

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační Technologie



## **Optimalizace systému SKS**

## **Optimalization of SKS system**

### **Diplomová práce**

Autor:	<b>Bc. Roman Gabriel</b>
Vedoucí práce:	RNDr. Klára Císařová, Ph.D.
Konzultant:	Mgr. Vladimír Malina

**V Liberci 03. 01. 2012**

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

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

Datum : 03. 01. 2012

Podpis: .....

Bc. Roman Gabriel

## **Poděkování**

Děkuji vedoucí diplomové práce paní RNDr. Kláře Císařové, Ph.D. za odborné vedení diplomové práce a cenné rady při konzultacích.

Dále nesmím opomenout poděkovat panu Mgr. Vladimíru Malinovi za konzultace, panu Mgr. Ondřeji Tomšů, Dis. a slečně Bc. Radce Mrázkové při identifikaci problémových míst, paní Ing. Ivě Prokopové, panu Mgr. Zbyňku Prokopovi a také ostatním zaměstnancům oddělení péče o zákazníky ve společnosti ŠKODA AUTO a.s., kteří mi umožnili zpracovat tuto diplomovou práci a poskytli mi nejen podklady, ale i vhodné prostředí pro její zhotovení.

## Abstrakt

Práce řeší problémová místa systému SKS, který je používán v oddělení péče o zákazníka ve společnosti ŠKODA AUTO a.s. Uživatelé se s těmito místy setkávají několikrát během dne a hodnotí práci se systémem jako velmi nepohodlnou. Diplomová práce vznikla z potřeby identifikace kritizovaných míst v systému. Jejím cílem je navrhnout zlepšení, poté schválené změny naprogramovat a integrovat do systému SKS. SKS je modul systému SAP, a proto byla realizace změn možná až po zvládnutí základních manipulací se SAP a jazyka ABAP. Kromě toho SKS eviduje data, která se v současnosti využívají pouze klasickými databázovými metodami. Protože množství dat v čase rychle narůstá, je možné využít metody dataminingu k predikci a řízení. V této práci byla provedena analýza dat a návrh na jejich zpracování pro dataminingové úlohy. Podle reálných dat byla vytvořena pomocným programem datová množina stejné struktury, ale změněna tak, aby data mohla být použita pro vytvoření vzorových školních úloh pro připravovaný předmět o dataminingu. Zadavateli diplomové práce bylo navrženo zpracování dat SKS pro predikci.

**Klíčová slova:** SAP, informační systém, SKS, ŠKODA AUTO a.s., NetWeaver, ABAP, JAVA, optimalizace, záznam, databáze, modul, business, datamining.

## **Abstract**

This work solves problem places of SKS system, which is using on customer care department in SKODA AUTO a.s. company. Users meet with these places several times during of the day and evaluate the work with the system as not very user-friendly. This diploma thesis created to critical problem places find, suggest better version and then implement authorized changes to SKS system. SKS system is a module of SAP and therefore realization was able after mastering a programming language ABAP and basic control of SAP. Besides this SKS system has a lot of interesting data, which the company using them via classic database methods for prediction and control. In this work was done an analysis of data and a proposal for processing for datamining's exercises. A dataset with the same structure was created by secondary application by real data. But the dataset was changed for use it in examples for school's exercises for prepared subject about datamining. The sponsor of diploma thesis was proposed prediction for data processing from SKS system.

**Keywords:** SAP, information system, SKS, SKODA AUTO, NetWeaver, ABAP, JAVA, optimalization, record, database, module, business, datamining.

# Obsah

Prohlášení .....	3
Poděkování.....	4
Abstrakt .....	5
Abstract.....	6
Seznam obrázků .....	10
Seznam tabulek .....	12
Seznam zkratk a termínů.....	13
Úvod .....	15
1 Charakteristika IS SAP ve společnosti ŠKODA AUTO a.s. ....	17
1.1 Historie společnosti ŠKODA AUTO a.s. ....	17
1.1.1 Základní údaje o společnosti ŠKODA AUTO a.s. ....	18
1.1.2 Organizační struktura společnosti ŠKODA AUTO a.s. ....	18
1.1.3 Oddělení péče o zákazníka .....	19
1.2 O společnosti SAP .....	20
1.2.1 Základní údaje o SAP.....	21
1.2.2 Důležitá data z historie společnosti SAP jako unikátního leadera na poli vývoje informačních systémů.....	22
1.3 Architektura IS SAP ve společnosti ŠKODA AUTO a.s. ....	23
1.3.1 Systém SKS pod komponentou CRM.....	25
2 Analýza systému SKS a pracovních postupů na oddělení péče o zákazníka (OPZ).....	26
2.1 Jak fungovalo a dnes funguje POZ.....	26
2.2 Problémová místa systému SKS neboli uživatelské nedostatky .....	28
2.2.1 Problémy s přihlášením .....	28
2.2.2 Výskyt duplicitních záznamů v hlavní databázi .....	29
2.2.3 Zdlouhavé odesílání e-mailů.....	29
2.2.4 Nemožnost přidat dalšího příjemce emailu .....	30
2.2.5 Sledované podněty představenstvem společnosti – automatická kopie ....	30
2.2.6 Zvýraznění dlouho nepřirazených podnětů v hlášení.....	31
2.2.7 Důkladnější vyhledávání .....	31
2.2.8 Samovolné odhlášení po delší době nečinnosti .....	31
2.2.9 Vyhledávání zákazníka v databázi .....	32
2.2.10 Nelze dočasně změnit status „Uzavřeno“ .....	32

3	Řešení popsanych nedostatku v systému SKS .....	33
3.1	Problémy s přihlášením.....	34
3.2	Výskyt duplicitních záznamů v hlavní databázi.....	35
3.3	Zdlouhavé odesílání e-mailů .....	37
3.4	Nelze přidat dalšího příjemce e-mailu .....	38
3.5	Sledované podněty představenstvem společnosti – automatická kopie .....	39
3.6	Zvýraznění dlouho nepřirazených podnětů v hlášení .....	40
3.7	Důkladnější vyhledávání.....	41
3.8	Samovolné odhlášení při delší době nečinnosti .....	43
3.9	Vyhledávání zákazníka v hlavní databázi.....	44
3.10	Nelze dočasně změnit status „Uzavřeno“ .....	45
4	Získávání dat pro statistické a ekonomické modelování .....	47
4.1	Úvodem pojem datamining .....	47
4.1.1	Proces dobývání znalostí .....	47
4.1.2	Používané metodiky pro realizaci DM úloh .....	48
4.1.3	Metody používané při dolování dat .....	52
4.1.4	Využití statistických metod v dataminingu .....	54
5	Příklad zpracování dat .....	61
5.1	Komunikace zákazníka s POZ .....	61
5.2	Definování cílů .....	63
5.3	Popis datové struktury systému SKS .....	64
5.4	Příklady .....	64
5.4.1	Který model vozu (modelový klíč) je nejvíce poruchový .....	65
5.4.2	Průměrná cena opravy na model vozu .....	67
5.4.3	Ve kterém měsíci je evidováno nejvíce závad.....	67
5.4.4	Která závada se vyskytuje nejčastěji.....	68
5.4.5	Jaké jsou výdaje na nejčastější závadu u různých servisů.....	69
5.4.6	Trvání opravy nejčastější závady u různých servisů.....	70
5.4.7	Který servis (tj. ID servisu) řeší kolik závad .....	70
	Závěr.....	72
	Seznam použité literatury .....	74
	Přílohy .....	75
	Příloha A.....	76
	A.1 - Problémy s přihlášením .....	76

A.2 – Výskyt duplicitních záznamů v hlavní databázi .....	79
A.3 – Zdlouhavé odesílání e-mailů .....	83
A.4 – Nemožnost přidat dalšího příjemce emailu .....	86
A.5 – Sledované podněty představenstvem společnosti – automatická kopie .....	88
A.6 – Zvýraznění dlouho nepřirazených podnětů v hlášení .....	91
A.7 – Důkladnější vyhledávání .....	96
A.8 – Samovolné odhlášení po delší době nečinnosti .....	100
A.9 – Vyhledávání zákazníka v databázi.....	102
A.10 – Nelze dočasně změnit status „Uzavřeno“ .....	105
Příloha B .....	108
B.1 – Náhled zdrojového souboru pro datamining .....	108
B.2 – Seznam servisů seřazených dle četnosti zásahů .....	109
Obsah příloženého CD .....	110



## Seznam obrázků

Obr. 1: Logo společnosti ŠA (používané od 1. 3. 2011) .....	17
Obr. 2: Organizační struktura ve společnosti ŠA.....	18
Obr. 3: Struktura oddělení péče o zákazníka – rozdělení FO a BO .....	19
Obr. 4: Logo společnosti SAP .....	21
Obr. 5: Současná architektura SAP .....	21
Obr. 6: Graficky zpracovaný náhled na propojení jednotlivých vrstev .....	24
Obr. 7: Propojení jednotlivých modulů .....	25
Obr. 8: Původní program Sorry! pro evidování podnětů.....	27
Obr. 9: Základní obrazovka SKS s hlášením pro určitého pracovníka .....	28
Obr. 10: Proces při odeslání e-mailu z SKS .....	29
Obr. 11: Zobrazovaná hláška v případě delší nečinnosti.....	32
Obr. 12: Obyčejné zobrazení přehledu podnětů pro uživatele.....	41
Obr. 13: Barevné rozlišení dlouho zpracovávaných podnětů (> 2 dny, datum 30.9.2011) .....	41
Obr. 14: Jednotlivé fáze procesu dobývání znalostí z databází .....	48
Obr. 15: Metodika SEMMA .....	50
Obr. 16: Metodologie CRISP-DM .....	51
Obr. 17: Zobrazení pro body se stejnou vzdáleností .....	58
Obr. 18: Dendrogram .....	59
Obr. 19: Popis datových toků znázorněný pomocí DFD .....	63
Obr. 20: Grafické zobrazení datové struktury pomocí ERD .....	64
Obr. 21: Stream vytvořený v dataminingovém nástroji PAWS Modeler.....	65
Obr. 22: Celkový přehled modelový klíč x četnost závad (řazeno vzestupně).....	66
Obr. 23: Detailní zobrazení modelový klíč x četnost závad .....	66
Obr. 24: Přehled modelů × průměrné náklady na opravu .....	67
Obr. 25: Přehled měsíců × počet servisních zásahů .....	68
Obr. 26: Přehled všech oprav × počet jejich výskytů.....	68
Obr. 27: Náklady na nejběžnější závadu u různých servisů .....	69
Obr. 28: Zobrazení doby opravy u nejčastější závady u různých servisů .....	70
Obr. 29: Přehled servisů × počet servisních zásahů .....	71
Obr. 30: Přihlašovací obrazovka (SAP standard) .....	76
Obr. 31: Akce pro přihlášení (úprava).....	76

Obr. 32: Dotaz po stisku tlačítka Vymazat .....	79
Obr. 33: Varovná zpráva, že u zákazníka se vyskytují podněty .....	79
Obr. 34: Informace o vymazání včetně jména zákazníka .....	79
Obr. 35: Stav tlačítek před schválením e-mailu .....	83
Obr. 36: Informace o schválení e-mailu .....	83
Obr. 37: Stav tlačítek po schválení před odesláním e-mailu .....	83
Obr. 38: Informace o předání e-mailu serveru .....	84
Obr. 39: Výběrová obrazovka e-mailových adres .....	86
Obr. 40: Výběrová obrazovka e-mailových adres (vybrána jen zákazníkova – na sekretariát G automaticky) .....	88
Obr. 41: Obyčejné zobrazení přehledu podnětů pro uživatele .....	91
Obr. 42: Barevné rozlišení dlouho zpracovávaných podnětů (> 2 dny, datum 30.9.2011) .....	91
Obr. 43: Základní selekční obrazovka pro vyhledávání .....	96
Obr. 44: Selekční obrazovka po výběru jedné možnosti .....	96
Obr. 45: Informační hláška o překročení času nečinnosti .....	100
Obr. 46: Další informační hláška .....	100
Obr. 47: Informační hláška o uložení všech dat .....	100
Obr. 48: vyhledávací obrazovka po ukončení práce se zákazníkem .....	102
Obr. 49: Tlačítko pro změnu statusu podnětu .....	105
Obr. 50: Varianta neuzavřeného podnětu .....	105
Obr. 51: Pop-up okno s dotazem .....	105
Obr. 52: Reakce na stisk tlačítka v případě již jednou otevřeného podnětu .....	105

## Seznam tabulek

Tab. 1: Popis generických úloh podle CRISP-DM .....	51
Tab. 2: Čtyřpolní tabulka – zjištěné četnosti kombinací .....	54
Tab. 3: Čtyřpolní tabulka – odhadnuté četnosti kombinací .....	55
Tab. 4: Deset nejporuchovějších modelů (modelových klíčů) .....	66
Tab. 5: Deset nejčastějších poruch .....	69
Tab. 6: Zobrazení souboru pro prvních 10 záznamů (1. 1/3) .....	108
Tab. 7: Zobrazení souboru pro prvních 10 záznamů (2. 1/3) .....	108
Tab. 8: Zobrazení souboru pro prvních 10 záznamů (3. 1/3) .....	108
Tab. 9: Seznam servisů seřazených sestupně podle četnosti zásahů (pouze část) .....	109

## Seznam zkratek a termínů

5A	Assess-Access-Analyze-Akt-Automate
ABAP	Advanced Business Application Programming
AG	Aktiengesellschaft
B2B	Business to Business
BC	Basic Components
BO	Back Office
CA	Cross Application Functions
CO	Controlling
CRISP-DM	Cross-Industry Solution Process for DataMining
CRM	Customer Relationship Management
EC	Enterprose Controlling
ERP	Enterprise Resource Planning
FI	Financial Accouting
FO	Front Office
GUI	Graphics User Interface
IM	Investment Management
IN	International Development
IS	Informační Systém
KUBA	Kundendatenbank
LO	Logistic General
MM	Material Management
PA	Personnel Administration
PD	Personnel Planning
PM	Plant Maintenance
PP	Product Planning
PS	útvár Servisní služby
PST	útvár Technika servisních služeb
PST/2	oddělení péče o zákazníka
QM	Quality Management
SAP	Systems - Applications - Products
SD	Sales and Distribution
SEMMA	Sample-Explore-Modify-Model-Assess

SKS	Small KUBA Solution
ŠA	ŠKODA AUTO a.s.
POZ	oddělení péče o zákazníka
TR	Treasury
VIN	Vehicle Indetification Number

# Úvod

Společnost ŠKODA AUTO a.s. je výrobcem osobních automobilů s velmi bohatou historií. Nabízí vozidla v mnoha modelových řadách, vyrábí příslušenství, doplňky a originální díly a poskytuje servisní služby prostřednictvím servisní sítě, prodejců a partnerů značky ŠKODA. Organizace dosahuje obchodních úspěchů nejen na území České republiky, ale také v mnoha ostatních zemích. Jedním z hlavních cílů společnosti je spokojenost zákazníka. Právě zákazník rozhoduje o úspěchu a prosperitě každé firmy, a to svou spokojeností s kvalitou výrobku, cenou, úrovní a rozsahem poskytovaných služeb (prodejní a servisní) a pozitivními zkušenostmi. Spokojený zákazník zůstává věrný značce.

Diplomová práce se zabývá jedním z modulů informačního systému SAP, který pomáhá udržet kvalitu služeb pro zákazníky na vysoké úrovni. IS SAP je implementován v celé společnosti. Konkrétně se práce věnuje systému Small KUBA Solution, dále SKS, provozovaným pod SAP (Systems – Applications – Products) v podsystému služeb označovaných CRM (Customer Relationship Management). Vedle přizpůsobení tohoto informačního systému potřebám oddělení péče o zákazníka je dalším zájmem této práce prozkoumat možnost zpracování dat vznikajících na tomto oddělení některou z dataminingových metod.

V první části je popsána společnost ŠKODA AUTO a.s. Od založení, přes základní organizační strukturu až po přesnější popis činnosti oddělení péče o zákazníka, které využívá systém SKS pro evidenci a vyřizování podnětů zákazníků. Dále se práce věnuje informačnímu systému SAP s detailnějším zaměřením na modul CRM, pod kterým je spuštěn a používán systém SKS.

V druhé části diplomové práce jsou po prostudování systému SKS analyzovány problémy spojené s jeho užíváním v oddělení péče o zákazníka, vytipována problémová místa a projednány jejich změny s vedením oddělení.

Realizace navržených změn je možné teprve po zvládnutí programovacího jazyka ABAP/4, který je stavebním nástrojem informačního systému SAP. Teprve poté bude možné navržené úpravy po detailnějším rozboru jednotlivých problémových míst naprogramovat tak, aby došlo ke zvýšení uživatelského komfortu. Úpravy jsou popsány teoreticky a zároveň jsou představeny částí zdrojového kódu s printscreeny. Všechny naprogramované úpravy jsou odzkoušeny na trialové verzi miniSAP NetWeaver 7.01.

Dále se práce zabývá zpracováním dat pomocí dataminingového softwarového nástroje. Data, která jsou dostupná v systému SKS a budou poskytnuta pro připravovaný předmět s pracovním názvem „*Datamining*“, budou upravena z důvodu ochrany dat zadavatele. Pro výuku budou mít význam především proto, že jejich datový model je odrazem skutečné reality v průmyslové praxi. Odkoušené metody budou inspirací pro oddělení péče o zákazníka. Skutečnost, že dnes různé organizace či průmyslové podniky nebo například banky disponují velkými datovými množinami, dává možnost hledat v datech skryté informace, tento jev se nazývá „nechat promluvit data“. Proto budou nastíněny některé postupy a metodiky dataminingu pro zpracovávání velkých datových množin a popsán dataminingový nástroj PAWS Modeler, který má fakulta mechatroniky k dispozici.

Poslední kapitola popisuje možné využití dat SKS v dataminingovém prostředí PASW Modeleru. Zabývá se přípravou dat vygenerovaných ze systému SKS jako jednou ze zásadních úloh dataminingu a následně zpracováním v dataminingovém nástroji. Modelová úloha je připravena pro výuku připravovaného předmětu.

# 1 Charakteristika IS SAP ve společnosti ŠKODA AUTO a.s.

Informační systém SAP, o kterém pojednává tato diplomová práce, je rozšířen v celé společnosti ŠKODA AUTO a.s. (dále jen ŠA). V průběhu první kapitoly je nejprve přiblížena stručná historie společnosti, základní údaje a organizační struktura, v další části je popsána společnost SAP AG a IS SAP aplikovaný v ŠA.

## 1.1 Historie společnosti ŠKODA AUTO a.s.

[1] V roce 2005 oslavila společnost ŠA 110 let od svého založení. Jedna z největších firem v České republice a třetí nejstarší automobilka na světě (starší jsou jen společnosti Mercedes a Saab) vznikla roku 1895, kdy knihkupec Václav Klement a mechanik Václav Laurin založili firmu zabývající se výrobou bicyklů, která se přes výrobu motocyklů dostala roku 1905 až k výrobě automobilů.

V dubnu 1991 se stala ŠA jednou ze značek koncernu Volkswagen, bylo tak zahájeno období transformace v suverénní, dynamickou a prosperující společnost s vlastním výrobním programem. Kompetentnost značky ŠKODA, jež je založena na místní tradici inženýrského a řemeslného umu, pak dává všechny předpoklady k vnímání ŠA jako svébytné společnosti v rámci koncernu VW s vlastním výrobním programem. Mezi další automobilové značky, které jsou součástí koncernu VW, patří Audi, Bentley, Porsche, Seat a další.

V současnosti vyrábí ŠA osobní automobily v několika modelových řadách: Superb II, Octavia II Tour, Octavia II facelift, Fabia II facelift, Roomster a Yeti, všechny modely jsou na světovém automobilovém trhu vážnými konkurenty ve svých třídách.

**ŠKODA**



Obr. 1: Logo společnosti ŠA (používané od 1. 3. 2011)<sup>1</sup>

---

<sup>1</sup> Zdroj: ŠKODA Corporate Design

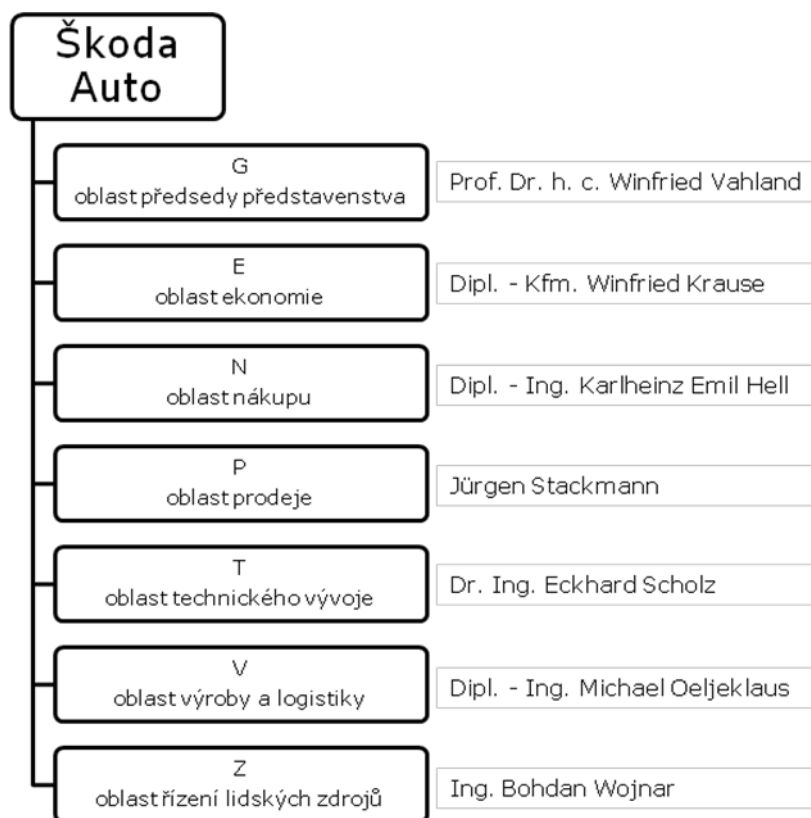


### 1.1.1 Základní údaje o společnosti ŠKODA AUTO a.s.

Společnost ŠKODA, automobilová a.s. byla zřízena dne 20. 11. 1990 jako právnická osoba. Dne 14. 11. 1997 se usnesením valné hromady změnilo obchodní jméno společnosti na ŠKODA AUTO a.s. s účinností od 1. 1. 1998. Hlavním předmětem činnosti je výroba a prodej vozidel včetně jejich příslušenství. Společnost v České republice tvoří tři závody na výrobu osobních automobilů. Mimo hlavní závod v Mladé Boleslavi se vyrábějí automobily též v pobočných závodech ve Vrchlabí a v Kvasinách. Odbyt a administrativa jsou soustředěny v Mladé Boleslavi.

### 1.1.2 Organizační struktura společnosti ŠKODA AUTO a.s.

Společnost ŠA je organizačně rozdělena do sedmi základních oblastí označených písmeny G, E, N, P, T, V a Z<sup>2</sup>. Značení je pouze vnitrofiremní a všichni nejvyšší zástupci jednotlivých oblastí jsou členy představenstva.



Obr. 2: Organizační struktura ve společnosti ŠA<sup>3</sup>

<sup>2</sup> Zdroj: ŠKODA zaměstnanecký portál

<sup>3</sup> Zdroj: vlastní zpracování a ŠKODA zaměstnanecký portál

Místo, kde se používá systém SKS, který je hlavním zájmem této diplomové práce, je útvar oblasti P – Prodej a Marketing, přesněji PS – Servisní služby, konkrétně PST – Technika servisních služeb, oddělení PST/2 – péče o zákazníka podle značení používaného v ŠA.

### 1.1.3 Oddělení péče o zákazníka

Toto oddělení spadá pod oblast PST – Technika servisních služeb a má na starosti vyřizování všech podnětů od zákazníků týkajících stížností, reklamací, dotazů, pochval či technických nápadů, a to jak pro tuzemské zákazníky, tak i zahraniční. Příchozí podněty mají nejčastěji formu e-mailu a jsou směrovány na Infoline ŠA ([info@skoda-info.cz](mailto:info@skoda-info.cz)). Přicházejí i podněty telefonické, nebo i písemné formou dopisů.

Pracovníci tohoto oddělení jsou rozděleni do dvou skupin – Front office (FO) a Back office (BO). Obě skupiny mají svého „team-leadera“ a jsou dále rozděleny na tuzemsko a zahraničí. Pracovníci, kteří vyřizují zahraniční podněty, obvykle podle jazykové kompetence, mají přidělenou korespondenci a komunikaci v anglickém a německém jazyce. Ostatní jazykové mutace jsou přeposílány specializovaným překladatelům.

Podněty jsou zpracovány do formulářů SKS a tvoří tabulku, která slouží k řízení vyřizování podnětů zákazníků.



Obr. 3: Struktura oddělení péče o zákazníka – rozdělení FO a BO<sup>4</sup>

---

<sup>4</sup> Zdroj: vlastní zpracování

## 1.2 O společnosti SAP

[2] Společnost SAP je v nynější době čtvrtou největší SW firmou na světě. Působí ve čtyřech geografických regionech: EMEA (Evropa, Střední Východ, Asie), AMERIKA (Spojené státy Americké a Kanada), LAC (Latinská Amerika a Karibik) a APJ (Asie – Pacifik – reprezentováno Austrálií, Indií a částí Asie a Japonskem). Především se specializuje na vývoj ERP (Enterprise Resource Planning) systému pro B2Bsegment(business to business). Nyní je SAP největším dodavatelem informačního systému pro podniky. Mezi jejich zákazníky patří nejvíce velké společnosti a organizace, ale tento SW si mohou zakoupit i střední a malé podniky. V České republice je společnost SAP v pozici absolutního leadera mezi dodavateli aplikačního software. 8 z TOP 10 zákazníků SAP ČR zaujímá první pozice ve svém zaměření a také to, že IS SAP je provozován v 51% z TOP 100 českých firem je známka jeho rozšířenosti. V dnešní době je nejznámějším produktem SAP NetWeaver. Tento produkt nahradil klasické aplikační servery a slouží jako integrační platforma, která obsahuje velké množství standardně dodávaného obsahu orientovaného především na webové aplikace. SAP NetWeaver je technologickým základem aplikací mySAP Business Suite, která obsahuje následující části:

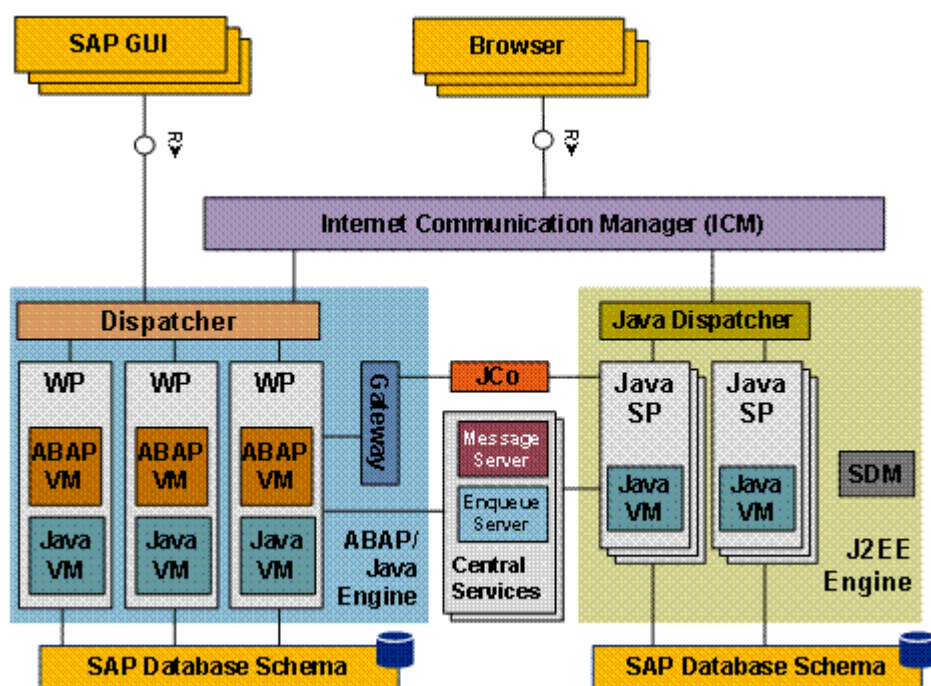
- SAP Customer Relationship Management (SAP CRM), sw pro řízení vztahů se zákazníky,
- SAP Enterprise Resource Planning (SAP ERP), sw pro plánování,
- SAP Supplier Relationship Management (SAP SRM), sw pro řízení vztahů s dodavateli,
- SAP Product Lifecycle Management (PLM), sw poskytující jednotný zdroj informací o produktech,
- SAP Supply Chain Management (SCM), sw pro spolupráci, plánování, fungování a koordinaci v zásobování.



