# EKONOMICKÁ UNIVERZITA V BRATISLAVE
# PODNIKOVOHOSPODÁRSKA FAKULTA
# SO SÍDLOM V KOŠICIACH


**Evidenčné číslo: 30600/D/2011/2723943226**

## MODEL FAKULTNÉHO KOMUNIKAČNÉHO SYSTÉMU NA UNIVERZITE EKONOMICKÉHO ZAMERANIA

**Dizertačná práca**

**2011** **Ing. Jaroslav Dugas**

# EKONOMICKÁ UNIVERZITA V BRATISLAVE
# PODNIKOVOHOSPODÁRSKA FAKULTA
# SO SÍDLOM V KOŠICIACH

**MODEL FAKULTNÉHO KOMUNIKAČNÉHO SYSTÉMU NA UNIVERZITE EKONOMICKÉHO ZAMERANIA**

**Dizertačná práca**

| | |
|---|---|
| **Študijný program:** | Ekonomika a manažment podniku |
| **Študijný odbor:** | 6284 9 00 Ekonomika a manažment podniku |
| **Školiace pracovisko:** | Katedra manažmentu |
| **Školiteľ:** | doc. Ing. Štefan Čarnický, PhD. |

**Košice 2011** **Ing. Jaroslav Dugas**

Zadanie záverečnej práce (vo vytlačenej verzii nahradiť stranou z AIS-u).

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum: .....

(podpis študenta)

Poďakovanie

Ďakujem týmto svojmu školiteľovi, doc. Ing. Štefanovi Čarnickému, PhD., za veľmi cenné rady, pripomienky a podporu pri vypracovaní práce. Moje veľké poďakovanie patrí aj Ing. Petrovi Zemanovi, ktorý nám umožnil odladiť obe prípadové štúdie - prvé priamo v jeho firme PeVyt a druhé v spoločnosti VALEO Slovakia, s. r. o. Košice, kde pôsobí ako externý konzultant na Informačné technológie.

# ABSTRAKT

DUGAS, Jaroslav: *Model fakultného komunikačného systému na univerzite ekonomického zamerania.* – Ekonomická univerzita v Bratislave, Podnikovohospodárska fakulta so sídlom v Košiciach, Katedra manažmentu – Vedúci dizertačnej práce: doc. Ing. Štefan Čarnický, PhD. – Košice : PHF EU, 2011, počet strán 220.

Cieľom dizertačnej práce bolo vytvoriť model fakultného komunikačného systému na univerzite ekonomického zamerania.

Práca je rozdelená do piatich kapitol. Obsahuje 46 obrázkov, 4 tabuľky a 9 príloh.

Dizertačná práca sa zaoberá problematikou implementácie komunikačného systému v univerzitnom prostredí. V prvej časti práce je spracovaná poznatková báza, ktorá obsahuje vybrané definície kľúčových pojmov, systémové vzťahy, analýzy a hodnotenia stavu rozvoja a trendov objektu skúmania. Ide najmä o vymedzenie stavu, v ktorom sa predmetná problematika nachádzala v čase riešenia dizertačnej práce. Hlavné zdroje informácií tvoria informácie z knižných publikácií, monografií, a najmä vedecké články z renomovaných domácich aj zahraničných odborných, vedeckých časopisov a z rôznych iných multidisciplinárnych časopisov.

Druhá časť uvádza formulácie stanovených cieľov dizertačnej práce vo vedecko-výskumnej oblasti rozvoja teórie, metodológie, metód a nástrojov vedného odboru a ciele zamerané do aplikačnej praxe, ako aj spôsob získavania údajov a poznatkov a ich informačné zdroje, je v nej popísaná štruktúra čiastkových cieľov, ktoré podporujú hlavný cieľ dizertačnej práce.

Tretia kapitola špecifikuje výsledky dizertačnej práce, ďalej vymedzuje zložky multidimenzionálneho objektu skúmania, pracovné postupy, metódy a techniky použité pri riešení dizertačnej práce, rieši tvorbu výkonnostných parametrov vhodných na posúdenie úspešnosti naimplementovaného modelu, ktoré umožnia kvantifikovať prínosy zo samotnej implementácie komunikačného systému.

Štvrtá kapitola vyhodnocuje výsledky práce, obsahuje diskusiu a polemizuje o možných ďalších smeroch implementácie modelu fakultného komunikačného systému.

V poslednej**,** piatej kapitole, sú zhrnuté všetky výsledky do prínosov, ktoré dizertačná práca nielen pre fakultu, ale aj pre iné univerzitné inštitúcie, taktiež aj pre jednotlivé vedné obory ako informačné systémy, manažment znalostí, projektový manažment a iné, môže prinášať.

**Kľúčové slová:**

komunikačný systém, fakultná komunikácia, implementácia, modelovanie, vývojový diagram, subjekty komunikácie

# ABSTRACT

DUGAS, Jaroslav: *Model Departmental communication system at a university economics.* - University of Economics in Bratislava, The Faculty of Business Economics with seat in Košice, Department of Management - Supervisor of doc. Ing. Štefan Čarnický, PhD. - Košice : PHF EU, 2011, p. 220.

The aim of the thesis is to develop a communication system for faculty at the University of economics. PhD thesis deals with the implementation of the communication system in the university environment.

In the first part is processed as theoretical review, that contains selected definitions of key concepts, interrelations, analysis and evaluation of development and trends in building investigations. In particular, the definition of the state in which the issue in question was situated at the time of solving of the dissertation. The main sources of information are details of books, monographs and scientific articles mainly from renowned domestic and foreign, scientific journals and various other multidisciplinary journals.

The second section gives the formulation of the objectives of dissertation research in the field of development of theory, methodology, methods and tools and also targets in the practical application and ways for data mining and knowledge gathering. The second part also contents the structure of partial goals, which supports the main objective of the dissertation thesis.

The third chapter specify results of dissertation thesis and also defines components of multidimensional object of investigation, working practices, methods and techniques used for solving dissertation thesis, deals with designing of performance parameters for assessing the success of implemented model, which allows to quantify the benefits of implementation of the communication system.

The fourth chapter evaluates the results of our work, contains discussion and take a look on other possible ways of further implementation of the model faculty communication system.

In the final, fifth chapter summarizes all results in benefits that thesis, not only for our faculty but also for other university institutions, as well as for individual disciplines such

as information systems, knowledge management, project management and others can bring.

**Keywords:**

communication system, faculty communication, implementation, modeling, flowchart, communications operators

# OBSAH

**Úvod** ..... **15**

**1 Súčasný stav riešenej problematiky doma a v zahraničí** ..... **18**

*1.1 Genéza implementácií komunikačných systémov* ..... *18*

*1.2 Vymedzenie základných pojmov* ..... *19*

*1.3 Tradičné modely implementácií komunikačných systémov* ..... *39*

*1.4 Porovnanie tradičných modelov implementácií komunikačných systémov* ..... *57*

*1.5 Alternatívne modely implementácií komunikačných systémov* ..... *62*

*1.6 Teória hodnotenia zo zavádzania komunikačných systémov* ..... *71*

*1.7 Fakultná komunikácia* ..... *84*

**2 Cieľ práce** ..... **87**

**3 Metodika práce a metódy skúmania** ..... **90**

*3.1 Popis objektu skúmania* ..... *90*

*3.2 Metodika práce* ..... *93*

*3.3 Postup riešenia dizertačnej práce* ..... *124*

*3.4 Tvorba výkonnostných parametrov vhodných na posúdenie prínosov z implementácie modelu fakultného komunikačného systému* ..... *139*

