedná se o jeden z nejstarších modelů vůbec. Hierarchická struktura databáze se skládá ze souborů vzájemných vztahů (tzv. „segmentů“), které dohromady tvoří hierarchický vztah. Ten je definován jako soubor vztahů spojených logickými asociacemi. Každý segment pak obsahuje několik vztahů zároveň a dále se větví. Segment, který je nadřazen dalším segmentům se nazývá parent-segment (master-segment). Podřízený child-segment (slave-segment) je hierarchicky pod parent-segmentem. Svým uskupením model připomíná například rodokmen, který se často interpretuje jako strom. Odtud také pochází název pro segment, „root“, který neobsahuje žádný nadřazený prvek (tedy parent-segment). Stejně tak jako poslední segment v řadě, neobsahující žádný podřazený prvek (tedy child-segment), takzvaný „leaf“. Výhodou tohoto modelu je zjevná jednoduchost vztahů mezi segmenty a tudíž jednoduchý design celé databáze. Dále pak bezpečnost a integrita dat, jednoduchost sdílení dat a efektivita ve vztahu 1:N. Mezi hlavní nevýhody patří neflexibilita spolu se složitou údržbou při změnách. Pokud se pokusíme vymazat segment, může se nám stát, že smažeme i všechny segmenty pod ním. Navíc pokud se nám povede

vše smazat bez problémů, nastanou potíže s aplikacemi, které se k databázi připojují. Je tedy nutné upravit i aplikace. Dalším problémem je složitost implementace vztahu M:N, který je v reálném životě mnohem častější než 1:M. Dalšími problémy jsou pak operační anomálie a absence jakýchkoliv standardů. (Singh, 2009)

Síťový model spatřil světlo světa v šedesátých letech devatenáctého století. Navrhla ho společnost Database Task Group. Tento model dat je v podstatě stejný jako hierarchický s tím rozdílem, že jednotlivé segmenty mohou nyní obsahovat více než jeden nadřazený segment. Jak již bylo řečeno realizace vztahu M:N byla v hierarchickém modelu dat poněkud problematická. Toto bylo síťovým modelem vyřešeno. V tomto modelu je předchozí vztah parent-segment pojmenovaný jako owner record a child-segment jako member record. Vztahy jsou v modelu realizovány pomocí pointerů. Stejně jako předchozí model se i tento dá členit na jednotlivá patra. Nyní se tak může stát, že segment obsahuje jeden vztah owner a dva či více vztahů member. Výhodou tohoto modelu je již zmiňovaná podpora vztahu M:N a jednoduchost po vzoru hierarchického modelu dat. Dalšími výhodami je například dostupná standardizace a integrita dat. Nevýhodou je již zmíněné užití pointerů v modelu. Celá databáze je tak velmi komplexní. Dalším problémem je absence strukturální nezávislosti. Za tím se skrývá skutečnost, že vymazat segment v prosířované struktuře je v tomto modelu takřka nemožné a databáze bez tohoto segmentu bude nefunkční. Zároveň jakýkoliv podobný update této struktury nebude kompatibilní s aplikacemi přistupujícími k těmto datům. (Tyrychtr, 2015)

Dalším datovým modelem je model relační. Vznikl v roce 1969, kdy ho uvedl doktor Codd. Vyznačuje se schopností uskupovat data do jednotlivých tabulek, které spolu souvisí na základě jednoho společného pole. Data tak nejsou hierarchicky uspořádána a dovolují uživateli použít velmi účinné operace, včetně algebraických výrazů. Jednotlivé vztahy segmentů dat jsou tak snadno odvoditelné, redundantní a konzistentní. Díky algebře se dají i složité vztahy reálného světa jednoduše popsat pomocí relačního modelu dat. Na rozdíl od dřívějších modelů, tento model klade důraz na vztah mezi segmenty dat, nikoliv na fyzické implementaci jako takové. To je zároveň výhodou tohoto modelu. Je díky tomu jednoduchý a vývojáři se tak mohou plně soustředit na logické vazby. Díky strukturální nezávislosti jednotlivých členů lze snadno modifikovat bez potíží s kompatibilitou

programů, které se připojují k datům databáze. Údržba i správa báze je pak také mnohem jednodušší. Díky svým vlastnostem umožňuje podporu vyšších a pokročilejších jazyků určených ke správě dat v bázi. Nevýhodou je hardwarová náročnost vykonávaných instrukcí spolu s faktem, že relační model databáze je jednoduchý na návrh i implementaci. To může samo o sobě vést k dalším problémům, jako jsou například přehlčení nekonzistentních informací na tzv. „informačních ostrůvcích“. To znamená, že se vytvoří spoustu skupin lidí s vlastními databázemi, které mezi sebou nejsou kompatibilní a nejsou schopné jakékoliv synchronizace. Může tak docházet ke ztrátě a duplicitě informací, což stojí společnost nemalé náklady. Dalším takovým problémem je situace, kdy se jednoduchá databáze navrhne nesprávným způsobem. Při malém obsahu dat v bázi se nemusí jednat o velký problém, ale v opačném případě se nároky na vykonání instrukcí radikálně zvětšují, dochází ke snižování výkonu systému jako celku a může dojít i k poškození informací. (Singh, 2009)

## 3 Programové vybavení pro tvorbu databází

### 3.1 Příklady softwarových řešení systémů řízení báze dat

Mezi nejznámější zástupce systémů řízení báze dat patří např. MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase a IBM DB2. (Pokorný, 2013)

MySQL je relační systém řízení báze dat, vyvinutý společností MySQL AB, Michaelem Wildeniusem, Davidem Axmarkem a Allanem Larssonem v roce 1995. Po bouřlivém rozvoji se tento systém stal jedním z nejpoužívanějších a říjnu roku 2015 vyšla již verze 5.7, která je dostupná široké veřejnosti. Výhodami systému je rychlost, spolehlivost a nenáročnost. Navíc je tento systém pro domácí použití zdarma a je dostupný pro různé platformy i různé aplikační softwary pracující v jazycích C, C++ a PHP. (DuBois, 2003) Nevýhodou je ztráta výkonu při náročnějších aplikacích.

PostgreSQL je takéž relační systém řízení dat, který spatřil světlo světa roku 1989, kdy byl spuštěn prototyp stavějící na Ingres projektu Univerzity v Kalifornii, týmem Michaela Stonebrakera. Výhodou tohoto systému je tvorba vlastních datových typů a procedur spolu s bezpečností, podporou SQL standardu a pokročilých technologií typu CTE a XML. Celkově se Postgre pokládá se lepší verzi MySQL, která však, a to je její hlavní nevýhodou, postrádá podporu webhostingů, jako je tomu u MySQL. (Momjian, 2003)

Microsoft SQL server je opět relační systém řízení báze dat vyvinutý společností Microsoft v roce 1989. Výhodou je naprosto profesionální systém, který zanechává nejbližší konkurenci (např. MySQL) daleko za sebou počtem funkcí, přehledností a velmi spolehlivou obnovou dat například při nechtěné odstávce hardwaru, či špatně navržené databázi, výsledkem čehož byla poškozena data. Díky protokolu souborů zálohování a výpisům z mezipaměti se mohou data spolehlivě obnovit. Nevýhodou je svázanost s platformou Microsoft Windows a vysoká cena pořízení systému. Zatímco Microsoft SQL server se pohybuje v řádech tisíců až desetitisíců dolarů (SQL Server, 2016), konkurenční MySQL je zcela zdarma, nebo vyjde komerčně velmi levně. (Stanek 2013)

Oracle Database je objektově orientovaný relační systém řízení báze dat vyvinutý společností Oracle Corporation v roce 1978. Tým Larryho Ellisona (Bob Miner a Ed Oates) založil roku 1977 společnost Software Development Laboratories, kterou financovala americká CIA pod krycím jménem Oracle. Odtud název společnosti, který si Software Development Laboratories po několika letech vzala za své. Výhodou Oracle Database je především její výkon díky pokročilým algoritmům při zadání instrukcí. Dále například absence dalších datových typů krom jednoho pro zajištění minimálního obsazení prostoru, podpora SQL standardů a programovacího jazyka a podpora objektového přístupu databáze. (Lacko, 2002) Nevýhodou je značně drahá licence, která způsobuje, že si většina malých i středních podniků licenci ani správce nemůže dovolit. (Oracle Price Lists, 2016)

Sybase byl vyvinut společností Systemware (později Sybase) roku 1986 v licenci s Microsoftem. Nepřekvapí snad, že se jedná opět o relační systém řízení báze dat. V nedávných letech byla společnost sloučena s německým SAPem. Provoz systému však stále trvá. (Taylor, 2006) Výhodou je vyšší výkon oproti konkurentům jako jsou MySQL a PostgreSQL. Nevýhodou je menší množství nástrojů ke správě databáze. (Verschoor, 1992)

IBM DB2 je skupina relačních systémů řízení báze dat, které v posledních letech začaly podporovat i objektově relační databázový přístup. Počátky této skupiny systémů sahají do roku 1974, kdy byla vyvinuta první verze systému, avšak až v roce 1983 byla nazvána DB2, při příležitosti vydání platformy, která systém již obsahovala. Výhody DB2 je například stabilita systému, podpora ze strany IBM a kompatibilita s různými platformami. Nevýhodou je jeho náročná správa a cena licence, která je opět nedostupná menším podnikům. (Neagu, 2012)

Zmíněné příklady jsou vhodné pro práci s rozsáhlejšími databázemi. V následující kapitole 3.2 je uveden softwarový balíček Microsoft Office, jehož nedílnou součástí je i Microsoft Excel, jež se hodí spíše pro menší databáze, bez nutnosti zabezpečit souběžný přístup více uživatelům zároveň.