Obr. 4: Logo společnosti SAP<sup>5</sup>

### 1.2.1 Základní údaje o SAP

SAP byl vytvořen v červnu roku 1972 jako „System Analysis and Program Development“ pěti zakládajícími inženýry z IBM v německém Mannheimu. Byli jimi D. Hopp, K. Tschira, H.-W. Hector, H. Plattner a C. Wellenreuther. Ve stejném roce byla jeho prvním zákazníkem firma ICI (Imperial Chemical Industries). V roce 1976 SAP GmbH přesunula hlavní sídlo do Walldorfu a v roce 1988 změnila název na SAP AG (podle německého zákona pro akciové společnosti).



Obr. 5: Současná architektura SAP

<sup>5</sup> Zdroj: webové stránky společnosti SAP AG – [www.sap.com](http://www.sap.com)

### **1.2.2 Důležitá data z historie společnosti SAP jako unikátního leadera na poli vývoje informačních systémů**

- 1973 - spuštěn systém SAP R/1,
- 1979 - spuštěn systém SAP R/2,
- 1981 - SAP kompletně mění design v oblasti prodeje. Nicméně tohoto SAP zatím nevyužívá do období mezi 1985 až 1990. Hlavní vylepšení nepřichází od zakladatelů nebo zaměstnanců, ale od partnerství se vzdělávacím institutem „Chico State“,
- 1985 - na 9000km vzdálené Kalifornské státní Univerzitě („Chico State“) student informoval pár svých profesorů o SW SAP. Požádal je o získání kopie pro zkušební účely. Po krátké době přešla konverzace z hodnotícího stupně na stupeň zdokonalování a vývoje. Obě strany souhlasily s tím, že se stanou partnery,
- 1988 - 4. listopadu byl zpřístupněn prodej veřejnosti,
- 1992 - SAP a „Chico State“ vyvinuly a vydaly několik verzí SAP R/3,
- 1995 - SAP následovala trendy z počítačového světa, tedy nutnost řešení informačních systémů architekturou klient – server, předělala koncept obchodního procesu a do řešení začlenila Internet,
- 1997 - SAP má partnerství s více jak 25 vzdělávacími instituty, včetně Massachusettského Institutu Technologii (MIT),
- 1999 - SAP získal cenu Industry Week's Best Managed Companies,
- 1993 – společnost SAP otevřela v pořadí 18. dceřinou společnost a to v České republice,
- 2000 – SAP vstupuje do světa elektronických tržišť a firemních portálů. To vše má na starosti její dceřiná společnost SAP Portals a zahajuje i partnerství se společností Commerce One,
- 2003 – v Číně se otevřelo vývojové centrum SAP Labs China (již deváté centrum mimo hlavní sídlo). Další centra jsou v Indii, Japonsku, Izraeli, Francii, Bulharsku, Kanadě a v USA. Společnost Sap chce těmito centry docílit co nejlepšího pochopení potřeb zákazníků.

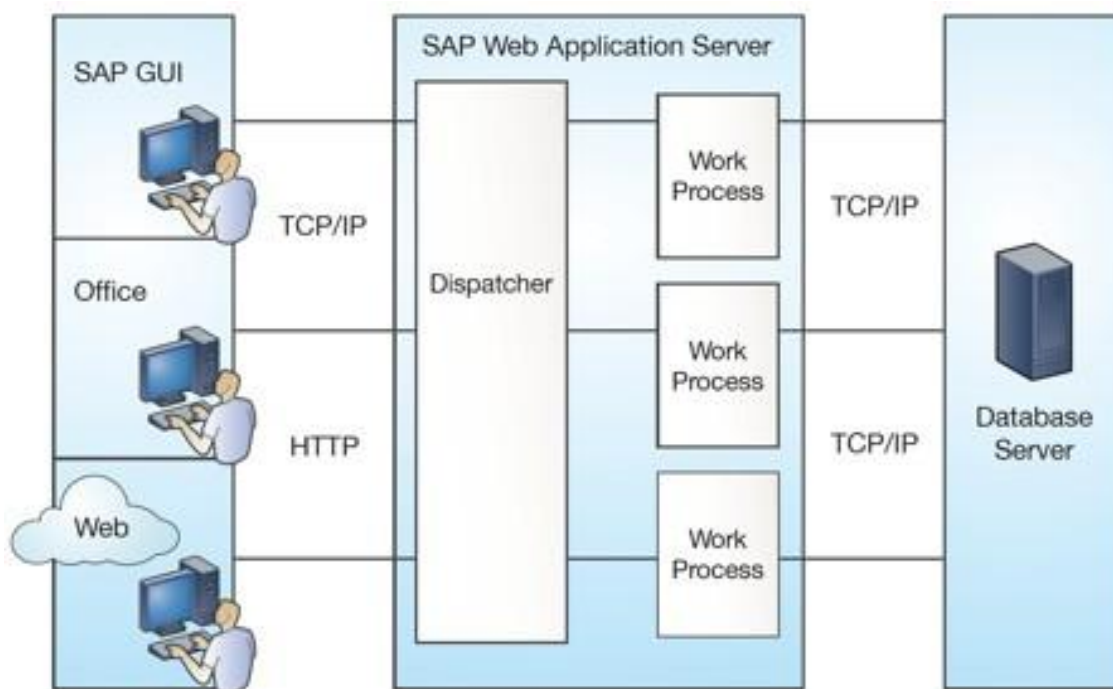
V tomto roce zaměstnávala společnost přes 30000 zaměstnanců – 17000 jich pracovalo mimo Německo,

- 2005 – v únoru společnost SAP oficiálně otevřela vývojové centrum v Budapešti v Maďarsku s více jak 50 vývojáři,
- 2009 – v únoru SAP koupil většinu firmy Coghead, ale prohlásil, že její technologie bude využívat pouze k vývoji vlastních produktů a nebude nabízet svým zákazníkům její produkty,
- 2009 – v květnu SAP oznámil, že kupuje výrobce databázového softwaru firmu Sybase za 5.8 miliardy dolarů v hotovosti. Firma Sybase nadále bude pracovat jako samostatná forma, ale SAP ji bude využívat pro vývoj svých produktů,
- 2010 – v listopadu společnost SAP ztratila 1.3 miliard dolarů ze svého majetku v souvislosti s akcí dceřiné společnosti SAP TomorrowNow. Tato částka připadla společnosti Oracle Corporation – citováno jako největší rozsudek v softwarovém pirátství v historii. SAP měl v úmyslu podat žádost o snížení náhrad škod společně s odvolání na rozsudek,
- 2011 – od chvíle, kdy společnost SAP začala spolupracovat se Státní univerzitou v Kalifornii (CSU, Chico), má v tomto roce již navázané partnerství s více jak 1000 vzdělávacími institucemi, z toho je 200 v severní Americe. Díky tomu má přes 250 000 studentů přístup k SAP systému díky SAP University Alliances.

### **1.3 Architektura IS SAP ve společnosti ŠKODA AUTO a.s.**

[3] Základní technická architektura je vícevrstvá, rozděluje se na tři vrstvy – prezentační, aplikační a databázovou. Prezentační vrstva zajišťuje vstup a výstup dat a komunikaci uživatele s počítačem pomocí dialogových obrazovek. V této vrstvě se také nachází SAP GUI (SAP Graphics User Interface – nutný pro komunikaci se systémem SAP) a pro webového klienta SAP lze použít libovolný webový prohlížeč. Aplikační vrstva, nyní má název SAP NetWeaver, je tvořena jedním nebo více aplikačními servery, které jsou rozděleny dle programovacího jazyka – ABAP nebo JAVA. Mezi jednotlivými servery jsou integrované komponenty, které provádějí konverzi dat, pokud je to nutné. Databázová vrstva je založená na společné databázi nad systémem

ORACLE., Je uložena na vlastním databázovém serveru. Mezi prezentační a aplikační vrstvou se nachází komponenta Web Dispatcher. Tato komponenta se především stará o možnost zobrazit SAP data na webových prohlížečích.



Obr. 6: Graficky zpracovaný náhled na propojení jednotlivých vrstev

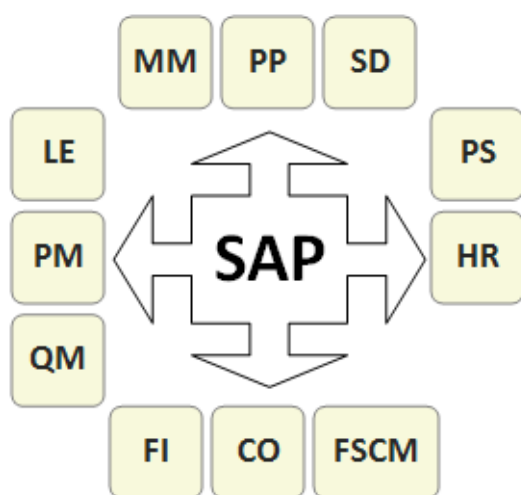
Ve společnosti ŠA se využívá velké množství SAP modulů. Všechny moduly jsou používány v maximální možné míře. Mezi tyto moduly patří:

- MM – Material Management (skladové hospodářství a logistika),
- FI – Financial Accounting (finanční účetnictví),
- HR – Human Resource,
- TR – Treasury (pokladní systém),
- CA – Cross-Application Functions (mezi aplikační funkce),
- CO – Controlling (kontroling),
- PP – Product Planning (plánování výroby),
- SD – Sales and Distribution (podpora prodeje).

A další, méně využívané:

IM – Investment Management, EC – Enterprise Controlling, LO – Logistics, QM – Quality Management, PM – Plant Management, PS – Project System, WF – WorkFlow, IS – Industry Solution.

Datové toky mezi moduly nejsou, všechny moduly jsou „napojené“ na databázi SAP. V databázi jsou uloženy tabulky pro jednotlivé moduly. Pokud nějaký modul potřebuje data uložená v jiném modulu, má je k dispozici právě v databázi. Stačí použít správně sestavený příkaz SQL jazyka a získat tak požadované informace. O konzistenci dat se stará serverová část systému klasickými databázovými prostředky jako jsou trigger, uložené procedury a UDF funkce.



Obr. 7: Propojení jednotlivých modulů

### 1.3.1 Systém SKS pod komponentou CRM

Zkratka *CRM* znamená *Customer Relationship Management*, ve volném překladu *Řízení vztahu se zákazníkem*. Tento pojem je v oblasti IT známý a používaný již několik let. Zpočátku nasazovaly tyto systémy pouze velké společnosti, které měly finance k jejich implementaci a provozování. V dnešní době je CRM finančně dostupný i pro malé a střední podniky.

V případě společnosti ŠA je kompletní systém CRM umístěn v hlavním městě koncernu VW, v německém Wolfsburgu a nazývá se KUBA. Toto umístění vychází z firemního standardu a je finančně méně nákladné. Pro členskou společnost VW koncernu ŠA bylo vyvinuto jednodušší a levnější řešení nazývané SKS. Tento název vznikl spojením prvních písmen *Small KUBA Solution* a systém je také implementován v CRM koncernu.



## **2 Analýza systému SKS a pracovních postupů na oddělení péče o zákazníka (OPZ)**

Na analýzu systému SKS se lze zaměřit z několika různých pohledů. Například je možné:

- Sledovat přínosy nasazení SKS, eventuálně je kvantifikovat. Například hledat odpověď na otázky: „Zlepšila se s SKS péče o zákazníka?“, pokud ano „Mělo zlepšení péče o zákazníka nějaký ekonomický efekt, kde a jak jej vyhodnotit?“
- Vyhodnotit způsob práce se systémem SKS na oddělení a sledovat, jestli systém poskytuje všechny potřebné funkcionality nebo ne. Hledat optimální ovládání systému tak, aby pracovník, který systém používá, nebyl obtěžován zbytečnými akcemi, a aby informace, které v daném čase potřebuje, měl k dispozici, a to v co nejlepší formě.

### **2.1 Jak fungovalo a dnes funguje POZ**

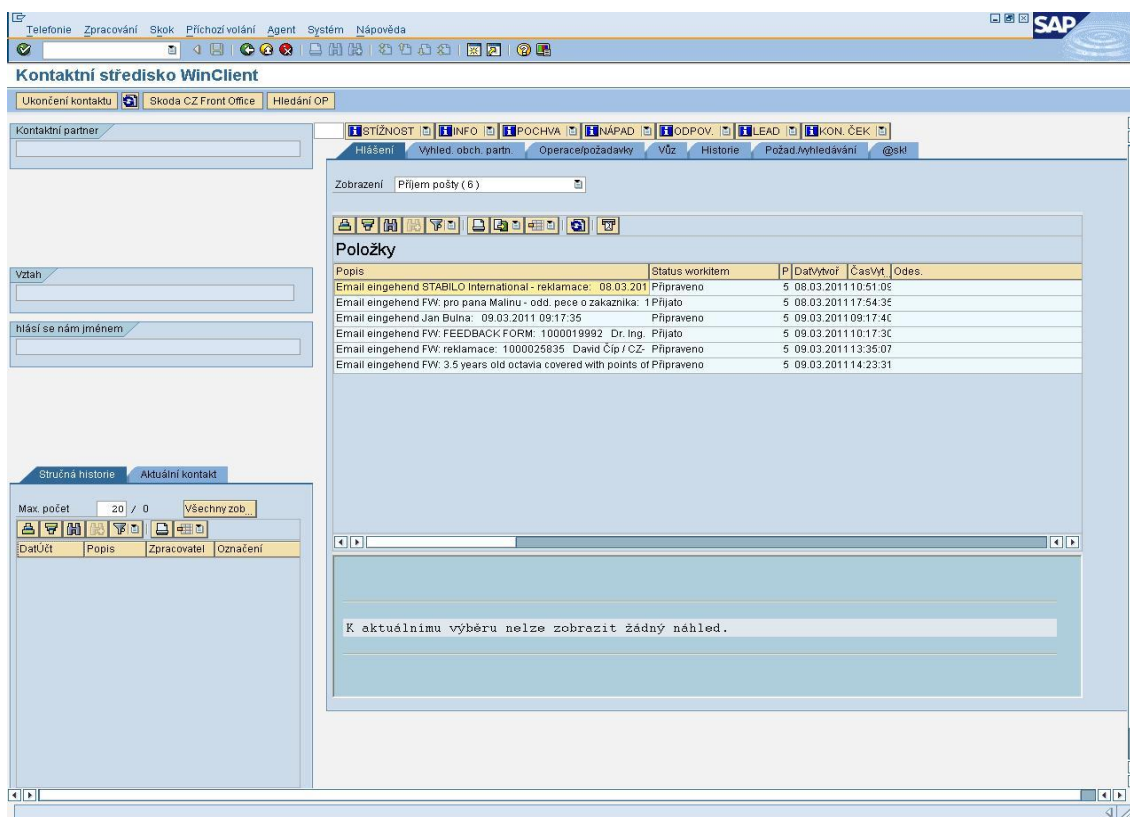
Původním programem pro evidenci podnětů zákazníků na oddělení péče o zákazníka v ŠA byl program Sorry! [Obr. 8] pracující pod OS Windows. Byl zde od roku 1999 a měl mnoho nedostatků. Například, veškerá korespondence musela být prováděna z osobních poštovních programů MS Outlook jednotlivých pracovníků oddělení. V program Sorry! pro každého zákazníka byla vytvořena elektronická složka, která však měla omezenou kapacitu. Pokud bylo zapotřebí zaevidovat mnoho materiálů a komunikace byla rozsáhlá, musela se zakládat další složka ke stejnému zákazníkovi. Data neúměrně, vzhledem k jejich organizaci a prezentaci, rostla a práce neodpovídala běžným standardům pro práci s „velkými“ daty. Běžné statistické a agregační funkce byly dostupné pouze ve velmi omezené míře. Po čase bylo nutné elektronické složky vytisknout a založit do archivu. Toto s sebou neslo nemalé finanční výdaje na tisk, skladování a následnou skartaci. Proto si společnost ŠA, v rámci úspory peněz, nechala vytvořit od společnosti Audi systém SKS.

Obr. 8: Původní program Sorry! pro evidování podnětů<sup>6</sup>

Systém SKS [Obr. 9] je používán již od roku 2007. Do programu mají přístup všichni pracovníci oddělení. Slouží jim k tomu jedinečné přístupové jméno a heslo. Je-li na oddělení doručen písemný podnět, je následně naskenován do elektronické podoby a zaslán na vnitřní e-mailovou adresu programu ([sks.service@skoda-auto.cz](mailto:sks.service@skoda-auto.cz)) jako příloha. Systém SKS obsahuje vnitřní e-mailovou schránku (tzv. *hlášení*), kde je tento podnět dostupný pro další zpracování a zobrazí se v seznamu *hlášení*. Je-li podnět zákazníka zaslán v elektronické podobě přímo do systému (buď přímo od zákazníka nebo přeposláním z Infoline), je zobrazen a dostupný také v *hlášení*. Pracovníci Infoline mají na starosti prvotní příjem e-mailů nebo telefonních podnětů směřovaných na [info@skoda-info.cz](mailto:info@skoda-info.cz). Základní dotazy vyřizují sami, ale složitější přeposílají do SKS.

Z tohoto *hlášení* zobrazeného v SKS si jednotlivé podněty vybírají pracovníci oddělení samostatně, a to podle svých kompetencí, znalostí a zkušeností. Zbylé podněty v *hlášení* přidělují k vyřízení team-leadeři. Oddělení řídí koordinátor, který má v SKS k dispozici roli team-leadera. Další uživatelské role pro do SKS v současnosti vytvořeny nejsou. Systém eviduje podněty od zákazníků. Tyto jsou kategorizované do FO – Front Office a BO – Back Office. Dělení určuje stupeň závažnosti podnětu, který ale není explicitně definován směrnici, či vytvořeným číselníkem. Pro koordinátora v čase vzniku této práce není dostupná řídicí funkcionality.

<sup>6</sup> Zdroj: vlastní printscreen programu



Obr. 9: Základní obrazovka SKS s hlášením pro určitého pracovníka<sup>7</sup>

## 2.2 Problémová místa systému SKS neboli uživatelské nedostatky

Systém SKS je používán 3 roky a již od počátku jsou vidět nedostatky nepříjemné pro uživatele. Postupem času se podařilo mnoho chyb odstranit, a tak eliminovat výskyt některých problémů či zdoluhavých procesů. Přesto doposud nejsou odstraněna všechna problémová místa, která práci se systémem komplikují. Tato kapitola je zaměřena na vytipování stávajících uživatelských nedostatků a návrh jejich řešení ke zvýšení pracovního komfortu. Mezi typické problémy patří např. duplicitní záznamy v tabulce zákazníků, zdoluhavé odesílání e-mailů, nemožnost přidat dalšího příjemce do kopie e-mailu, atd.

### 2.2.1 Problémy s přihlášením

Jedním z nedostatků, na které je potřeba poukázat, je zdoluhavé odblokování účtu v případě, že si jej uživatel zablokuje např. zadáním třikrát špatného hesla. Problém dlouhé doby odblokování vzniká z důvodu velkého počtu zaměstnanců v koncernu VW. Zde je řešením lepší správa účtů, ale jelikož vlastníkem systému SKS je firma Audi, je

<sup>7</sup> Zdroj: vlastní printscreen programu

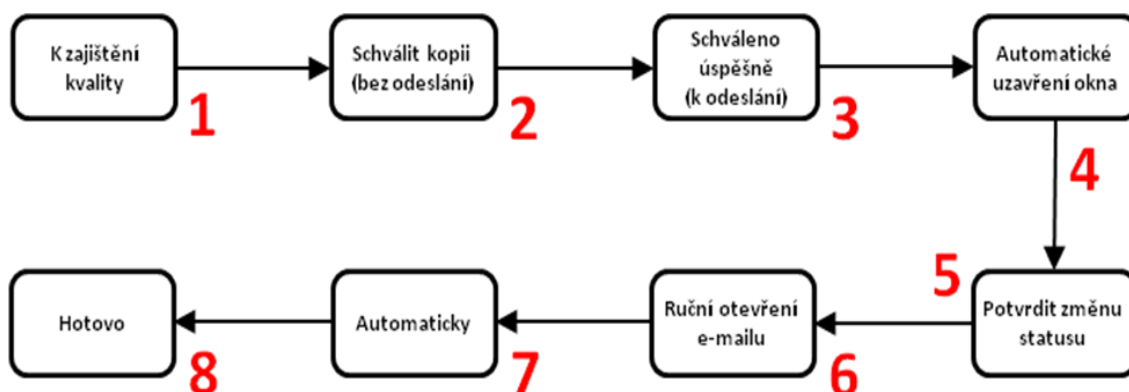
kompletní administrace a správa všech účtů prováděna z Německa. Alternativa řešení tohoto problému v ŠA bude popsána v kapitole 3.1.

### 2.2.2 Výskyt duplicitních záznamů v hlavní databázi

Díky občasné chybné komunikaci na oddělení se stane, že jeden business partner (tj. běžný zákazník, firma nebo ŠKODA dealer) je do hlavní tabulky zákazníků zaevidován dvakrát. Toto je zásadní chyba, protože pokud si zaměstnanec chyby nevšimne, může dojít k chybným analýzám i dalším provozním nedostatkům. Například přehlídnutí záznamu podnětů zákazníka sledováním jeho evidence na jiném záznamu, na kterém zdánlivě může být vše vyřízeno, na rozdíl od skutečnosti. Sice je možné přesunutí složek a podnětů od jednoho výskytu zákazníka ke druhému, ale není možné po kompletním přesunu prázdný výskyt smazat.

### 2.2.3 Zdlouhavé odesílání e-mailů

Systém SKS obsahuje integrovaného e-mailového klienta pro příjem a odesílání e-mailů (odpovědi zákazníkům). E-mail se vytváří v sekci **Operace / požadavky** a dále v záložce **Odkazy**. Po stisknutí tlačítka **Vytvoření individuálního dokumentu** je možno vybrat e-mail. Otevře se okno, uživatel napíše odpověď, a poté musí zahájit sérii úkonů, které nejsou složité, ale zabere čas a uživatel je nucen věnovat se tomuto procesu až do chvíle, než je e-mail odeslán zákazníkovi. Následuje schéma procesu:



Obr. 10: Proces při odeslání e-mailu z SKS<sup>8</sup>

<sup>8</sup> Zdroj: vlastní zpracování

Popis jednotlivých kroků:

- 1 – uživatel vytiskne odpověď na podnět a donese ji svému vedoucímu, který zkontroluje kvalitu odpovědi, navrhne změny nebo potvrdí odeslání Celý tento postup se provádí ve společnosti VW, na oddělení POZ zaveden není, ale je nutné tlačítko stisknout, aby mohl pokračovat proces odesílání,
- 2 – uživatel musí vytisknout „nanečisto“ e-mail. Všichni ho tisknout přes virtuální tiskárnu, která vytvoří dokument ve formátu PDF,
- 3 – po vytisknutí systém vyzve uživatele, aby potvrdil schválení,
- 4 – program sám od sebe uzavře okno s e-mailem (vše uloženo v paměti, takže nedojde ke ztrátě),
- 5 – po uzavření okna se objeví potvrzovací dialogové okno, kde uživatel musí stisknout tlačítko OK,
- 6 – po změně statusu na „úspěšně schváleno“ je nutno otevřít e-mail,
- 7 – po stisknutí „Automaticky“ se e-mail předává hlavnímu odesílacímu serveru, informační ikonka u e-mailu má tvar žlutého trojúhelníčku,
- 8 – celý proces končí tím, že se ikonka změní na zelené kolečko (toto musí uživatel zkontrolovat před tím, než vystoupí z „karty“ business partnera – musí mít jistotu, že e-mail byl odeslán). Zelené kolečko značí, že e-mail opustil odesílací server.

#### **2.2.4 Nemožnost přidat dalšího příjemce emailu**

Při odesílání e-mailu zákazníkovi je ve většině případů, kdy je znám i autorizovaný servis a jedná se o stížnost či reklamací, nutno kontaktovat i vedoucího servisu. To se provádí nejprve telefonicky, poté se elektronicky přeposílají veškeré podklady. Zde se ukazuje další nedostatek – není možné odeslat odpověď zákazníkovi a zároveň odeslat stejný text v kopii servisu. Uživatel je nucen odeslat podklady ze svého poštovního klienta nebo vytvořit nový e-mail v rámci SKS.

#### **2.2.5 Sledované podněty představenstvem společnosti – automatická kopie**

Mezi běžnými podněty se také vyskytují takové, které jsou přeposlány z vedení společnosti ŠA (označeno G – oblast předsedy představenstva společnosti) nebo vedení chce být informováno o jejich průběhu. Pokud se eviduje takovýto podnět, uživatel

mění jeho prioritu na vysokou. Jelikož předchozí nedostatek je nemožnost zasílat kopie, musí se veškerá komunikace mezi zákazníkem, příp. servisem a specializovanými odděleními ŠA, zasílat na sekretariát G prostřednictvím poštovního klienta MS Outlook.

### **2.2.6 Zvýraznění dlouho nepřirazených podnětů v hlášení**

Pokud je uživateli přidělen nějaký zákaznický podnět, v *hlášení* se změní status na „*Přijato*“ (pokud má uživatel A přidělen podnět X, uživatel B tento podnět v *hlášení* nevidí. Oba pracovníci vidí pouze podněty, které doposud nebyly rozděleny – mají status „*Připraveno*“). Při větším počtu přijatých i nepřijatých podnětů dochází k nepřehlednosti v zobrazení. Přirazení více stavů a jejich implementace do SKS, a to podle dohody s uživateli.

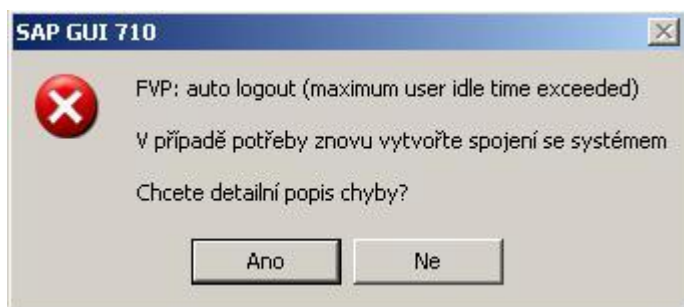
### **2.2.7 Důkladnější vyhledávání**

Vyhledávání provádějí pracovníci mnohokrát za den. Existuje několik případů, kdy je potřeba vyhledat například vůz, ale uživatel nemá k dispozici VIN vozu (jedinečné výrobní číslo) a je mu známa např. jen SPZ. Bohužel vyhledávání je možné pouze pomocí VIN. Ukazuje se jako velmi podstatné upravit vyhledávací algoritmy.

### **2.2.8 Samovolné odhlášení po delší době nečinnosti**

Pokud je uživatel přihlášen do systému, zpracovává podnět zákazníka (tj. vkládá do jeho složky určité informace) a je nucen na delší dobu odejít, program začne na pozadí počítat dobu nečinnosti. Pokud tento čas dovrší 45 minut a uživatel poté přijde k počítači, zobrazí se mu informační hláška [**Obr. 11**] „*FVP: auto logout (maximum user idle time exceeded). V případě potřeby znovu vytvořte spojení se systémem. Chcete detailní popis chyby? Ano – Ne*“ (v překladu: automatické odhlášení – byl překročen maximální čas nečinnosti). V tu chvíli má uživatel dvě možnosti jak postupovat. Pokud stiskne tlačítko *Ano*, zobrazí se výpis chyby (tj. je zde popsáno, že byl překročen čas připojení) a celý systém SKS se uzavře bez jakékoli nabídky na uložení a všechna data, která nebyla uložena, se vymažou. Po stisknutí na tlačítko *Ne* je akce podobná jako v případě stisknutí prvního tlačítka, ale bez zobrazení výpisu chyby. Pokud ale nestiskne žádné tlačítko, má možnost si nějaký text v programu zkopírovat do schránky (pouze ale text, který je vidět – nelze se přepnout do jiné záložky). Poté je nucen hlášku potvrdit. U tohoto nedostatku je zobrazen i další text, který není zde ve výpisu, ale provází uživatele každý den při práci v SKS, a tím je nejednotnost jazyka – na obrázku je vidět spojení anglického

a českého jazyka. Dalšími kombinacemi jsou např. německý jazyk – český jazyk (velmi časté) a anglický jazyk – německý jazyk. Systém velmi necitlivě řeší ukládání rozpracovaných úloh a komunikace s uživatelem je neprofesionální.