**4 Výsledky práce** ..... **155**

*4.1 Prípadová štúdia 1* ..... *155*

*4.2 Prípadová štúdia 2* ..... *177*

*4.3 Úpravy navrhovaného modelu fakultného komunikačného systému* ..... *184*

**5 Diskusia** ..... **192**

*5.1 Ďalšie smery vedeckého výskumu implementácií komunikačných systémov* ..... *192*

*5.2 Prínosy pre fakultu* ..... *195*

*5.3 Prípadné prínosy pre iné univerzitné inštitúcie* ..... *198*

*5.4 Prípadné prínosy pre teóriu manažmentu znalostí, teóriu informačných technológii, teóriu komunikačných systémov* ..... *198*

**Záver** ..... **200**

**Zoznam použitej literatúry** ..... **201**

**Prílohy** ..... **211**

# Zoznam obrázkov

Obrázok 1 Miesto BI a dvoch základných vrstiev v pyramíde ISP ..... 27

Obrázok 2 Hlavné komponenty BI a ich väzby v riešení BI ..... 29

Obrázok 3 Všeobecná koncepcia architektúry BI ..... 30

Obrázok 4 Vzťah subject - model - original ..... 96

Obrázok 5 Grafické znázornenie metód systémovej analýzy ..... 99

Obrázok 6 Základné procesy, operácie a funkcie prebiehajúce v rámci systému ..... 104

Obrázok 7 Všeobecný model KS ..... 107

Obrázok 8 Životný cyklus informácie v KS ..... 110

Obrázok 9 Príklad vzťahov medzi prvkami v systéme ..... 114

Obrázok 10 Kroky riešenia dizertačnej práce ..... 124

Obrázok 11 Metodika tvorby ERA diagram ..... 127

Obrázok 12 Kontextový diagram ..... 131

Obrázok 13 Diagram dátových tokov ..... 132

Obrázok 14 Príklad objektovej metodológie ..... 133

Obrázok 15 Hierarchické členenie vývojového diagram ..... 154

Obrázok 16 Názorné nastavenie e-mailového konta na školský poštový server ..... 158

Obrázok 17 Názorné nastavenie e-mailového konta na prihlasovací mimoškolský poštový server ..... 159

Obrázok 18 Nastavenie internetového e-mailu - server odosielanej pošty ..... 160

Obrázok 19 Nastavenie internetového e-mailu - server prichádzajúcej pošty ..... 160

Obrázok 20 Test nastavenia internetového konta ..... 161

Obrázok 21 Zadanie termínu splnenia úlohy ..... 162

Obrázok 22 Možnosti spracovania úlohy ..... 163

Obrázok 23 Vyhľadávanie e-mailov pomocou kľúčových slov ..... 164

Obrázok 24 Nastavenie možností vyhľadávania ..... 165

Obrázok 25 Vyhľadávanie kontaktov ..... 166

Obrázok 26 Zoznam kontaktov ..... 167

Obrázok 27 Možnosti poslania správy alebo úlohy konkrétnej osobe ..... 168

Obrázok 28 Zoznam aktuálnych úloh ..... 169

Obrázok 29 Zadanie novej úlohy a jej riešiteľa ..... 170

Obrázok 30 Triedenie úloh podľa kategórií ..... 171

Obrázok 31 Zadanie konkrétnej úlohy ..... 173

Obrázok 32 Triedenie úloh podľa zadefinovaných kategórií ..... 174

Obrázok 33 Logo softvéru Windows Small Business Server ..... 176

Obrázok 34 Print screen medzinárodnej webovej stránky spoločnosti VALEO ..... 178

Obrázok 35 Úvodné užívateľské okno Lotusu ..... 179

Obrázok 36 Nástroj na zadávanie úloh Help Desk VSS ..... 180

Obrázok 37 Zoznam riešiteľov úloh ..... 181

Obrázok 38 Skrátený zoznam úloh konkrétnych riešiteľov ..... 182

Obrázok 39 Presné zadefinovanie úlohy ..... 184

Obrázok 40 Možnosť zaslania úlohy na e-mail konkrétho užívateľa ..... 186

Obrázok 41 Odkaz na riešenie úlohy pri spolupráci Lotusu a googlu ..... 187

Obrázok 42 Znázornenie riešiteľov úloh pri spolupráci Lotusu a googlu ..... 188

Obrázok 43 Skrátený popis zadanej úlohy ..... 189

Obrázok 44 Znázornenie popisu konkrétnej úlohy ..... 190

Obrázok 45 Komentár riešiteľa úlohy zadávateľovi ..... 191

Obrázok 46 LUCID - súčasti balíka modelovacích techník ..... 194

# Zoznam tabuliek

Tabuľka 1 Fakultná komunikácia ..... 86

Tabuľka 2 Možné bariéry komunikácie ..... 92

Tabuľka 3 Metódy systémovej analýzy ..... 98

Tabuľka 4 Inžinierske disciplíny ..... 101

# Úvod

Akákoľvek komunikácia by mala byť základným a integrujúcim nástrojom riadenia akéhokoľvek podniku, firmy či inštitúcie, ktorý by mal slúžiť na „spájanie ľudí“ a ich „zjednocovanie“, za účelom dosahovania stanovených cieľov. So samotným vstupom Slovenskej republiky do Európskej únie sa slovenské inštitúcie dostali do globálnejšieho, náročnejšieho, a hlavne aj konkurenčného prostredia, a ak chcú v tomto náročnom prostredí aj úspešne obstáť, musia sa naplno zapojiť do širokého európskeho vzdelávacieho veľkopriestoru a zároveň musia dobiehať rozbehnuté trendy implementácií komunikačných systémov často podľa vzoru skúsených západoeurópskych partnerov.

Ešte v marci 2000 sa najvyšší predstavitelia štátov a vlád EÚ na rokovaní v Lisabone dohodli na ambicióznom cieli: urobiť z EÚ do roku 2010 najkonkurencieschopnejšiu a najdynamickejšiu poznatkovo orientovanú ekonomiku sveta (http://kiwiki.fmtnuni.sk). V roku 2005 bola táto Lisabonská stratégia revidovaná a jej cieleným výberom a zoskupením tém z viacerých politík EÚ okrem iných sú hlavne výskum, vzdelávanie a informačná spoločnosť (http://www.pcworld.sk). Od r. 2008 sa v rámci EÚ na úrovni národných koordinátorov diskutovalo aj o stratégii po r. 2010. Do diskusií prispela aj Slovenská republika, v júni 2009 aj písomne - v Predbežnom stanovisku SR ku stratégii po r. 2010. Na základe stanovísk a diskusií vznikol návrh stratégie po r. 2010 - Európa 2020.

Uvedomujeme si, že v posledných rokoch sa čoraz častejšie stretávame s názorom, že samotným kľúčom k úspechu v akomkoľvek konkurenčnom prostredí sa stáva čo efektívnejšie využívanie schopností, kreativity, znalostí, vnorené do svojho vybudovaného komunikačného systému. Vo vyspelejších západných štátoch, na rozdiel od bývalých postkomunistických krajín, už skôr a hlavne vo väčšej miere dochádzalo k obrovskej množine dôležitých významných technických, technologických, podnikateľských, sociálnych, ekonomických, ale aj vzdelávacích zmien, ktoré mali a majú nepostrádateľný vplyv na akékoľvek manažérske myslenie, ktoré sa vyvíjalo v silnej závislosti od politicko-spoločenského diania, od svojho blízkeho okolia, v ktorom museli samotní títo manažéri žiť, pôsobiť, konať a hlavne komunikovať. Časom silnejúca globalizácia a zároveň zvyšujúca sa konkurencia zrodili novú filozofiu aj inštitucionálneho komunikačného

manažmentu. Zrodil sa nový fenomén systému získavania bohatstva založený práve na informáciách a naimplementovaných komunikačných systémoch, ktorý začína stále viac a viac ovplyvňovať vývoj každej inštitúcie (Jablonský, 2002).