## 3.2 Microsoft Office

Balíček kancelářských aplikací Microsoft Office je dnes již naprostým standardem při práci na počítači. Obsahuje spoustu užitečných programů a funkcí, které jsou nutností pro každého, kdo pracuje například s textem, výpočty, databázemi a prezentacemi. Jedním ze společných znaků všech programů obsažených v kancelářském balíku je intuitivní ovládání pomocí karet, které pomáhá i naprostým začátečníkům rychle si osvojit základy používání nástrojů. Dalším výrazným znakem balíku je jednotlivá provázanost programů Microsoft. Je například možné importovat tabulku do sešitu v Excelu z úplně jiného sešitu a vytvořit tak permanentní link k této tabulce. Změní-li se něco ve zdrojovém souboru, výsledky se reflektují i v propojeném sešitě. Tohoto spojení je realizováno technologií OLE, která existuje v balíčku již s vydáním verze 2003. Díky této technologii se při užití správného postupu může i z jednoduché Excelovské tabulky stát výkonná databáze, obsahující aktuální informace ze souborů umístěných buď na disku, nebo v síti. Mezi největší klady balíčku Office je však podpora Visual Basic for Applications, díky kterým lze psát makra, která nejen ulehčují práci, ale umožňují naprogramovat jinak únavnou nebo složitou práci tak, aby se dělala sama. Mezi hlavní zástupce kancelářského balíčku patří bezesporu MS Word, MS Excel, MS PowerPoint a MS Access. Mezi další méně známé aplikace balíčku patří například MS OneNote, MS Outlook, MS Visio nebo MS Publisher. Programový balíček MS Office je nejvíce používaným kancelářským softwarem na světě spolu se 1,2 miliardami uživatelů, vlastnících alespoň základní balíček MS Office (Microsoft by the numbers, 2016). V této práci si uvedeme alespoň dva základní programy tohoto balíčku.

### 3.2.1 MS Word

Program MS Word je textovým procesorem, jehož počátky sahají až do šedesátých let devatenáctého století, kdy fungoval jako samostatný počítač. Ten uměl pouze tisknout a psát a to jen v tučném formátování a kurzívou. Spolu s rozmachem osobních počítačů a tiskáren se Word jako textový procesor rapidně vyvíjel. Postupně obsahoval stále více



funkcí jako například kontrolu pravopisu, rozšířené možnosti formátování, správu a možnosti přidání netextových elementů nebo správu různých stylů formátování textu. Program je založen na grafickém uživatelském rozhraní what-you-see-is-what-you-get (WYSIWYG). To znamená, že se procesor snaží při tisku stránek více či méně přiblížit tomu co uživatel během tisknutí vidí na své obrazovce monitoru. Dnes je MS Word nejvíce užívaným textovým procesorem na světě.

### **3.2.2 MS Excel**

Program MS Excel je tabulkovým procesorem, jehož první verze pro Macintosh vyšla v roce 1985. Byl prvním tabulkovým procesorem, jež podporoval ovládání rozbalovacích menu pomocí ovládání myši. Zároveň podporoval na svou dobu enormní množství fontů. Již od svého počátku se od svých konkurentů rychle vzdaloval a své definitivní vítězství potvrdil v roce 1993 podporou VBA (Visual Basic for Applications), čímž umožnil široké využití tvorby maker. Dnes je MS Excel standardem v oblasti kancelářských nástrojů. Umožňuje ukládání tabulkových záznamů, správu a práci s uloženými daty, provádění výpočtu pomocí vestavěných funkcí, používání filtrů, kontingenčních tabulek a mnoho dalšího.

### **3.2.3 VBA**

VBA je zkratkou pro Visual Basic for Applications, jež vyvinula společnost Microsoft. Programovací jazyk je schopen přímo komunikovat s Windows API a zajišťovat propojení mezi aplikacemi vyvinutými Microsoftem. Pokud program má svou objektovou knihovnu, lze s ním ve VBA pracovat. Tento programovací jazyk také sdružil a rozšířil některé programovací jazyky, specifické pro jednotlivé programy. Například WordBasic, který dříve fungoval jako programovací jazyk v textovém procesoru Word, je nyní plně obsažen ve Visual Basic for Applications a je jím obohacen o nové funkce. Název také napovídá, že VBA je velice úzce spjata s Visual Basic, jež sdílí se svým příbuzným stejnou běhovou

knihovnu Visual Basic Runtime Library. Program napsaný ve VBA však potřebuje ke svému běhu hostující aplikaci, ve které byl napsán, a nelze ho tedy spustit jako samostatný program. Je tedy nutné nejdříve otevřít například sešit Excelu a v tomto prostředí svůj program spustit.

### 3.3 Porovnání softwarových řešení SŘBD

Tabulka 1 shrnuje výhody a nevýhody SŘBD probraných v kapitolách 3.1 a 3.2.

Tab. 1: Výhody a nevýhody SŘBD

SŘBD	Výhody	Nevýhody
MySQL	Rychlost, spolehlivost, nenáročnost, nízká cena, meziplatformní	Ztráta výkonu v náročnějších aplikacích
PostgreSQL	Počet funkcí a nástrojů, spolehlivá obnova dat, přehlednost	Svázanost s platformou MS Windows.
Oracle database	Výkon, datová nenáročnost	Cena, cena údržby
Sysbase	Výkon	Počet funkcí
IBM DB2	Stabilita systému, meziplatformní	Cena, cena údržby
MS Excel	Cena, cena údržby, VBA, nízké nároky na prostor, přehlednost, vysoké množství uživatelů	Omezený počet předprogramovaných nástrojů, omezená správa databáze, pouze jeden uživatel

Zdroj: Vlastní tvorba.

Jak bude dále zdůvodněno v kapitole 5.1, z možností v Tabulce 1 byl pro řešení úkolu, kterým se zabývá tato práce, vybrán MS Excel.

## **4 Analýza současného stavu FMEA dokumentace ve ŠKODA AUTO a.s.**

Dokumentací FMEA se ve ŠKODA AUTO a.s. zabývá oddělení GQZ, jež má na starosti metody kvality a statistické metody. FMEA se skládá z jednotlivých protokolů, které vznikly během jednání FMEA a byla ukládána dle stromové struktury. Zapisování každého protokolu do stromové struktury, kontrolu jeho správnosti a kontrolu plnění již zajistil Ing. Roman Čejka v rámci své bakalářské práce. (Čejka, 2013) Dokumenty jsou uspořádané dle vozů, projektů a typů FMEA. Dalším krokem tedy bylo navázat na tento projekt a vytvořit jednotnou databázi, která by se dokázala sama aktualizovat a umožnila by jednoduchý přehled všech doposud vytvořených protokolů.

### **4.1 Kvalita**

Co je to kvalita? Jak uvádí norma (ČSN EN ISO 9000, 2006, s 19) „jakost, jejímž synonymem v českém jazyce je slovo kvalita, je definována, jako je stupeň splnění požadavků souborem inherentních znaků“.

Kvalita je velice abstraktním pojmem. Pro každého tento termín znamená něco jiného. Každý zákazník má určitý počet požadavků, které jsou pro něj definicí kvality. Tato množina požadavků se mění, například kvůli subjektivním požadavkům, kultuře, geografické poloze a podobně. Kvalita je pak průnikem těchto množin požadavků zákazníků. Kvality se tak dosáhne přiblížením vlastností výrobku průniku této množiny a tím k průměrné představě kvalitního produktu. Cíli dosažení maximální kvality se lze přiblížit, ale nikoliv ho splnit, jelikož se neustále pohybuje s rozvojem nových technologií, trendů a neustále se měnícím požadavkům zákazníků.

## 4.2 FMEA

Zkratka FMEA stojí za termínem Analýza možnosti vzniku vad a jejich následků (Failure Mode and Effects Analysis, dále jen FMEA). Tato analýza je součástí metod popsaných v příručce VDA. VDA (z němčiny Verband der Automobilindustrie) je svaz automobilového průmyslu, který vydává standardy a doporučení ve formě příruček VDA. Zejména se pak zaměřuje na zajištění kvality a prevenci vad ve výrobě, avšak nikoliv výhradně. Jedná se o analýzu, která se provádí před vstupem produktu do sériové výroby. Jako důležitý metodický nástroj slouží k včasnému rozpoznání potencionálních vad tak, aby jim bylo možné před vstupem do výroby zamezit. Analyzuje tak produkt na všech kritických místech a navrhuje vhodná opatření, která mohou rizika výskytu chyb snížit či zcela eliminovat. Uplatní se tak zejména při návrhu nových produktů či upgradu těch stávajících. Založením dokumentace FMEA do databáze získá společnost výraznou a efektivní podporu při probíhajících i budoucích projektech. (ČSJVDA 4, 2006)

Počátky FMEA zasahují až do roku 1949, kdy byla vyvinuta jako vojenský předpis. Roku 1963 vyvinula NASA analýzu FMEA pro projekt vesmírného letu Apollo. O dva roky později byla tato metoda přejata pro účely letecké techniky a kosmonautiky. Kolem roku 1975 se začala uplatňovat i v jaderném průmyslu. Prvním milníkem v automobilovém průmyslu bylo nasazení metody FMEA společností FORD (USA) v roce 1977 z důvodu prevence vad a zajištění kvality. Roku 1980 byla metoda zmíněna v podtitulu německé normy DIN 25448. Poprvé byla v příručce VDA 4 FMEA popsána o šest let později a navždy se tak stala součástí souboru metod zajišťujících kvalitu v automobilovém průmyslu. Metoda našla uplatnění i v jiných oborech jako jsou lékařské a sdělovací techniky či facility management. Metoda FMEA klade důraz na zvyšování kvality v zájmu zvýšených požadavků zákazníků a státní i mezistátní legislativy. Spolu s tím jsou čím dál více zvyšovány nároky na úsporu v oblasti produktů a procesů, což dlouhodobě vede k optimalizaci nákladů. (ČSJVDA 4, 2006)

Příčiny konání jednání FMEA jsou změny podmínek, nové návrhy či změny dílů, technologií nebo celých systémů. Tu pak řídí a moderuje moderátor a zároveň se stará o zápis celého jednání do protokolu FMEA. FMEA dokumentace je v této práci chápána

jako množina všech FMEA protokolů. Moderátorem může být jak interní tak i externí pracovník. Odpovědný pracovník zpočátku uvede týmu FMEA návrh nového produktu nebo jeho změny a hlediska, ze kterých bude návrh posuzován.