Obr. 11: Zobrazovaná hláška v případě delší nečinnosti<sup>9</sup>

### 2.2.9 Vyhledávání zákazníka v databázi

Několikrát denně uživatel využívá vyhledávání zákazníků v hlavní databázi. V základním okně, kde se vyplňuje požadované jméno, se nachází i tlačítko RESET. Toto tlačítko je nutné stisknout několikrát za sebou, aby se našel správný zákazník. Pokud se stiskne tlačítko pouze jednou, nalezne se například jiný zákazník nebo vyhledávání oznámí, že zákazník neexistuje. Pokud se ale tlačítko stiskne například dvakrát nebo třikrát, vyhledávání nalezne již požadovaného zákazníka (je-li uložen v tabulce). Ovládání tímto způsobem je zmatené a vyžaduje od uživatele nečekané a zbytečné akce.

#### 2.2.10 Nelze dočasně změnit status „Uzavřeno“

Každý uživatel má na starosti několik podnětů, které si hlídá, a pracuje s nimi, jak je potřeba a má dostupné informace. Pokud má již všechny podklady u sebe a problém mezi ŠA, respektive smluvním servisem ŠKODA a zákazníkem je vyřízen k jeho spokojenosti, uživatel má možnost celý podnět (aktivitu) uzavřít, tzv. změnit status z „Přijato“ na „Uzavřeno“. Po této změně se celý podnět uzavře a je k dispozici pouze pro čtení. Pokud ale uživateli přijde dodatečná informace, od zákazníka či servisu, musí otevřít novou aktivitu, jelikož nelze přiložit žádnou informaci k již uzavřenému podnětu. Zavírání podnětů tak nerespektuje přirozené řízení, které s danou činností v denní praxi probíhá. Proto je nutné upravit tento stav tak, aby byla možná dodatečná editace.

---

<sup>9</sup> Zdroj: vlastní printscreen informační hlášky

### 3 Řešení popsaných nedostatků v systému SKS

V této kapitole budou detailně rozepsána jednotlivá problémová místa, k nim navržená příslušná řešení včetně grafického náhledu a bude zobrazena i část kódu, který je v úpravě zajímavý. Změny jsou naprogramovány v jazyku ABAP, kterým se programují jednotlivé moduly pro SAP. Popis řešení je pro každý nedostatek rozdělen do následujících 5 částí:

1. popis projevu problémového místa,
2. popis návrhu úpravy,
3. grafický náhled bude zobrazen pouze u složitějších úprav, všechny náhledy budou zobrazeny v příloze (všechny printscreeny jsou získány vlastním zpracováním),
4. ukázka nejzajímavější části kódu,
5. vyhodnocení nutnosti implementace.

Programování jednotlivých návrhů řešení nedostatků bude testováno na počítači (konfigurace počítače: procesor Intel i5-760 2,8GHz, operační paměť 4GB, grafická karta ATI Radeon 256MB, OS Windows 7 64b Professional s XP Mode 32b režimem), kde je nainstalována aplikace miniSAP NetWeaver 7.01 32b. Tento SW je volně dostupný jak z internetových stránek firmy SAP AG, tak i z přiloženého média u knihy [5] a je určen pro zkušební vývoj reportů v jazyce ABAP/4. Hlavní součást instalace je podpora toho programovacího jazyka. Součástí této instalace je MaxDB Manager pro správu databáze a SAP GUI pro přihlášení do SAP.

Základní informace o jazyce ABAP/4:

Původně byl vyvinut společností SAP pro databázové aplikace, proto má vestavěnou podporu SQL příkazů, kolem roku 2000 bylo provedeno objektové rozšíření jazyka (tzv. ABAP Objects).

Uživatel (koncový uživatel, programátor, atd.) může otevřít až 6 paralelních připojení k ABAP serveru. Interní paměť obsahuje ABAP programy a související data. V programech napsaných v jazyce ABAP je možné používat export a import dat pro přístup ke sdílené paměti.



Pro otevření programů v jazyce ABAP používá programátor transakční kód SA38 a pro jejich úpravu SE38. Tyto kódy znamenají "odkazy" na příslušné podprogramy v rámci SAP.

### **3.1 Problémy s přihlášením**

#### **Popis projevu problémového místa:**

V případě, že uživatel má zablokovaný účet, je nutné kontaktovat administraci ve VW, kde se zadá požadavek na jeho odblokování. Samotné zprovoznění účtu se neděje automaticky, ale pověřeným pracovníkem správou účtu v centru, a to ručně, což je z důvodu velkého počtu uživatelů v celém koncernu VW pomalé. Negativním důsledkem je, že postižený uživatel nemůže pokračovat v práci a zpracovávat podněty zákazníků.

#### **Popis návrhu úpravy:**

Zlepšením, pro plynulejší práci, je přesun administrace a správy účtů z VW do ŠA. Administrace se nemusí týkat jen chybových stavů účtů, ale celé správy těchto dat. Z důvodu zastupitelnosti mají pověření vykonávat tuto činnost dva zaměstnanci IT. V případě, že by toto řešení nebylo možné dojednat, schůdnějším řešením je zavedení několika virtuálních účtů pro řešení popsaných situací. Virtuální účet má možnost paralelního přihlášení. Pokud je nutné paralelní přihlášení použít a jeden uživatel je již pod tímto účtem přihlášen, systém ho informuje, že je již používán a nabídne 3 možnosti:

- a) zrušení přihlašovací akce,
- b) zrušení aktuálního přihlášení a náhrada jiným přihlášením,
- c) možnost přihlásit se paralelně s aktuálním přihlášením. – přidělení tohoto režimu s vyššími právy je nutné domluvit s IT oddělením ŠA.

Pro potřebu zajištění plynulého provozu oddělení i v době dočasného zablokování účtu, vyhovuje třetí varianta, a to především proto, že nelze vyloučit, že bude stačit jedno „krizové“ přihlášení. Po přihlášení do systému tímto způsobem, bude nový modul kontrolovat „práci“ obou přihlášení. Pokud chce uživatel pod druhým přihlášením pracovat se stejným zákazníkem, systém opět otevře informační hlášku o tom, že v tomto zákazníkovi je již uživatel se stejným jménem přihlášen, tudíž není možné vstup do detailů zákazníka. Řešení kolizních situací jinak řeší databázový engine svými

„zámky“ nad poli a recordy tabulek v databázové serverové části systému, do které ŠA z pochopitelných důvodů nemůže vstupovat, a proto kolize v tomto speciálním případě bude řešit nový modul. Postup přesměrování uložených změn od virtuálního uživatele, ke skutečnému uživateli „postiženého zablokováním účtu“ je možný při dalším přihlášení. Virtuální účet bude mít takto nastavená práva pro manipulaci s jeho daty.

**Grafický náhled úpravy** se nachází v příloze A.1 - Problémy s přihlášením.

**Ukázka části kódu:**

```
START-OF-SELECTION.  
IF user = sy-uname.  
    clear text. "definování informačního textu pro pop-up okno  
    CONCATENATE 'Uživatel' SY-UNAME 'je již přihlášen. Zvolte další  
                krok:' INTO text SEPARATED BY space.  
    PERFORM pop_up.  
ELSE.  
    Write: / 'Probíhá přihlášení do systému, čekejte...'.  
ENDIF.
```

**Vyhodnocení nutnosti implementace:**

Toto řešení není urgentní, lze implementovat časem.

## 3.2 Výskyt duplicitních záznamů v hlavní databázi

**Popis projevu problémového místa:**

V hlavní tabulce zákazníků se vyskytuje duplicita záznamů – tj. dvakrát nebo vícekrát zaevidovaný stejný zákazník. Duplicita vznikla konverzí dat a tabulek z předešlého programu Sorry!. U každého výskytu je možné nalézt podněty, které lze přesunout k jednomu, ale bohužel již nelze smazat ten výskyt, který neobsahuje žádný podnět. Je jasné, že díky velkému počtu obyvatel na celém světě včetně České republiky se mohou vyskytnout dva či více zákazníků, kteří mají stejná jména či dokonce bydlí na stejné adrese (příbuzní nebo jen náhoda). Ale nikdy není možné, aby měli například stejnou kombinaci telefonu a e-mailové adresy.

**Popis návrhu úpravy:**

Způsob, jak předcházet duplicitám, je správná kontrola záznamů v databázi již při vyhledávání zákazníka. V případě, kdy je nutné zaevidovat nového zákazníka, program zkontroluje všechny relevantní identifikační informace ze vstupního formuláře. A to takto: pokud se zjistí, že se jedná o stejné příjmení a jméno, jsou zkontrolovány i ostatní identifikační informace a v případě, že se jedná o stejného zákazníka, program vhodně a důrazně informovuje uživatele, že již není potřeba vytvořit nový záznam,

že zákazník již v tabulce existuje. Je možné, že některé informace budou při vyhledávání chybět (např. telefon, e-mail, adresa, atd.), a tudíž nebude možné jednoznačně určit zákazníka – tento problém povede v budoucnu k možnému výskytu duplicit. Je to z důvodu toho, že ne vždy zákazník podá uživateli všechny své údaje.

Dalším způsobem, jak odstranit duplicitní záznamy, je mazání jednotlivých záznamů. To probíhá v několika krocích tak, aby se významná data nemohla ztratit. Pokud uživatel při vyhledávání zjistí, že existuje další výskyt toho stejného zákazníka, kterého vyhledává, bude mít možnost označit řádek ve výsledném výběru, stisknout tlačítko „Nahlásit“, a tak informovat svého team-leadera o duplicitním záznamu v tabulce zákazníků. Team-leader si na základě tohoto nahlášení zobrazí stejný výsledek vyhledávání jako uživatel (při nahlášení se předají i přesné hodnoty formulářových polí, které vyplnil uživatel), určí, který výskyt se stane hlavním, označí zbylé a stiskne tlačítko „Vymazat“. Po stisku program nejprve zkontroluje, zda výskyty neobsahují podněty. Pokud ano, zobrazí team-leaderovi informační okno s touto informací a dotazem, zdali je možné přesunout podněty od duplicitních výskytů do hlavního. Team-leader tento přesun, který se vykoná automaticky na pozadí, potvrdí a vyčká na další informační hlášku o úspěšném přesunutí. Po této informaci opět program zkontroluje podněty u vymazávaných výskytů. V případě, že se již žádný podnět nenalezne, provede se vymazání z tabulky zákazníků.

**Grafický náhled úpravy** je k dispozici v příloze A.2 – Výskyt duplicitních záznamů v hlavní databázi.

#### **Ukázka části kódu:**

```
IF ODPOVED = '1'.
* kontrola, zdali jsou u zákazníka nějaké záznamy.... formou zavolání
  formu
  IF i_zcustomers-podnet = 'X'.
    Message W000(38) with 'Tabulka záznamů není prázdná, přesuňte vše
      a poté opakujte akci!!! '.
    EXIT.
  ELSE.
    DELETE FROM zcustomers WHERE prijmeni = surname AND jmeno = name.
    CONCATENATE text-011 surname name text-012 INTO text SEPARATED BY
      space.
    WRITE text.
  ENDIF.
```

#### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned. Duplicitní záznamy jsou hrubou databázovou chybou a jsou příčinou chybných výsledků různých agregačních funkcí v reportech nad zákaznickými daty.

### 3.3 Zdlouhavé odesílání e-mailů

#### Popis projevu problémového místa:

Systém SKS má výhodu oproti staršímu programu, že lze zde odesílat e-maily zákazníkům nebo dealerům prostřednictvím integrovaného e-mailového klienta. Po napsání obsahu zprávy, kterou chce uživatel odeslat na určitou e-mailovou adresu, musí absolvovat sérii úkonů, které jsou ale zbytečně zdlouhavé. Pro odeslání jednoho e-mailu musí uživatel udělat 8 úkonů, které zaberou čas a některé silně obtěžují. Uživatele to také omezuje v plynulé práci (přesný popis a obrázek viz nadpis 2.2.3 *Zdlouhavé odesílání e-mailů*).

#### Popis návrhu úpravy:

Odstraněním tohoto problémového místa je realizace sloučení 8 úkonů do dvou tak, aby uživatel v nejlepším případě stiskl pouze jedno až dvě tlačítka a již jen zkontroloval výsledek. Ten se projeví změnou informační ikony v seznamu příloh. Pokud zůstane úkon pro tisk, systém nabídne uživateli pouze výběr tiskárny a uložení na disk. Zbytek se provede automaticky a uživatel se již nebude o nic starat.

První tlačítko zastává funkce těchto následujících úkonů (jsou vysvětleny v kapitole 2.2.3): k zajištění kvality, schválení kopie, schváleno úspěšně, uzavření okna, schválení změny statusu a opětovné otevření e-mailu. Druhé má za úkol jen předat e-mail serveru, uzavřít e-mailového klienta a změnit informační ikonu v seznamu.

**Grafický náhled úpravy** je k dispozici v příloze A.3 – Zdlouhavé odesílání e-mailů.

#### Ukázka části kódu:

```
FORM schvaleni.  
  CALL FUNCTION 'OPEN_FORM'  
    EXPORTING  
      DEVICE      = 'PRINTER'  
      LANGUAGE    = SY-LANGU  
      OPTIONS     = ' '  
    IMPORTING  
      LANGUAGE    = SY-LANGU  
    EXCEPTIONS  
      DEVICE      = 2.  
  IF SY-SUBRC <> 0.
```

```

        Message W000(38) WITH 'Chyba tiskárny!!!'.
        approved = 'SCNO'.
    ELSE.
        Message I000(38) WITH 'Úspěšně schváleno'.
        approved = 'SCOK'.
    ENDIF.
    PERFORM check.
ENDFORM.                                "schvaleni

```

### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned, chyba nenarušuje konzistenci dat, ale způsob práce neodpovídá současným standardům IT a uživatele velmi obtěžuje.

## **3.4 Nelze přidat dalšího příjemce e-mailu**

### **Popis projevu problémového místa:**

Ve většině případů je potřebné, aby uživatel zaslal kopii e-mailu, který je určen zákazníkovi jako odpověď, také vedoucímu servisu. Toto bohužel není možné provést v jednom kroku tak, jak jsou uživatelé zvyklí u jiných poštovních klientů. V SKS je uživatel nucen poslat daný text ve dvou oddělených e-mailech, tedy odeslat odpověď zákazníkovi, a poté si text překopírovat do nového e-mailu, který je adresován vedoucímu servisu.

### **Popis návrhu úpravy:**

Úprava se týká případů, kdy je u zákazníka zaevidován i servis a je potřeba zaslat informaci i vedoucímu servisu. Při vytváření odpovědi zákazníkovi a výběru e-mailové adresy, uživatel vybere nejen adresu zákazníka, ale i servisu. E-mail na servis je v poli skryté kopie, tudíž zákazník nevidí dalšího příjemce. I když z hlediska důvěry v kvalitu vyřízení zákaznického podnětu je dobré, aby zákazník věděl, že kopie je odeslána také servisu, z důvodu interních stanov se kopie zasílá jako skrytá. Pouze ve vybraných případech se pošle běžná kopie. Rozeslání obstarává e-mailový server.

**Grafický náhled úpravy** je k dispozici v příloze A.4 – Nemožnost přidat dalšího příjemce emailu.

### **Ukázka části kódu:**

```

IF p_name02 IS INITIAL.
    CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
        I_TITLE                = text_header
        I_EMAIL                 = p_name01

```

```

TABLES
  T_LINES                                = text_email.
ELSE.
  CALL FUNCTION 'ZMM_SEND_EMAIL'
  EXPORTING
    I_TITLE                                = text_header
    I_EMAIL                                = p_name01
    I_EMAIL_COPY_CC                        = p_name02
  TABLES
    T_LINES                                = text_email.
ENDIF.

```

### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned pro optimální a efektivní využití tohoto systému.

## **3.5 Sledované podněty představenstvem společnosti – automatická kopie**

### **Popis projevu problémového místa:**

V některých případech zákazník zašle podnět (dopis, e-mail nebo fax) adresovaný přímo na předsedu představenstva společnosti. Pokud jeho sekretariát rozhodne předat podnět na oddělení péče o zákazníka, přiřazuje se tomuto podnětu při zaevidování nejvyšší priorita. V tu chvíli si je uživatel SKS vědom, že veškerou komunikaci s tímto zákazníkem musí přeposílat na sekretariát předsedy představenstva společnosti. Uživatel musí příchozí korespondenci vložit do nového e-mailu (vytvořeného buď v SKS nebo ve svém e-mailovém klientu – poté je nutné vložit odeslaný e-mail jako přílohu k podnětu do systému do záložky Odkazy) a odeslat.

### **Popis návrhu úpravy:**

Řešení tohoto nedostatku je velmi podobné předchozímu případu s tím rozdílem, že uživatel nemusí vybírat v seznamu více adres, ale pouze tu zákaznickou, případně adresu servisu. V průběhu evidování všech detailů podnětu vybere pouze vyšší prioritu, což znamená, že podnět je sledován sekretariátem předsedy představenstva. V tomto výběru bude veškerá odchozí komunikace směřovat automaticky jako skrytá kopie i na tento sekretariát.

**Grafický náhled úpravy** je k dispozici v příloze A.5 – Sledované podněty představenstvem společnosti – automatická kopie.

### **Ukázka části kódu:**

```

IF t_zcustomers-priority = 'X'.
IF p_name02 IS INITIAL.
CALL FUNCTION 'ZMM_SEND_EMAIL'
EXPORTING
  I_TITLE                = text_header
  I_EMAIL                = p_name01
  I_EMAIL_COPY_CC        = 'sekretariat_g@skoda-auto.cz'
TABLES
  T_LINES                = text_email.
ELSE.
  CONCATENATE p_name2 'sekretariat_g@skoda-auto.cz' INTO p_name2
              SEPARATED BY ';'.
  " pokud se vybere i servis, spojují se adresy a rozděleny jsou
  středníkem
CALL FUNCTION 'ZMM_SEND_EMAIL'
EXPORTING
  I_TITLE                = text_header
  I_EMAIL                = p_name01
  I_EMAIL_COPY_CC        = p_name02
TABLES
  T_LINES                = text_email.
ENDIF.

```

#### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned pro optimální a efektivní využití tohoto systému.

### **3.6 Zvýraznění dlouho nepřirazených podnětů v hlášení**

#### **Popis projevu problémového místa:**

Při pohledu na záložku *hlášení* v systému SKS je uživatel schopen rozeznat na první pohled podněty, které zpracovává a které nemá zatím nikdo přiřazené. V dalším sloupci je zobrazen datum příjmu podnětu do systému. Pokud ale nastane okamžik, kdy se v *hlášení* objeví více podnětů, stává se, že uživatel některý přehlédne nebo přidělí osobní prioritu jinému než tomu, který vyžaduje nejrychlejší zpracování.

#### **Popis návrhu úpravy:**

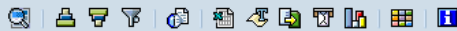
Základní barevné rozdělení podnětů v *hlášení* se provede například jednou denně. Systém kontroluje aktuální datum a odečte to, které nalezne v došlém podnětu. Rozdíl v nepřidělených podnětech více jak 2 pracovní dny, označí červenou barvou.

Druhou variantou barevného rozlišení je obarvení nepřidělených podnětů tak, aby bylo zřejmé, které jsou tuzemské a které zahraniční. Systém zkontroluje určitý záznam v tabulce – nadpis podnětu. Poté dle nalezeného a předem definovaného dvou písmenného kódu vyhodnotí, o který typ se jedná, a provede obarvení. České a slovenské, reprezentovány kódem CZ, by označil červenou barvou, zahraniční,

reprezentovány kódem EN, žlutou. Toto vylepšení v kódu je ve formě návrhu a diskusí, není zatím implementováno.

### Grafický náhled úpravy:












Označení podnětů



Popis	Status	WorItem	P	Datum	Čas	Odeslané
EN PROBLEM WITH ENGINE SKODA OCTAVIA	PŘIJATO	X		20110925	112211	
CZ NETĚŠTNOST MOTOROVÉHO BLOKU	PŘIPRAVENO			20110929	145212	
CZ VŮLE V ŘÍZENÍ	PŘIPRAVENO			20110927	105841	

Obr. 12: Obyčejné zobrazení přehledu podnětů pro uživatele

Označení podnětů



Popis	Status	WorItem	P	Datum	Čas	Odeslané
EN PROBLEM WITH ENGINE SKODA OCTAVIA	PŘIJATO	X		20110925	112211	
CZ NETĚŠTNOST MOTOROVÉHO BLOKU	PŘIPRAVENO			20110929	145212	
CZ VŮLE V ŘÍZENÍ	PŘIPRAVENO			20110927	105841	

Obr. 13: Barevné rozlišení dlouho zpracovávaných podnětů (> 2 dny, datum 30.9.2011)

### Ukázka části kódu:

```

SELECT * FROM zmailtab
        INTO CORRESPONDING FIELDS OF TABLE it_zmailtab.
LOOP AT it_zmailtab INTO wa_zmailtab.
    LOOP AT it_zmailtab.
        CLEAR ld_color.
        IF wa_zmailtab-status = 'PŘIPRAVENO'.
            "difference = ( difference - wa_zmailtab-datum ).
            IF wa_zmailtab-difference > '2'.
                ld_color = 6. "světle červená barva v ALV
                CONCATENATE 'C' ld_color '10' INTO wa_zmailtab-line_color.
            ENDIF.
        
```

### Vyhodnocení nutnosti implementace:

Nezbytné implementovat ihned pro optimální a efektivní využití tohoto systému.

## 3.7 Důkladnější vyhledávání

### Popis projevu problémového místa:

Vyhledávání je nejpoužívanější proces během dne. Nejčastěji se využívá při vyhledávání zákazníka, o tomto problému pojednává kapitola 2.2.9 (resp. v návrhu



řešení v kapitole 0). V některých případech je nutné vyhledávání vozidla, které funguje pouze, je-li známo výrobní číslo nebo VIN. Pokud uživatel zná pouze SPZ nebo modelový klíč<sup>10</sup>, SKS již není schopen vozidlo dohledat. Modelový klíč se nepoužívá k vyhledání konkrétního vozu, ale uživatel může zjistit, kolik je problémových vozů se stejným údajem. Uživatelé na konci každého měsíce vytvářejí přehledy zpracovaných podnětů. Při takovémto zpracování využívají jiné záložky než pro vyhledávání zákazníků.

### Popis návrhu úpravy:

Do detailů každého podnětu zadávají uživatelé i informace o vozidle, kterého se podnět týká. Zadané informace se ukládají do databáze a komfortnější vyhledávání je tedy realizovatelné rozšířením programového kódu Nabídka v rozhraní SKS je rozšířená o další detaily, podle kterých a které lze vyhledávat. Vlastní hledání je pak realizováno v klientském modulu. Rychlejší je ale volání nové uložené procedury na databázovém serveru. O takovou implementaci je nutné požádat vedení koncernu VW. Navrhnout celý kód pro uloženou proceduru je možné jen po získání celé datové struktury databáze VW.

**Grafický náhled úpravy** je k dispozici v příloze A.7 – Důkladnější vyhledávání.

### Ukázka části kódu:

```
FORM vin.
  CONCATENATE text-034 vin INTO text SEPARATED BY space.
  SELECT * FROM zvehicles
        INTO CORRESPONDING FIELDS OF TABLE t_zvehicles
        WHERE idvin = vin.
ENDFORM.
*&-----
FORM model.
  CONCATENATE text-035 vin INTO text SEPARATED BY space.
  SELECT * FROM zvehicles
        INTO CORRESPONDING FIELDS OF TABLE t_zvehicles
        WHERE model_code = model.
ENDFORM.
*&-----
FORM spz.
  CONCATENATE text-036 vin INTO text SEPARATED BY space.
  SELECT * FROM zvehicles
        INTO CORRESPONDING FIELDS OF TABLE t_zvehicles
        WHERE spz = spz.
ENDFORM.
```

---

<sup>10</sup> kombinace čísel a písmen (např. 5L71D5 označuje vůz Yeti s určitou výbavou a určitým motorem)

### **Vyhodnocení nutnosti implementace:**

Toto řešení není urgentní implementovat ihned.

## **3.8 Samovolné odhlášení při delší době nečinnosti**

### **Popis projevu problémového místa:**

Často se stává, že ve chvíli, kdy uživatel eviduje podnět zákazníka, je nucen na delší dobu odejít. Zamkne si přístup do počítače, ale ze systému SKS se neodhlásí. V tento moment se začne odečítat čas, který kontroluje nečinnost připojení. Pokud se vrátí k počítači po více jak 45 minutách a chce pokračovat v práci, zobrazí se mu informační hláška o překročení maximálního času nečinnosti. V tu chvíli má možnost zkopírovat text, který je viditelný na aktuální záložce (nelze se přepnout do jiné) nebo systém SKS ukončit a znovu přihlásit. Je celkem pochopitelné, že z důvodu velké síťové vytíženosti v rámci společnosti ŠA, je nutné kontrolovat aktivnost připojení. Pokud je nějaké spojení nečinné po delší dobu, je zbytečné jej udržovat, a proto dochází k přerušení po uplynutí určité doby. Problém pro uživatele nastává při ztrátě neuložených dat.

### **Popis návrhu úpravy:**

Řešením je automatické uložení rozpracovaného podnětu. Po uplynutí času neaktivity systém zobrazí informační hlášku a odpojí uživatele od spojení. Uložení proběhne před zobrazením informační hlášky a uživatel nepřichází o rozpracovaná a neuložená data.

**Grafický náhled úpravy** je k dispozici v příloze A.8 – Samovolné odhlášení po delší době nečinnosti.

### **Ukázka části kódu:**

```
* logout čas je nastaven na 45 minut = 2700 vteřin
IF rdisp/gui_auto_logout = 2500.
* uložení všech tabulek, které mají souvislost se zákazníkem a
                                                                podnětem
ENDIF.
* otevření info okna, že překročen čas a spojení zrušeno
PERFORM info.
PERFORM potvrzení.

FORM info.
CALL FUNCTION 'POPUP_TO_DISPLAY_TEXT'
  EXPORTING
```

```

TITEL = 'Byl překročen maximální čas nečinnosti'
TEXTLINE1 = 'V případě potřeby vytvořte spojení se systémem
                                                    znovu.'
TEXTLINE2 = 'Všechna data budou uložena.'
START_COLUMN = 13
START_ROW = 2.
ENDFORM.

```

### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned pro optimální a efektivní využití tohoto systému.

## **3.9 Vyhledávání zákazníka v hlavní databázi**

### **Popis projevu problémového místa:**

Během dne je třeba několikrát vyhledat zákazníka. V základním okně pro zadávání vyhledávacích slov se nachází i tlačítko *Reset*. Pokud uživatel před vyplňováním nestiskne toto tlačítko minimálně 3x, stane se, že s největší pravděpodobností nedohledá správného zákazníka nebo mu systém oznámí, že zákazník neexistuje. Když ale tlačítko 3x stiskne, zákazníka nalezne, pokud byl již v minulosti zaevidován nebo je v tabulce uveden jiný se stejným jménem. Tento postup je velmi nepraktický. Navíc mezi jednotlivými stisknutími tlačítka musí být prodleva jednu vteřinu, což výrazně zvyšuje riziko duplicity v databázi.

### **Popis návrhu úpravy:**

Při stisku tlačítka *Reset* se vymazává virtuální schránka programu a obsah textových polí pro zápis při vyhledávání. Toto tlačítko se vypustí a jeho funkce se přesune pod tlačítko „Ukončení kontaktu“, čímž dojde k tomu, že až uživatel vystoupí z detailů jednoho zákazníka, dojde k vymazání paměti a obsahu relevantních polí. Uživatel je tedy schopen ihned vyhledávat dalšího zákazníka bez nutnosti stisknout vícekrát tlačítko *Reset*.

**Grafický náhled úpravy** je k dispozici v příloze A.9 – Vyhledávání zákazníka v databázi.

### **Ukázka části kódu:**

```

INITIALIZATION.
CLEAR smp_dyntxt01.
CLEAR smp_dyntxt02.
smp_dyntxt01-icon_id = '@13@'.
smp_dyntxt01-icon_text = text-098.

```

```

smp_dyntxt01-quickinfo = text-099.
sscrfields-functxt_01 = smp_dyntxt01.

smp_dyntxt02-icon_id = '@02@'.
smp_dyntxt02-icon_text = text-097.
smp_dyntxt02-quickinfo = text-096.
sscrfields-functxt_02 = smp_dyntxt02.

AT SELECTION-SCREEN.
  IF sscrfields-ucomm = 'FC02'.
    PERFORM clearing.
  ELSEIF sscrfields-ucomm = 'FC01'.
    PERFORM finding.
  ENDIF.