Aby akákoľvek inštitúcia mohla prijímať tie najsprávnejšie efektívne strategické rozhodnutia, je potrebné, aby veľmi dobre poznala svoje interné a externé prostredie, t. j. aby poznala možné obmedzenia svojich strategických rozhodnutí. Z takto formulovaných obmedzení pre efektívne prijímanie možných strategických rozhodnutí je jasná prvoradá úloha neustáleho skúmania a vyhodnocovania interného a externého prostredia a riadeného využívania informácií a znalostí s cieľom ich maximálne účelného využitia vo svojom komunikačnom systéme. A to je možné len pri naimplementovanom, vybudovanom a praxou overenom efektívnom komplexnom, spravodajskom, informačnom systéme.

Aj pri akomkoľvek dokonalom naimplementovanom komunikačnom systéme s obrovským nárastom informácií rastie aj potreba zadefinovaných rozlišovacích kritérií. Často je problémom rozlíšiť medzi informáciou spoľahlivou a manipulovanou, medzi zdrojom serióznym a napríklad aj nezmyselným. Schopnosť informácie hodnotiť patrí síce k schopnostiam orientovania sa vo svete informácií, hodnotenie informácií je preto tiež jednou z podstatných zložiek informačnej gramotnosti, ale stručne povedané, stále ide o rozpoznanie a vytriedenie potrebných informácií a to len a len pri správnom zavedení komunikačného systému.

Na druhej strane - pokiaľ je dostupnosť kvalitných informácií pre jednotlivých vedúcich manažérov rôzna, tak tí manažéri, ktorí majú k takýmto informáciám prístup, majú lepšie predpoklady pracovať kvalitnejšie. Je veľký predpoklad, že vedúci manažéri pomocou týchto informácií môžu získať konkurenčnú výhodu, kde cieľom môžu byť nielen nové vzácne príležitosti, ale v dnešnej hospodárskej recesii bohužiaľ veľmi aktuálne - a to eliminácia a zníženie rôznych negatívnych rizík. Ten, kto včas vlastní informácie a vie ich aj náležite využiť, vie aj tvoriť podmienky prísunu ďalších informácií. A tak teda včasné, relevantné, to znamená využiteľné kvalitné informácie prúdiace z komunikačného systému sú liahňou nových konkurenčných východ a vysnívaných pracovných úspechov. K tomu je však potrebné vytvoriť také komunikačné informačné prostredie, v ktorom je získavanie informácií, rovnako ako aj ich vzájomná výmena tou najvyššou prioritou.

Tematiku zavádzania modelu fakultného komunikačného systému na univerzite ekonomického zamerania rieši aj predkladaná dizertačná práca. Východiskom pre riešenie sú pôvodné odborné práce, teórie, analýzy, metodiky a postupy.

V prvej časti spracovaná poznatková báza obsahuje vybrané definície kľúčových pojmov, systémové vzťahy, analýzy a hodnotenia stavu rozvoja a trendov objektu skúmania. Ide najmä o vymedzenie stavu, v ktorom sa predmetná problematika nachádzala v čase riešenia dizertačnej práce. Hlavné zdroje informácií tvoria informácie z knižných publikácií, monografií, a najmä vedecké články z renomovaných domácich a zahraničných odborných a vedeckých časopisov, z rôznych iných multidisciplinárnych časopisov, ďalej to sú zdroje z teórie informačných technológií, teórie komunikačných systémov, ako aj zdroje z personálneho manažmentu.

Druhá časť uvádza formulácie stanovených cieľov dizertačnej práce vo vedecko - výskumnej oblasti rozvoja teórie, metodológie, metód a nástrojov vedného odboru a ciele zamerané do aplikačnej praxe, ako aj spôsob získavania údajov a poznatkov a ich informačné zdroje, je v nej popísaná štruktúra čiastkových cieľov, ktoré podporujú hlavný cieľ dizertačnej práce.

Tretia kapitola vymedzuje zložky multidimenzionálneho objektu skúmania, pracovné postupy, metódy a techniky použité pri riešení dizertačnej práce, rieši tvorbu výkonnostných parametrov vhodných na posúdenie úspešnosti naimplementovaného modelu, ktoré umožnia kvantifikovať prínosy zo samotnej implementácie komunikačného systému.

Štvrtá kapitola vyhodnocuje výsledky dizertačnej práce a polemizuje o možných ďalších smeroch implementácie modelu fakultného komunikačného systému. Z tohto hľadiska bolo potrebné podrobne rozviesť súčasný nekomunikačný stav v oblasti fakultnej informačnej sústavy a popísať aj navrhnúť prípadové štúdie modelu fakultného komunikačného systému.

V poslednej, piatej kapitole, sú zhrnuté všetky výsledky do prínosov, ktoré dizertačná práca nielen pre fakultu, alebo iné univerzitné inštitúcie, ale aj pre jednotlivé vedné obory ako manažment znalostí, informačné technológie, komunikačné systémy a iné, môže prinášať.

# 1 Súčasný stav riešenej problematiky doma a v zahraničí

Žijeme v pretechnizovanom svete, máme k dispozícii nevídané možnosti komunikácie. Nové komunikačné technológie nám ponúkajú rôzne nové riešenia. Aj preto je komunikácia v akejkoľvek inštitúcii čoraz dôležitejšia a dnes už zďaleka presahuje rámec osobnej komunikácie, poprípade telefónneho komunikovania. Bolo by ideálne, keby všetci zamestnanci PHF EU, nielen vedenie, prodekani a vedúci katedier, mali vytvorenú takú komunikačnú sieť, ktorá by zodpovedala všetkým informačným potrebám a požiadavkám akéhokoľvek účastníka tejto siete.

## 1.1 Genéza implementácií komunikačných systémov

Akýkoľvek komunikačný systém je definovaný ako množina ľudí, technických prostriedkov, aplikačného programového vybavenia a metód, zabezpečujúcich zber, prenos, uchovávanie a spracovanie dát za účelom tvorby a prezentácie informácií pre potreby užívateľov činných v systémoch riadenia (Ivanička, 2000).

Rok 1989 bol zlomom aj v oblasti tvorby a využitia komunikačných systémov. Pred týmto obdobím sa u nás, podobne ako v celom východnom bloku, veľmi výrazne prejavoval nedostatok spolupráce s vyspelými štátmi. Izolácia tejto časti sveta zo strany západu vyprovokovala vlastný výskum a vývoj, bohužiaľ aj kopírovanie hardvérových a softvérových prostriedkov. V rámci RVHP vznikla jednotná rada počítačov JSEP a SMEP, vyvíjalo sa pre ne aj základné programové vybavenie od operačných systémov, vyvíjali sa rôzne programovacie jazyky.

Musíme konštatovať, že tieto plagiáty však nemohli dosiahnuť v žiadnom prípade kvalitatívnu úroveň svojich originálov. Veľkým prínosom bol však fakt, že naši výskumní inžinieri dokonale prenikli do podstaty hardvérového a základného softvérového vybavenia počítačov renomovaných značiek, ako v tom čase boli napríklad IBM, DIGITAL a ďalšie. Na poli užívateľských aplikácií bola situácia lepšia, v početných výskumných tímoch vznikali odvetvové typové aplikačné projekty, ktoré boli analyticky dobre zvládnuté, no

žiaľ v technickom prevedení dosť zaostávali. Pre ich označenie sa vžil názov ASR - automatizované systémy riadenia, alebo aj AIS - automatizované informačné systémy.