Poté tým začne se stanovováním možných vad a rizik v pěti následujících krocích. Prvním je analýza struktury, při níž se zachycují související systémové prvky, a probíhá stanovení systémové struktury. Dalším krokem je analýza funkcí, při které se tým FMEA snaží přiřadit funkce systémovým prvkům a propojuje je. Následně proběhne analýza chybných funkcí, kde tým přiřazuje funkcím chybné funkce a ty chybné opět propojuje. V dalším kroku, jímž je Analýza opatření, probíhá dokumentace aktuálních opatření k eliminaci a odhalování vad spolu s vyhodnocením aktuálního stavu. V posledním kroku, zvaném optimalizace, snižuje tým FMEA riziko dalšími opatřeními a vyhodnocuje změněný stav. Vyhodnocení probíhá pomocí úrovně priority rizika, neboli RPZ a vypočítá se součinem tří proměnných A, B, E, určených při analýze potenciální vady produktu. Bodování probíhá v rozmezí od jedné do deseti. Proměnná A charakterizuje pravděpodobnost vzniku závady, přičemž deset označuje nejvyšší pravděpodobnost, proměnná B zobrazuje závažnost vady, přičemž deset je nejvyšší stupeň závažnosti a proměnná E znázorňuje pravděpodobnost odhalení vady, přičemž deset je nejnižší pravděpodobnost odhalení. Teoreticky tak RPZ může dosáhnout hodnoty až jednoho tisíce nebo jedničky. Na základě tohoto hodnocení tým FMEA rozhoduje, zda se tímto problémem dále zabývat. Jestliže se na základě RPZ hodnocení rozhodne, že je potřeba učinit příslušná opatření, předloží se jeho návrh k dalšímu projednání. Nejdříve se provede hodnocení nákladů a realizace. Poté se předvede znázornění předpokladů k rozhodnutí a vyhodnocují se vhodně alternativy. Jestliže FMEA tým dojde k závěru, že daná varianta opatření je vhodná, přejde se k plánování realizace a zpracování podkladů ke zpracování. Výsledkem je tedy realizace opatření spolu s dokumentací o rozhodnutí a rezervování potřebných zdrojů. V opačném případě se opatření zamítne a může dojít k žádosti o nové zpracování. Při průběhu realizace se průběžně hodnotí a měří stav realizace, a jestli přinesla očekávaný úspěch. Jestliže je úspěch potvrzen, opatření se uzavírá. Pokud však nespĺňuje dané požadavky, dochází novému startu opatření nebo k vyhodnocení dalšího opatření. Po ukončení fáze realizace nebo při překročení termínu realizace příp. rámce nákladů je potřeba vypracovat závěrečnou/dílčí zprávu FMEA. Poté dochází k předávání získaných informací při řešení

problému metodou FMEA dalším pracovníkům v organizaci, případně dodavatelům. (ČSJVDA 4, 2006)

### **4.3 Dosavadní způsob uložení FMEA dokumentace**

Makra Ing. Romana Čejky obsažená v protokolech zajistila, že protokol se po splnění všech podmínek a termínu přesunul do složky dle hodnoty v buňce „Projekt“ a dále do složky dle hodnoty v buňce „Typ FMEA“. Výsledná cesta tak vypadala například takto: (...)\vozy\SK001\procesní\2.06 Vnitřní plech.xls. V těchto protokolech se tak v rámci hlavičky objevovaly často duplicitní údaje, jelikož všechny protokoly musely mít stejnou vypovídací hodnotu, ať už byly ve složce s ostatními dokumenty nebo byly rozesílány jako kopie odpovědným plnitelům. Zároveň byl každý dokument doplněn stejnými makry pro třídění do složek a kontrolu plnění opatření. (Čejka, 2013)

### **4.4 Tvorba FMEA dokumentace ve ŠKODA AUTO**

Jak již bylo zmíněno v kapitole 4.2, dokumentace FMEA vzniká při uvedení nového produktu či jeho změně. S tímto procesem je spjato založení nové složky na sdíleném disku a pojmenováním projektu. Dle témat se přidělí odpovědné osoby nebo oddělení ve společnosti. V případě, že odpovědnost za produkt či komponent nese externí firma, vyzve oddělení GQZ firmu, aby vyslala zástupce s odbornými znalostmi a kompetencemi k řešení daného tématu. Poté se na určitý termín svolá jednání FMEA, jež je pro všechny osoby závazné.

Na začátku jednání se protokol nadepíše. To zahrnuje především vyplnění hlavičky. Hlavička obsahuje název téma, datum konání, moderátor, typ FMEA, předmět, zodpovědná oblast a stav FMEA. Dále se zapíše Tým FMEA, jež zahrnuje pouze účastníky FMEA jednání. Moderátor FMEA je kvalifikovaný pracovník z oddělení GQZ, který zabezpečuje metodiku kvality a zároveň vede celé jednání. Svými zásahy do jednání zajišťuje plynulý chod jednání a objektivitu diskuze. Moderátor musí být vybaven

laptopem a prázdným elektronickým protokolem FMEA.

Během jednání se doplňuje protokol dle toho, jak tým FMEA narazí na možné problémy ať už při funkci produktu/komponentu nebo jeho konstrukci. Jakmile tým na takový problém narazí, vyplní se funkce problémové části, možné příčiny a důsledky vady, doporučené opatření a charakteristika RPZ. Identifikace možných vad je možná díky dlouholeté zkušenosti a vysoké profesionalitě všech zúčastněných.

Po ukončení jednání se všem zodpovědným osobám stanoví termíny, do kterých je nutné závadu odstranit nebo zamezit jejímu vzniku. Jestliže se týmu FMEA podaří tomuto vzniku závad zamezit, je celá záležitost uzavřena. V opačném případě se na dalším jednání stanoví nová opatření, což znamená vznik nového protokolu FMEA.

Název FMEA		Moderátor	Datum konání FMEA	Projekt	FMEA-Typ	Číslo protokolu											
Name der FMEA		Moderator	Datum der Umsetzung	Projekt	Procesní	Protokollnummer											
Multifunkční Tiskárna		Loupolský DEA	13.6.2016	XM01/3	Procesní	02.01											
Předmět FMEA		Zodpovědná oblast	FMEA-Stav	Datum kontroly protokolu													
Gegenstand der FMEA		Verantwortlicher Bereich	FMEA-Status	Datum der Protokoll-Kontrolle													
Drucker AC514		Petr Dvořák A1C	Průběžná	29.7.2016													
FMEA Tým / FMEA Team																	
Petr Dvořák																	
Loupolský Vlastimil																	
Petra Malátná																	
Funkce	Možná vada	Možný důsledek	S	Příčina	Kont. prev opatření	Vznik	Význam	Možné riziko	Doporučená opatření	Řešitel	Termín	Provedená opatření	Vznik	Význam	Možné riziko	Stav	
Skrýt / Zobrazit hlavičku						A	B	E									
Funktion	Potentielle Fehler	Potentielle Fehlerfolge	Ursache	Kontrollmaßnahmen				RPZ odzn	Empfohlene Abstellmaßnahmen	zu erledigen durch	Termin odznač	Getroffene Maßnahmen	A	B	E	RPZ	Stand
OP 20 Tisk	Netiskne	Nespokojený uživatel	Chybí papír	Kontrolka chybějících o papíru, periodické doplňování papíru	7	7	3	147	školení ekonomického tisku	Petr Dvořák	30.6.2016	Provedeno školení dne 25.6.2016, dále vypracován návod na ekonomický tisk	3	7	3	63	
			nemá datové připojení	kontrolka datového připojení, mechanické zajištění dat. Kabelu	4	7	2	56	Optimalizovat umístění přípojky proti vykopnutí	Petra Malátná	30.6.2016	Kabel zafixován dne 27.6.2016	3	7	2	42	
									výměna problematické síťové karty tiskárny	Loupolský Vlastimil	20.6.2016	Karta vyměněna dne 14.6.2016	3	7	2	42	
			papír je zaseklý	Pogumovaná kolečka podavače, Obsluha proškolena	7	7	1	49	Výměna pohybového ústrojí podavače	Loupolský Vlastimil	30.7.2016	Byla provedena výměna pohybového ústrojí podavače dne 29.7.2016	4	7	1	28	
								252	4	4		0			400		

Obr. 1: Vyplněný FMEA protokol

Zdroj: Interní dokument ŠKODA AUTO a.s.