```

### **Vyhodnocení nutnosti implementace:**

Nezbytné implementovat ihned pro optimální a efektivní využití tohoto systému.

## **3.10 Nelze dočasně změnit status „Uzavřeno“**

### **Popis projevu problémového místa:**

Každý uživatel sleduje své zaevidované podněty. Pokud je veškerá komunikace mezi ním, zákazníkem a případně servisem vyřízena a uživatel si je jist, že další komunikace probíhat nebude, může daný podnět ukončit. To se provádí v záložce detaily změnou statusu z „Přijato“ na „Uzavřeno“. Uživatel bohužel poté nemá možnost tento status změnit. To znamená, že pokud mu například zákazník zašle děkovný e-mail, musí vytvořit novou aktivitu, vložit příchozí e-mail, vyplnit všechna potřebná pole, a poté uzavřít změnou statusu.

### **Popis návrhu úpravy:**

Uživatelsky příjemnější je možnost status změnit, protože nejde o tak vážnou situaci, kdy status nesmí být změněn. Při změně statusu systém zkontroluje hodnotu změnového statusu a v případě, že nalezne hodnotu „0“, zobrazí informační hlášku, že daný podnět je již uzavřen, a zeptá se, zdali chce opravdu přejít ke změně statusu. Po odsouhlasení se změní na „Přijato“. Uživatel doplní potřebné informace a stiskem ikony pro uložení se změní status automaticky zpět na „Uzavřeno“. Poté se změní hodnota změnového statusu na „1“. Pokud systém před zobrazením dotazu zjistí, že hodnota je rovna „1“, uživateli nedovolí následnou změnu a informuje ho. Možnost dodatečného otevření podnětu je možné pouze jednou pro naprosto výjimečné případy.

**Grafický náhled úpravy** je k dispozici v příloze A.10 – Nelze dočasně změnit status „Uzavřeno“.

**Ukázka části kódu:**

```
IF cstatus = 'uzavren'.
  IF cindex = '0'.
    CALL FUNCTION 'POPUP_TO_CONFIRM' "standardní fce pro pop-up okno"
    ELSEIF cindex = '1'.
      Message W000(38) WITH
        'Podnět již nelze otevřít, již byl upravován.'.
      EXIT.
    ENDIF.
  ELSEIF cstatus >< 'uzavren'.
    Message I000(38) WITH 'Podnět je otevřen, není zapotřebí měnit
      status.'.
    EXIT.
  ENDIF.
```

**Vyhodnocení nutnosti implementace:**

Toto řešení není urgentní implementovat ihned.

## 4 Získávání dat pro statistické a ekonomické modelování

[8] Data, která má oddělení péče o zákazníka k dispozici, jsou rozsáhlá a ukrývají v sobě také významné informace pro řízení výroby, prodeje, či při výběru nových smluvních servisů. Pro hledání skrytých souvislostí a netriviálních informací se dnes používají dataminingové metody. Modifikovaná data oddělení péče o zákazníka ŠA jsou použita v návrhu na jejich zpracování dataminingovým nástrojem. Data a příklady budou použity pro připravovaný předmět s touto tematikou.

### 4.1 Úvodem pojem datamining

[9] O dobývání znalostí z databází (zkráceně KDD – *Knowledge Discovery in Databases*) se začalo více mluvit na počátku 90. let minulého století, kdy se v různých elektronických informačních systémech nahromadila velká data, s tímto jevem vznikla i potřeba využívat nashromážděná data ve prospěch firem. O zpracování dat se z počátku začaly zajímat velké, finančně silné společnosti, které tak podpořily vznik nových algoritmů a postupů. Jedna z používaných definic říká: „*jedná se o netriviální proces identifikace pravdivých, dosud neznámých, potenciálně využitelných a naprosto srozumitelných vzorů v datech*“<sup>11</sup>. Datamining hledá v datech takové vztahy, které na první pohled nejsou viditelné a patrné, ale po nalezení pomohou k dosažení nějakého cíle.

Název pro tuto vědní činnost se vyvíjel z různých názvů, a to: informatic harvesting, data archeology, data destillery. Nakonec se ustálil název vyplývající z hornictví, a to dobývání znalostí a dolování dat (datamining). Statistika nabízí doporučení, jak zlepšit procesy, ale dataminingové řešení zahrnuje integraci postupů do firemních rozhodovacích procesů.

#### 4.1.1 Proces dobývání znalostí

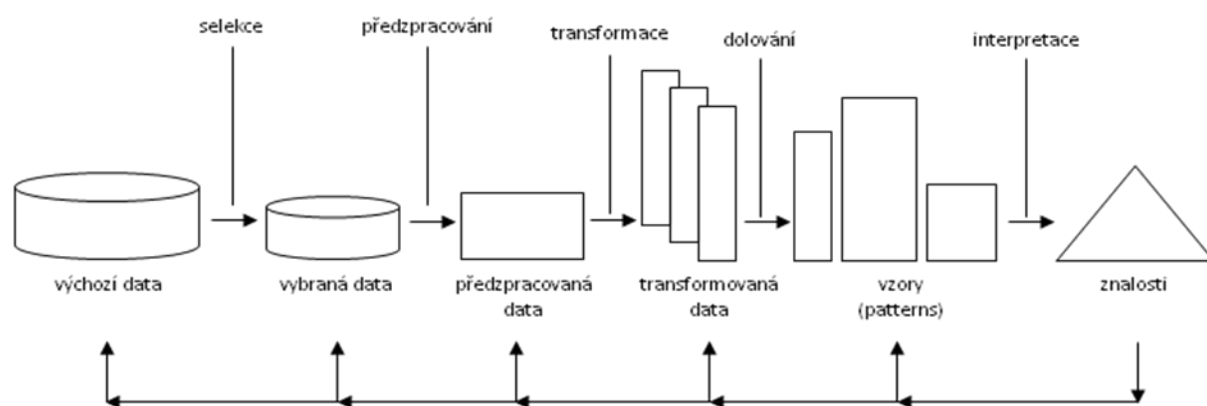
Aby bylo možné začít dolovat data, je nutná existence reálného problému. Mezi takové problémy dnes patří např. hledání určité skupiny zákazníků – segmentu trhu, pro který chystá firma nabídku. Tito zákazníci, s jistou pravděpodobností, budou nabídkou osloveni. Mohou to být klienti banky osloveni speciálním produktem,

---

<sup>11</sup> Definice Fayyad et al.

zákazníci obchodního domu uspořádáním nabízeného zboží, klienti telefonních společností nabídkou vhodných balíčků apod.

Na rozdíl od využití obvyčejného zpracování dat statistickými metodami se v dataminingu kladе velký důraz mimo jiné také na přípravu dat a interpretaci výsledků. Ve většině případů jsou datové zdroje pro dataminingovou úlohu uchovávány ve složité struktuře (datový sklad, apod.) a pro zpracování se vytvoří zpravidla jedna tabulka, která obsahuje potřebné informace. To znamená, že od databázové struktury, která je normalizací optimalizována na minimální redundanci dat, pro dataminingovou úlohu musíme data často složitě přeskupit, a to i přes obnovenou redundanci.



Obr. 14: Jednotlivé fáze procesu dobývání znalostí z databází<sup>12</sup>

Na Obr. 14 v horní části jsou vidět jednotlivé činnosti a jejich souslednost, kterými se datamining realizuje, a to dle popisu v knize od Fayyada. Datová vrstva a její proměny jsou ve spodní části obrázku. Nalezené „vzory“ se stávají znalostmi teprve ve chvíli, kdy se interpretují.

#### 4.1.2 Používané metodiky pro realizaci DM úloh

Dataminingové úlohy jsou manažersky i odborně náročné. Pokud tým řešitelů nemá dostatečnou zkušenost prověřenou praxí, pomáhají si známými a ověřenými metodologiemi. Metodologii chápeme jako pracovní postup nebo plán, který zahrnuje všechny potřebné kroky k úspěšné realizaci projektu. Použití známé metodologie umožní projekt efektivně naplánovat, řídit řešení a srozumitelně předat výsledky uživatelům.

<sup>12</sup> dle knihy Fayyad a kol. 1996

#### 4.1.2.1 Metodika 5A

Mezi doporučenými postupy pro datamining je několik metodik. Jedna z nich se nazývá metodika **5A**. Název je složen z počátečních písmen jednotlivých kroků, které ji popisují:

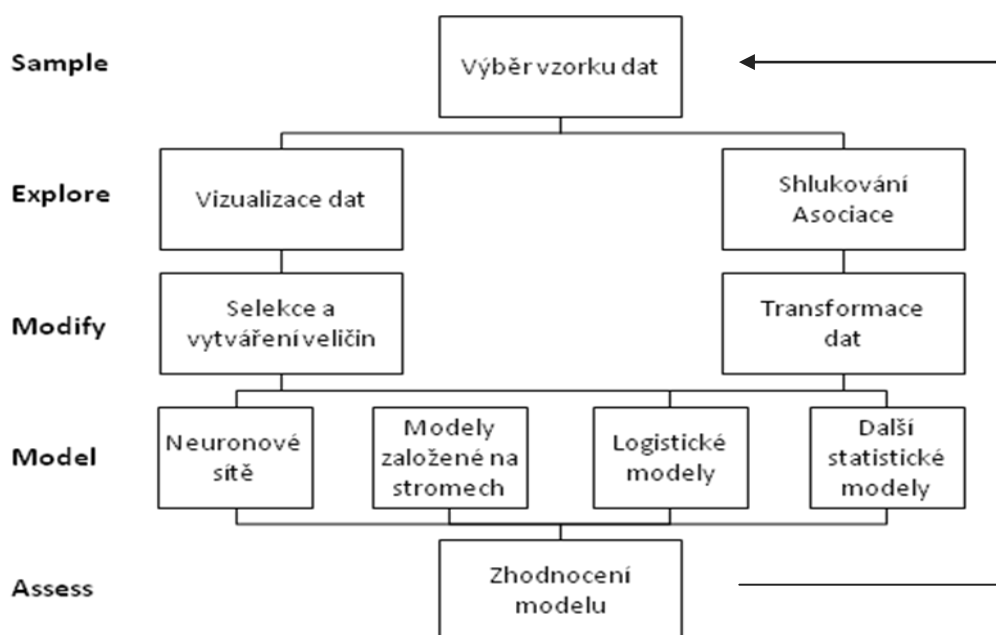
- **Assess** – posouzení, zdali je projekt potřebný.
- **Access** – shromažďování dat.
- **Analyze** – analýza nashromážděných dat.
- **Akt** – přeměna doposud nasbíraných znalostí na funkční znalosti.
- **Automate** – převod výsledků analýzy do praxe.

#### 4.1.2.2 Metodika SEMMA

Další metodikou je **SEMMA**. Název je opět složen z počátečních písmen jednotlivých kroků:

- **Sample** – výběr vhodných dat.
- **Explore** – redukce dat na potřebné.
- **Modify** – různé datové transformace, které se v pozdější době využívají pro analýzy.
- **Model** – analýza dat pomocí neuronových sítí, rozhodovacích stromů a dalších postupů.
- **Assess** – porovnání modelů a přechod k interpretaci (neboli k použitelnému výsledku).





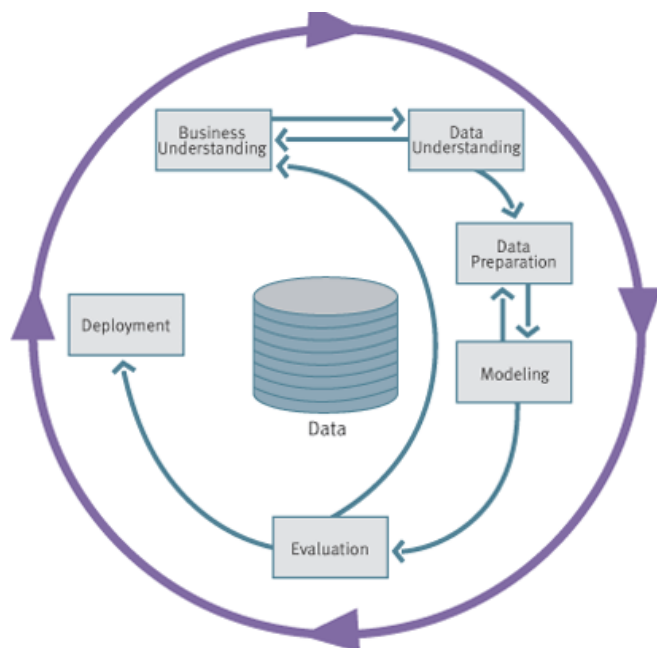
Obr. 15: Metodika SEMMA

#### 4.1.2.3 Metodika CRISP-DM

Třetí metodika je známá jako **CRISP-DM**, jedná se o zkratku slov Cross – Industry Solution Proces for DataMining. Patří mezi nejúspěšnější metodiky a vznikla jako projekt financovaný EU. Její postupy jsou tak veřejně dostupné. Je složena z 6 částí, které jsou velmi úzce spojené. Každá část se rozpadá na několik generických úloh. Generické úlohy popisují jednotlivé akce obecně a poskytují na úloze nezávislý výčet kroků pro realizaci dataminingového projektu. V konkrétním projektu se generické úlohy rozvedou do specializovaných úloh. Obecný postup pak dostává konkrétní podobu plánovaných kroků spojených těsně se zadáním a cíli projektu.

- **Definování cílů** – důležitá část, pokud zadavatel nesdělí projektantovi přesně definované cíle, není projektant schopen data poté adekvátně zpracovat.
- **Porozumění datům – analýza dat** – zde musí projektant porozumět datům, vědět, v jakém jsou formátu, co obsahují jednotlivé tabulky, atd.
- **Příprava dat** – nejdůležitější, ale zároveň nejdelší část celého projektu, jelikož dostupná data jsou například v různých formátech, nemají vždy stejné sloupce nebo některé hodnoty dokonce chybí.
- **Modelování** – výběr vhodného algoritmu(-ů) pro analýzu dat.

- **Hodnocení výsledků** – rozhodování o výsledcích modelování, jejich správnosti nebo nutnosti opakovat některé kroky a jejich kontrola jako prevence opomenutí důležitých faktorů.
- **Implementace modelu** – závěrečná část, která zahrnuje plány na údržbu a kontrolu dat.



Obr. 16: Metodologie CRISP-DM

Tab. 1: Popis generických úloh podle CRISP-DM

Porozumění úloze	Analýza dat	Příprava dat	Modelování	Hodnocení	Uvedení do praxe
<b>Určení obchodního cíle</b>	<b>Přístup k datům</b>	<b>Určení Datová matice</b>	<b>Výběr modelovacích algoritmů</b>	<b>Posouzení kvality modelů</b>	<b>Plánování zavedení řešení</b>
Popis situace	Popis datových zdrojů	Popis datové matice	Vybrané metody	Analýza kvality modelů	Plán zavádění
Obchodní cíle	Popis dat	<b>Výběr dat</b>	Předpoklady metod	Vyhodnocené modely	<b>Sledování a udržování</b>
Kritérium obchodního úspěch	Popis atributů	Důvody pro výběr	<b>Návrh způsobu hodnocení</b>	<b>Rekapitulace projektu</b>	Plán monitorování a údržby
<b>Zhodnocení situace</b>	<b>Analýza</b>	<b>Čištění dat</b>	Evaluační design	Shrnutí dosavadní práce	<b>Příprava závěrečné zprávy</b>

Dostupné zdroje	Analytická zpráva	Zpráva o čištění dat	<b>Budování modelu</b>	<b>Stanovení dalších akcí</b>	Závěrečná zpráva
Požadavky, předpoklady a omezení	<b>Zjištění kvality dat</b>	<b>Vytváření dat</b>	Nastavení parametrů	Návrh dalších kroků	Prezentace projektu
Možná nebezpečí	Zpráva o kvalitě dat	Odvozené atributy	Modely	Vybraný postup	<b>Zhodnocení projektu</b>
Terminologie		Generované záznamy	Popis modelu		Záznam zkušeností
Náklady a zisky		<b>Integrace datových zdrojů</b>	<b>Zhodnocení modelu</b>		Odhad ekonomických dopadů
<b>Určení DM cílů</b>		Spojená data	Interpretace		
Cíle projektu		<b>Formátování</b>	Změny v nastavení parametrů		
Kritéria úspěšnosti		Zformátovaná data			
<b>Plánování projektu</b>					
Časový plán					
Dostupné nástroje					

Dataminingový projekt vytvořením modelu získává rozhodující nástroj pro odvození výsledku pro nasazení DM projektu do praxe. Bez nasazení řešení do praxe projekt nemá smysl. Znalosti získané z dat je třeba vhodným způsobem interpretovat koncovému uživateli, tak aby posloužily definovaným cílům. Forma výstupu je různá, závislá na cílech. Jedná se např. o report dané struktury nebo modifikaci rozhodovacího procesu ve firmě koncového uživatele. V některých projektech je též závěrečná fáze realizována nikoli dataminerem, ale pracovníky IT oddělení nebo systémovým integrátorem.

#### 4.1.3 Metody používané při dolování dat

- **Klasifikační problémy** – metoda funguje na principu rozdělování dat do skupin podle určitých kritérií. Jestliže jsou kritéria předem známá, alespoň pro část dat, vyvine se model, který má výstup ve formě klasifikační

proměnné. Nejčastěji bohužel kritéria předem známá nejsou a cílem klasifikace je jejich nalezení.

- **Regrese** – standardní statistická metoda, pomocí které se popisuje důležitost vstupních proměnných na výstupu. Silnou stránkou je teoretická propracovanost odhadu chyb modelu a schopnost hledat závislosti i na kombinaci vstupních proměnných. Použité regrese je omezeno časovou náročností a pracností při vývoji složitějšího modelu.
- **Nejbližší soused** – v tomto případě jsou třídy reprezentovány svými zástupci. V průběhu klasifikace se poté nový příklad přiřadí do příslušné třídy podle podobnosti.
- **Rozhodovací stromy** – populární díky snadné interpretaci. Popis modelu pomocí rozhodovacího stromu je sled rozhodovacích pravidel, která jsou nejčastěji zobrazena ve formě grafu. Při tvorbě se postupuje metodou „*rozděl a panuj*“ (divide and conquer). Tato metoda postupně rozdělí trénovací data na menší podmnožiny tak, aby v těchto podmnožinách byly nejlépe data jedné třídy.
- **Asociační pravidla** – metodu lze nejlépe popsat na analýze nákupního košíku. Ta se zajímá o produkty, které zákazníci nakupují. Hledá souvislosti v nákupu produktů A a C a jejich vliv na při zakoupení produktů B nebo D. To napomáhá prodejcům vhodně organizovat zboží v prodejnách, případně připravovat výhodné balíčky.
- **bagging & boosting** – metodu lze rozdělit na dvě části:
  - **bagging** – z trénovacích dat se pomocí náhodného výběru vytvoří například  $k$  souborů. Z každého souboru se vytvoří klasifikační strom a výsledný klasifikační les je dán většinovým hlasováním se stejnými vahami.
  - **boosting** – (název vychází s anglického slova boost = zesilovat). Při vytváření klasifikačního stromu je každému prvku dána váha. Všem správně zařazeným prvkům je tato váha snižována, naopak špatně zařazeným je zvyšována. Je důležité zaměřit se na prvky s vysokými vahami.

Kromě těchto metod existuje mnoho dalších, kterými lze řešit problémy při dolování dat, ale cílem diplomové práce není prezentovat úplný přehled dataminingových postupů a principů. Jedním z cílů je hledat možnosti využití dat oddělení POZ.

#### 4.1.4 Využití statistických metod v dataminingu

##### 4.1.4.1 Kontingenční tabulka

[9] Metoda je využívána v oblasti dolování dat. Slouží ke zjištění závislostí mezi dvěma kategoriálními veličinami. Řádky tabulky odpovídají hodnotám jedné veličiny, sloupce druhé. V příslušné buňce je zobrazena hodnota, která odpovídá hodnotě první veličiny v řádku a zároveň druhé veličiny ve sloupci. Příkladem je kontingenční tabulka o 2 řádcích (muži a ženy) a 2 sloupcích (vysokoškolský titul – ano a ne). Tato tabulka 2x2 je nazývána 4polní tabulkou.

Mezisoučty v jednotlivých buňkách (tj. v řádku a v sloupci) nesou informaci o počtu mužů, respektive žen s vysokoškolským titulem, respektive bez titulu. Z tabulky je možné nejen zjistit četnosti kombinací hodnot dvou znaků, ale i otestovat, zdali je mezi oběma znaky vztah. V následující tabulce jsou uvedeny řádkové a sloupcové součty:

$$a + d = g; b + e = h; a + b = c; d + e = f; a + b + d + e = i$$

Tab. 2: Čtyřpolní tabulka – zjištěné četnosti kombinací

	Vysokoškolský titul ANO	Vysokoškolský titul NE	$\Sigma$
Muži	35 (a)	80 (d)	115 (g)
Ženy	29 (b)	58 (e)	87 (h)
$\Sigma$	64 (c)	138 (f)	202 (i)

Při zpracovávání dat v kontingenční tabulce se aplikuje **nulová hypotéza**: *Rozložení znaků se navzájem neovlivňuje*. Za předpokladu **nezávislosti znaků** platí, že například podíl mužů a žen bude stejný jak v celkovém souboru, tak i ve skupině vysokoškolského titulu ANO a vysokoškolského titulu NE, analogicky i podíl vysokoškolský titul ANO a vysokoškolský titul NE ve skupině mužů nebo žen je stejný jako v celém souboru. Pak lze sestavit tabulku odhadnutých hodnot:

Tab. 3: Čtyřpolní tabulka – odhadnuté četnosti kombinací

	Vysokoškolský titul ANO	Vysokoškolský titul NE	$\Sigma$
Muži	36,44	78,56	115
Ženy	27,56	59,44	87
$\Sigma$	64	138	202

Výpočet odhadu hodnoty v  $i$ -tém sloupci a  $j$ -tém řádku lze spočítat například vynásobením součtu hodnot v  $i$ -tém sloupci se součtem hodnot v  $j$ -tém řádku a podílem všech prvků v tabulce. Například pro odhad hodnoty (muži – vysokoškolský titul ANO) vypadá vzorec následovně:

$$n_{11} = \frac{(64 * 115)}{202}$$

Zjištěné četnosti výskytu se obvykle označují  $n_{ij}$ , kde  $i$  je příslušný řádek a  $j$  příslušný sloupec kontingenční tabulky. Četnosti odhadnuté za předpokladu nezávislosti znaků se označují  $m_{ij}$ . Pro testovou statistiku  $\chi^2$  platí:

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{(n_{ij} - m_{ij})^2}{m_{ij}}$$

Pro data z příkladu platí:

$$\chi^2 = \frac{(35 - 36,44)^2}{36,44} + \frac{(80 - 78,56)^2}{78,56} + \frac{(29 - 27,56)^2}{27,56} + \frac{(58 - 59,44)^2}{59,44} = 0,1936$$

Hodnotu je třeba porovnat s kritickou hodnotou  $\chi^2_{\nu}(\alpha)$ . Parametr  $\nu$  (někdy též  $df$ ) se označuje jako **počet stupňů volnosti** a má hlubší matematický význam. V případě kontingenční tabulky se počet stupňů volnosti určí jako **součin počtu řádků zmenšeného o jedna s počtem sloupců zmenšeným o jedna**. Počet stupňů volnosti čtyřpolní tabulky je tedy roven jedné. Parametr  $\alpha$  je požadovanou nejistotou prvního druhu, v biomedicínské statistice se značí jako hladina významnosti  $p$ .

Z pravidla je předem stanovena hladina významnosti  $p$  a podle ní se vyhodnocuje výsledek. Stanoví-li se tedy v modelovém příkladu v rovno 0,05 (hladina významnosti je 5 %) a počet stupňů volnosti je u čtyřpolní tabulky 1, pak platí:

$$\chi_1^2(0.05) = 3.84$$

Testová statistika je nižší než kritická hodnota, nelze tedy zamítnout nulovou hypotézu, tj. hypotézu, že vysokoškolský titul není závislý na pohlaví.

#### 4.1.4.2 Regresní analýza

Statistická metoda slouží pro odhad hodnoty náhodné veličiny (*cílové proměnné, vysvětlované proměnné, regresand nebo závislé proměnné*) na základě znalosti jiné veličiny (*regresor, kovariát nebo nezávislé proměnné*).

Pokud jsou k dispozici vhodná pozorování (určité dvojice hodnot  $x_i, y_i$ ), lze parametry rovnice  $y = q_1x + q_0 + \varepsilon$  odhadnout pomocí metody nejmenších čtverců – tato metoda slouží pro minimalizaci rozdílů mezi očekávanou a pozorovanou hodnotou. Očekávaná hodnota se spočítá na základě funkce

$$f(x) = q_1x + q_0.$$

Příkladem regresní analýzy v praxi je odhadování očekávané délky zatížení mobilní sítě v době Vánoc nebo konce roku. Na základě zkušeností z minulých let, ze kterých se shromáždily údaje o vytiženosti sítě z různých krajů republiky, například velikost města, počet krátkých zpráv a hovorů, u paušálních telefonních čísel i pohlaví majitele čísla apod. (regresory), jakož i záznamy o délce zatížení (regresand), je možné pomocí správně vybraného typu regresní analýzy určit takový vzorec. S jehož pomocí bude možné před Vánoci nebo oslavou Nového roku odhadnout střední dobu délky zatížení. Jsou-li k dispozici i informace o případném výpadku sítě z předešlých let, lze i předem předložit návrhy na eliminaci výpadku sítě. Pokud hledaná závisle proměnná  $Y$  je vektor nebo skalár, je možné její střední hodnotu dohledat pomocí funkce nezávisle proměnných  $X_1, \dots, X_p$ :

$$\mathbb{E}(Y|X_1, \dots, X_p) = f(X_1, \dots, X_p),$$

Znak  $\mathbb{E}$  je symbol označující střední hodnotu, znak „|“ znamená „se znalostí“ a  $f$  je *regresní funkce*, kterou je třeba odhadnout. Nejčastěji se tato funkce předpokládá v obecném tvaru závislém na neznámých *regresních koeficientech*, tyto koeficienty se poté odhadují na základě pozorovaných dat. Nejčastějším případem je lineární regresní funkce, ze které lze vyvodit regresní rovnici (kde  $\beta^j$  jsou regresní koeficienty):

$$\mathbb{E}(Y|X_1, \dots, X_p) = \beta^0 + \sum_{j=1}^p \beta^j X_j$$

Tato regresní rovnice se nazývá lineární regresí. S nelineární regresí se lze setkat například u některých typů neuronových sítí.

V případě, že závisle proměnná  $Y$  je diskrétní, nenabývá střední hodnoty, ale určitých hodnot z množiny. Příkladem je situace, kdy se na základě věku a pohlaví zákazníka odhaduje, jaký nejvhodnější telefonní tarif si vybere *{výhodná volání ve všední dny, výhodná volání o víkendech, zlevněné zprávy, nejvhodnější kombinace zpráv a volání}*. Regresní analýza se v tomto případě nazývá diskriminační analýzou. Pomocí této analýzy se podaří zjistit, zdali zkoumaná veličina patří do jednotlivých tříd ( $p_k$  udává pravděpodobnost, že veličina patří do  $k$ -té třídy):

$$p_k(Y|X_1, \dots, X_p) = f(X_1, \dots, X_p),$$

Typickými metodami, které se používají pro řešení úloh tohoto typu, jsou Pearsonova lineární diskriminační analýza, logistická regrese a metody z nich odvozené.

#### 4.1.4.3 Shluková analýza

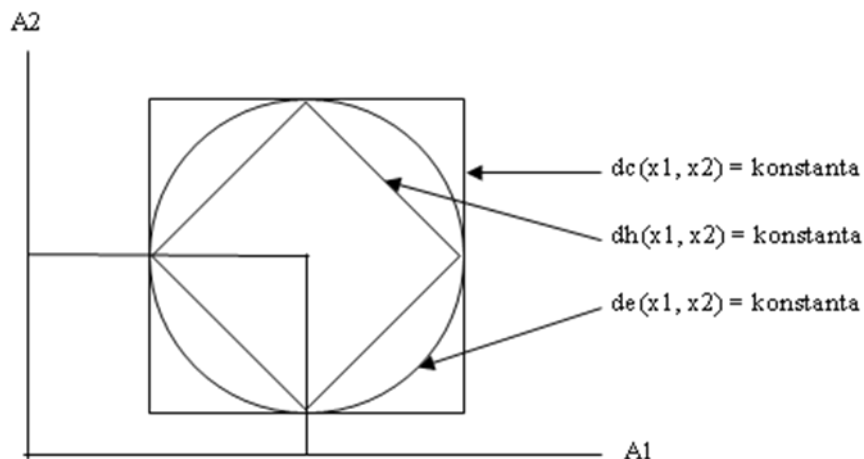
Shluková analýza slouží pro nalezení skupin příkladů, které jsou si určitým způsobem podobné. Důležitou součástí je měření vzdáleností mezi příklady. Každý příklad je vyjádřen  $m$  numerickými veličinami (tj.  $x_1 = [x_{11}, x_{12}, \dots, x_{1m}]$  a  $x_2 = [x_{21}, x_{22}, \dots, x_{2m}]$ ). Pro tento výpočet existují tři postupy:

- Hammingova vzdálenost  $d_H(x_1, x_2) = \sum_{j=1}^m |x_{1j} - x_{2j}|$ ,
- eukleidovská vzdálenost  $d_E(x_1, x_2) = \sqrt{\sum_{j=1}^m (x_{1j} - x_{2j})^2}$ ,



- Čebyševova vzdálenost  $d_C(x_1, x_2) = \max_i |x_{1j} - x_{2j}|$ .