Po roku 1989 sa nám otvorili hranice a dovtedy nechránený a hlavne neskúsený východný trh sa stal odpadkovým košom vyspelých a „skúsených“ veľkých firiem. Začali sa dovážať repasované počítače, ktoré už boli tak morálne zastarané, že sa hodili už len na počítačové hry, alebo na smetisko. Na západe vtedy začínal „PC boom“, no bolo to oveľa rozumnejšie ako u nás. Brala sa do úvahy kontinuita vývoja cez veľké sálové počítače vo výpočtových strediskách, až postupne k prvotným počítačovým sieťam s PC (personal computer).

Po čase sa aj k nám dostal „PC boom“, ale aplikoval sa „ HURÁ SYSTÉM“, ktorý odsúdil na zánik výpočtové strediská. PC sa začali považovať za jediný správny prostriedok pre technické zabezpečenie komunikačných systémov. Do vývoja týchto systémov sa pustili pracovníci bývalých výpočtových stredísk a analytických skupín, vznikalo veľa tzv. garážových firiem, ktoré si nemohli dovoliť dávať prostriedky do vývoja, ale čerpali ešte z minulých skúseností. Musíme pripomenúť, že v tejto oblasti sa dali získať pomerne slušné finančné prostriedky, pretože neskúsení a dôverčiví, ba možno niekedy aj podplatiteľní manažéri v nových štátnych ale aj v súkromných firmách začali budovať svoj vlastný informačný, komunikačný systém práve na platforme izolovaných PC s riešením od garážových firiem. Neuvedomili si, že ich systém musí mať aj nejaké predpoklady ďalšieho vývoja. A práve tieto firmy neboli toho zárukou. Preto v dnešnom období, po skonsolidovaní perspektívnych firiem v jednotlivých výrobných odvetviach, vystupuje do popredia kvalita informačných, komunikačných systémov.

Kvalitné informačné zabezpečenie sa podieľa podstatným vplyvom na úspechu firmy. Úspešné a solventné firmy sa preto dnes už orientujú na renomovaných dodávateľov informačných technológií a komunikačných systémov. A práve v tejto ich snahe im výrazne môže pomôcť rozpracovaná metodika noriem riadenia kvality ISO pre podmienky vývoja, výberu, implementácie, prevádzky a údržby informačných systémov.

## 1.2 Vymedzenie základných pojmov

**Dáta** - sú skratkové profesionálne označenie pre čísla, text, zvuk, obraz, prípadne ďalšie

zmyslové vnemy, všeobecne výroky popisujúce realitu. Dáta sú základom informačného bohatstva firmy alebo organizácie. Dáta do IS vstupujú, sú v ňom uložené, uchovávané, spracovávané a z nich sa okamžite alebo neskôr získavajú najrôznejšie informácie.

**Typy dát** - všetky IS zvyčajne obsahujú:

- aktuálne dáta - opisujú súčasný skutočný stav. Akonáhle stratia svoju aktuálnu platnosť, stávajú sa z nich dáta archívne,
- archívne dáta - vznikajú z aktuálnych dát po preklenutie ich platnosti. Sú uchovávané pre neskoršie analýzy histórie, vývoja a zmien,
- prognostické dáta - sú plánovanými dátami, vznikajú z rôznych plánov, výhľadov, návrhov a neskôr sa môžu stať dátami aktuálnymi.

**Uloženie dát** - dáta sú uložené v dátových bázach. Spôsob ich uloženia ovplyvňuje neskoršie spracovanie týchto dát a má obrovský vplyv na neskoršiu efektivitu práce s dátami. Vzhľadom na rozsah uchovaných dát majú prehľadové informácie získané z aktuálnych a archívnych dát nedoceniteľnú hodnotu pre manažment podniku.

**Informácie** - sú to dáta, ktorým užívateľ v procese interpretácie prisudzuje určitý význam, ktoré mu slúžia napríklad k riešeniu konkrétneho problému, napomáhajú pri určitých rozhodnutiach. Sú získavané z dát rôznymi operáciami, napríklad: zaraďovaním, výberom, súhrnom a podobne. Je to veličina znižujúca neurčitosť rozhodovateľa a to tak, že uľahčujú rozhodovanie v zmysle lepšieho výsledku konečného rozhodnutia. Hodnota informácie má individuálny charakter pre daného príjemcu, pretože ju posudzuje sám, podľa svojich znalostí.

Kvalitná práca s informáciami, dátami a znalosťami je jednou z najdôležitejších konkurenčných výhod každej organizácie. S informáciami je tiež nutné nakladať veľmi obozretne, aby nedošlo ku zdieľaniu informácií ľuďmi, ktorým sa informácie dostať nemali. Je nutné tiež dbať na ochranu citlivých informácií, ktoré sa môžu stať až cieľom priemyselnej špionáže. Súbory rôzne usporiadaných dát sú označované ako databázy či dátové zdroje, pre ich jednoduchšie použitie a manipuláciu sú využívané informačné systémy.

**Atribúty informácií** - pri práci s informáciami musíme sledovať nasledujúce atribúty:

- včasnosť, dostupnosť, spoľahlivosť prísunu informácií,
- obsah (aktuálnosť, relevantnosť, pravdivosť, objektívnosť),

- formát,
- cenu a úžitkovú hodnotu,
- legálnosť nadobudnutia.

**Informačný šum** - je to náhodná, alebo nechcená, skresľujúca informácia, ktorá sprevádza vo väčšej či menšej miere každú prenášanú alebo odovzdávanú informáciu. Šum má rôznu intenzitu, od nepatrnej, ktorá prenášanú informáciu prakticky vôbec neovplyvní, až po tak vysokú, že v nej odovzdávané informácie úplne zaniknú. Technický šum je rušivý signál v prenosovom kanáli.

**Znalosti** - na rozdiel od definícií dát a informácií, ktoré sú pomerne jednoduché a logické, so znalosťami je to trochu zložitejšie. Chápeme ich ako premenný systém so vzájomnou interakciou skúseností, faktov, vzťahov, myšlienkových pochodov a významov. Sú výsledkom aktívneho učenia sa na základe vlastného poznania, skúseností, celoživotného vzdelávania, a v menšej miere pôsobením výučby.

**Informačný systém** - pochopenie čo Informačný systém (IS) vlastne je, sa menilo nielen historicky ale aj v súčasnej dobe závisí na uhle pohľadu, respektíve na priestore, v ktorom sa IS nachádzajú. Obecne je IS chápaný ako systém na zber, prechovávanie, triedenie a distribuovanie informácií. Z tohto hľadiska nemusí byť nutne IS automatizovaný pomocou rôznych Informačných a komunikačných technológií (IKT), ale môže byť výnimočne celý zrealizovaný manuálne - napríklad vo forme nejakej kartotéky. V prípade takejto kartotéky sú procesy IS realizovávané zamestnancami danej organizácie pracujúcimi s touto kartotékou a dáta sú uchovávané na konkrétnych kartách v papierovej podobe. Tento príklad nie je vymyslený, pretože IS existovali dlho pred vznikom prvých počítačov alebo iných nástrojov IKT, ktoré automatizáciu zabezpečovali. Informácie boli zhromažďované a spracovávané už v dávnej minulosti a automatizácia sa do takýchto informačných systémov dostala postupne s vývojom vedeckotechnickej revolúcie.