## 4.5 RPZ

RPZ, neboli Míra možného rizika je ukazatel (z německého Risikoprioritätzahl), který má upozorňovat na závažnost dané vady. Jedná se o součin tří parametrů vady. Parametry se číslují od 1 do 10. Tato čísla mají každé hodnotící tabulku s přesně určeným popisem. Prvním je pravděpodobnost vzniku a určuje pravděpodobnost, za které daná vada na voze/komponentu vznikne, kde 1 je pro nejmenší pravděpodobnost a 10 pro nejvyšší. Druhým je význam a zobrazuje závažnost vady a její důsledky na funkci vozu/komponentu, kde 1 je nejmenší závažnost a 10 je nejvyšší. Třetím je odhalitelnost vady, kde na rozdíl od předchozích parametrů je nejvyšší odhalitelnost ohodnocena stupněm 1 a v případě, že lze vadu odhalit jen těžko, boduje se stupněm 10. Tento ukazatel je nejdůležitějším parametrem pro moderátora FMEA, jelikož může určovat, v jakém pořadí budou jednotlivé předměty porady FMEA řešeny a na jeho základě se pak stanovují opatření. Tedy zda-li stačí mírné opatření, nebo je nutné se tomuto problému více věnovat. Výsledná databáze, která díky autorovi vznikne, bude pak schopna filtrovat RPZ dle jejich závažnosti a tím i navrhnout nejčastější a nejzávažnější vady právě díky tomuto parametru.

Pro lehčí představu lze uvést příklad. Na klávesnici je nefungující klávesa Q. Pravděpodobnost vzniku vady, tedy špatné zacházení s klávesnicí, které má za následek disfunkci dané klávesy je relativně nízké. Klávesnice se dnes vyrábějí z odolných materiálů, aby se těmto dříve častým vadám předešlo, a při výrobě se předpokládá vysoký fyzický stres tohoto komponentu. Pravděpodobnost vzniku závady bychom ohodnotili tedy stupněm 1. Druhý parametr je závažnost vady. Klávesa Q v českém jazyce není moc hojně využívána a v operačních systémech vždy existuje možnost jak si klávesnici nebo daný znak vyvolat jiným způsobem (například na obrazovku). To však s sebou nese určité zdržení a frustraci uživatele, jelikož musí udělat pro jediné písmeno spoustu úkonů navíc. Ohodnotili bychom tento parametr stupněm 5. Třetím je odhalitelnost vady. Klávesa Q je, jak již bylo řečeno velmi málo používaným písmenem v českém jazyce. Je tedy malá pravděpodobnost, že na nefunkční písmeno narazíme v běžném psaní. Tento parametr bychom ohodnotili stupněm 9, tedy velmi špatně odhalitelný. Výsledná charakteristika RPZ je tedy  $1 \cdot 5 \cdot 9 = 45$ . Vzhledem k povaze charakteristiky a možným mezím, kde se výsledná hodnota může pohybovat v rozmezí 1 až 1000, lze říci, že 45 je velmi nízká



hodnota, tedy nízká míra možného rizika. Tato charakteristika nám však jen říká jak závažná vada to je. Ve ŠKODA AUTO a.s. je nutné řešit všechny potenciální závady pro zajištění maximální spokojenosti zákazníků.

## **4.6 Zhodnocení současného stavu dokumentace**

Současný stav dokumentace FMEA zajišťuje úplnost a dostupnost veškerých dat, která kdy vznikla za účelem předcházení vad. Dokumenty jsou rozříděné dle projektů a typu FMEA a lze dohledat jednání FMEA podle daného modelu. Není však možné filtrovat výsledky hledání dle jedné vady, která se projevuje u každého modelu. FMEA tým se skládá z vysoce kvalifikovaných a zkušených expertů v daném poli. Je proto možné, aby si tým zpětně díky své zkušenosti promítnul výsledky předchozích jednání do toho současného. Není to však sázka na jistotu. Protokoly FMEA jsou ukládány v programovém prostředí Microsoft Excel. Excel neukládá jen data v sešitech obsažená, nýbrž i další informace o dokumentu, jako například formáty, výšky a šířky buněk, metadata, VBA každého dokumentu zvlášť atd. V našem případě, kde je protokolů na sdíleném disku i několik tisíc, může tato archivace hrát podstatný problém například z hlediska kapacity nebo přehlednosti. Navíc, jak je zmíněno v kapitole 4.3, sešity obsahují často duplicitní údaje, které by bylo možné eliminovat vhodnou volbou databáze. V případě, že systém archivace dokumentů zůstane v dosavadním stavu, bude hrát velkou roli také přehlednost. S přibývajícím počtem projektů totiž začne růst počet složek v hlavním adresáři.

## 5 Tvorba FMEA databáze pomocí Microsoft Office

Řešením situace popsané v kapitole 4.6 by bylo vytvoření jednotné databáze obsahující všechna jednání FMEA a kategorizace všech vad a opatření dle modelů. Bylo by pak například možné filtrovat záznamy podle určité vady. Tímto způsobem by tým získal informace o dalších modelech, které v historii měly identickou vadu. V případě potřeby by také bylo možné filtrovat záznamy podle všech charakteristik uvedených v protokolu. Každý záznam v databázi by byl jedinečný, jelikož by se během importu zapsalo jedinečné identifikační číslo databázového záznamu. Databázový záznam tato práce chápe jako jeden řádek excelovského sešitu protokolu, viz Obr. 1 a 2, obsahující relevantní informace o jednání FMEA. Hlavička protokolu je pro všechny jeho záznamy stejná a proto se opisuje do každého záznamu, aby bylo možné jednotlivé záznamy bezpečně identifikovat. V případě, že se v záznamu vyskytne sloučený řádek, je informace relevantní k oběma záznamům a proto se opisuje do každého zvlášť.

Zpočátku probíhalo programování pomalu, jelikož bylo žádoucí, aby v databázovém sešitě nebyly žádné pomocné buňky, které by si pamatovaly hodnoty. Vše se při spuštění muselo znovu počítat a toto ošetření znamenalo spoustu kompromisů během programování. Makro vznikalo ve verzích podle toho, čeho již bylo schopno, aby se zamezilo záměně. Zároveň bylo nutné zaznamenat postup vypracování, k čemuž verze makra posloužily výborně. Jelikož je možné, že makro bude někdo v budoucnu upravovat, bylo nutné popisovat jednotlivé operace pomocí komentářů.

Celé testování aplikace probíhalo na lokálním disku, aby se zamezilo ztrátě nebo poškození skutečných dat. Pracovalo se tedy se zkušebními množstvím protokolů, což zároveň urychlovalo proces výpisu dat z protokolů během testování. Při spuštění aplikace „naostro“ dosahovala databáze stejných výsledků jako ve zkušebním prostředí.

## 5.1 Výběr databáze

Po důkladné analýze všech nabízených možností programového vybavení k vytvoření a správě databázi a po zhodnocení možností v prostředí oddělení kvality ve ŠKODA AUTO a.s. autor této bakalářské práce vybral program Microsoft Excel z programového balíčku Microsoft Office a to hned z několika důvodů. Vzhledem k faktu, že většina protokolů je po projektu Ing. Romana Čejky v podobě excelovských sešitů, bude tento přechod pro zaměstnance pracující v tomto prostředí již nějakou dobu daleko snazší. Do 7. 1. 2005 byly protokoly FMEA vypracovávány v aplikaci MS Word bez stálého formátu. Jedná se o protokoly, které se během projektu pana Čejky sjednotit nepodařilo. Počet těchto protokolů je přibližně 40. Tyto protokoly se do FMEA databáze popisované v následujících kapitolách zařazovat nebudou. Moderátor může dle svého uvážení do těchto starých FMEA protokolů nahlédnout přímo v souborovém systému. V současnosti již protokoly FMEA vznikají výhradně v aplikaci MS Excel, díky jejím nesporným kladům v práci s protokoly oproti aplikaci MS Word. Aplikace MS Word pro tvorbu tabulek není vhodná a přehledná a pracovníci vnímali přechod z textového procesoru na tabulkový pozitivně. Odpadlo složité formátování a vkládání tabulek na stránky symbolizující list papíru. V MS Excel jsou tabulky velmi jasně strukturované a je možné použít velké množství funkcí pro extrakci požadovaných dat z tabulek, jako je například výpočet ukazatele RPZ, zmíněný v kap. 4.5.

Jestliže by byl vybrán například MySQL, bylo by nutné důkladně proškolení veškerý personál, který by s programem přišel do styku. Navíc by bylo nezbytné zakoupit průmyslové licence pro používání programu, zprostředkujícího přístup k MySQL, ke komerčním účelům. Výběrem programu MS Excel tak dojde k ušetření času a nákladů na školení a na zakoupení licencí k odpovídajícímu programovému vybavení. Licence k Microsoft Office patří ke standardnímu vybavení kancelářských počítačů a laptopů ve společnosti. Z finanční i časové náročnosti se tak jedná o nejlepší volbu. Prostředí Microsoft Excel je velmi intuitivní, a proto bude i transfer správy databáze na jiného pracovníka velmi snadný a odborně nenáročný, jelikož v přijímacích požadavcích na pracovníka je schopnost práce v Microsoft Office.