Rozdíly mezi  $d_H(x_1, x_2)$ ,  $d_E(x_1, x_2)$  a  $d_C(x_1, x_2)$  jsou zobrazeny níže. Vzdálenost mezi  $x_1$  a  $x_2$  je pro všechny tři postupy stejně velký, ale výsledek bude jiný. Pro Hammingovu vzdálenost je vyobrazen čtverec „na špičce“, pro eukleidovskou vzdálenost kruh a pro Čebyševovu vzdálenost klasický čtverec.



Obr. 17: Zobrazení pro body se stejnou vzdáleností

Nejznámějšími metodami shlukové analýzy jsou hierarchické shlukování a metoda  $K$ -středů ( $K$ -means clustering).

Hierarchické shlukování má definovaný algoritmus pro postup výpočtů. Využívá se metody směrem „zdola-nahoru“, což znamená shlukování jednotlivých příkladů do větších shluků. Algoritmus je rozdělen do dvou částí. Nejprve se provádí *Inicializace*, během které se určí vzájemné vzdálenosti mezi jednotlivými příklady, a poté se každý příklad zařadí do samostatného shluku. Druhá část, *Hlavní cyklus*, definuje již samotné shlukování. Pokud existuje více než jeden shluk, naleznou se dva, které mají nejmenší vzdálenost, a spojí se do jednoho. Po spojení se spočítají nové vzdálenosti mezi tímto novým a ostatními shluky.

Vzdálenosti mezi shluky lze spočítat pomocí následujících metod:

- metoda nejbližšího souseda (vzdálenost je dána minimem ze vzdálenosti mezi příklady)  $D(\mathcal{U}, \mathcal{V}) = \min_{k,l} d(x_k, x_l)$ ;  $x_k \in \mathcal{U}, x_l \in \mathcal{V}$ ,

- metoda nejvzdálenějšího souseda (vzdálenost je dána maximem ze vzdálenosti mezi příklady)

$$D(\mathcal{U}, \mathcal{V}) = \max_{k,l} d(x_k, x_l); \quad x_k \in \mathcal{U}, x_l \in \mathcal{V},$$

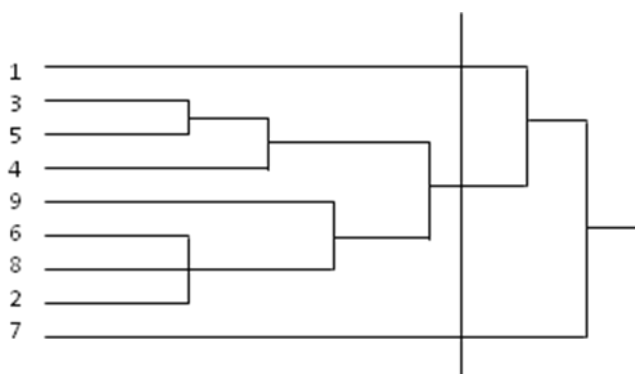
- metoda průměrné vzdálenosti (vzdálenost je dána průměrem ze vzdálenosti mezi příklady)

$$D(\mathcal{U}, \mathcal{V}) = \frac{1}{n_U n_V} \sum_{k=1}^{n_U} \sum_{l=1}^{n_V} d(x_k, x_l); \quad x_k \in \mathcal{U}, x_l \in \mathcal{V},$$

- centroidní metoda (vzdálenost představuje vzdálenosti mezi centroidy<sup>13</sup>)

$$D(\mathcal{U}, \mathcal{V}) = d(\mathbf{u}, \mathbf{v}), \quad \mathbf{u} \text{ je střed shluku } \mathcal{U} \text{ a } \mathbf{v} \text{ je střed shluků } \mathcal{V}$$

Celý proces hierarchického shlukování se graficky znázorní pomocí tzv. dendrogramu, který zobrazuje postupné spojování shluků do menšího počtu. Konečný počet shluků není předem znám, proto se odvodí rozbořem výsledků (tzn., že se dendrogram někde „rozřízne“).



Obr. 18: Dendrogram

Druhá metoda shlukování pomocí  $K$ -středů předpokládá, že konečný počet shluků je předem znám. Konečný počet se během výpočtů nemění, ale mění se pouze rozřazení příkladů do shluků. Tato metoda je méně náročná na výpočet než hierarchické shlukování, a proto je vhodnější pro větší datové soubory. Definovaný algoritmus pro výpočet má následující postup:

1. náhodné zvolení rozkladu do  $K$  shluků,
2. určení centroidů pro všechny shluky v aktuálním rozkladu,
3. pro každý příklad  $\mathbf{x}$  se určí vzdálenost vzdálenosti  $d(\mathbf{x}, \mathbf{c}_k), k = 1, \dots, K$ , kde  $\mathbf{c}_k$  je centroid  $k$ -tého shluku

<sup>13</sup> Centroid – průměr proměnných v jednotlivých shlucích.

4. vypočítaná vzdálenost bude minimem (metoda nejbližšího souseda)  
 $d(x, c_k) = \min_k d(x, c_k),$
5. není-li  $x$  součástí nějakého shluku  $l$  (jehož centroidu  $c_l$  je nejbliže),  
přesune se  $x$  do shluku  $l$ ,
6. pokud došlo k nějakému přesunu, pokračovat bodem 2, jinak konec.

Při použití této metody jsou výsledné shluky reprezentovány centroidy.

## 5 Příklad zpracování dat

Pro praktický příklad byl vybrán soubor s daty, který se vygeneruje ze systému SKS. Tento soubor je ve formátu \*.xls a pro použití definovaných úloh jej bylo nutné nejdříve zpracovat. To zásadní činnost pro dataminingové úlohy. Logická struktura popsaná datovým modelem SKS a tokem dat, je uvedena v **Obr. 19**. Její odvození je výsledkem analýzy dostupných údajů a analýzy činností, které oddělení provozuje. Vytvořený soubor byl upraven například: počestěním názvů sloupců, odstraněním sloupců, které obsahovaly důvěrná data zákazníků a servisů, odstraněním ID vozu ve tvaru VIN vozu (výrobní číslo), a nejsou pro toto zpracování vhodná a použitelná. Upravený soubor obsahuje pro modelování následující data:

- ID vozu – vygenerované číslo pro tento příklad (7mi místné číslo).
- Kód vozu – kombinace čísla a písmene (např. 1U – Octavia 1. generace).
- Model vozu – slovní název model (např. Superb).
- Modelový klíč – kombinace čísel a písmen (např. 5L71D5 označující model Yeti s určitou výbavou).
- Rok závady – YYYY.
- Měsíc závady – MM.
- Počet závad – tj. kolikrát se závada odstraňovala do doby, než se přestala vyskytovat u konkrétního vozu.
- Kód závady – 4 místné číslo (např. 9794 – blok pojistkový).
- Popis závady – slovní popis.
- Datum předání – datum předání vozu zákazníkovi – YYYY-MM-DD.
- Datum přijetí – datum předání vozu do servisu – YYYY-MM-DD.
- ID servisu – 5 místné číslo v kombinaci čísel (např. 27081).
- Náklady na opravu – částka udaná v eurech.

Pro zpracování je použita metodika CRISP-DM z důvodu volné dostupnosti a univerzálnosti. Náhled souboru je k dispozici v příloze B.1 – Náhled zdrojového souboru pro datamining a je součástí přiloženého CD.

### 5.1 Komunikace zákazníka s POZ

Vyskytují se případy, kdy se zákazník obrací na oddělení péče o zákazníka. Základními případy jsou podněty, kdy se zákazník dotazuje na například

na kompatibilitnost náhradního dílu či příslušenství ke svému vozidlu. Dalším typem podnětů jsou technická vylepšení či pochvaly na pracovníky oddělení, kteří ochotně pomohli zákazníkovi s vyřešením jeho problému. Důležitými typy podnětů, díky kterým bylo možno použít zdrojový soubor pro příklad, jsou takové, kdy se zákazník obrací na POZ s žádostí o pomoc při řešení problému s vozidlem. To nastává při nespokojenosti s vyjádřením servisu před nebo po opravě vozidla. Jedná-li se o opakovanou závadu nebo nemá-li dobré zkušenosti se servisem, kontaktuje zákazník oddělení před návštěvou servisu. Pokud se obrátí zákazník na POZ s prosbou o pomoc, uživatel se pokusí získat od zákazníka co nejvíce informací o vozidle a poruše či problému. Na základě těchto informací uživatel zkontaktuje vedoucího servisu a zjistí jeho pohled na daný problém<sup>14</sup>. Po zkontaktování vytvoří odpověď zákazníkovi s informacemi o předání problému vedoucímu daného servisu. Když uživatel vyřizuje podnět, kdy musel kontaktovat servis, dostává zpětné informace o průběhu řešení právě od něj, případně i od zákazníka.

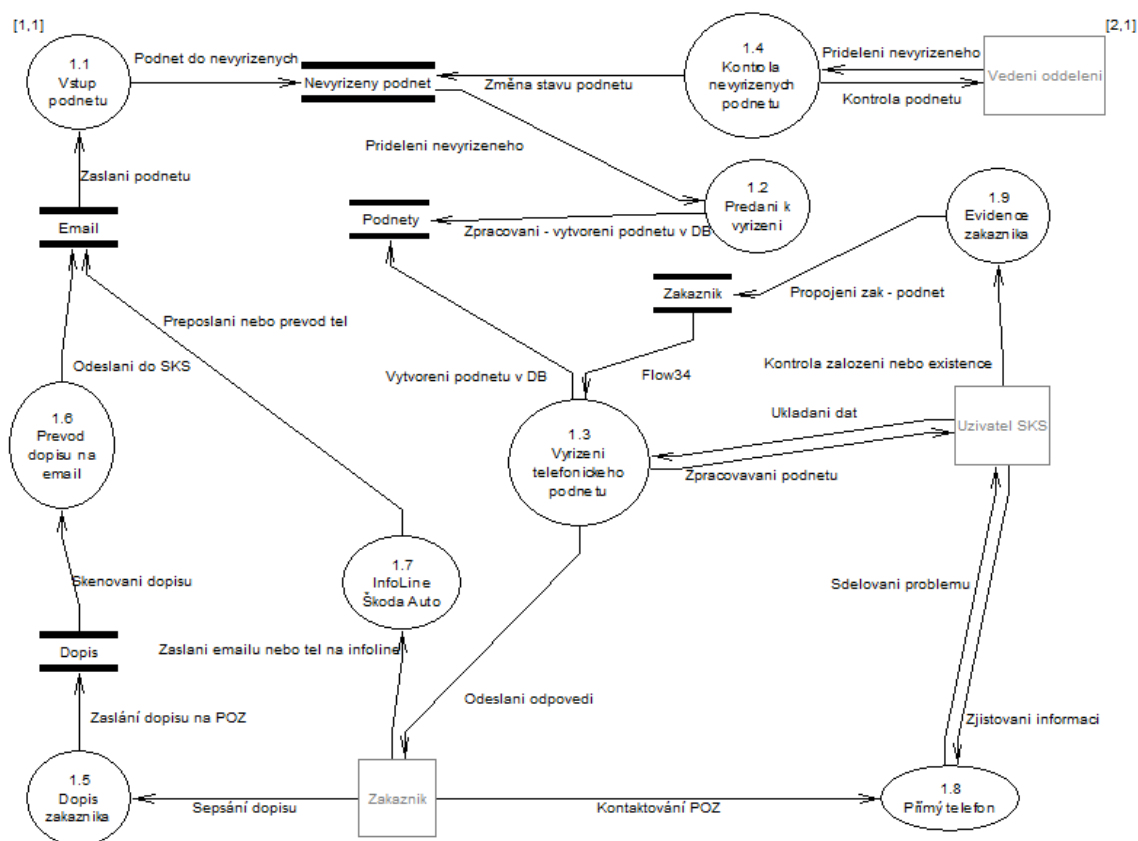
Pokud je zákazník spokojený se servisem, který navštěvuje, není povinen před nebo po opravě kontaktovat POZ s informacemi o průběhu, **proto data získaná ze systému SKS neobsahují všechny servisní zásahy uskutečněné v daném období, ale pouze takové, které jsou popsány výše.**

Aby zdrojový soubor a následné výsledky získané pomocí programu PAWS Modeler byly 100% relevantní, musí se soubor rozšířit o data ve stejné struktuře získaná z jiné aplikace, která obsahuje všechny servisní zásahy. Bohužel systém SKS neobsahuje propojení s touto aplikací a uživatelé oddělení POZ k této aplikaci nemají přístup, protože je v kompetenci jiného oddělení.

V tomto případě budeme pracovat se zdrojovým souborem tak, jako by obsahoval všechny servisní zásahy.

---

<sup>14</sup> Jelikož všechny servisní místa ŠA jsou pouze smluvními partnery společnosti, uživatel nemá právo nařizovat vedoucímu servisu, jak má postupovat nebo jakou případnou slevu musí nabídnout zákazníkovi. Společnost ŠA působí v tomto vztahu jako technicko-poradenský partner.



Obr. 19: Popis datových toků znázorněný pomocí DFD

## 5.2 Definování cílů

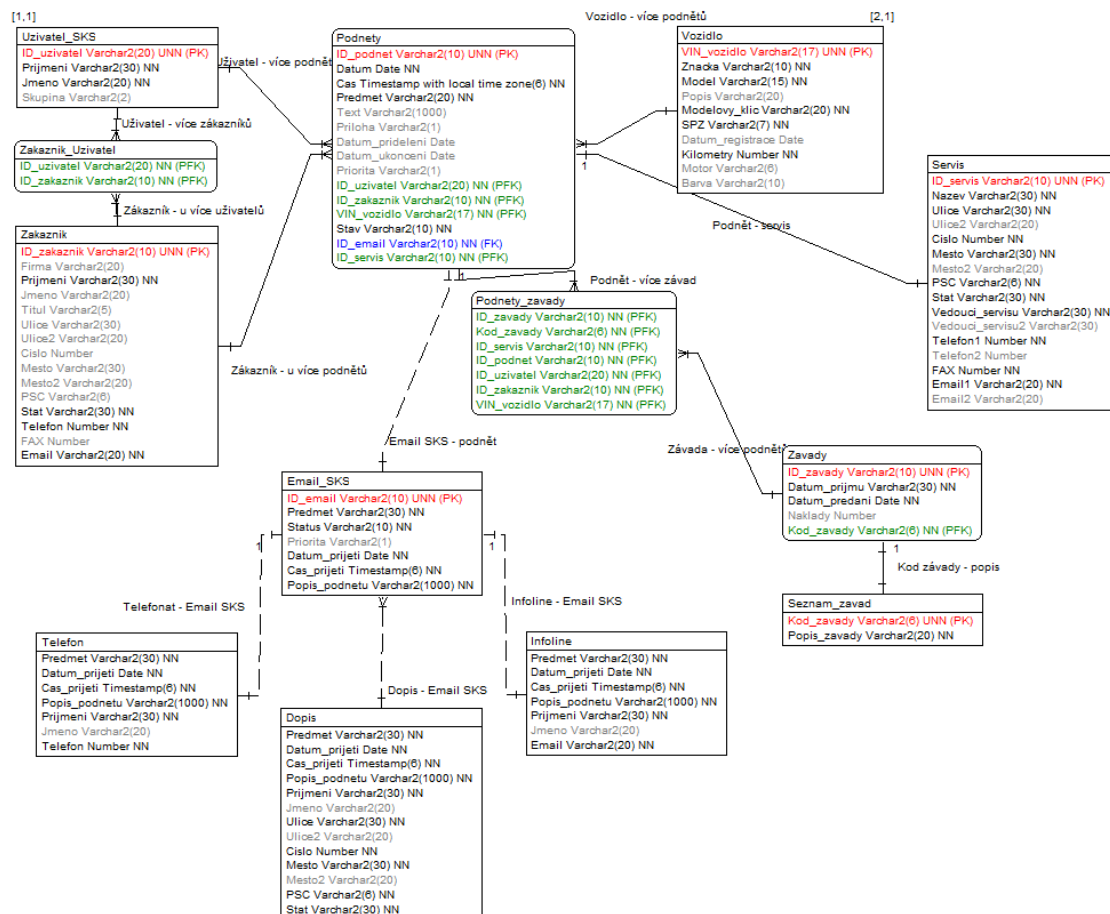
Dostupná data poskytují informace, které se definují v cílech pro možné nasazení dataminingových nebo databázových technik. Z výše uvedeného souboru se zjistí:

- který modelový klíč se nejvíce vyskytuje v souboru, tudíž který model vozu (modelový klíč) je nejvíce poruchový,
- jaká je průměrná cena opravy na model vozu (modelový klíč),
- ve kterém měsíci bylo evidováno nejvíce závad,
- která závada se nejčastěji vyskytovala,
- jaké byly výdaje na nejčastější závadu u různých servisů,
- jak dlouho trvala oprava nejčastější závady u různých servisů,
- který servis (tj. ID servisu) řešil kolik závad.

Ze souboru lze pochopitelně získat i další informační vazby.

## 5.3 Popis datové struktury systému SKS

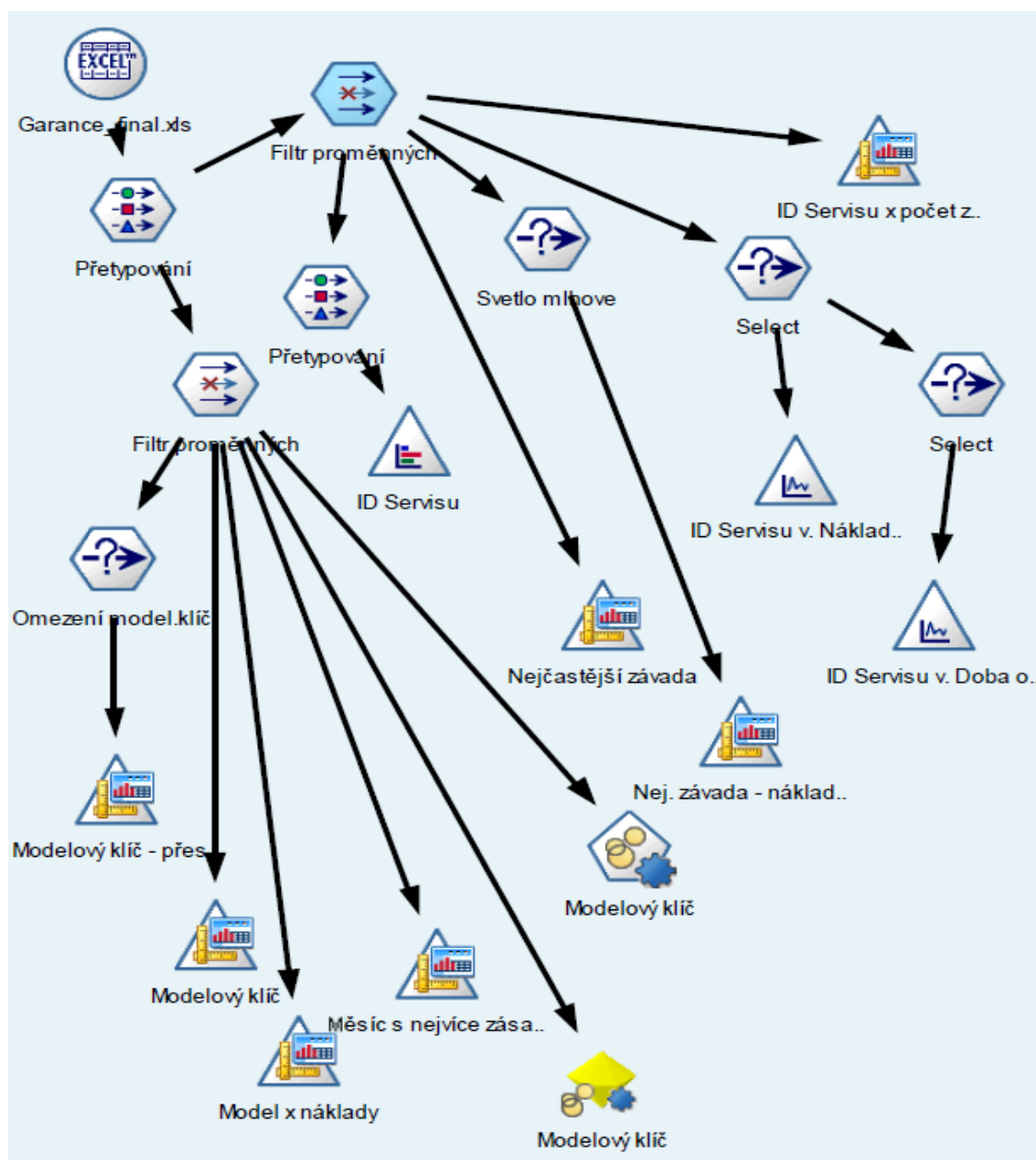
Datová struktura systému SKS je tvořena několika entitami a vazbami mezi nimi. Mapováním do relací pro relační databázi pak vznikají tabulky *Uživatel SKS*, *Zákazník*, *Podněty*, *Závada*, *Servis*, *Email\_SKS*, *Vozidlo*. Vedlejší tabulky jsou *Seznam závad*, *Telefon*, *Dopis*, *Infolinka*.



Obr. 20: Grafické zobrazení datové struktury pomocí ERD

## 5.4 Příklady

Příklady jsou realizovány dataminingovým nástrojem PAWS Modeler. Odpovědi lze získat přímo z databáze, ale návazné použití dat pro řízení a predikci, databázové enginy poskytují jen sporadicky. Zpracování dat pomocí modeleru má podobu streamu, který je složený z několika bloků. Jeho části jsou v logických blocích, které představují jednotlivé fáze CRISP-DM. Modularita a flexibilita práce se streamem poskytuje velmi dobrý nástroj pro práci s velkými datovými množinami. Další použití PASW Modeleru je uvedeno v příloze.

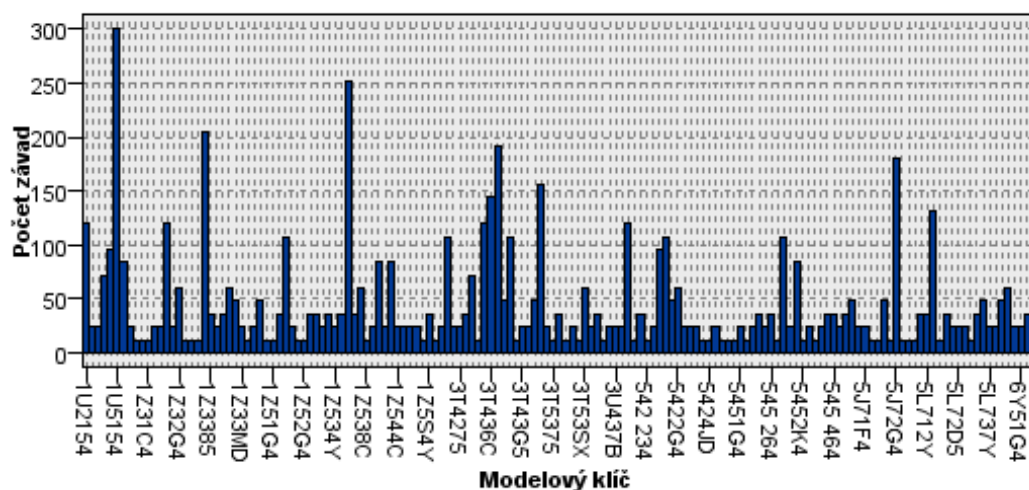


Obr. 21: Stream vytvořený v dataminingovém nástroji PAWS Modeler

#### 5.4.1 Který model vozu (modelový klíč) je nejvíce poruchový

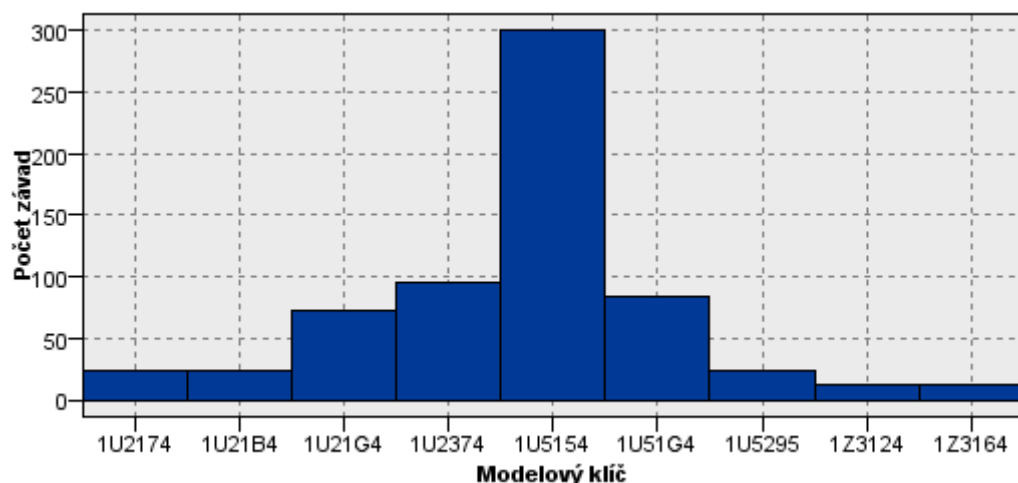
Na základě grafického výstupu je zřejmé, že nejvíce závad je evidováno na modelovém klíči (1U2154 až 1Z31C4).





Obr. 22: Celkový přehled modelový klíč x četnost závad (řazeno vzestupně)

Detail okolí maxima ukazuje, který modelový klíč je nejporuchovější – **1U5154**.



Obr. 23: Detailní zobrazení modelový klíč x četnost závad

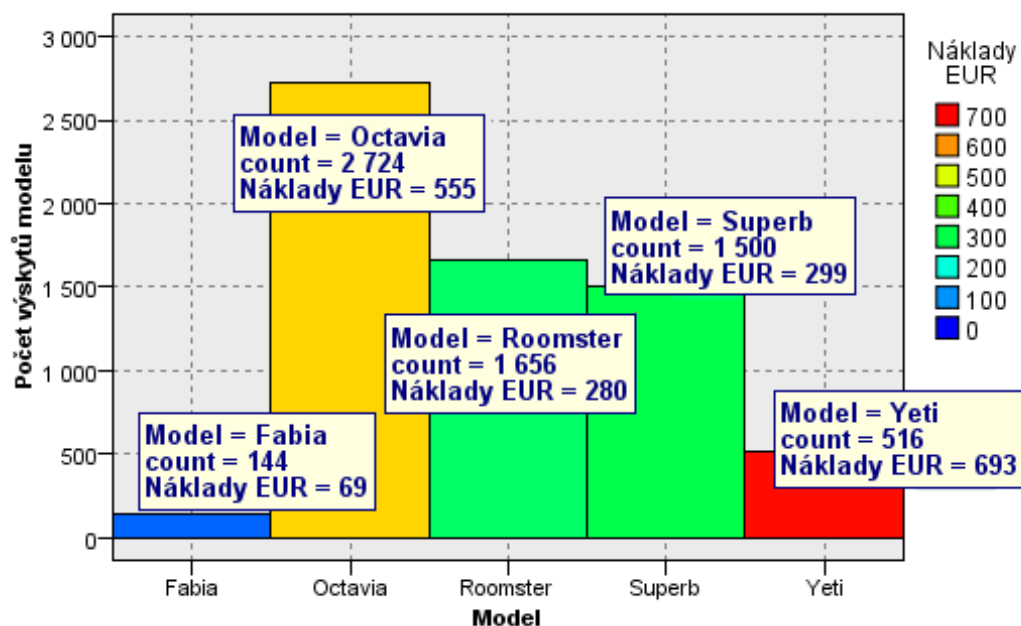
Tab. 4: Deset nejporuchovějších modelů (modelových klíčů)

Pořadí	Modelový klíč	Počet závad	Pořadí	Modelový klíč	Počet závad
1.	1U5154 (Octavia)	300	6.	3T43SX (Superb)	156
2.	1Z5364 (Octavia)	252	7.	3T436C (Superb)	144
3.	1Z3364 (Octavia)	204	8.	5L71D5 (Yeti)	132
4.	3T4375 (Superb)	192	9.	1U2154 (Octavia)	120
5.	5J72G4 (Roomster)	180	10.	5421B4 (Roomster)	120

### 5.4.2 Průměrná cena opravy na model vozu

Pomocí streamu z dataminingového nástroje se získá grafický výstup, kde lze vidět přehled všech pěti modelů vozů Škoda s informacemi o počtu oprav a průměrné ceně za opravu během určitého období.