Aj z pohľadu tejto dizertačnej práce sú však zaujímavé práve IS, ktoré už používajú PC na zber a spracovávanie dát, aj keď stále platí, že mnohé procesy v IS sú a aj naďalej môžu byť z rôznych dôvodov zrealizovávané manuálne. Príkladom takéhoto nevyhnutného manuálneho spracovania môžu byť napríklad došlé listové zásielky, kde je nutný ľudský zásah pre spracovanie takýchto zásielok.

Ďalšia definícia IS nás posúva dopredu, bližšie k ich súčasnému chápaniu: IS je systém zberu, uchovávania, analýzy a prezentácie dát určený pre poskytovanie informácií

mnohým užívateľom rôznych profesií. IS teda môže a nemusí byť podporovaný počítačmi. IS často využíva počítačom podporované a teda automatizované, ale aj neautomatizované, manuálne spôsoby práce. Voľba optimálnej kombinácie automatizovaných a neautomatizovaných činností je základným problémom návrhu konkrétneho IS (Ivanička, 2000). Táto definícia už explicitne zdôrazňuje veľmi dôležitú rolu automatizácie procesov v IS, ale vôbec nenaznačuje ich prevahu.

S manuálnymi aktivitami v IS sa stretávame vždy, aj keď vo väčšine prípadov je možné tieto manuálne činnosti zanedbať pomocou stanovenia hraníc IS. Zanedbanie týchto hraníc je možné iba v takom prípade, keď nie sú dôležité v životnom cykle zadefinovaného systému. Príkladom pre takúto činnosť môže byť napríklad prepisovanie došlej pošty v papierovej forme do IS. Z pohľadu IS sú tieto „papiere“ nedôležité, keďže skutočný vstup informácií do systému je zadávanie dát zodpovedným pracovníkom inštitúcie, z hľadiska návrhárov tohto IS sú však tieto papiere mimo definovaných hraníc systému a preto sú zanedbané.

Preto aj konkrétna množina manuálnych a automatizovaných procesov IS závisí od toho, ako zúžime priestor IS z najobecnejšieho chápania tohto pojmu. Toto zmenšenie priestoru sa riadi tým, čo je potrebné pre samotný životný cyklus tohto IS, respektíve to čo je potrebné. S IS, v ktorých prevažná väčšina procesov je automatizovaná pomocou IKT, sa dnes stretávame na každom kroku vo všetkých oblastiach, profesiách, a ich používanie je v dnešnej dobe bežné v najrôznejších organizáciách od škôl, vládnych inštitúcií, až po výrobné podniky.

V nasledujúcom texte zmapujeme typy IS práve z hľadiska organizácie a účelu v organizácii, čo je aj najčastejšie miesto ich naimplementovania. Takto zadefinované IS sú už podmienené množinou rôznych špecifikácií, ktoré sú popísané v nasledujúcej definícií: IS budeme nazývať celok, ktorý je softvérovým vybavením schopným komplexne riadiť veľké podniky. Základom takýchto produktov budú systémy pre spracovanie dát – Databázové systémy. Toto jadro bude obklopené ďalšími komponentmi – podpornými systémami riadenia tímovej práce, systémami automatizovaného riadenia procesov, grafickými systémami a ďalšími potrebnými zadefinovanými časťami (Basl - Blažíček, 2008). Táto definícia nám približuje, čo realizujú a aké procesy podporujú IS v organizácii.

Musíme zdôrazniť, že takýchto IS môže byť v organizácii aj viacero. Pritom podporujú organizáciu a jej činnosť z rôznych strán, a tieto IS majú vždy jediný cieľ - slúžiť a pomáhať všetkým používateľom v ich každodennej práci. V typickej organizácii

môžeme nájsť množinu funkcií, ktoré sú predmetom podnikania, alebo sú doplnkové a zároveň podporné pre samotnú činnosť organizácie. IS majú za úlohu podporovať všetky funkcie podniku, ktoré si to vyžadujú, alebo tie funkcie, ktoré sa majú zefektívniť. Tieto IS môžu, ale aj nemusia navzájom zdieľať spoločné dáta či spolupracovať a stupeň ich vzájomnej previazanosti je závislý na konkrétnych podmienkach a potrebách organizácie.

Všeobecne existuje mnoho druhov IS realizovaných pomocou IKT a líšiacich sa podľa účelu, alebo oblasti nasadenia. Môžu to byť napríklad Študijné informačné systémy, Kamerové, monitorovacie informačné systémy, Dochádzkové systémy, Informačné systémy plynulej dopravy a ďalšie. Oblasti životného cyklu všetkých IS - analýzy, návrhy a vývoje - sú dynamicky sa rozvíjajúce, a preto presné všeobecné rozdelenie či nejaká kategorizácia informačných systémov v podstate neexistuje.

V nasledujúcom texte podrobnejšie rozoberieme typy IS v organizáciách, účel ich implementácie a niektoré charakteristiky. V každej organizácii môžeme nájsť dva základné typy IS: Operačný a Manažérsky IS. Ďalšie IS môžeme začleniť do kategórie expertných alebo znalostných IS a nakoniec sú to špeciálne IS, plniace špecifické úlohy najčastejšie v závislosti na konkrétnej činnosti tej ktorej organizácie, pritom toto zaradenie do týchto kategórií nemusí byť konečné, pretože je možné nájsť rôzne prelínajúce sa a kombinované v závislosti na podmienkach organizácie.

**Manažérske informačné systémy** - Manažérsky informačný systém (MIS) je možné definovať ako systém na podporu riadenia organizácie, ktorý slúži funkciám plánovania, kontroly a rozhodovania poskytovaním pravidelných prehľadov a hlásení o výnimočných situáciách. MIS sú odlišné od obyčajných IS v tom, že používajú odlišné IS a aplikujú ich v operatívnych aktivitách organizácie (Čarnický - Mesároš, 2009).

Niektorí autori u nás chápu MIS skôr užšie ako IS len na strategickej úrovni riadenia podniku, teda na vrchole riadiacej pyramídy. Najmä v zahraničí však prevláda širšie chápanie MIS, pri ktorom sa tieto IS využívajú nielen na strategickej, ale aj na manažérskej úrovni riadenia organizácie. Pojem MIS sa používa pre skupinu metód riadenia informácií spojených s automatizáciou a podporou rozhodovania a teda prinášajú strategické výhody tým, že zvyšujú rýchlosť a kvalitu rozhodovania najvyšších úrovní riadenia organizácie. K tomu je nutné zabezpečiť prístup k databázam, k on-line výpočtom a k iným ukazovateľom, kde zdrojom dát môžu byť napríklad aj operatívne informačné systémy a celkove v MIS sa používajú informácie globálneho charakteru „vypočítané“ z veľkých objemov získaných celoorganizačných dát.

Typickým príkladom MIS sú systémy na podporu rozhodovania (DSS – Decision Support Systems), ktoré sú nápomocné všetkým užívateľom pri plnení riadiacich a rozhodovacích činností. Jednotlivému užívateľovi umožňujú porovnávať dielčie výsledky s jeho predstavami a na základe toho má tento užívateľ možnosť ovplyvňovať ďalší priebeh svojho riešenia. DSS môžu teda poskytnúť užívateľovi ponuku riešení a prípadne pomocou ďalších presne zadefinovaných otázok usmerňovať jeho postup. Tieto systémy však nedokážu nahradiť konkrétnu osobu, ktorá vykonáva rozhodnutia, a výsledkom týchto systémov teda nie je rozhodnutie, ale iba poskytujú súbor možných variant, ďalej spresňujú a urýchľujú výpočty a kvantifikujú možné riziká.