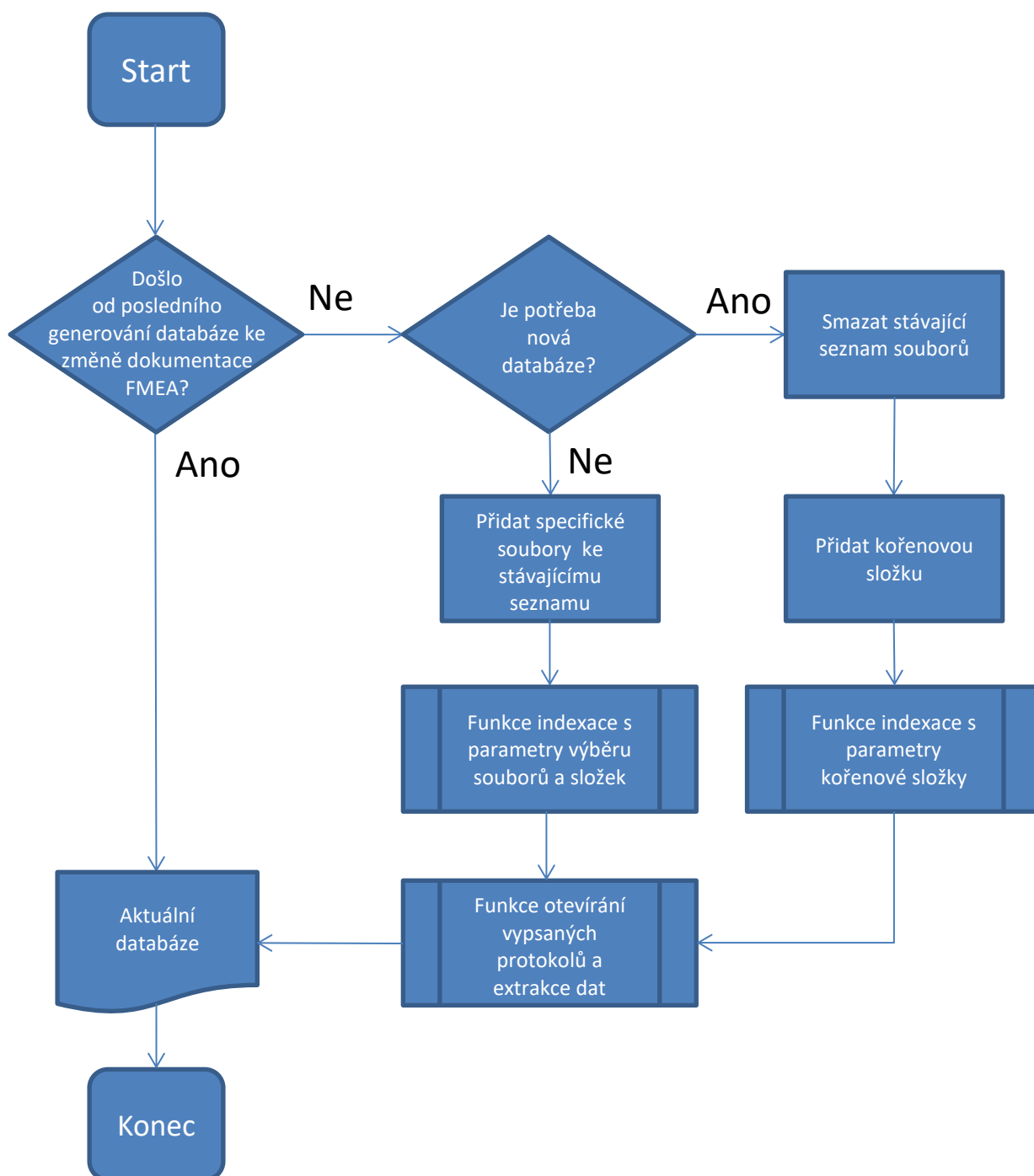
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
Chl. SET	Téma FMEA	Projekt	Jméno FMEA	datum konání FMEA	Kdo změnil FMEA	Funkce	Měrná vada	Měrný důsledek	PŘčina	Kont. opatření	A	B	C	D	E	F
#	Thema der FMEA	Projekt	Jméno FMEA	Datum	Wer hat die FMEA geändert	Funktion	Wahrscheinliche Ursache	Mögliche Folgerung	Ursache	Kontrol-Maßnahme	A	B	C	D	E	F
1	Multi-funkční tiskárna	XHE03	pracovní	13.6.2016	Leopoldský DEA	Op 20 Tisk	Netiskání	Nespojený uživatel	Chybí papír	Kontrola chybných papírů, periodické doplňování papíru	7	7	3	147		
2	Multi-funkční tiskárna	XHE03	pracovní	13.6.2016	Leopoldský DEA	Op 20 Tisk	Netiskání	Nespojený uživatel	není datové připojení	Kontrola datového připojení a mechanické zajištění datového kabelu	4	7	2	56		
3	Multi-funkční tiskárna	XHE03	pracovní	13.6.2016	Leopoldský DEA	Op 20 Tisk	Netiskání	Nespojený uživatel	není datové připojení	Kontrola datového připojení a mechanické zajištění datového kabelu	4	7	2	56		
4	Multi-funkční tiskárna	XHE03	pracovní	13.6.2016	Leopoldský DEA	Op 20 Tisk	Netiskání	Nespojený uživatel	papír je zaseklý	Popumná kolečka podavače. Osluha proběhla	7	7	1	49		

Obr. 2: Databáze FMEA vytvořená z dat protokolu z Obr. 1

Zdroj: Vlastní tvorba

## 5.2 Řešení úloh pomocí VBA v Microsoft Excel

Microsoft Office podporuje programovací jazyk VBA, díky kterému bylo možné naprogramovat import veškerých protokolů dle zadaných parametrů. Mezi parametry patřila například unikátnost každého zápisu, prohledání všech složek a podsložek v umístění a extrakce dat z každého protokolu, sjednocený formát všech protokolů, možnost aktualizace dle jediného souboru a dle souborů ve složce, oddělení protokolů ve výčtu všech dat v databázi, skrytí hlavičky ve výčtu protokolů, zvýšená rychlost extrakce a možnost otevření protokolu dle adresy, ze které byl vypsán. Postup, jímž se řídí rozhodování uživatele při tvorbě databáze, je popsán na Obr. 3.



Obr. 3: Vývojový diagram tvorby databáze v naprogramovaném prostředí Microsoft Excel

Zdroj: Vlastní tvorba

### 5.2.1 Unikátnost zápisu

Každý záznam v databázi musí být unikátní. Identifikační číslo bude pro zápis permanentní. Identifikační číslo se generuje pomocí jednoduché funkce, která při každém novém zápisu databázového záznamu přiřadí číslo dle pořadí již vygenerovaných záznamů. Každý další zápis dat bude s ohledem na již zapsaná data a při této operaci se neobjeví duplicitní identifikační číslo. Bude ošetřena situace, kde v souborovém systému budou 2 soubory se stejným jménem a zapíše se tak duplicitní údaje z obou protokolů.

Unikátnost zápisu byla docílena kontrolou duplikátů ve výčtu vyhledaných souborů v adresářové struktuře popsané v kapitole 4.3. Funkce zajišťující indexaci relevantních souborů protokolů kontroluje jejich názvy. Jestliže funkce narazí na opakující se název, z indexace ho vynechá. Každý soubor vlastní unikátní jméno je unikátní i svým obsahem. Projekt kolegy Čejky právě tuto skutečnost zajistil. Jak již bylo v úvodu kapitoly 5 popsáno, bylo zároveň nutné zajistit číslování jednotlivých záznamů. Záznamy jsou číslovány od jedné do posledního možného řádku v Microsoft Excel. Při generaci nové databáze protokolů se tak smaže celá databáze a začíná se odznova. V případě přidání dodatečného zápisu si program přečte poslední identifikační číslo záznamu a naváže na něj.

### 5.2.2 Indexace obsahu a extrakce dat

Program musí být schopen projít celou stromovou strukturu včetně podsložek a vypsát cestu k souborům obsahujícím protokoly. Poté je nutné všechny soubory otevřít a vypsát z nich data do databáze aniž by při otevření přepsal datum poslední úpravy, zabudovaném pomocí makra v protokolech.

To aby byl program schopen prohledat složky včetně veškerých podsložek byl poněkud problém. Jelikož zpočátku program ani uživatel přesně neví, jak daleko stromová struktura souborového systému sahá, není možné dopředu naprogramovat vložené funkce pro prohledávání. Pro tuto úlohu je nutné použít rekurzivní funkci, která sama sebe spouští jakmile narazí na další „větev“ souborového systému. Celkem se tento vyhledávací

program skládá ze tří modulů. Prvním je hlavní modul „index“, který odstartuje další dva rekurzivní. V tomto modulu se ověří, zda se jedná o přidání dalších souborů nebo tvorbu kompletně nové databáze. Poté se definují obě funkce realizující vypisování souborů. V poslední fázi se nejdříve spustí druhý modul „GetAllFiles“, který vypíše všechny soubory ve složce. Zde také lze nastavit podmínky výpisu souborů. Jakmile byly vypsané všechny soubory ve složce, spustí se třetí modul „GetAllFolders“, který hledá všechny složky v té současné a poté znovu spustí druhý modul. Tento postup se může zdát poněkud komplikovaný, ale jedná se v podstatě o velice jednoduchý stále se opakující postup. Po výběru složky, ve které chce uživatel soubory hledat, včetně podsložek, se jedná už jen o následující: „Najdi soubory a poté složky ve stávající složce a totéž udělej i ve všech dalších nalezených složkách.“ Na každý vyhledaný soubor ve funkci „GetAllFiles“ byla navázána funkce na výpis umístění vyhledaného protokolu do tabulky na druhém listu databáze. Po vysání veškerého obsahu adresáře byla pak zavolána funkce pro extrakci všech vypsaných FMEA protokolů. Jelikož předchozí funkce vypisovala přesné umístění v diskové jednotce, je nyní možné pomocí příkazu `Workbooks.Open` otevřít FMEA protokol ze zadané adresy. Tento příkaz se umístí do cyklu a výsledkem je, že program otevírá veškeré vypsané protokoly a extrahuje z nich informace do databáze. Tato procedura zpočátku vyžadovala pevnou strukturu protokolů. Implementací funkce `range.find`, která je součástí VBA, bylo dosaženo mírné tolerance strukturálních odlišností jednotlivých protokolů. Program si tedy vyhledá potřebný nadpis buňky. Je však nutné, aby v buňce nějaká hodnota byla pro správné vypsaní do databáze. Pro uzavření protokolů bez změny data poslední úpravy bylo použito příkazu `Workbooks.Close` s parametrem `„SaveChanges:=False“`.