Z výstupu je zřejmé, že nejporuchovější je model Octavia s 2724 opravami, ale model s průměrně nejnákladnějšími opravami je model Yeti.

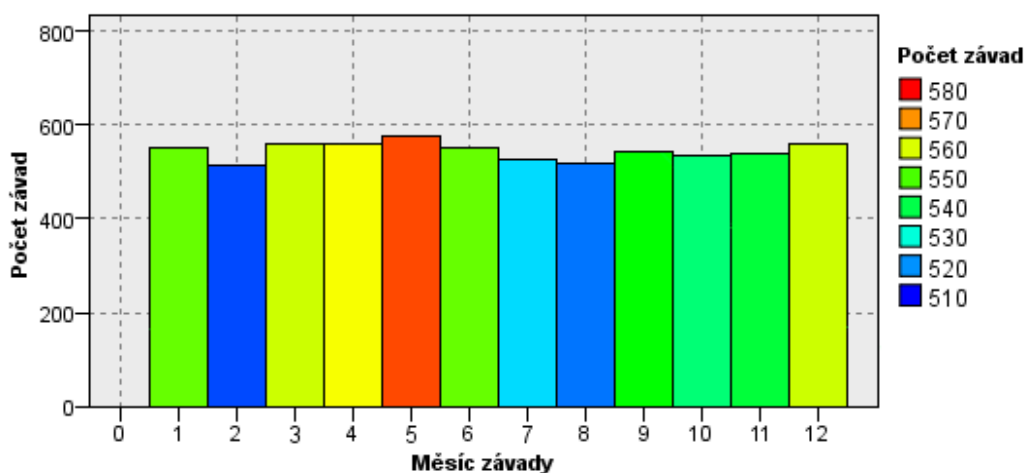


Obr. 24: Přehled modelů × průměrné náklady na opravu

### 5.4.3 Ve kterém měsíci je evidováno nejvíce závad

Pro získání odpovědi na tuto otázku jsou na ose x jednotlivé měsíce vyskytující se ve zdrojovém souboru a na ose y počet závad, které jsou v měsících evidovány.

Z grafického výstupu lze sledovat, že 5. měsíc (tj. květen) byl měsícem s největším počtem servisních zásahů.

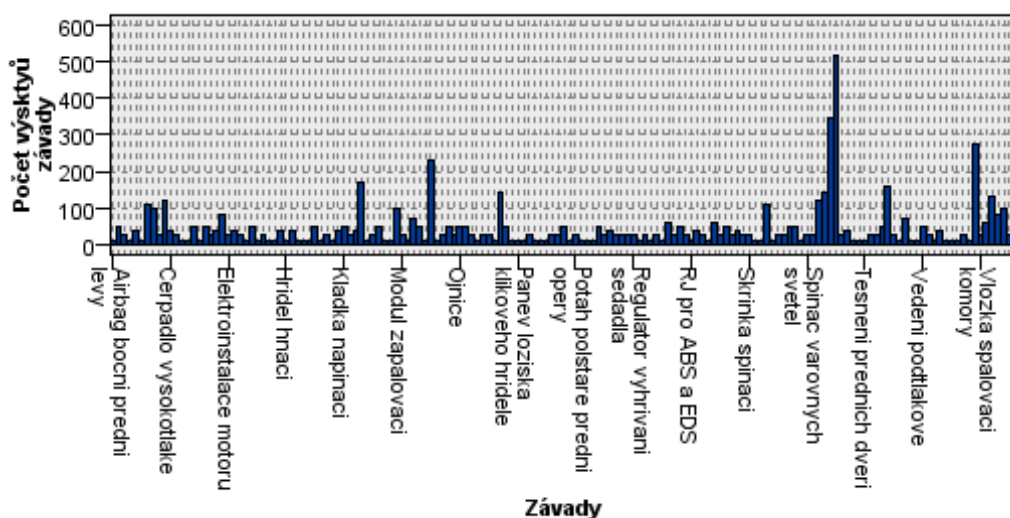


Obr. 25: Přehled měsíců × počet servisních zásahů

Pro využití výsledků této analýzy pro řízení podniku je nutné propojení s daty o prodeji vozidel. Bohužel systém SKS tato data neobsahuje. Jejich dostupnost umožní zjistit, zda vozidla vyrobená v určitém měsíci jsou více poruchová po určité době provozu nebo vozidla vyrobená v jiném. Je vhodné mít dostupná data i o době mezi prodejem, první opravou vozu a počtem najetých kilometrů pro předcházení větším problémům na cestách. Tato prevence by se provedla v rámci pravidelných servisních kontrol.

#### 5.4.4 Která závada se vyskytuje nejčastěji

Zpracování této otázky je identické s prvním případem. Z výsledného výstupu je čitelné, že nejčastější závada je v rozmezí *Spínač varovných světel* až *Těsnění předních dveří*. Závady jsou řazeny vzestupně podle abecedy.



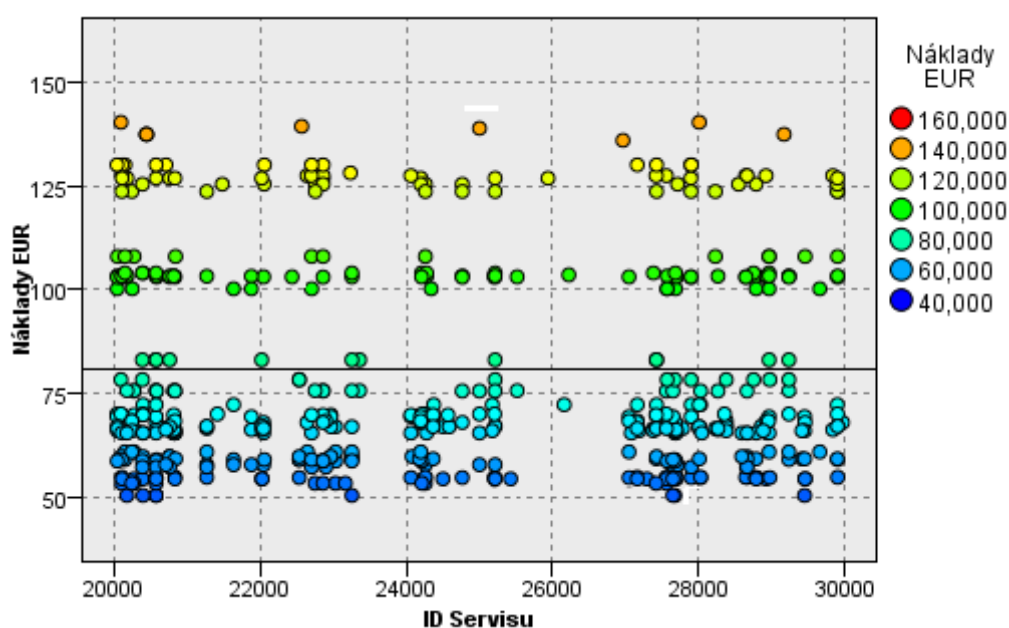
Obr. 26: Přehled všech oprav × počet jejich výskytů

Tab. 5: Deset nejčastějších poruch

Pořadí	Modelový klíč	Počet závad	Pořadí	Modelový klíč	Počet závad
1.	Světlo mlhové	516	6.	Turbodmychadlo	156
2.	Světlo met	348	7.	Výbojka	132
3.	Ventil vstřikovací	276	8.	Čerpadlo paliva vysokotlaké	120
4.	Napínák	228	9.	Snímač parkování	108
5.	Kotouč brzdy přední	168	10.	Blok válců	108

#### 5.4.5 Jaké jsou výdaje na nejčastější závadu u různých servisů

Z předchozího výstupu je známa nejčastější závada. Ke zjištění, zdali nějaký servis účtuje větší částky za tuto opravu než ostatní servisy, je nutné porovnat ID Servisu s cenami za tuto opravu.



Obr. 27: Náklady na nejběžnější závadu u různých servisů

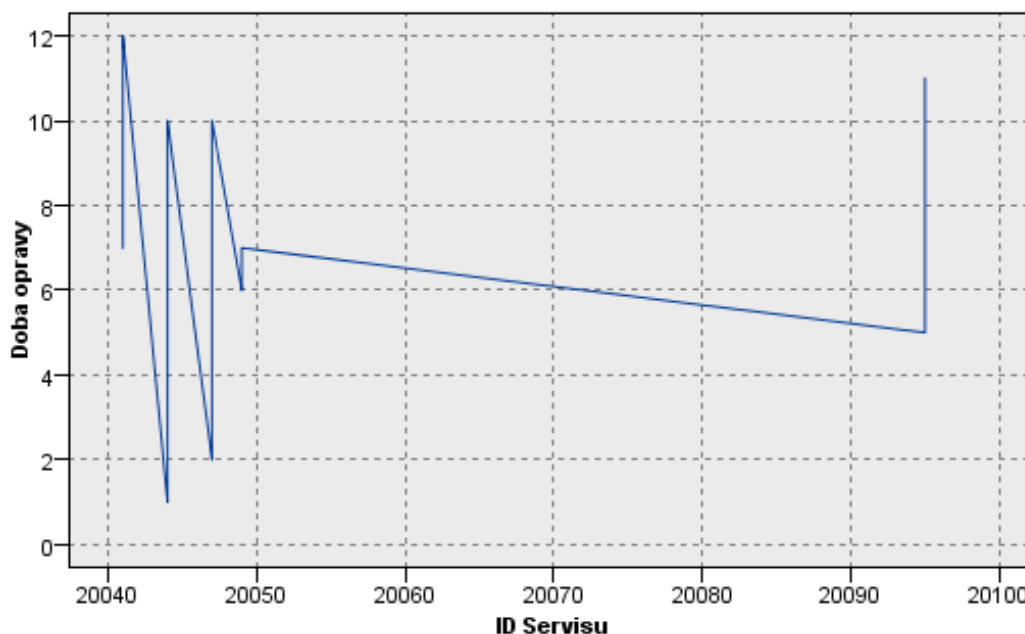
Na grafickém výstupu je znázorněna průměrná cena nejčastější závady (79,79EUR – Světlo mlhové) pomocí tenké čáry. Lze pozorovat, že některé servisy opravu nadceňují, jiné podceňují. Změna ceny oproti průměru je zapříčiněna individuální slevou pro zákazníka nebo naopak delší dobou opravy či potřebou vyměnit důležité součástky spojené se závadou.

Zobrazení v Obr. 27 je nečitelné, protože počet zobrazovaných servisů na ose x je příliš velký, data splývají a není možné je správně rozlišit. V příloženém streamu pro

PASW Modeler lze zobrazit detaily a „čist“ data podle potřeby. Stream je přílohou této práce.

#### 5.4.6 Trvání opravy nejčastější závady u různých servisů

Ve zdrojovém souboru jsou dostupné informace o příjmu vozidla do servisu, předání zpět zákazníkovi a době strávené v servise (vypočteno na celé dny). Pomocí těchto informací lze zjistit, jaká byla průměrná doba opravy u nejčastější závady.

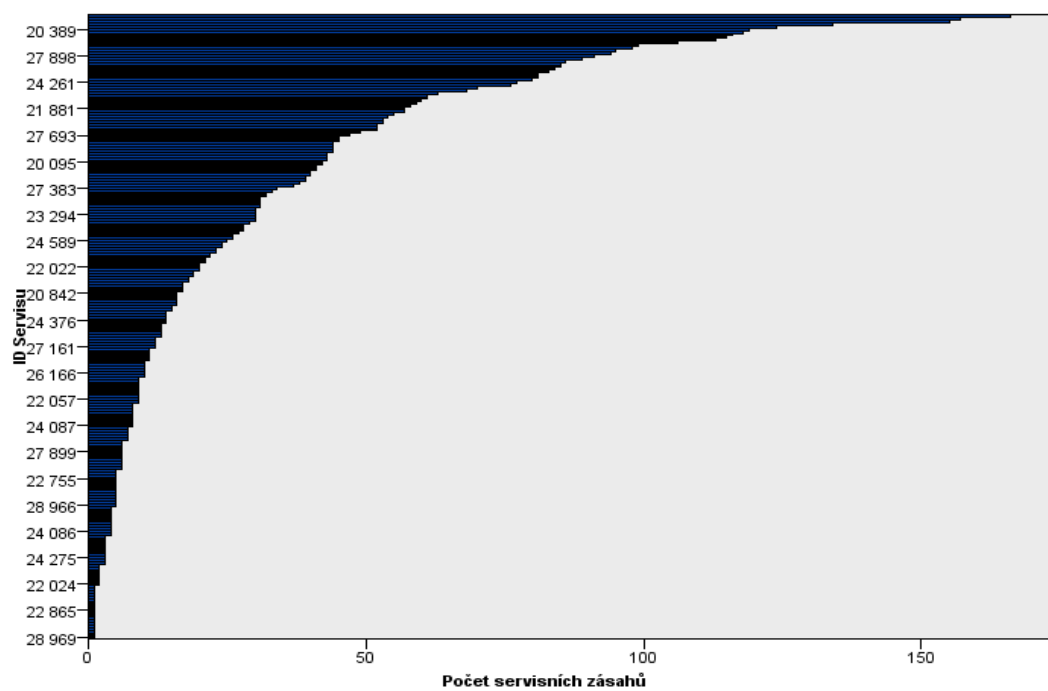


Obr. 28: Zobrazení doby opravy u nejčastější závady u různých servisů

U tohoto grafického výstupu je zobrazena vybraná část zpracovaných dat (20040 až 20100). Všechna data i výsledek této analýzy je uveden v streamu v příloze.

#### 5.4.7 Který servis (tj. ID servisu) řeší kolik závad

Na základě grafického výstupu se zjistilo, kolik každý servis evidoval přibližně servisních zásahů. Část detailního seznamu je k dispozici k nahlédnutí v příloze B.2 – Seznam servisů seřazených dle četnosti zásahů. Kompletní je k dispozici na přiloženém CD.



Obr. 29: Přehled servisů × počet servisních zásahů

## Závěr

Je třeba mít na mysli, že právě zákazníci jsou ti, kteří rozhodují o úspěšnosti a výši zisku podniku. Čím vyšší je úspěšnost prodeje, tím roste i riziko reklamací a stížností ze stran zákazníků. Automobily jsou velmi složitá zařízení a je velká pravděpodobnost, že porucha nastane. V tu chvíli je vedle autorizovaného servisu ŠKODA i oddělení péče o zákazníka, které přijímá stížnosti, reklamace, dotazy i pochvaly ze strany zákazníků. Jelikož je těchto podnětů mnoho a uživatelé na oddělení s nimi pracují po celou pracovní dobu, je důležité, aby jejich správa byla bezchybná, pohodlná a vždy dostupná.

V úvodu jsem stanovil hlavní i dílčí cíle, které jsem splnil a všechna řešení popsal v příslušných kapitolách této práce.

V první části diplomové práce jsem popsal společnost Škoda Auto pomocí stručných informací. Krom této společnosti jsem popsal i společnost SAP AG, která IS SAP vynalezla, a který je provozován ve ŠKODA AUTO a.s. Dále je možné se zde dočíst, které moduly a systémy jsou používány ve společnosti a konkrétně v oblasti spojených s péčí o zákazníka.

Ve druhé kapitole jsem nejprve popsal a přiblížil čtenáři činnost a strukturu oddělení péče o zákazníka. Poté jsem se již zaměřil na analýzu systému Small KUBA Solution, zkráceně SKS, který je denně využíván na tomto oddělení. Postupně jsem identifikoval 10 problémových míst, která v další kapitole podrobněji popsal a pomocí programovacího jazyka upravil.

Třetí kapitola je praktickou částí a strávil jsem nad ní nejvíce času. Odstranění všech předem nalezených problémových míst předcházelo seznámení se s programovacím jazykem ABAP/4, který byl vynalezen pro účely společnosti SAP a je jím naprogramován celý systém. U každého problémového místa je napsáno jak teoretické řešení, tak i praktické ve zdrojovém kódu a je zde i informace o nutnosti implementace do systému. Zdrojový kód je pouze částí celého řešení u každého problému. Celé kódy včetně grafického náhledu řešení je možné nalézt v příloze této práce.

V předposlední části jsem přiblížil aktuálnost využití získaných rozsáhlých dat pro potřeby řízení společnosti. Nejprve jsem stručně popsal proces dobývání znalostí z databáze, dále nejznámější používané metodiky a poté i některé metody

pro zpracování dat. Datamining je velmi rozsáhlé téma, o kterém Škoda Auto teprve uvažuje a nejvhodnější by bylo zpracovat toto téma ve vlastní bakalářské či diplomové práci.

Závěrečná část obsahuje některá možná zpracování dat SKS systému pomocí dataminingového nástroje PAWS Modeler. Důležitá část celého procesu byla úprava souboru dat, aby bylo možné tento soubor použít i mimo areál závodu ŠKODA AUTO a.s. jelikož obsahoval citlivá data. Tato data byla buď vymazána, pokud nebyla nutná pro další zpracování, nebo nahrazena uměle vygenerovanými daty, podobnými datům skutečným. Poté jsem stanovil cíle, kterých jsem docílil pomocí softwarového nástroje pro datamining. Dále jsem čtenáři přiblížil komunikaci mezi zákazníky a oddělení péče o zákazníka, aby bylo jasné, jak jsou data získávána. V poslední části této kapitoly jsem zpracoval zdrojový soubor a docílil tak odpovědí na výše uvedené stanovené cíle. Zdrojový soubor bohužel neobsahoval všechna data ze všech servisů z celé republiky a to z důvodů, které jsou popsány v kapitole [5.1]. Pro zpracování v této práci jsem ale pracoval se souborem tak, jako by obsahoval veškerá data za určité období i z jiných částí informačního systému podniku. Pokud by společnost ŠKODA AUTO a.s. chtěla využití těchto a dalších metod využít, musela by rozšířit soubor získaný z SKS o data, která jsou dostupná pouze v jiné aplikaci, kam nemá systém SKS přístup. Nesporně by ale aplikací těchto technologií mohla získat cenné informace pro řízení výroby a servisu záručních oprav.



## Seznam použité literatury

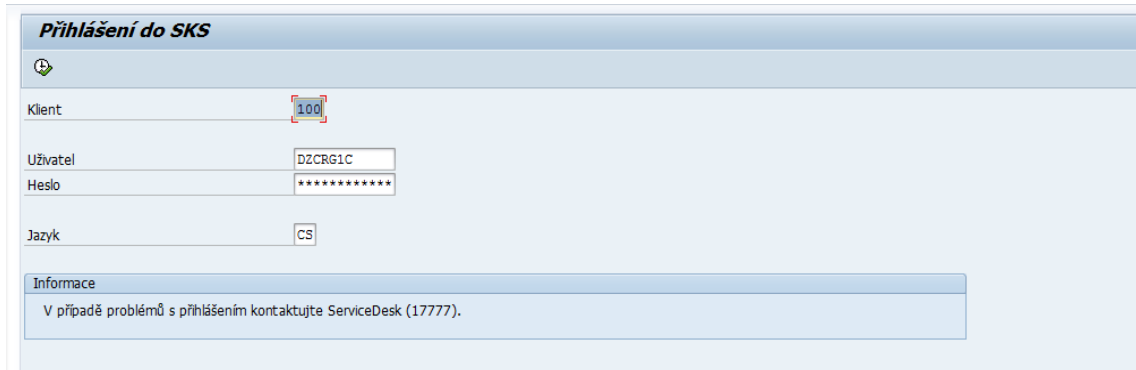
- [1] KRÁLÍK, J. Škoda Auto: *100 let historie automobilů*. 1. vydání. 2005. ISBN není.
- [2] KŘIVÁK, V. *Návrh a realizace komplexního monitorovacího systému pro podnikový IS na bázi SAP*. Praha, 2008. Diplomová práce. ČVUT. URL: <[https://dip.felk.cvut.cz/browse/pdfcache/krivav1\\_2008dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/krivav1_2008dipl.pdf)>
- [3] MAASSE, A., GADATSCH, A., FRICK, D., SCHOENEN, M. *SAP R/3: Kompletní průvodce*. 1. vydání. 2007. ISBN 80–251-1750-2.
- [4] MATZKE, B. *ABAP/4: Programming the SAP R/3 System*. 2. vydání. 2000. ISBN 978–201675153.
- [5] KÜHNHAUSER, K. – H. *ABAP: Výukový kurz*. 1. vydání. 2009. ISBN 978-80-251-2117-7.
- [6] MCCONNELL, S. *Dokonalý kód*. 1. vydání. 2006. ISBN 80-251-0849-X.
- [7] HENDL, J. *Přehled statistických metod zpracování dat*. 3. vydání. 2006. ISBN 80-7178-820-1.
- [8] RUD, O. R. *Data Mining: praktický průvodce dolováním dat pro efektivní prodej, cílený marketing a podporu zákazníků (CRM)*. 1. vydání. 2001. ISBN 80-7226-577-6.
- [9] BERKA, Petr. *Dobývání znalostí z databází*. 1. Vydání. 2003. ISBN 80-200-1062-9.
- [10] PLCHÚT, Martin. *Dobývání znalostí z databází* [online]. Publikováno 9. 2. 2004. URL: <<http://www.fit.vutbr.cz/study/courses/ZZD/public/seminar0304/Uvod.pdf>>.
- [11] BERKA, Petr. *Statistika* [online]. Publikováno 13. 10. 2009. URL: <[http://sorry.vse.cz/~berka/docs/izi456/kap\\_3.pdf](http://sorry.vse.cz/~berka/docs/izi456/kap_3.pdf)>.
- [12] Tutoriály a firemní dokumentace SAP a ŠKODA AUTO a.s.
- [13] Tutoriály a firemní dokumentace PASW Modeler, společnost Predictive Analytics Software (PASW).

## **Přílohy**

## Příloha A

### A.1 - Problémy s přihlášením

#### Grafický náhled úpravy



Obr. 30: Přihlašovací obrazovka (SAP standard)



Obr. 31: Akce pro přihlášení (úprava)

#### Zdrojový kód

TYPE-POOLS: slis.

DATA: text TYPE LVC\_TITLE,  
ODPOVED TYPE c.

SELECTION-SCREEN BEGIN OF BLOCK b0.

PARAMETERS: client(3) TYPE n OBLIGATORY DEFAULT '100'.

SELECTION-SCREEN SKIP.

PARAMETERS: user TYPE sy-uname OBLIGATORY,  
P\_PASS(12).

SELECTION-SCREEN SKIP.

PARAMETERS: jazyk(2) TYPE c.

SELECTION-SCREEN END OF BLOCK b0.

SELECTION-SCREEN SKIP.

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-010.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT (79) text-011.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN END OF BLOCK b1.

AT SELECTION-SCREEN OUTPUT.

LOOP AT SCREEN.

IF SCREEN-NAME = 'P\_PASS'.

SCREEN-INVISIBLE = '1'.

MODIFY SCREEN.

ENDIF.

ENDLOOP.

START-OF-SELECTION.

*\* definování textu pro pop-up okno*

IF user = sy-uname.

clear text.

CONCATENATE 'Uživatel' SY-UNAME 'je již přihlášen. Zvolte další krok:' INTO  
text SEPARATED BY space.

PERFORM pop\_up.

ELSE.

Write: / text-012.

ENDIF.

*\* zobrazení pop up okna a skupiny radiobuttonů*

FORM pop\_up.

CALL FUNCTION 'POPUP\_TO\_CONFIRM'

EXPORTING

titlebar = text-001

text\_question = text

text\_button\_1 = text-002

text\_button\_2 = text-003

default\_button = '1'

display\_cancel\_button = 'X'

userdefined\_f1\_help = SPACE

start\_column = 15

start\_row = 4

IMPORTING

answer = ODPOVED.

*\* po stisku tlačítka*

IF ODPOVED = '1'. *"stisknutí Nahrazení*

PERFORM nahradit.

ELSEIF ODPOVED = '2'. *"stisknutí Paralérní*

PERFORM paraler.

ENDIF.

ENDFORM.

*\* zrušení stávajícího přihlášení a nahradit ho novým*

FORM nahradit.

CALL 'SYST\_LOGOFF'.

ENDFORM.

*\* paralérní přihlášení*

FORM paraler.

```
DATA : lv_skscr(1) TYPE C VALUE 'X',  
      lv_vbeln LIKE LIKP-VBELN,  
      l_st_prscr TYPE tpara,  
      l_it_scrprm TYPE TABLE OF tpara.
```

```
CLEAR l_st_prscr.  
CLEAR l_it_scrprm[].
```

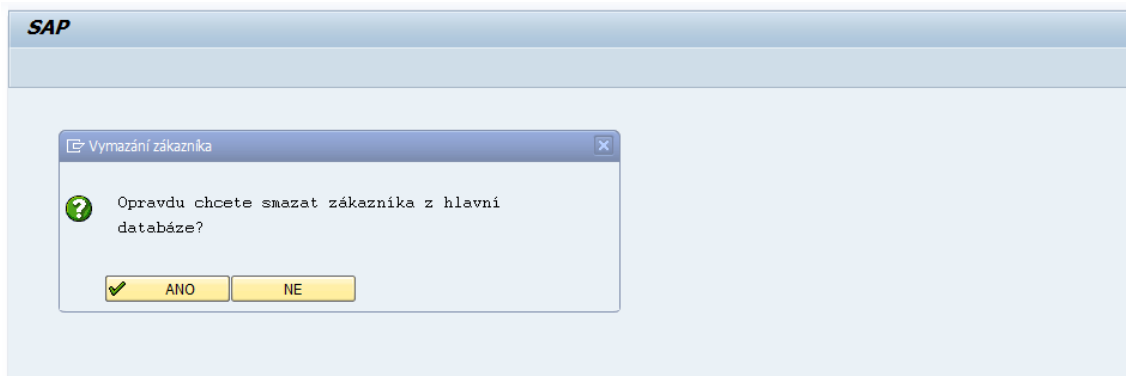
```
l_st_prscr-paramid = 'VL'.  
l_st_prscr-partext = lv_vbeln.  
APPEND l_st_prscr TO l_it_scrprm.
```

```
CALL FUNCTION 'CC_CALL_TRANSACTION_NEW_TASK'  
EXPORTING  
  TRANSACTION      = sy-uname  
  SKIP_FIRST_SCREEN = 'X'  
TABLES  
  PARAMTAB         = l_it_scrprm  
EXCEPTIONS  
  communication_failure = 97  
  system_failure = 98  
  OTHERS = 99.  
IF sy-subrc = 97. " Komunikační chyba  
  EXIT.  
ELSEIF sy-subrc = 98. " Systémová chyba  
  EXIT.  
ENDIF.
```

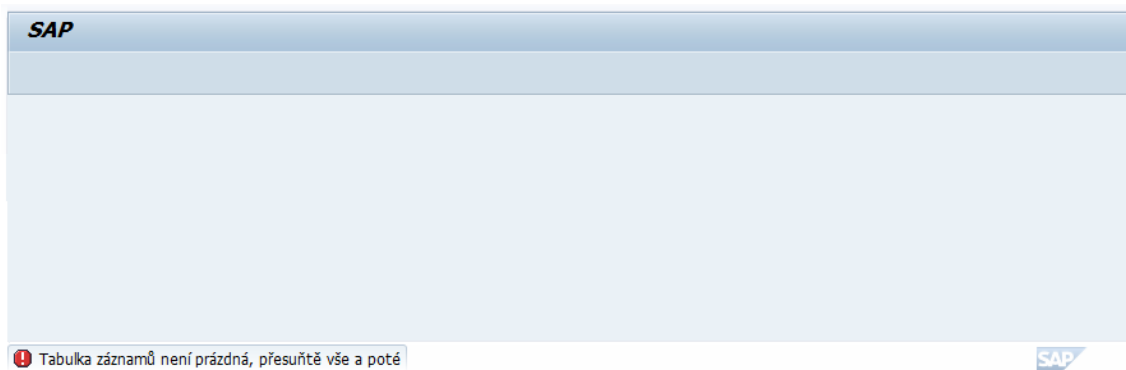
```
* tímto se vytvořilo paralelní vlákno a samotné přihlášení se provede zavoláním  
podprogramu, který jej má na starosti  
ENDFORM.
```

## A.2 – Výskyt duplicitních záznamů v hlavní databázi

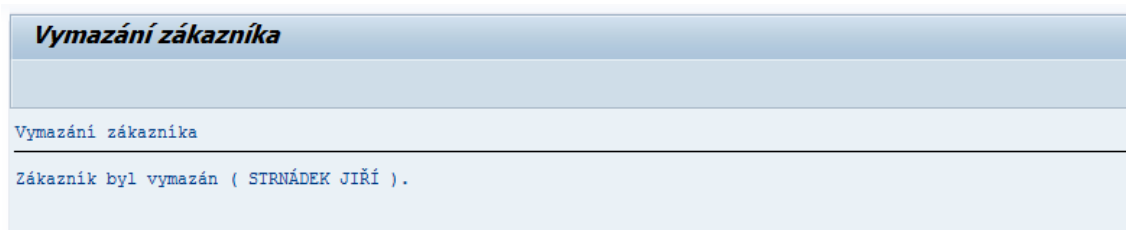
### Grafický náhled úpravy:



Obr. 32: Dotaz po stisku tlačítka Vymazat



Obr. 33: Varovná zpráva, že u zákazníka se vyskytují podněty



Obr. 34: Informace o vymazání včetně jména zákazníka

### Zdrojový kód

TABLES: ZCUSTOMERS.