Systém plánovania zdrojov (ERP – Enterprise Resource Planning) je MIS, ktorý integruje a automatizuje veľké množstvo procesov súvisiacich s produkčnými činnosťami výrobného podniku. Prakticky sa jedná o výrobu, logistiku, distribúciu, predaj, fakturáciu, účtovníctvo, správu majetku, a podobne. Kľúčovým prvkom ERP systémov je použitie unifikovanej databázy na uchovávanie dát pre rôzne systémové moduly. Pojem ERP vznikol ako rozšírenie systémov pre plánovanie zdrojov pre výrobu (MRP – Manufacturing Resource Planning) (Čarnický - Mesároš, 2009).

**Operačné informačné systémy** - táto dôležitá skupina IS v podniku sú operačné, respektíve operatívne informačné systémy. OIS sú používané pre riadenie každodenných operácií v podniku. Operatíva je súbor denných činností zahrňujúcich napríklad účtovníctvo, vedenie skladovej evidencie, finančné operácie, riadenie výrobných procesov, sledovanie majetku, knihy jázd služobných automobilov, dochádzka a podobne.

Pre OIS sú typické nasledujúce charakteristické vlastnosti: údaje s ktorými sa v určitom okamihu pracuje, nemajú príliš veľký rozsah a tvoria určitý uzavretý celok - môže to byť napríklad zoznam hnuteľného majetku a priamo súvisiace informácie o položkách na zozname, ako je geografická poloha a vlastník položky na zozname. Určitý typ dát sa vyskytuje v mnohých inštanciách – je viacero zoznamov majetku. Úlohou OIS je vyhľadať relevantné dáta.

Ďalšou typickou vlastnosťou týchto systémov je, že výber a význam operácií je zadefinovaný už vo fáze špecifikácie požiadaviek na systém. OIS prinášajú skôr taktické výhody, ktoré sa obvykle týkajú vnútorného chodu podniku, majú preň krátkodobý efekt a nemajú nejaký významný vplyv napríklad na správanie organizácie na trhu. Príkladom takéhoto vnútroorganizačného a zároveň aj taktického prínosu je zníženie stavu zásob. Oproti MIS sa líši pri OIS pohľad na konkrétne dáta v systéme. U OIS sa potreba dát

odvíja od konkrétnych presne zadefinovaných operácií, ale u MIS a ich zdokonalenej variante EIS – (Exekutívne Informačné Systémy) nie je možné dopredu potrebné dáta len tak jednoducho vymedziť. Preto sa u MIS často zhromažďujú všetky dáta, u ktorých je čo i len malý predpoklad zmysluplného využitia.

Príkladom jednoduchých OIS je Rezervačný informačný systém pre skriptá a učebnice, alebo iné publikácie ako v knižnej, tak v elektronickej podobe (CD, DVD). Príkladom celistvejšieho a hlavne komplexnejšieho OIS je systém pre podporu riadenia vzťahov so zákazníkmi (CRM – Customer Relationship Management), ktorý pomáha organizácii riadiť celý cyklus kontaktu so zákazníkom, ktorý podporuje efektívnu koordináciu väzieb na zákazníka či starostlivosť o tohto zákazníka. Oddelenia, ktoré využívajú takýto systém, sú napríklad marketing, predaj, zákaznícke služby, školenia, personalistika, a podobne. Mnohokrát pod pojmom CRM rozumieme nielen nejaký IS pre podporu uvedených funkcií ale častokrát aj akúsi metodiku prístupu ku zákazníkom, ktorá má za účel pomôcť organizáciám pochopiť a predvídať potreby aktuálnych ale aj ďalších - potenciálnych zákazníkov.

Iným príkladom komplexnejšieho OIS je systém na riadenie štúdia (LMS – Learning Management System). Tento OIS slúži na podporu organizácie a administratívy vzdelávania v organizácii. Príklady činností podporovaných takýmto systémom sú napríklad evidencia študentov, rezervačný systém výukových miestností, podpora plánovania výukových kurzov a podobne. V poslednej dobe slúžia LMS hlavne pre organizáciu a administratívu on-line štúdia – eLearningu. LMS je teda súbor aplikácií, ktoré v sebe integrujú aj nástroje pre komunikáciu medzi účastníkmi kurzov, riadenie kurzov a mnohé ďalšie ako výveska, diskusné fóra, informačné tabule, evidencie, testy atď., a zároveň sprístupňujú účastníkom týchto kurzov učebné materiály a iný výukový obsah a to ako on-line, tak aj off-line. Požitie LMS v kombinácií s nástrojmi eLearningu sa stávajú stále populárnejšou formou vzdelávania v organizáciách a to vďaka mnohým významným výhodám, ako sú nízke náklady, neustála dostupnosť, rôznorodá a neustále obmeňujúca sa forma obsahu a podobne.

**Exekutívne informačné systémy a Business intelligence -** MIS a OIS je možné používať ako samostatné systémy. IS, ktorý plní obe funkcie OIS aj MIS, umožňuje podstatné zvýšenie kvality riadenia organizácie. Niekedy sa takýto IS nazýva EIS (Exekutívny informačný systém) a považuje sa za špeciálny typ systému na podporu rozhodovania.

Obecne sa dá povedať, že to sú celoorganizačné DSS systémy, ktoré pomáhajú analyzovať, porovnávať ale hlavne nachádzať správne smery, napomáhajú monitorovať výkonnosť v organizácii a môžu identifikovať nové príležitosti, a hlavne dokážu predchádzať akýmkoľvek problémom.

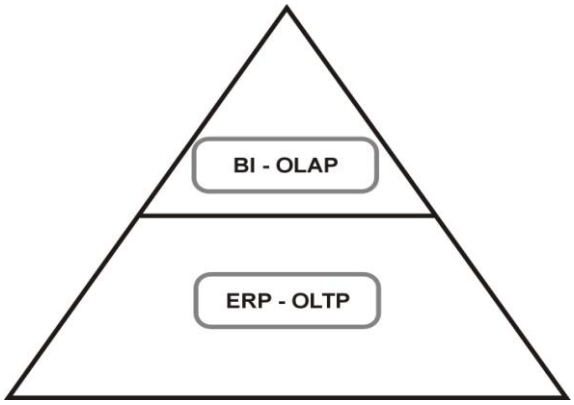
V poslednej dobe je pojem EIS nahradzovaný novým pojmom: BI (Business Intelligence) - je to súbor plánov, konceptov a konkrétnych metód na zlepšenie rozhodovania v organizácii za použitia podporných systémoch, založených na všeobecných obchodných faktoch a je nadradený procesom spravodajstva v obchodnom segmente a pôvodnému konceptu EIS, keďže sú sledované, zhromažďované a analyzované údaje o obchodnom prostredí ako celku, nielen o zákazníkoch, trhu a konkurencií.

Pojem BI sa používa, ak hovoríme o rôznych moderných nástrojoch správy, analýzy, zhromažďovania a vyhodnocovania veľkých objemov dát, ich ukladaním, správou a dolovaním znalostí, a ich potreba vyplýva najmä z nutnosti účinnejšieho využívania súčasného veľkého informačného potenciálu organizácií na podporu ich riadenia v podmienkach čoraz viac zvyšujúcej sa konkurencie. V súčasnosti totiž organizácie tento potenciál využívajú veľmi málo. Všetky dáta, ktoré sú uložené v štandardnom ERP, ako aj v aplikáciách typu CRM a SCM, sú použiteľné na zlepšenie rozhodovania v podniku.