### **5.2.3 Urychlení procedury**

Bylo třeba dosáhnout toho, aby program extrahoval data z protokolů za přijatelnou dobu a nároky na výpočetní výkon byly co nejmenší. Eliminace výpočetně náročných operací a jejich nahrazení efektivnějšími substituty byly hlavní prioritou.

Celá procedura tvorby databáze znamenala nejdříve spuštění funkce, zodpovědné

za indexaci dat, a potom spuštění extrakce z protokolů. Takových protokolů bylo například ve zkušební verzi již 1500. Celý proces tak časově odpovídal zhruba půl hodině. Redukce času potřebného na provedení procedury bylo docíleno pomocí několika jednoduchých příkazů. Prvním je například „ScreenUpdating = false“. Tento příkaz zajišťuje, že při každé provedené změně se aktualizuje obrazovka tak, aby uživatel viděl, co se v sešitě mění. Deaktivací a znovuaktivací na konci procedury došlo k výraznému zvýšení výkonu programu. Toto zvýšení však nebylo dostatečné a způsobilo problém s objektem zobrazujícím průběh operace – tento problém však bude probrán později. Dalším výrazným zvýšením výkonu bylo dosaženo vypnutím rekalkulace po každé změně příkazem „Calculation = xlCalculationManual“. Stejně jako obnovování obrazovky pomocí „ScreenUpdating“ i tato vlastnost objektu Application provádí svou činnost při každé sebemenší změně v sešitě. Jedná se o rekalkulaci všech buněk, vzorců, vztahů a propojení. Jelikož při naší proceduře pouze zapisujeme a nevyužíváme žádné vestavěné funkce v sešitě je možné dočasně „Calculation“ vypnout a zapnou znovu až na konci procedury. Díky vypnutí této funkce došlo neočekávaně k obrovskému nárůstu výkonu programu. Časová náročnost na provedení se tak díky těmto krokům snížila na jednu třetinu, tedy zhruba 10 minut. Byla také provedena substituce neefektivních operací. Nahraditelné „řádkování“ (procházení řádků dokud nebude podmínka splněna) bylo vyměněno funkcí „Range.Find“. V některých příkladech bylo zajištěno „vyskočení z cyklu“ za splnění určitých podmínek. Byly přidány podmínky pro vstup do cyklů pro eliminaci „prázdných“ operací. Žádná z těchto úprav však neměla tak dramatický dopad, jako funkce „ScreenUpdating“ a „Calculation“.

#### **5.2.4 Sjedení formátu**

Při importu dat do databáze bude ošetřena možnost, že některé tabulky s protokoly se mohou nepatrně lišit z důvodu většího počtu moderujících pracovníků v teamu FMEA jednání.

Díky projektu mého kolegy Ing. Čejky (Čejka, 2013) jsou nyní protokoly vypracovávány dle jednotného vzoru. Avšak není možné zajistit, aby ani jeden protokol neobsahoval



nějaký ten „řádek navíc“ nebo aby všechny nadpisy a termíny byly stejně nadepsány. Každý moderátor píše protokol „po svém“ a dělá tak v přesvědčení, že právě jeho zápis je ten správný. Program se tak musí přizpůsobit všem těmto abnormalitám. Pro tento účel je nejlepší funkce „Range.Find“. Jestliže se z nějakého důvodu posunula celá tabulka o řádek níže nebo moderátoři používají některá slova jiná, lze pomocí funkce „Range.Find“ hledat množinu výrazů odpovídající slovníku moderátorů. Dle vyhledané buňky se poté hledá hodnota pod nadepsaným termínem.

### **5.2.5 Jednotlivé přidávání souborů a složek**

Databáze bude doplněna o možnost přidání jednotlivých protokolů či složek s protokoly k dosavadní datové základně databáze, aniž by došlo k přepisu databáze jako celku.

Další výzvou bylo upravit kód programu tak, aby umožňoval přidání souboru nebo složky do stávajícího datového obsahu databáze. Toto bylo vyřešeno úpravou programu tak, aby si jednotlivé moduly předávaly parametr dle stisknutého tlačítka v sešitě. Jestliže uživatel stiskl tlačítko pro přidání souboru či složky, část kódu, která zajišťovala vyčištění sešitu od stávajících záznamů, se vynechala. Započala procedura pro vyhledání souborů v určené složce. Oproti normálnímu průběhu se v tomto případě nalezené soubory a cesta k nim zapisovala do druhého listu o pár sloupců vedle původních. Odtud se poté četly cesty pro proceduru otevírání protokolů a extrakce dat. Pro vybrání objektu složky a souborů jak v klasickém případě tak dodatečném přidávání protokolů do databáze bylo využito příkazu „FileDialog“. Konkrétně tedy „msoFileDialogFolderPicker“ pro výběr složky a „msoFileDialogFilePicker“ pro výběr souborů.

### **5.2.6 Link z databáze na protokoly**

Databáze bude umožňovat samostatně otevřít importované protokoly přímo z excelového sešitu databáze.

Tato funkce byla zajištěna prostou implementací již zmíněné funkce „Workbooks.Open“, kde cesta k souboru je vypisována již při extrakci dat. Stačí tedy jen označit řádek se záznamem daného protokolu a po stisknutí tlačítka otevřít se protokol otevře z adresy, ze které byl vypsán, pokud je samozřejmě pořad dostupný.

### **5.2.7 Oddělení protokolů**

Jednotlivé protokoly budou ve výčtu všech protokolů znatelně odděleny pro lepší přehlednost v databázi.

Po extrakci vznikla databáze obsahující obrovské množství záznamů. Aby se mohl zaměstnanec lépe orientovat v záznamech, byla do programu vložena funkce pro oddělení jednotlivých protokolů v rámci zobrazení výčtu všech extrahovaných protokolů. Jedná se o funkci, která kontroluje změnu názvu protokolu a v případě, že se název změní, podtrhne poslední záznam černou linkou, kterou lze snadno odlišit od implicitní šedé. Pro další zpřehlednění bylo potřeba vytvořit tlačítko pro skrývání hlavičky protokolů. Tato funkce je zajištěna pomocí příkazu „EntireColumn.Hidden“, který skryje nebo odkryje specifické sloupce dle potřeby uživatele. V našem případě jde právě o hlavičku protokolů.

## **5.3 Řešení problémů**

Během vypracovávání databáze se objevilo hned několik problémů, které musely být vyřešeny. Některým chybám bylo možné předejít a jiným se zase vyhnout. Následující výčet problémů zobrazuje právě takové případy.

### 5.3.1 Workbooks.Open

Prvním problémem byla skutečnost, že když běží makro v hostitelském sešitě, do kterého se bude vypisovat databáze ze FMEA protokolů, a to samé makro otevře jiný sešit, který obsahuje trigger makra pro spuštění „Workbook.Open“, došlo ke konfliktu těchto maker zejména, jestliže v obou makrech nebyly přesně definované příkazy pro hostitelský a otevíraný sešit. Makra v otevíraných protokolech zajišťovala funkčnost a pomocné funkce protokolů, jak je uvedeno v práci Ing. Čejky. (Čejka, 2013) Otevřený protokol naštěstí nic neměnil v databázi. Ztěžoval však práci programátora tím, že odstartoval celou proceduru bez možnosti „krokování“. Jindy například vyskočil z makra úplně bez jediné chybové hlášky. Jedná se o známou chybu ve Visual Basic for Applications, která byla v dalších verzích již opravena. Jelikož je ale veškerý update softwaru ve společnosti ŠKODA AUTO a.s. řízen IT oddělením, bylo nutné předejít možným chybám. Tuto chybu lze obejít přenesením pozornosti zpět na hostitelský sešit. To znamená umístit do cyklu příkaz týkající se hostitelského sešitu, obsahující definovanou proměnnou, ve které je hostitelský sešit uložen. „Krokování“ stále možné není klasickým způsobem, ale chod programu probíhá bez problému. Krokování lze vyvolat umístěním dvou „breakpointů“ za příkaz „Workbooks.Open“ a spuštění procedury. Jakmile se chod programu zastaví na „breakpointu“ je možné dále krokovat do dalšího otevření protokolu.