TYPE-POOLS: slis.

DATA: odpoved TYPE c,  
kontrola TYPE c,  
i\_zcustomers TYPE TABLE OF zcustomers WITH HEADER LINE,  
fieldcat TYPE slis\_t\_fieldcat\_alv WITH HEADER LINE,  
"definice pro výstupní pole  
layout TYPE slis\_layout\_alv, " definice pro vzhled ALV  
text TYPE LVC\_TITLE. " definice pro název tabulky

```

SELECTION-SCREEN BEGIN OF BLOCK name WITH FRAME TITLE text-001.
PARAMETERS: surname LIKE zcustomers-prijmeni,
             name  LIKE zcustomers-jmeno,
             telef  LIKE zcustomers-telefon.
SELECTION-SCREEN END OF BLOCK name.
* Výběr z tabulky na základě příjmení a jména
SELECT * FROM zcustomers INTO CORRESPONDING FIELDS OF TABLE
i_zcustomers
      WHERE prijmeni = surname
      AND jmeno = name
      AND telefon = telef.
PERFORM build_fieldcatalog.
PERFORM build_disp.
* akce po označení řádku a stiku tlačítka pro vymazání ("odpadkový koš")
CALL FUNCTION 'POPUP_TO_CONFIRM' "standartní funkce pro pop-up okno
EXPORTING
  titlebar = 'Vymazání zákazníka'
  text_question = 'Opravdu chcete smazat zákazníka z hlavní databáze?'
  text_button_1 = 'ANO'(001)
  icon_button_1 = 'ICON_OKAY'
  text_button_2 = 'NE'(002)
  icon_button_2 = 'ICON_NO'
  default_button = '1'
  display_cancel_button = "
  userdefined_f1_help = SPACE
  start_column = 25
  start_row = 6
IMPORTING
  answer = ODPOVED.
IF ODPOVED = '1'.
  " kontrola, zdali jsou u zákazníka nějaké záznamy.... formou zavolání form
  IF i_zcustomers-podnet = 'X'.
    Message W000(38) with text-010 'Tabulka záznamů není prázdná, přesuňte vše a
    poté opakujte akci !!! '
    EXIT.
  ELSE.
    DELETE FROM zcustomers WHERE prijmeni = surname AND jmeno = name.
    CONCATENATE text-011 surname name text-012 INTO text SEPARATED BY
    space.
    WRITE text.
  ENDIF.

ELSEIF ODPOVED = '2'.
  SET SCREEN 0.
  LEAVE TO SCREEN 1000.
ENDIF.
*-----
FORM build_fieldcatalog.
  fieldcat-fieldname = 'ID'.
  fieldcat-seltext_m = 'ID'.

```

```
fieldcat-col_pos = 0.  
fieldcat-outputlen = 10.  
fieldcat-emphasize = 'X'.  
fieldcat-key = 'X'.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'PRIJMENI'.  
fieldcat-seltext_m = 'Příjmení'.  
fieldcat-col_pos = 1.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'JMENO'.  
fieldcat-seltext_m = 'Jméno'.  
fieldcat-col_pos = 2.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'TITUL'.  
fieldcat-seltext_m = 'Titul'.  
fieldcat-col_pos = 3.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'ADRESA1'.  
fieldcat-seltext_m = 'Adresa'.  
fieldcat-col_pos = 4.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'ADRESA2'.  
fieldcat-seltext_m = 'Dodatek adresy'.  
fieldcat-col_pos = 5.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'MESTO'.  
fieldcat-seltext_m = 'Město'.  
fieldcat-col_pos = 6.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'MESTO_CAST'.  
fieldcat-seltext_m = 'Část města'.  
fieldcat-col_pos = 7.  
append fieldcat to fieldcat.  
clear fieldcat.
```

```
fieldcat-fieldname = 'PSC'.
```



```
fieldcat-seltext_m = 'PSČ'.
fieldcat-col_pos   = 8.
append fieldcat to fieldcat.
clear fieldcat.
```

```
fieldcat-fieldname = 'STAT'.
fieldcat-seltext_m = 'Stát'.
fieldcat-col_pos   = 9.
append fieldcat to fieldcat.
clear fieldcat.
```

```
fieldcat-fieldname = 'TELEFON'.
fieldcat-seltext_m = 'Telefon'.
fieldcat-col_pos   = 9.
append fieldcat to fieldcat.
clear fieldcat.
```

```
fieldcat-fieldname = 'FAX'.
fieldcat-seltext_m = 'Fax'.
fieldcat-col_pos   = 9.
append fieldcat to fieldcat.
clear fieldcat.
```

```
fieldcat-fieldname = 'EMAIL'.
fieldcat-seltext_m = 'E-mail'.
fieldcat-col_pos   = 9.
append fieldcat to fieldcat.
clear fieldcat.
```

```
ENDFORM.
```

```
*-----
```

```
FORM build_disp.
```

```
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
EXPORTING
```

```
  I_GRID_TITLE = 'Zobrazení výsledků vyhledávání'
```

```
  IS_LAYOUT    = layout
```

```
  IT_FIELDCAT  = fieldcat[]
```

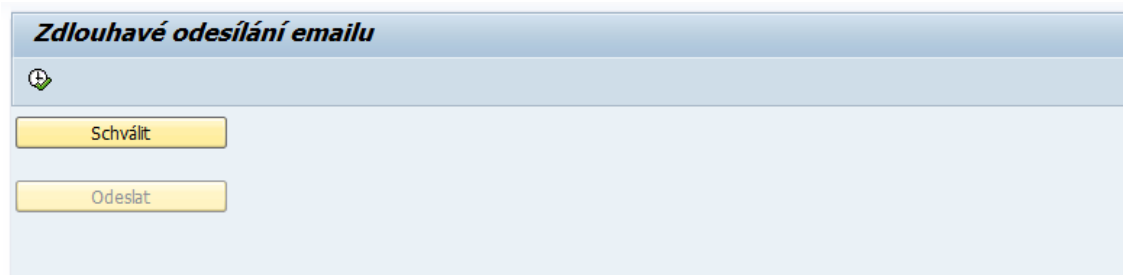
```
TABLES
```

```
  T_OUTTAB     = i_zcustomers[].
```

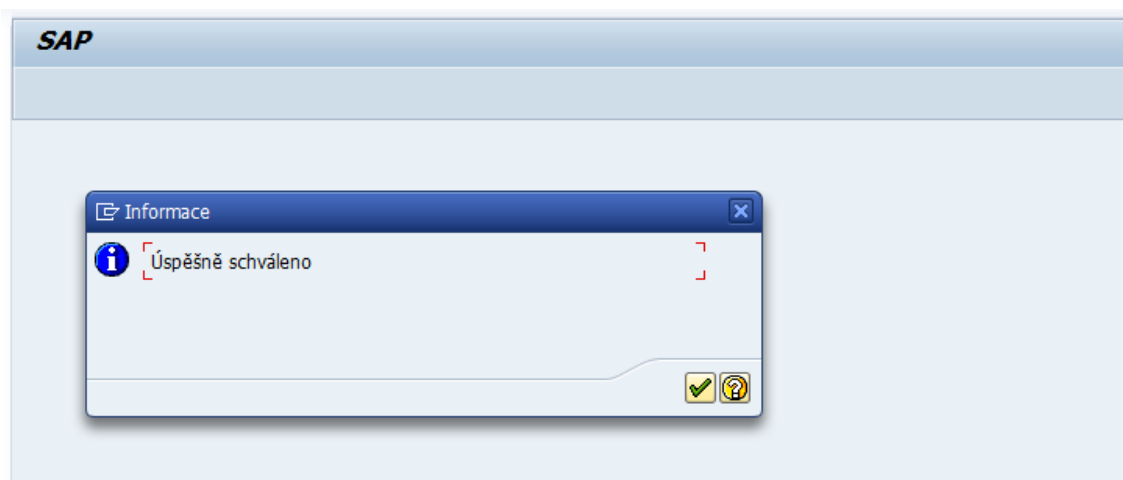
```
ENDFORM.
```

### A.3 – Zdlouhavé odesílání e-mailů

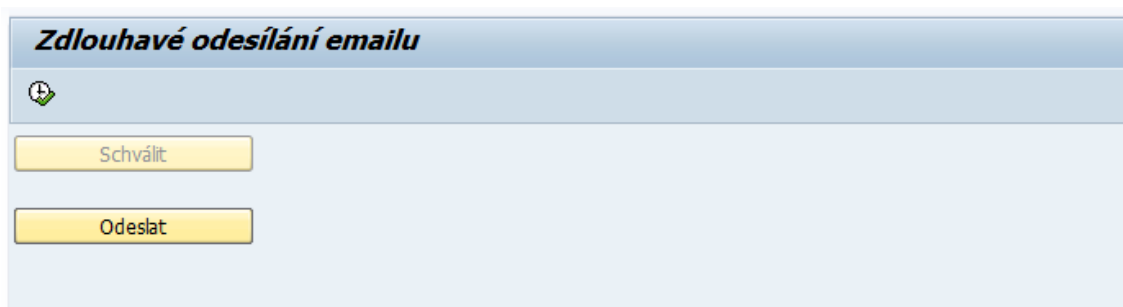
#### Grafický náhled úpravy:



Obr. 35: Stav tlačítek před schválením e-mailu



Obr. 36: Informace o schválení e-mailu



Obr. 37: Stav tlačítek po schválení před odesláním e-mailu



Obr. 38: Informace o předání e-mailu serveru

### **Zdrojový kód**

TABLES: zmailtab, SSCRFIELDS.

DATA: approved LIKE zmailtab-approved,  
output TYPE c VALUE '1'.

DATA: it\_message type standard table of SOLISTI1 initial size 0 with header line.

SELECTION-SCREEN PUSHBUTTON /1(20) schvalit USER-COMMAND EXEC  
MODIF ID AAA.

SELECTION-SCREEN skip.

SELECTION-SCREEN PUSHBUTTON /1(20) odeslat USER-COMMAND EXEC  
MODIF ID BBB.

INITIALIZATION.

schvalit = 'Schválit'.

odeslat = 'Odeslat'.

approved = 'SCNO'.

\*-----

AT SELECTION-SCREEN OUTPUT.

FORM check.

IF approved = 'SCNO'.

LOOP AT SCREEN.

IF screen-group1 = 'AAA'.

screen-input = '1'.

output = '1'.

MODIFY SCREEN.

ELSEIF screen-group1 = 'BBB'.

screen-input = '0'.

MODIFY SCREEN.

ENDIF.

ENDLOOP.

ELSEIF approved = 'SCOK'.

LOOP AT SCREEN.

IF screen-group1 = 'AAA'.

```

        screen-input = '0'.
        MODIFY SCREEN.
    ELSEIF screen-group1 = 'BBB'.
        screen-input = '1'.
        output = '2'.
        MODIFY SCREEN.
    ENDIF.
ENDLOOP.
ENDIF.

IF output = '1'.
    PERFORM schvaleni.
ELSEIF output = '2'.
    PERFORM odeslani.
ENDIF.
ENDFORM.
*-----
START-OF-SELECTION.
FORM schvaleni.
    CALL FUNCTION 'OPEN_FORM'
        EXPORTING
            DEVICE = 'PRINTER'
            LANGUAGE = SY-LANGU
            OPTIONS = ''
        IMPORTING
            LANGUAGE = SY-LANGU
        EXCEPTIONS
            DEVICE = 2.
    IF SY-SUBRC <> 0.
        Message W000(38) WITH 'Chyba tiskárny!!!'.
        approved = 'SCNO'.
    ELSE.
        Message I000(38) WITH 'Úspěšně schváleno'.
        approved = 'SCOK'.
    ENDIF.
    PERFORM check.
ENDFORM.
*-----
FORM odeslani.
    PERFORM populate_message_table.
    PERFORM send_email_message.
    Message I000(38) WITH 'E-mail předán serveru'.
    PERFORM initiate_mail_execute_program.
ENDFORM.

```

## A.4 – Nemožnost přidat dalšího příjemce emailu

### Grafický náhled úpravy:

**Odeslání e-mailu**

Odeslat

Vyberte se seznamů adresu zákazníka, případně servisu.

Zákazník: 1 petr.novak@p... Servis: 2 servis@servis-s...

Obr. 39: Výběrová obrazovka e-mailových adres

### Zdrojový kód

TYPE-POOLS: VRM.

TABLES: sscrfields, zcustomers.

DATA: t\_zcustomers TYPE STANDARD TABLE OF zcustomers.

DATA: smp\_dyntxt01 TYPE smp\_dyntxt.

DATA: param01 TYPE vrm\_id,  
values01 TYPE vrm\_values,  
value01 LIKE LINE OF values01.

DATA: param02 TYPE vrm\_id,  
values02 TYPE vrm\_values,  
value02 LIKE LINE OF values02.

DATA: mail1(25) TYPE c,  
mail2(25) TYPE c,  
text\_header TYPE c.

SELECTION-SCREEN FUNCTION KEY 1.

SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME.

SELECTION-SCREEN COMMENT 1(54) text-060.

SELECTION-SCREEN skip.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT 1(9) text-061.

SELECTION-SCREEN COMMENT 30(9) text-062 for field p\_name02.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT 1(1) text-069 for field p\_name01.

PARAMETERS p\_name01(50) AS LISTBOX VISIBLE LENGTH 16.

SELECTION-SCREEN COMMENT 30(1) text-068 for field p\_name02.

PARAMETERS p\_name02(50) AS LISTBOX VISIBLE LENGTH 16.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN END OF BLOCK b0.

INITIALIZATION .

CLEAR smp\_dyntxt01.

```
smp_dyntxt01-icon_id = '@0T@'.
smp_dyntxt01-icon_text = text-098.
smp_dyntxt01-quickinfo = text-099.
sscrfields-functxt_01 = smp_dyntxt01.
```

```
mail1 = t_zcustomers-email.
mail2 = t_zservis-email1.
```

AT SELECTION-SCREEN OUTPUT.

```
param01 = 'P_NAME01'.
value01-key = '1'.
value01-text = mail1.
APPEND value01 TO values01.
```

```
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    id    = param01
    values = values01.
```

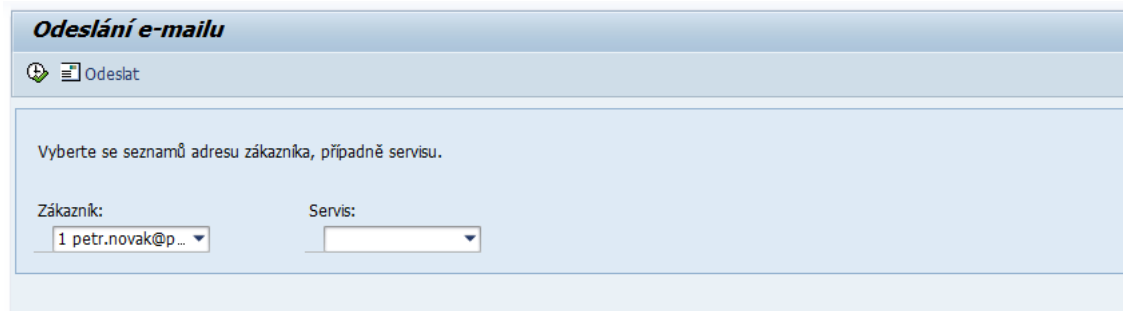
```
param02 = 'P_NAME02'.
value02-key = '2'.
value02-text = mail2.
APPEND value02 TO values02.
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    id    = param02
    values = values02.
```

```
CASE sscrfields-ucomm.
  WHEN 'FC01'.
    CALL SCREEN 0100.
ENDCASE.
```

```
IF p_name02 IS INITIAL.
  CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
      I_TITLE           = text_header
      I_EMAIL           = p_name01
    TABLES
      T_LINES           = text_email.
ELSE.
  CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
      I_TITLE           = text_header
      I_EMAIL           = p_name01
      I_EMAIL_COPY_CC   = p_name02
    TABLES
      T_LINES           = text_email.
ENDIF.
```

## A.5 – Sledované podněty představenstvem společnosti – automatická kopie

### Grafický náhled úpravy:



Obr. 40: Výběrová obrazovka e-mailových adres (vybrána jen zákazníkova – na sekretariát G automaticky)

### Zdrojový kód

TYPE-POOLS: VRM.

TABLES: sscrfields, zcustomers.

DATA: t\_zcustomers TYPE STANDARD TABLE OF zcustomers.

DATA: smp\_dyntxt01 TYPE smp\_dyntxt.

DATA: param01 TYPE vrm\_id,  
values01 TYPE vrm\_values,  
value01 LIKE LINE OF values01.

DATA: param02 TYPE vrm\_id,  
values02 TYPE vrm\_values,  
value02 LIKE LINE OF values02.

DATA: mail1(25) TYPE c,  
mail2(25) TYPE c,  
text\_header TYPE c.

SELECTION-SCREEN FUNCTION KEY 1.

SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME.

SELECTION-SCREEN COMMENT 1(54) text-060.

SELECTION-SCREEN skip.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT 1(9) text-061.

SELECTION-SCREEN COMMENT 30(9) text-062 for field p\_name02.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT 1(1) text-069 for field p\_name01.

PARAMETERS p\_name01(50) AS LISTBOX VISIBLE LENGTH 16.

SELECTION-SCREEN COMMENT 30(1) text-068 for field p\_name02.

PARAMETERS p\_name02(50) AS LISTBOX VISIBLE LENGTH 16.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN END OF BLOCK b0.

```

INITIALIZATION .
CLEAR smp_dyntxt01.
smp_dyntxt01-icon_id = '@0T@'.
smp_dyntxt01-icon_text = text-098.
smp_dyntxt01-quickinfo = text-099.
sscrfields-functxt_01 = smp_dyntxt01.

mail1 = t_zcustomers-email.
mail2 = t_zservis-email1.

AT SELECTION-SCREEN OUTPUT.
  param01 = 'P_NAME01'.
  value01-key = '1'.
  value01-text = mail1.
  APPEND value01 TO values01.

  CALL FUNCTION 'VRM_SET_VALUES'
    EXPORTING
      id    = param01
      values = values01.

  param02 = 'P_NAME02'.
  value02-key = '2'.
  value02-text = mail2.
  APPEND value02 TO values02.
  CALL FUNCTION 'VRM_SET_VALUES'
    EXPORTING
      id    = param02
      values = values02.

CASE sscrfields-ucomm.
  WHEN 'FC01'.
    CALL SCREEN 0100.
ENDCASE.

IF t_zcustomers-priority = 'X'.
IF p_name02 IS INITIAL.
  CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
      I_TITLE           = text_header
      I_EMAIL           = p_name01
      I_EMAIL_COPY_CC   = 'sekretariat_g@skoda-auto.cz'
  TABLES
    T_LINES            = text_email.
ELSE.
  CONCATENATE p_name2 'sekretariat_g@skoda-auto.cz' INTO p_name2
  SEPARATED BY ';'.
  " pokud se vybere i servis, spojují se adresy a rozděleny jsou středníkem
  CALL FUNCTION 'ZMM_SEND_EMAIL'

```



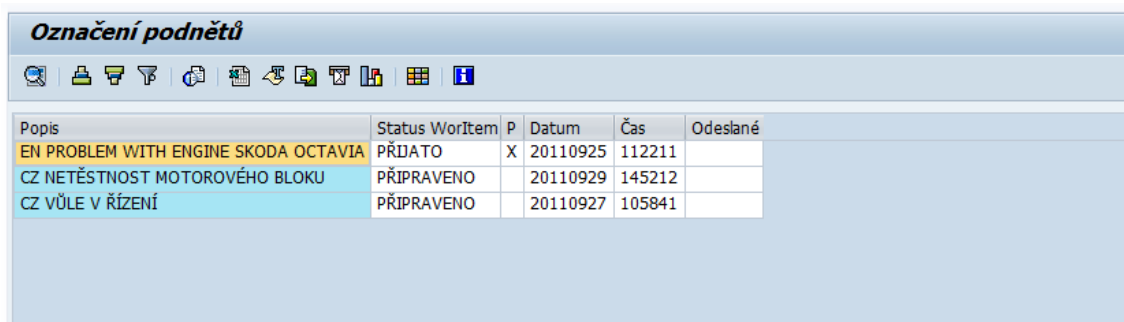
```

EXPORTING
  I_TITLE           = text_header
  I_EMAIL           = p_name01
  I_EMAIL_COPY_CC   = p_name02
TABLES
  T_LINES           = text_email.
ENDIF.
ELSE.
  IF p_name02 IS INITIAL.
    CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
      I_TITLE           = text_header
      I_EMAIL           = p_name01
    TABLES
      T_LINES           = text_email.
  ELSE.
    CALL FUNCTION 'ZMM_SEND_EMAIL'
    EXPORTING
      I_TITLE           = text_header
      I_EMAIL           = p_name01
      I_EMAIL_COPY_CC   = p_name02
    TABLES
      T_LINES           = text_email.
  ENDIF.
ENDIF.

```

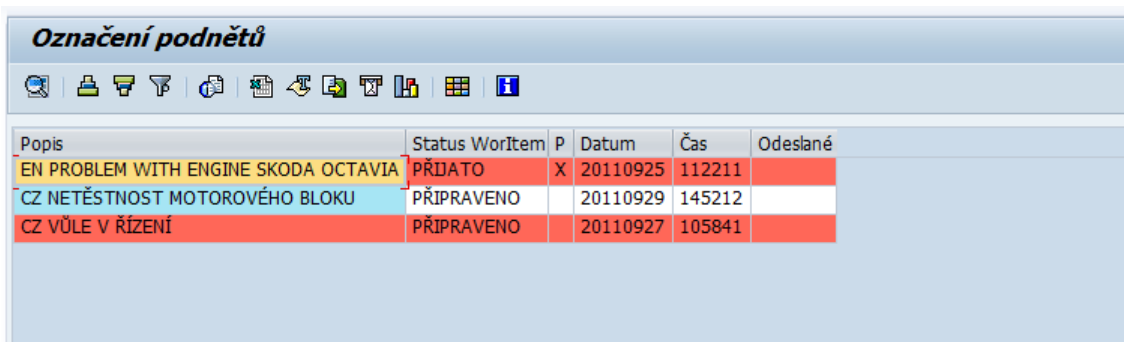
## A.6 – Zvýraznění dlouho nepřiřazených podnětů v hlášení

### Grafický náhled úpravy:



Popis	Status	WorItem	P	Datum	Čas	Odeslané
EN PROBLEM WITH ENGINE SKODA OCTAVIA	PŘIJATO	X	20110925	112211		
CZ NETĚSTNOST MOTOROVÉHO BLOKU	PŘIPRAVENO		20110929	145212		
CZ VŮLE V ŘÍZENÍ	PŘIPRAVENO		20110927	105841		

Obr. 41: Obyčejné zobrazení přehledu podnětů pro uživatele



Popis	Status	WorItem	P	Datum	Čas	Odeslané
EN PROBLEM WITH ENGINE SKODA OCTAVIA	PŘIJATO	X	20110925	112211		
CZ NETĚSTNOST MOTOROVÉHO BLOKU	PŘIPRAVENO		20110929	145212		
CZ VŮLE V ŘÍZENÍ	PŘIPRAVENO		20110927	105841		

Obr. 42: Barevné rozlišení dlouho zpracovávaných podnětů (> 2 dny, datum 30.9.2011)

### Zdrojový kód

TABLES: zmailtab.

TYPE-POOLS: slis, VRM.

TYPES: BEGIN OF t\_zmailtab,

title TYPE zmailtab-title,

status TYPE zmailtab-status,

priorita TYPE zmailtab-priorita,

datum TYPE zmailtab-datum,

cas TYPE zmailtab-cas,

send TYPE zmailtab-send,

difference TYPE zmailtab-difference,

user\_name TYPE zmailtab-user\_name,

line\_color(4) TYPE c, "Used to store row color attributes

END OF t\_zmailtab.

DATA: it\_zmailtab TYPE STANDARD TABLE OF

t\_zmailtab INITIAL SIZE 0 WITH HEADER LINE,

wa\_zmailtab TYPE t\_zmailtab OCCURS 0 WITH

HEADER LINE.

DATA: param TYPE vrm\_id,

```
values      TYPE vrm_values,
value       LIKE LINE OF values.
DATA: fieldcatalog TYPE slis_t_fieldcat_alv with
```

```
header line,
  gd_tab_group TYPE slis_t_sp_group_alv,
  layout       TYPE slis_layout_alv,
  ld_color(1)  TYPE c,
  difference    LIKE sy-datum,
  difference1   TYPE sy-datum,
  gd_repid     LIKE sy-repid.
```

SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME

TITLE text-001.

PARAMETERS: p\_name(16) AS LISTBOX VISIBLE LENGTH

16 DEFAULT '1'.

SELECTION-SCREEN END OF BLOCK b0.

\*\*\*\*\*

AT SELECTION-SCREEN OUTPUT.

```
param = 'P_NAME'.
value-key = '1'.
value-text = 'Všechny podněty'.
APPEND value TO values.
value-key = '2'.
value-text = 'Moje práce'.
APPEND value TO values.
CALL FUNCTION 'VRM_SET_VALUES'
  EXPORTING
    id    = param
    values = values.
```

START-OF-SELECTION.