Softvérové nástroje BI ponúkajú plnohodnotné informácie za dlhšie časové obdobie napríklad formou prehľadných tabuliek a grafov, ktoré ukazujú správne ale aj nesprávne trendy a korelácie rôznych javov v organizácii. BI pomáhajú v situáciách, keď základný ERP alebo riešenia CRM a SCM nie sú dostatočne účinné, pretože manažéri pri svojich úvahách myslia v skutočnosti multidimenzionálne, a teda vzájomne porovnávajú vzťahy a vyžadujú na prijatie svojich rozhodnutí len jasné a stručné výsledky, s možnosťou pozrieť sa na prípadný detail a pre toto detailné spracovávanie všetkých súvislostí by bol potrebný zase veľký počet zostáv (Basl - Blažíček, 2008). Podobným spôsobom však potrebujú pracovať s dátami nielen manažéri vrcholového manažmentu, od ktorých vlastníci a akcionári podnikov vyžadujú napríklad presné a hlavne aktuálne prehľady a analýzy hospodárenia, ale aj pracovníci na nižších úrovniach riadenia pracujúci s vhodne agregovanými informáciami.

BI predstavujú súbor procesov, aplikácií a technológií, ktorých cieľom je účinne a účelne podporovať riadiace aktivity v podniku. Podporujú analytické a plánovacie činnosti podnikov a sú postavané na princípoch multidimenzionálnych (viacrozmerných)

pohľadov na podnikové dáta. Aplikácie BI zahŕňajú analytické a plánovacie funkcie väčšiny oblastí podnikového riadenia, t. j. marketingu, nákupu, výroby, predaja, finančného riadenia, controllingu, riadenia ľudských zdrojov, správy majetku plus ďalších, zároveň zaujímajú z hľadiska svojej pozície v ISP vrchol rozhodovacej pyramídy a tvoria určitú nadstavbu nad plne integrovanými informačnými systémami typu ERP vrátane CRM a SCM, ktoré sa nachádzajú na nižšej úrovni riadenia organizácie (Čarnický - Mesároš, 2009). Základné umiestnenie BI a dvoch základných vrstiev v pyramíde ISP je znázornené na obrázku. 1.




Obrázok 1 Miesto BI a dvoch základných vrstiev v pyramíde ISP (vlastné spracovanie)

BI a ERP ako dva piliere softvérových aplikácií vytvárajú dve základné vrstvy v ISP. K ich funkcionálnemu a databázovému odlíšeniu sú často používané i dôležité technologické pojmy. O oblasti ERP sa často hovorí ako prevádzkovom, operačnom alebo transakčnom systéme (OLTP – On-Line Transaction Processing) a nadstavba BI slúži na ďalšie analytické spracovanie dát (OLAP – On-line Analytical Processing). V rámci BI sú v rozsiahlej miere využívané databázové komponenty, najmä dátové sklady (Data Warehouses). Okrem toho sú v BI využívané tiež analytické komponenty, ku ktorým patria reporting, technológia OLAP a dolovanie dát (Data Mining), ktoré umožňujú zadávanie rôznych „ad hoc“ vopred nešpecifikovaných otázok formulovaných podľa aktuálnej potreby používateľov.

Pri popise prínosov aplikácií BI je potrebné vychádzať z toho, že BI využívajú inú databázovú technológiu a hlavne štruktúru ako systémy ERP, a to viacrozmernú databázu reprezentovanú obvykle tzv. dátovou kockou, na rozdiel do systémov ERP pracujúcich len s relačnou databázou. To umožňuje systémom BI viacrozmerné pohľady na podnikové

dáta a zároveň zmenu spôsobu samotného spracovania, podľa ktorého aplikácie BI obsahujú agregované a často predspracované informácie zobrazované v dlhšom časovom období formou prehľadných tabuliek a najmä formou rôznych grafov.

Medzi základné možnosti využitia nástrojov BI pri podpore rozhodovania vrcholového manažmentu patria:

- vykonávanie rôznych analýz – ide o možnosti spracovávať komplexnejšie i detailnejšie viacrozmerné analýzy celopodnikových dát ale aj dát z okolia tohto podniku,
- zadávanie otázok (query) – ide o nástroje manažérov na kladenie náhodných ad hoc otázok podľa ich aktuálnej potreby ale aj možnosti použitia rôznych preddefinovaných otázok.

Efektívne využívanie prináša používateľom viaceré výhody, ako napríklad:

- poskytovanie aktuálnych informácií:

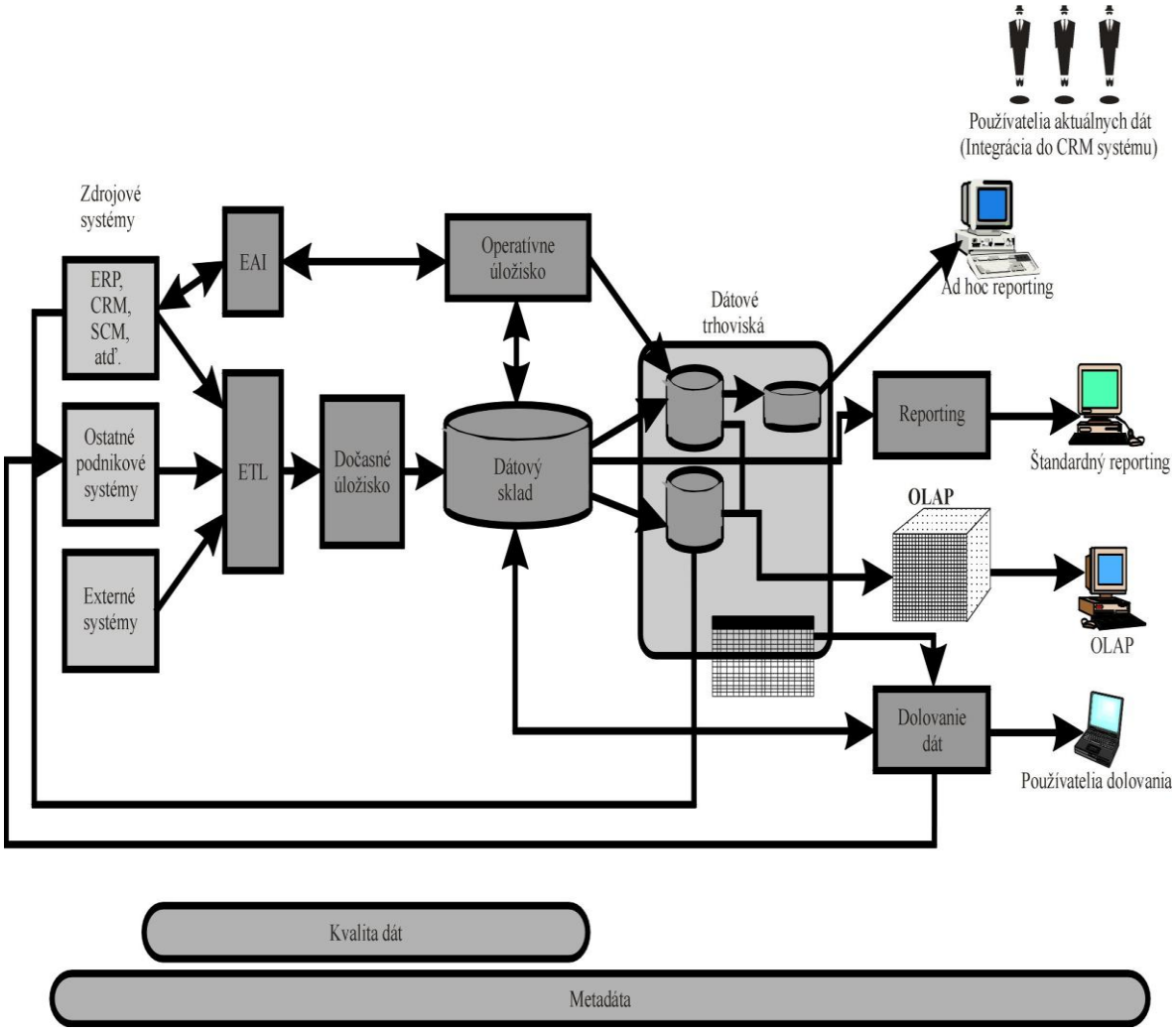
  napríklad poskytovanie informácií o stave zásob, predaji, rozpracovanosti projektov a podobne a to bez čakania na konkrétne dni, kedy sa spúšťajú v transakčných systémoch spracovania periodických uzávierok,