### 5.3.2 Ukazatel průběhu

Při testování aplikace často docházelo k úvahám, zda program běží a importuje data nebo se někde zacyklil. Dalším problémem byl stav, kdy program extrahoval do nekonečna a uživatel neměl tušení, jestli někdy skončí. Tyto úvahy byly naprosto oprávněné, jelikož celý proces dříve trval, jak již bylo zmíněno v kapitole 5.2.3, i půl hodiny na laptopu s SSD diskem. Bylo potřeba zobrazit průběh celé procedury spolu s předpokládaným časem dokončení, zbývajících položkami a těmi dokončenými. Zároveň by měl ukazatel obsahovat právě extrahující řádek a protokol. Celý ukazatel průběhu byl naprogramován do „Userform“, kde se využilo vestavěné funkce „Progressbar“. Pro maximální počet položek se vypočítal počet obsazených řádků po výpisu indexace souborů v souborovém

systemu před extrakcí dat. Pro číslo souboru bylo použito jednoduchého počítadla. Název souboru a obsah právě importovaného řádku byl odvozen z objektu otevřeného souboru. Ukazatel průběhu obsahoval celkem dva „Progressbary“ ten první ukazoval průběh extrakce z aktuálně otevřeného protokolu a druhý zobrazoval celkový průběh procedury. Zbývající čas byl však nezlomný úkol, který zabral největší dobu. Zbývající čas měla mít na starost funkce, která neustále počítá průměrný čas všech již extrahovaných protokolů a dle počtu těch zbývajících dopočítá předpokládaný konec procedury. Tato funkce se velmi složitě programuje a bylo nutné si ji nejdříve graficky znázornit. Jakmile byla funkce hotova, objevil se další problém. Vyvolaný ukazatel zabraňoval dalšímu průběhu procedury. Tato skutečnost byla ošetřena přepnutím na nemodální „Userform“. Už v tento moment však bylo jasné, že jakékoliv změny se na ukazateli průběhu přestanou vykreslovat, jakmile se procedura spustí. Po průzkumu možností bylo zvoleno manuální překreslení ukazatele pomocí funkce „repaint“. Po urychlení celé extrakce, jak je vysvětleno výše, se ukazatel průběhu občas během procedury „ztratil“. Tuto skutečnost se však již opravit nepodařilo. Jedná se o další chybu Microsoftu, kterou již nijak obejít nelze. Jediným řešením je odznačit a znovu označit sešit databáze.

### **5.3.3 Opisování prázdných řádků**

Během procedury se občas v protokolu objeví prázdné nebo skryté řádky, a jelikož je program nastaven tak, aby opisoval v případě prázdného pole to, co je nad ním, začaly se objevovat duplicitní údaje s jiným identifikačním číslem. Toto bylo vyřešeno domluvou s Ing. Mikulášem Koukolským, že řádek se opíše pouze v případě, že bude protokol obsahovat na prvním místě prázdnou hodnotu a na dalších místech hodnoty odlišné od vyššího řádku, zapíše se tento zápis unikátně. V opačném případě se řádek vynechá pomocí příkazu „Goto“.

### 5.3.4 Výpis neprotokolových souborů

Během výpisu souborů v souborovém systému se začaly vypisovat i materiály, které souvisely s protokoly, ale v databázi neměly co dělat. Po otevření pomocí „Workbooks.Open“ se pochopitelně program zastavil a nahlásil chybu. Bylo potřeba tyto soubory vyloučit z extrakce. Během výpisu je tak nastavena podmínka zápisu souborů do seznamu sloužícímu k pozdější extrakci. Podmínka se týkala koncovky souboru. Vzal se tedy konec názvu souboru a porovnal se s koncovkami excelových sešitů. Některé výsledky zahrnovaly například i přehledy protokolů. Ty byly vyloučeny pomocí funkce „InStr“, která hledá pozici výrazu v proměnné. Jestliže funkce nahlásí nulovou hodnotu, výraz v proměnné není. Mezi hledanými výrazy tedy byly například „Přehled“ a jim podobné ekvivalenty a ty byly spolu s neprotokolovými soubory vyloučeny ze zápisu, jelikož díky práci Ing. Čejky (Čejka, 2013) mají protokolové soubory uniformní název lišící se pouze projektem a datem konání. Stejným způsobem byl řešen i přístup do složek, které neobsahovaly protokoly. Tato podmínka byla přepsána i do hledání složek v souborovém systému.

### 5.3.5 Změna metadat

Po instalaci balíčku aktualizací jak operačního systému, tak nové verze MS Office ve společnosti ŠKODA AUTO a.s. se změnil způsob, jakým VBA vyhledávalo v metadatech souborů. V dřívější verzi „Shortname“ obsahoval název souboru jen s koncovkou. Po aktualizaci byl ten samý název pod „Name“. Pro vyřešení tohoto problému bylo nutné přepsat všechny podmínky týkající se metadat protokolů.

### 5.3.6 Velikost databáze

Soubor, kde je databáze tvořena, se postupem času zvětšoval navzdory konstantnímu množství zapsaných protokolů. Tento fenomén je způsoben neustálou úpravou kódu

ve Visual Basic for Applications, protože tento soubor obsahuje i výkonný kód VBA, zodpovědný za extrakci záznamů z protokolů. Každá změna znamenala malý přírůstek dat, jelikož Microsoft Excel si ukládá soubor ve vrstvách. Každá změna znamená přidání nové vrstvy. Průběh procedury i otevření samotné databáze a práce v ní začnou být postupně čím dál pomalejší. Řešením je export všech modulů ve VBA obsahujících kód a import do jiného sešitu. Moduly se uloží jako celek a při importu do nového sešitu se vrstvy „stmelí“ v jeden kompaktní celek. Databáze obsahující vypsání protokolů má tak z například 16 MB, jež obsahovaly změny VBA kódu za půl roku, pouhých 2100 KB, přičemž databáze je identická s původní.

## **6 Zhodnocení výsledků tvorby FMEA databáze ve ŠKODA AUTO a.s.**

Vytvoření databáze FMEA v prostředí Microsoft Office usnadnilo přechod pracovníků na tuto formu dokumentace FMEA. V případě, že by databáze byla v SQL nebo v jiném systému řízení báze dat, znamenalo by to, že se musí veškerý personál, který s databází přijde do styku, proškolit a naučit se pohybovat v jednom z uživatelských rozhraní programového vybavení pro práci s SQL databází. Volba databáze ve formě excelovské tabulky je dle názoru autora této bakalářské práce pro tuto úlohu nejlepší. Uživatelské prostředí Microsoft Excel totiž dovoluje s takovouto databází lépe pracovat. Umožňuje vyhledávat určité termíny, filtrovat výsledky podle zadaných parametrů a řadit je dle libosti. Časová a finanční náročnost budování databáze v SQL, s ohledem na fakt, že Microsoft Excel umí nahradit většinu SQL příkazů, spolehlivě vyřadila tohoto slibného kandidáta ze seznamu. Ostatní systémy řízení báze dat nebyly brány v potaz buď z důvodu drahé licence programových prostředí zajišťujících práci s těmito systémy, nedostupnosti kvalifikovaných programátorů či personálu s kompetencí správy takové databáze nebo finanční náročnosti takového pracovníka zaměstnávat.

Každý FMEA protokol je nyní nahrazen několika řádky v excelovské tabulce, veliké 2 MB a ušetřil se tak prostor, jež zaplňovaly všechny protokoly v souborovém systému. Je jednodušší tento 2MB soubor vzít s sebou na jednání FMEA než přesouvat všechny protokoly, které dosahují celkové velikosti až 1 GB. Z tabulky lze čerpat jako ze zdroje všech předchozích jednání a reflektovat tak jejich výsledky do právě vytvářeného FMEA protokolu. Databáze umožňuje vyhledávání podle všech parametrů, které v protokolu jsou. Například tak lze filtrovat protokoly FMEA podle moderátora, který jednání FMEA vedl, poté může uživatel odfiltrovat všechny výsledky starší minulého roku a získá tak své protokoly z jednání za poslední dobu. Lze také filtrovat neuzavřené protokoly, co se týče neúspěšných nápravných opatření. Možností, jak konfigurovat databázi tak, aby ukázala to, co žádáme, je skutečně mnoho. Jelikož databáze jen ve zkušební verzi obsahovala okolo deseti tisíc záznamů, je z čeho vybírat.

Tato databáze však zatím není náhradou za původní způsob ukládání protokolů. Je spíše doplňkovým zdrojem informací. Jedná se v podstatě o studnici znalostí všech proběhlých jednání FMEA, ze které lze čerpat inspiraci na témata FMEA, nejčastější druhy vad a nejčastější druhy opatření. V případě, že dost podobný komponent byl již během jednání FMEA posouzen, databáze výrazně urychlí návrh témat, zjišťování vad apod. Původní dokumenty sjednocené díky projektu kolegy Čejky tak zůstanou uloženy na síťovém disku pro pozdější využití dle uvážení moderátora.



## Závěr

Cílem této bakalářské práce je návrh vhodné databáze pro ukládání protokolů z jednání FMEA, jejich úpravu a aktualizaci. Databáze shromažďuje veškeré záznamy, vedené během jednání FMEA a umožňuje tak další využití v budoucích jednáních FMEA.

Díky tomuto projektu je práce s protokoly kompaktnější a jednodušší. Oproti původnímu konceptu se velikost dat veškerých vytvořených protokolů snížila z 1 GB dat velkého množství souborů na 2 MB představující databázi FMEA v jediném souboru. Jedná se tedy o snížení velikosti o 99,8 %.

Stěžejním úspěchem však je zjednodušení vyhledávání v protokolech. V případě potřeby lze tento malý soubor databáze rychle otevřít a vyhledat např. jednu vadu u všech vozů, které kdy byly vyrobeny. Hledaný výraz se tak vepíše do filtru a během okamžiku má uživatel k dispozici relevantní údaje. Oproti původnímu konceptu tak došlo ke značnému ušetření času, jelikož by se v tomto případě musely všechny protokoly po jednom otevírat a hledat v nich podobný výraz. Během testování časové náročnosti před v této práci prezentovaným řešením bylo zjištěno, že vyhledání jednoho termínu v jednom protokolu i s otevřením souboru zabere 15 sekund. Počet protokolů se aktuálně pohybuje kolem jednoho a půl tisíce. Časová náročnost prohledání všech protokolů je tak 22 500 sekund, což je přes 6 hodin, přičemž není započten čas nutný k vypisování výsledků hledání. Ve verzi používané pracovníky bude protokolů přibližně dvakrát více a počet se bude neustále zvyšovat s narůstajícím počtem nových modelů, faceliftů a nových palet dílů. Aktuální databáze sestavená z výše zmíněného množství 1500 protokolů obsahuje okolo 10 000 záznamů, jak je uvedeno v kapitole 6. Po realizaci projektu databáze se během testování zjistilo, že vyhledání výrazu (vady) v celé databázi FMEA protokolů, tedy zadání vady do filtru a zjištění výsledků hledání, trvá 12 sekund. Došlo tedy ke zlepšení časové náročnosti o 99,9 %.