*\* listbox s výběrem - Všechny podněty, Moje práce...*

```
IF p_name = '1'.
  PERFORM data_complete.
  PERFORM build_fieldcatalog.
  PERFORM build_layout.
  PERFORM display_alv_report.
ELSEIF p_name = '2'.
  PERFORM data_personal.
  PERFORM build_fieldcatalog.
  PERFORM build_layout.
  PERFORM display_alv_report.
ENDIF.
```

\*-----

FORM build\_fieldcatalog.

```

fieldcatalog-fieldname = 'TITLE'.
fieldcatalog-seltext_m = 'Popis'.
fieldcatalog-col_pos   = 0.
fieldcatalog-outputlen = 10.
fieldcatalog-emphasize = 'X'.
fieldcatalog-key       = 'X'.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'STATUS'.
fieldcatalog-seltext_m = 'Status WorItem'.
fieldcatalog-col_pos   = 1.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'PRIORITA'.
fieldcatalog-seltext_m = 'P'.
fieldcatalog-col_pos   = 2.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'DATUM'.
fieldcatalog-seltext_m = 'Datum'.
fieldcatalog-col_pos   = 3.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'CAS'.
fieldcatalog-seltext_m = 'Čas'.
fieldcatalog-col_pos   = 4.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'SEND'.
fieldcatalog-seltext_m = 'Odeslané'.
fieldcatalog-col_pos   = 5.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.
Endform.
*-----
FORM build_layout.
  layout-no_input      = 'X'.
  layout-colwidth_optimize = 'X'.
  layout-totals_text   = 'Totals'(201).
  layout-info_fieldname = 'LINE_COLOR'.
ENDFORM.
*-----
FORM display_alv_report.
  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING

```

```

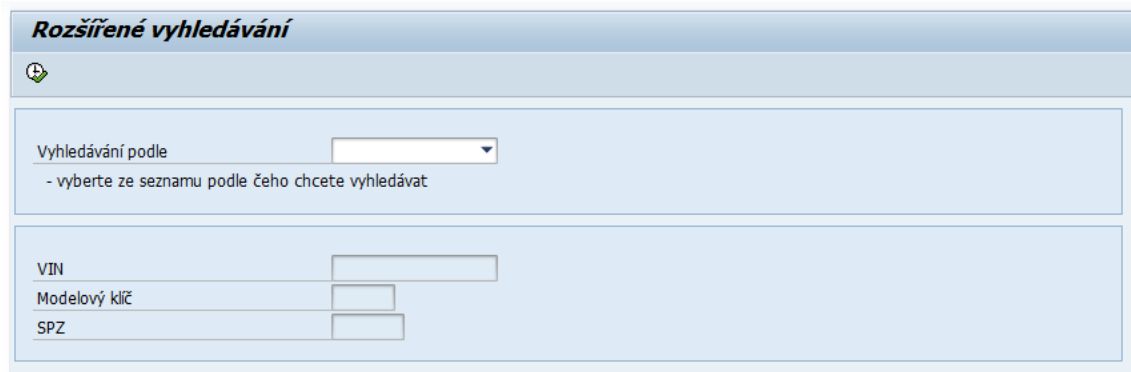
        i_callback_program = gd_repid
*       i_grid_title       = 'Položky'
        is_layout          = layout
        it_fieldcat        = fieldcatalog[]
        i_save              = 'X'
TABLES
        t_outtab           = it_zmailtab
EXCEPTIONS
        program_error      = 1
        others              = 2.
if sy-subrc <> 0.
endif.
ENDFORM.
*-----
FORM data_personal.
    difference = sy-datum.
    SELECT * FROM zmailtab
        INTO CORRESPONDING FIELDS OF TABLE it_zmailtab
        WHERE status = 'PŘIJATO'
        AND user_name = sy-uname.
    LOOP AT it_zmailtab INTO wa_zmailtab.
        LOOP AT it_zmailtab.
            CLEAR ld_color.
            IF wa_zmailtab-status = 'PŘIJATO'.
                difference = ( difference - wa_zmailtab-datum ).
                IF wa_zmailtab-difference > '2'.
                    ld_color = 6.
                    CONCATENATE 'C' ld_color '10' INTO wa_zmailtab-line_color.
                ENDIF.
            ENDIF.
        ENDLOOP.
    ENDLOOP.
    MODIFY it_zmailtab FROM wa_zmailtab.
ENDLOOP.
ENDFORM.
*-----
FORM data_complete.
    difference = sy-datum.
    SELECT * FROM zmailtab
        INTO CORRESPONDING FIELDS OF TABLE it_zmailtab.
    LOOP AT it_zmailtab INTO wa_zmailtab.
        LOOP AT it_zmailtab.
            CLEAR ld_color.
            IF wa_zmailtab-status = 'PŘIPRAVENO'.
                IF wa_zmailtab-difference > '2'.
                    ld_color = 6.  "světle červená barva v ALV
                    CONCATENATE 'C' ld_color '10' INTO wa_zmailtab-line_color.
                ENDIF.
            ELSEIF wa_zmailtab-status = 'PŘIJATO'.
                difference = ( difference - wa_zmailtab-datum ).
                IF it_zmailtab-user_name = sy-uname.

```

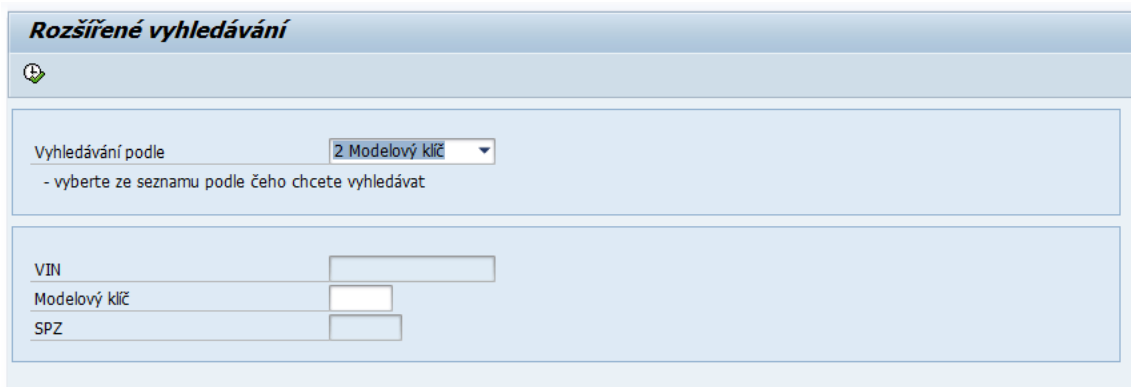
```
IF wa_zmailtab-difference > '2'.  
  ld_color = 6. "světle červená barva v ALV"  
  CONCATENATE 'C' ld_color '10' INTO wa_zmailtab-line_color.  
ENDIF.  
ENDIF.  
ENDLOOP.  
MODIFY it_zmailtab FROM wa_zmailtab.  
ENDLOOP.  
ENDFORM.
```

## A.7 – Důkladnější vyhledávání

### Grafický náhled úpravy:



Obr. 43: Základní selekční obrazovka pro vyhledávání



Obr. 44: Selekční obrazovka po výběru jedné možnosti

### Zdrojový kód

TABLES: zcustomers, zvehicles, SSCRFIELDS.

TYPE-POOLS: slis, VRM.

\* *výběr z listboxu to, co chce uživatel hledat*

DATA: t\_zvehicles TYPE STANDARD TABLE OF zvehicles.

DATA: param TYPE vrm\_id,  
values TYPE vrm\_values,  
value LIKE LINE OF values.

DATA: layout TYPE slis\_layout\_alv,  
gd\_repid LIKE sy-repid,  
fieldcatalog TYPE slis\_t\_fieldcat\_alv with header line,  
gd\_tab\_group TYPE slis\_t\_sp\_group\_alv,  
text TYPE LVC\_TITLE.

SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME TITLE text-002.

PARAMETERS: p\_name(17) TYPE c AS LISTBOX VISIBLE LENGTH 17

DEFAULT '3'.

SELECTION-SCREEN COMMENT /2(50) 'Vyberte ze seznamu podle čeho chcete vyhledávat:'

```

SELECTION-SCREEN END OF BLOCK b0.
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.
PARAMETERS: vin(17) TYPE c MODIF ID AAA,
              model(6) TYPE c MODIF ID BBB,
              SPZ(7) TYPE c MODIF ID CCC.
SELECTION-SCREEN END OF BLOCK b1.
* v příkladě vybrány pouze dva další typy
AT SELECTION-SCREEN OUTPUT.
  param = 'P_NAME'.
  value-key = '1'.
  value-text = 'VIN'.
  append value to values.
  value-key = '2'.
  value-text = 'Modelový klíč'.
  append value to values.
  value-key = '3'.
  value-text = 'SPZ'.
  append value to values.
  CALL FUNCTION 'VRM_SET_VALUES'
    EXPORTING
      id    = param
      values = values.

LOOP AT SCREEN.
  IF screen-group1 = 'AAA'.
    screen-input = '1'.
    MODIFY SCREEN.
  ELSEIF screen-group1 = 'BBB'.
    screen-input = '0'.
    MODIFY SCREEN.
  ELSEIF screen-group1 = 'CCC'.
    screen-input = '0'.
    MODIFY SCREEN.
  ENDIF.
ENDLOOP.
CASE value.
  WHEN '1'.
    PERFORM vin.
  WHEN '2'.
    PERFORM model.
  WHEN '3'.
    PERFORM spz.
ENDCASE.

INITIALIZATION.
*-----
FORM vin.
  CONCATENATE text-034 vin INTO text SEPARATED BY space.
  SELECT * FROM zvehicles
    INTO CORRESPONDING FIELDS OF TABLE t_zvehicles

```



```

        WHERE idvin = vin.
ENDFORM.
*-----
FORM model.
    CONCATENATE text-035 vin INTO text SEPARATED BY space.
    SELECT * FROM zvehicles
        INTO CORRESPONDING FIELDS OF TABLE t_zvehicles
        WHERE model_code = model.
ENDFORM.
*-----
FORM spz.
    CONCATENATE text-036 vin INTO text SEPARATED BY space.
    SELECT * FROM zvehicles
        INTO CORRESPONDING FIELDS OF TABLE t_zvehicles
        WHERE spz = spz.
ENDFORM.
"spz
* následné vybudování ALV zobrazení - tabulka t_zvehicles
layout-zebra          = 'X'.
layout-no_input       = 'X'.
layout-colwidth_optimize = 'X'.

fieldcatalog-fieldname = 'IDVIN'.
fieldcatalog-seltext_m = 'ID - VIN'.
fieldcatalog-col_pos   = 0.
fieldcatalog-outputlen = 4.
fieldcatalog-emphasize = 'X'.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'ZNACKA'.
fieldcatalog-seltext_m = 'Značka'.
fieldcatalog-col_pos   = 1.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'MODEL'.
fieldcatalog-seltext_m = 'Model'.
fieldcatalog-col_pos   = 2.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'MODEL_CODE'.
fieldcatalog-seltext_m = 'Modelový klíč'.
fieldcatalog-col_pos   = 3.
append fieldcatalog to fieldcatalog.
clear fieldcatalog.

fieldcatalog-fieldname = 'DESCRIPTION'.
fieldcatalog-seltext_m = 'Popis'.
fieldcatalog-col_pos   = 4.

```

```
append fieldcatalog to fieldcatalog.  
clear fieldcatalog.
```

```
fieldcatalog-fieldname = 'SPZ'.  
fieldcatalog-seltext_m = 'SPZ'.  
fieldcatalog-col_pos   = 5.  
append fieldcatalog to fieldcatalog.  
clear fieldcatalog.
```

```
fieldcatalog-fieldname = 'KM'.  
fieldcatalog-seltext_m = 'Kilometry'.  
fieldcatalog-col_pos   = 6.  
append fieldcatalog to fieldcatalog.  
clear fieldcatalog.
```

```
fieldcatalog-fieldname = 'MOTOR'.  
fieldcatalog-seltext_m = 'Motor'.  
fieldcatalog-col_pos   = 7.  
append fieldcatalog to fieldcatalog.  
clear fieldcatalog.
```

```
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
```

```
EXPORTING
```

```
  i_callback_program = gd_repid
```

```
  i_grid_title       = text
```

```
  is_layout          = layout
```

```
  it_fieldcat        = fieldcatalog[]
```

```
  i_save              = 'X'
```

```
TABLES
```

```
  t_outtab           = t_zvehicles
```

```
EXCEPTIONS
```

```
  program_error      = 1
```

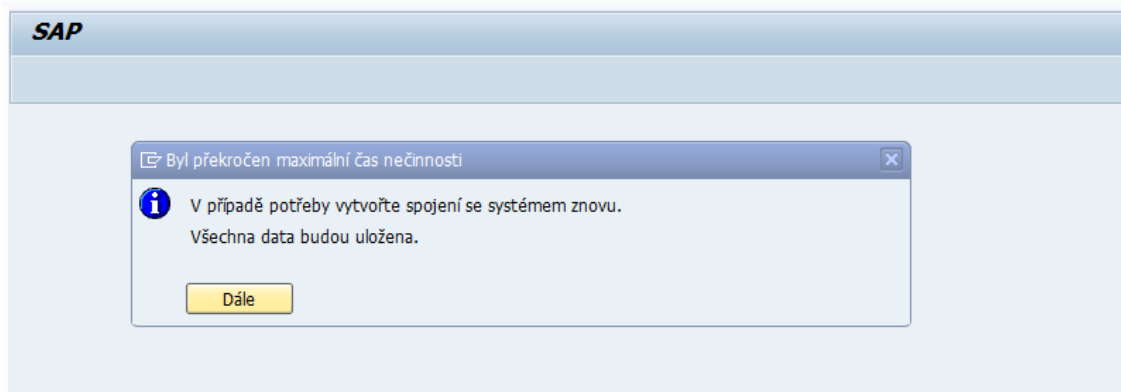
```
  others              = 2.
```

```
if sy-subrc <> 0.
```

```
endif.
```

## A.8 – Samovolné odhlášení po delší době nečinnosti

### Grafický náhled úpravy:



Obr. 45: Informační hláška o překročení času nečinnosti



Obr. 46: Další informační hláška



Obr. 47: Informační hláška o uložení všech dat

### Zdrojový kód

DATA: ODPOVED TYPE c.

\* logout čas je nastaven na 45 minut = 2700 vteřin

IF rdisp/gui\_auto\_logout = 2500.

\* uložení všech tabulek, které mají souvislost se zákazníkem a podnětem

ENDIF.

*\* otevření info okna, že překročen čas a spojení zrušeno*

PERFORM info.

PERFORM potvrzení.

FORM info.

CALL FUNCTION 'POPUP\_TO\_DISPLAY\_TEXT'

EXPORTING

TITEL = 'Byl překročen maximální čas nečinnosti'

TEXTLINE1 = 'V případě potřeby vytvořte spojení se systémem znovu.'

TEXTLINE2 = 'Všechna data budou uložena.'

START\_COLUMN = 13

START\_ROW = 2.

ENDFORM.

FORM potvrzení.

CALL FUNCTION 'POPUP\_TO\_CONFIRM'

EXPORTING

TITLEBAR = 'Automatické odhlášení'

TEXT\_QUESTION = 'Byl překročen maximální čas nečinnosti.'

TEXT\_BUTTON\_1 = 'Zavřít'

icon\_button\_1 = 'ICON\_CANCEL'

START\_COLUMN = 13

START\_ROW = 2

IMPORTING

ANSWER = ODPOVED.

IF ODPOVED = '1'.

POSITION 13.

Message I000(38) with text-022. " text-022 Vše uloženo.

SET SCREEN 0.

LEAVE TO SCREEN 1001.

*" samotné odhlášení provádí gui\_auto\_login po uplynutí nastaveného času*

ELSE.

Message W000(38) with text-023. " text-023 Chyba při ukládání.

ENDIF.

ENDFORM.

## A.9 – Vyhledávání zákazníka v databázi

### Grafický náhled úpravy:

**Vyhledat zákazníka**

Vyhledat Ukončit zákazníka

Příjmení: ☒ Jméno: Titul:

Adresa: Podadresa: Číslo:

Město: Část: PSČ:

Stát:

Telefon: FAX: E-mail:

Obr. 48: vyhledávací obrazovka po ukončení práce se zákazníkem

### Zdrojový kód

TABLES: zcustomers, sscrfields.

DATA: smp\_dyntxt01 TYPE smp\_dyntxt,  
smp\_dyntxt02 TYPE smp\_dyntxt.

SELECTION-SCREEN FUNCTION KEY 1.

SELECTION-SCREEN FUNCTION KEY 2.

SELECTION-SCREEN BEGIN OF BLOCK b0 WITH FRAME.

SELECTION-SCREEN BEGIN OF LINE.

SELECTION-SCREEN COMMENT (9) text-041 for field prijmeni.

PARAMETERS prijmeni TYPE zcustomers-prijmeni OBLIGATORY.

SELECTION-SCREEN COMMENT (6) text-042 for field jmeno.

PARAMETERS jmeno TYPE zcustomers-jmeno.

SELECTION-SCREEN COMMENT (6) text-043 for field titul.

PARAMETERS titul TYPE zcustomers-titul.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN END OF BLOCK b0.

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.

SELECTION-SCREEN begin of LINE.

SELECTION-SCREEN COMMENT (6) text-044 for field adresa1.

PARAMETERS adresa1 TYPE zcustomers-adresa1.

SELECTION-SCREEN COMMENT (10) text-045 for field adresa2.

PARAMETERS adresa2 TYPE zcustomers-adresa2.

SELECTION-SCREEN COMMENT (6) text-046 for field adrescis.

PARAMETERS adrescis TYPE zcustomers-adresa\_cislo.

SELECTION-SCREEN END OF LINE.

SELECTION-SCREEN begin of LINE.

SELECTION-SCREEN COMMENT (6) text-047 for field mesto.

PARAMETERS mesto TYPE zcustomers-mesto.

SELECTION-SCREEN COMMENT (10) text-048 for field mestcast.

PARAMETERS mestcast TYPE zcustomers-mesto\_cast.

SELECTION-SCREEN COMMENT (4) text-049 for field psc.

```

PARAMETERS psc TYPE zcustomers-psc.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN begin of LINE.
  SELECTION-SCREEN COMMENT (6) text-053 for field telefon.
  PARAMETERS stat TYPE zcustomers-stat.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN skip.
SELECTION-SCREEN begin of LINE.
  SELECTION-SCREEN COMMENT (8) text-050 for field telefon.
  PARAMETERS telefon TYPE zcustomers-telefon.
  SELECTION-SCREEN COMMENT (4) text-051 for field fax.
  PARAMETERS fax TYPE zcustomers-fax.
  SELECTION-SCREEN COMMENT (7) text-052 for field email.
  PARAMETERS email TYPE zcustomers-email.
SELECTION-SCREEN end of LINE.
SELECTION-SCREEN END OF BLOCK b1.

```

\*-----

```

INITIALIZATION .
CLEAR smp_dyntxt01.
CLEAR smp_dyntxt02.
smp_dyntxt01-icon_id = '@13@'.
smp_dyntxt01-icon_text = text-098.
smp_dyntxt01-quickinfo = text-099.
sscrfields-functxt_01 = smp_dyntxt01.

```

```

smp_dyntxt02-icon_id = '@02@'.
smp_dyntxt02-icon_text = text-097.
smp_dyntxt02-quickinfo = text-096.
sscrfields-functxt_02 = smp_dyntxt02.

```

\*-----

```

AT SELECTION-SCREEN.
  IF sscrfields-ucomm = 'FC02'.
    PERFORM clearing.
  ELSEIF sscrfields-ucomm = 'FC01'..
    PERFORM finding.
  ENDIF.

```

FORM clearing. *" vyčištění polí a proměnných při stisku ukončení*

```

CLEAR prijmeni.
CLEAR jmeno.
CLEAR titul.
CLEAR adresa1.
CLEAR adresa2.
CLEAR adrescis.
CLEAR mesto.
CLEAR mestcast.
CLEAR psc.
CLEAR stat.
CLEAR telefon.
CLEAR fax.

```

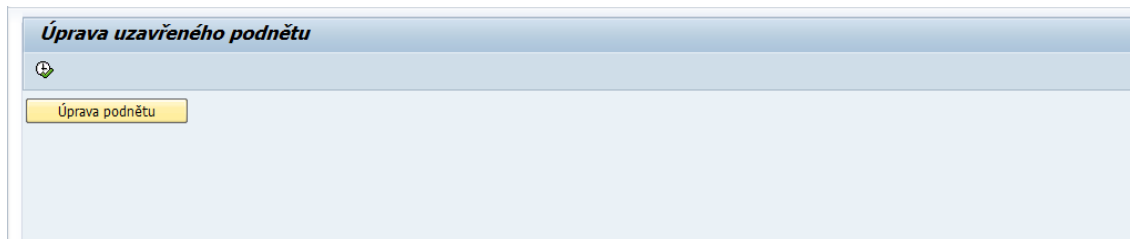
CLEAR email.  
ENDFORM.

FORM finding.

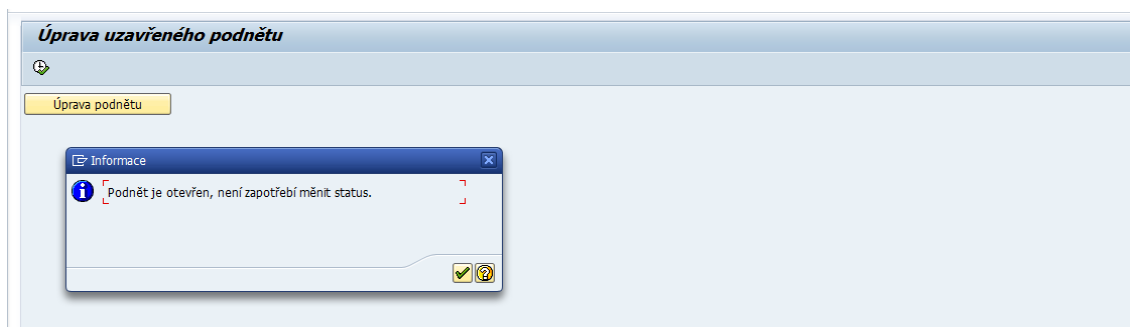
ENDFORM.

## A.10 – Nelze dočasně změnit status „Uzavřeno“

### Grafický náhled úpravy:



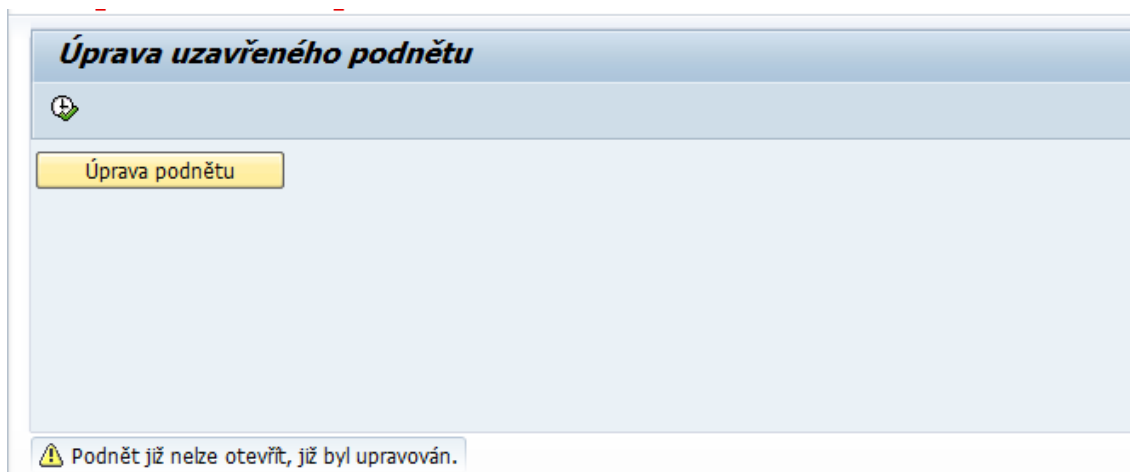
Obr. 49: Tlačítko pro změnu statusu podnětu



Obr. 50: Varianta neuzavřeného podnětu



Obr. 51: Pop-up okno s dotazem



Obr. 52: Reakce na stisk tlačítka v případě již jednou otevřeného podnětu



## Zdrojový kód

TABLES: SSCRFIELDS.

DATA: cstatus TYPE c LENGTH 7,  
      cindex TYPE c LENGTH 1,  
      ODPOVED TYPE c,  
      text TYPE LVC\_TITLE.

SELECTION-SCREEN PUSHBUTTON /1(20) Tlactko USER-COMMAND EXEC.

INITIALIZATION.

  Tlactko = 'Úprava podnětu'.

\*-----

AT SELECTION-SCREEN.

  IF sscrfields-ucomm = 'EXEC'.

    PERFORM check.

  ENDIF.

\*-----

START-OF-SELECTION.

FORM check.

\*cstatus = 'uzavren'.

\*cindex = '1'.

IF cstatus = 'uzavren'.

  IF cindex = '0'.

    CALL FUNCTION 'POPUP\_TO\_CONFIRM' "standardní fce pro pop-up okno

      EXPORTING

        titlebar = 'Změna statusu podnětu'

        text\_question = 'Podnět je uzavřen,

chcete jej dočasně otevřít?'

        text\_button\_1 = 'ANO'(001)

        icon\_button\_1 = 'ICON\_OKAY'

        text\_button\_2 = 'NE'(002)

        icon\_button\_2 = 'ICON\_CANCEL'

        default\_button = '1'

        display\_cancel\_button = "

        userdefined\_f1\_help = SPACE

        start\_column = 50

        start\_row = 10

      IMPORTING

        answer = ODPOVED.

  IF ODPOVED = '1'. "stisknutí ANO

    PERFORM change\_status.

  ELSEIF ODPOVED = '2'. "stisknutí NE

    exit.

  ENDIF.

ELSEIF cindex = '1'.

  Message W000(38) with 'Podnět již nelze otevřít, již byl upravován.'.

  exit.

ENDIF.

```
ELSEIF cstatus >< 'uzavren'.  
  Message I000(38) WITH 'Podnět je otevřen, není zapotřebí měnit status.'  
  EXIT.  
ENDIF.  
ENDFORM.  
FORM change_status.  
  cstatus = 'dočasný'.  
  cindex = '1'.  
  CONCATENATE 'Status je ' cstatus ' a změnový status je nyní ' cindex INTO text  
    SEPARATED BY space.  
  Message I000(38) WITH 'Nyní je možná úprava podnětu.'  
  Message I000(38) WITH text.  
ENDFORM.
```

## Příloha B

### B.1 – Náhled zdrojového souboru pro datamining

Tab. 6: Zobrazení souboru pro prvních 10 záznamů (1. 1/3)

ID vozu	Kód vozu	Model	Modelový klíč	Závada RRRRMM	Rok závady	Měsíc závady
0000001	1U	Octavia	1U21G4	201007	2010	7
0000002	1U	Octavia	1U5154	201003	2010	3
0000003	1Z	Octavia	1Z31D5	201007	2010	7
0000004	1Z	Octavia	1Z32G4	201003	2010	3
0000005	1Z	Octavia	1Z3364	201010	2010	10
0000006	1Z	Octavia	1Z3364	201008	2010	8
0000007	1Z	Octavia	1Z3364	201006	2010	6
0000008	1Z	Octavia	1Z33L5	201004	2010	4
0000009	1Z	Octavia	1Z33L5	201009	2010	9
0000010	1Z	Octavia	1Z34M5	201002	2010	2

Tab. 7: Zobrazení souboru pro prvních 10 záznamů (2. 1/3)









































Počet závad	Kód závady	Popis závady	Datum předání
2	5717	Zamek dveri	29.7.2010
1	1908	Ventilator	9.3.2010
1	9710	Jednotka ridici centralni	17.7.2010
1	2807	Vlozka zamku	22.3.2010
1	2338	Cerpadlo/ tryska	8.11.2010
1	2338	Cerpadlo/ tryska	1.9.2010
2	7057	Panel sloupku A	15.6.2010
2	2783	Ovladani regulace rychlosti	25.4.2010
2	5766	Tesneni pridavne	5.10.2010
1	1536	Napinak retezu	5.2.2010

Tab. 8: Zobrazení souboru pro prvních 10 záznamů (3. 1/3)

Datum přijetí	Doba opravy	ID Servisu	Náklady EUR
22.7.2010	7	20397	143,12
6.3.2010	3	22110	24,00
10.7.2010	7	25216	63,82
19.3.2010	3	28380	33,40
29.10.2010	10	20575	1818,83
24.8.2010	8	20575	2369,17
9.6.2010	6	22021	26,81
21.4.2010	4	27161	63,62
26.9.2010	9	29971	67,78
3.2.2010	2	21636	3756,00

## B.2 – Seznam servisů seřazených dle četnosti zásahů

Tab. 9: Seznam servisů seřazených sestupně podle četnosti zásahů (pouze část)

Pořadí	ID Servisu	Grafické zpracování	%	Počet zásahů
1	20834		2,54	166
2	20575		2,40	157
3	22861		2,37	155
4	27685		2,05	134
5	28380		1,90	124
6	20389		1,82	119
7	21270		1,80	118
8	24767		1,77	116
9	25216		1,76	115
10	22756		1,73	113
11	23256		1,62	106
12	22705		1,51	99
13	29238		1,50	98
14	27651		1,45	95
15	20397		1,44	94
16	27898		1,39	91
17	29459		1,36	89
18	28665		1,31	86
19	27561		1,30	85
20	27162		1,30	85
21	29904		1,28	84
22	21636		1,27	83
23	25003		1,24	81
24	20273		1,24	81
25	28789		1,22	80
26	24261		1,18	77
27	20761		1,16	76
28	28649		1,07	70
29	27424		1,04	68
30	27427		0,96	63
31	22969		0,93	61
32	24368		0,93	61
33	29453		0,92	60
34	20818		0,90	59
35	28967		0,89	58
36	24064		0,87	57
37	21881		0,87	57
38	22535		0,84	55
39	24198		0,83	54
40	20176		0,81	53

## Obsah přiloženého CD

CD přiložené k této diplomové práci obsahuje následující soubory:

- elektronická verze diplomové práce ve formátu PDF,
- zdrojové kódy řešení jednotlivých problémových míst v souborech TXT,
- zdrojový soubor použitý pro datamining,
- zdrojový soubor pro DFD a ERD (vytvořeno v programu CASE Studio),
- soubor datového proudu (streamu) pro program PAWS Modeler,
- grafický pohled datového streamu ve formátu PDF,
- celkový přehled všech servisů seřazených podle četnosti zásahů.