Při přesouvání adresáře s FMEA protokoly ze sdíleného síťového disku na disk lokální, byl v původním konceptu naměřen čas 340 sekund. Po implementaci projektu databáze byla

naměřena jedna sekunda pro přesun databáze FMEA namísto přesunu jejích zdrojových FMEA protokolů. Bylo tedy zaznamenáno zvýšení rychlosti přesunu o 99,1 %. Toto zrychlení se uplatní v případě, že se jednání FMEA neodehrává v závodě ŠKODA AUTO a.s., ale například v závodě u dodavatele nebo v místě, kde není možnost připojit se k síti Škoda. V tomto případě lze databázi jednoduše přesunout na paměťové medium a přenést ji na místo jednání FMEA

Díky vytvoření databáze je nyní možné objektivně posoudit klesající či rostoucí tendence vad v průběhu vývoje automobilů společnosti ŠKODA AUTO a.s. dle modelů, časového období nebo podobných komponentů použitých během výroby. Je nyní možné daleko lépe předpokládat výskyt některých vad u budoucích modelů a díky účinné prevenci tak zajistit lepší kvalitu vozů značky ŠKODA. Při přípravě jednání FMEA je nyní jednodušší pro pracovníky připravit si hlavní body jednání extrahované z výsledků hledání vad v databázi. Je nyní například možné okamžitě vyhledat vady související s náběhem nového faceliftu díky zapsaným vadám z faceliftu minulého.

Do budoucna by bylo vhodné propojit makra kolegy Čejky se sešitem s makry sestavujícími FMEA databázi, kterou popisuje tato práce, a zajistit, aby se databáze aktualizovala uložením protokolu do souborového systému. Tedy při uložení protokolu do určitého umístění v síti by se aktivovala makra kolegy Čejky a spustila aktualizaci databáze FMEA. Tato databáze by tak každým dnem rostla a tvořila by studnici znalostí všech možných vad na veškerých modelech vozů, které se kdy ve ŠKODA AUTO a.s. vyrobily.

# Seznam literatury

## Citace

ARBEITSKREIS VW. 2015. *Ansprechpartner FMEA Konzern Europa*. Wolfsburg. (Interní materiál)

AWAD, Amar Ali a Sha'ri Mohd YUSOF. 2012. A Methodology for Integrating Web Based FMEA and TRIZ. *International Journal of Systematic Innovation*. 2(1), 33-44. ISSN 2077-7973.

BURLESON, Donald K. 1999. *Inside the database object model*. Boca Raton, FL.: CRC Press, xiii, 226 p. ISBN 0849318076.

CHEN, Ying, Cui YE a Guoqi LI. Failure Mode Databases and Their Knowledge-Based Management. In: *2014 International Conference on Reliability, Maintainability and Safety (ICRMS)*. Beijing: Beihang University, 2014, s. 732-736. ISBN 978-1-4799-6632-2.

ČEJKA, Roman. 2013. *Standardizace FMEA Dokumentace ve ŠKODA AUTO a.s. s využitím Microsoft Visual Basic for Applications*. Liberec. Bakalářská práce. Technická Univerzita v Liberci. Vedoucí práce Ing. Dana Nejedlová, Ph.D.

ČSJ, *VDA 4: Zajišťování kvality před sériovou výrobou. Kapitola: FMEA produktu, FMEA procesu*. 2006. 2. přepracované a aktualizované vydání. Praha: ČSJ. ISBN 80-02-01682-3.

ČSN EN ISO 9000: *Systémy managementu kvality - Základní principy a slovník*. 2006. Praha: Český normalizační institut.

ČSN ISO/IEC 2382-17: *Informační technologie - Slovník*. 1999. Praha: Český normalizační institut.

DAŘENA, František. 2005. *Myslíme v jazyku PERL*. 1. vyd. Praha: Grada, 700 s. Myslíme v--. ISBN 8024711478.

DORSEY, Jim. 2013. Big Data in the Driver's Seat of Connected Car Technological Advances: New IHS Automotive study forecasts more than 150 million connected cars on global highways by 2020. IHS [online]. [cit. 2016-02-03]. Dostupné z: <http://press.ihs.com/press-release/country-industry-forecasting/big-data-drivers-seat-connected-car-technological-advance>

DUBOIS, Paul. 2003. *MySQL profesionálně: komplexní průvodce použitím, programováním a správou MySQL*. Vyd. 1. Praha: Mobil media, 1071 s. ISBN 808659341x.

HUANG, George, M. NIE a Kai Ling MAK . 1999. Web-based failure mode and effect analysis (FMEA). *Computers & Industrial Engineering*. **37**(1-2), 177-180. ISSN 03608352.

KOMINÁCKÁ, Jitka. 2007. *Prostorově orientované systémy pro podporu manažerského rozhodování*. 1. vyd. Praha: C.H. Beck, xv, 141 s. C.H. Beck pro praxi. ISBN 9788071794639.

KUKKAL, Puneet a Ronald D. BONNEL. Database Design for Failure Modes and Effects Analysis. In: *Annual Reliability and Maintainability Symposium 1993 Proceedings*. Columbia: University of South Carolina, 1993, s. 231-239. ISBN 0-7803-0943-X.

LACKO, Luboslav. 2002. *Oracle: správa, programování a použití databázového systému : platné pro verze Oracle9i, 8i a 8*. Vyd. 1. Praha: Computer Press, xiv, 464 s. Rychle a jistě. ISBN 8072266993.

LUCKOW, Andre, Ken KENNEDY, Fabian MANHARDT, Emil DJEREKAROV, Bennie

VORSTER a Amy APON. Automotive Big Data: Applications, Workloads and Infrastructures. In: *2015 IEEE International Conference on Big Data (Big Data)*. Clemson: Clemson University, 2015, s.1201-1210. ISBN 978-1-4799-9926-2

Microsoft by the numbers. 2016. Microsoft News [online]. [cit. 2016-12-02]. Dostupné z: [www.news.microsoft.com/bythenumbers/](http://www.news.microsoft.com/bythenumbers/)

MOMJIAN, Bruce. 2003. *PostgreSQL: praktický průvodce*. Brno: Computer Press, xxii, 402 s. ISBN 8072269542.

NEAGU, Adrian a Robert PELLETIER. 2012. *IBM DB2 9.7 advanced administration cookbook: over 100 recipes focused on advanced administration tasks to build and configure powerful databases with IBM DB2 : [quick answers to common problems]*. 1. vydání. Olton Birmingham [England]: Packt Pub., v, 464 p. ISBN 978-1-84968-332-6.

Oracle Price Lists. 2016. Oracle [online]. [cit. 2016-12-02]. Dostupné z: <http://www.oracle.com/us/corporate/pricing/price-lists/index.html>

PERNET, Vojtěch, Ing., bývalý zaměstnanec ŠKODA AUTO, Budova AC tř. Václava Klementa 869, 293 60 Mladá Boleslav, ústně podaná informace z 8. 2. 2016.

POKORNÝ, Jaroslav a Michal VALENTA. 2013. *Databázové systémy*. 1. vyd. Praha: České vysoké učení technické v Praze, 265 s. ISBN 9788001052129.

SINGH, Pushpendra Kumar. 2009. *Database Management System Concepts*. New Delhi - 2: V.K.(India) Enterprises. ISBN 978-93-80006-33-8.

SKLENÁK, Vilém. 2001. *Data, informace, znalosti a Internet*. Vyd. 1. Praha: C.H. Beck, xvii, 507 s. C.H. Beck pro praxi. ISBN 8071794090.

SQL Server. 2016. Microsoft [online]. [cit. 2016-12-02]. Dostupné z: <http://www.microsoft.com/cs-cz/server-cloud/products/sql-server/default.aspx>

STANEK, William R. 2013. *Microsoft SQL Server 2012: kapesní rádce administrátora*. 1. vyd. Brno: Computer Press, 544 s. Microsoft (Computer Press). ISBN 9788025137970.

TYRYCHTR, Jan. 2015. *Provozní a analytické databáze: Teoretické základy*. Praha: ČSVIZ. ISBN 978-80-87968-02-4.

VERSCHOOR, Rob. 1992. What is Sybase ASE ? In: Sypron [online]. Haag: Sypron B. V., 2012-01-29 [cit. 2016-01-08]. Dostupné z: [http://www.sypron.nl/whatis\\_ase.html](http://www.sypron.nl/whatis_ase.html)

Výroční zpráva. 2015. *ŠKODA AUTO* [online]. Mladá Boleslav: ŠKODA AUTO [cit. 2016-05-05]. Dostupné z: <http://cs.skoda-auto.com/company/investors/pages/annual-reports.aspx>

ŽID, Norbert. 1998. *Orientace ve světě informatiky*. Vyd. 1. Praha: Management Press, 391 s. ISBN 8085943581.

## **Bibliografie**

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. 2009. *Podniková informatika*. 2. přeprac. a aktual. vyd. Praha: Grada Publishing. ISBN 978-80-247-2615-1.

DVOŘÁK, Drahošlav. 2008. *Řízení projektů: Nejlepší praktiky s ukázkami v Microsoft Office*. Brno: Computer Press. ISBN 978-80-251-1885-6.