- zrýchlenie spracovania agregovaných informácií:

  odstránenie potreby zisťovania agregovaných informácií cez viac úrovní riadenia v transakčných systémoch, pri ktorom môže dochádzať k nežiaducim skresleniam a zbytočnému časovo náročnému spracovaniu,

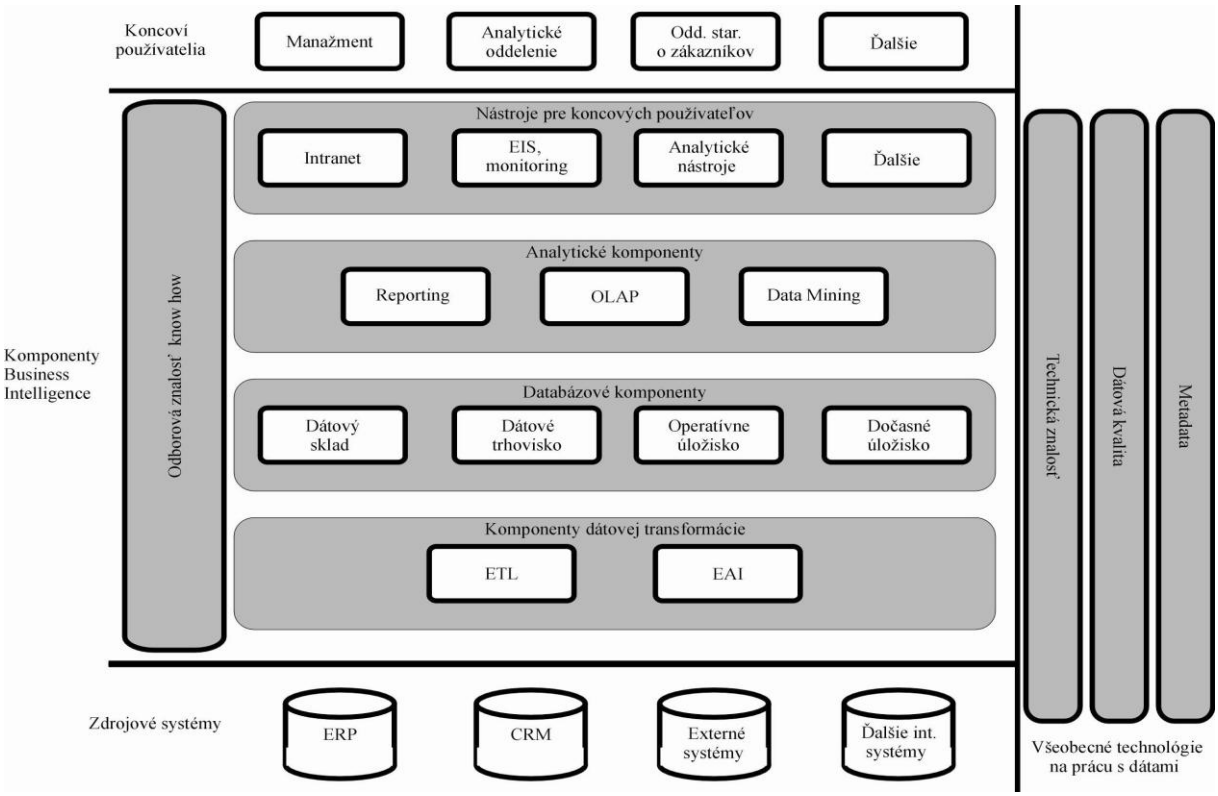
- zvýšenie pružnosti poskytovania informácií:

  vysoká pružnosť sa dosahuje pri kladení otázok na informácie, ktoré nemožno vopred špecifikovať.




Obrázok 2 Hlavné komponenty BI a ich väzby v riešení BI (Čarnický - Mesároš, 2009)

Všeobecná koncepcia architektúry BI je znázornená na obrázku 3 (Čarnický - Mesároš, 2009).




Obrázok 3 Všeobecná koncepcia architektúry BI (Čarnický - Mesároš, 2009)

Množina nástrojov BI sa zaoberá získavaním dôležitých informácií z nepreberného množstva dát, ktoré sa v organizácii hromadia. Jedná sa o informácie, potrebné pre rozhodovanie manažmentu, ako napríklad kde, za akých okolností a kedy organizácia na nejakom procese stráca. Tieto informácie nie sú v dátach nikde explicitne uvedené, analytik musí dáta obsiahnuť na oveľa vyššej úrovni, než sú jednotlivé záznamy, a objaviť v nich skryté súvislosti. BI mu k tomu poskytuje rad predmetov, bez ktorých by nemohol uspieť. Ponúka techniky od tých základných, akou sú agregácie dát, až po tak sofistikované, ako štatistické metódy alebo neurónové siete. BI sa čiastočne prelína s dataminingom, ktorý sa všeobecne zaoberá získavaním informácií z dát.

***Business Process (BP)*** - je abstraktný opis nejakej dlhotrvajúcej činnosti (úloh, tokov informácií, rozhodovacích blokov, a ďalšie), ktorá zvyčajne vyžaduje ľudskú interakciu. Procesy v organizácii majú väčšinou vlastné dáta, ktoré si so sebou nesie každá odštartovaná inštancia procesu, rôzne stavy riešenia, a je pre ne typická existencia viacerých alternatívnych tokov riadenia. Takéto procesy je nereálne realizovať vo vnútri jednotlivých aplikácií, ktoré fungujú ako čierne skrinky poskytujúce vymedzené služby, preto je zvyčajne nasadený tzv. procesný server. Zavedenie a formálne definície organizačných procesov umožňuje ich modelovanie, simuláciu, choreografiu, monitoring

a optimalizáciu.

***Business Process Execution Language (BPEL)*** - je jazyk založený na XML, ktorý slúži k popisu organizačných procesov. Definuje súbor prvkov, konkrétnych jazykových konštrukcií ako napr. sekvencie, priradenie, vyvolanie vlákna, čakanie, vetvenia, prijatie správy a pod. Manažér alebo konzultant nepracuje priamo s XML, namiesto toho využíva vizuálnu reprezentáciu v nejakom modelovacom nástroji. Výsledný BPEL opis možno naimportovať do procesného servera, alebo nejakého vývojového nástroja, kde je abstraktný opis transformovaný na softvérové komponenty a napojený na reálnu infraštruktúru.

***Business Process Modeling (BPM)*** - je súbor techník a štandardov, definujúcich návrh, implementáciu, administráciu a monitorovanie podnikových procesov. Rovnako popisuje aktivitu, vykonávanú podnikovými analytikmi vnútri spoločnosti za účelom vytvárania business process modelu. Tento model slúži pre popis procesov v organizácii a poskytuje prostriedky na ich optimalizáciu. Najprv vzniká na základe skúseností model aktuálneho stavu organizácie, ktorý umožňuje odhaliť neefektívne oblasti v existujúcich organizačných procesoch. Na jeho základe sa následne vykonáva optimalizácia týchto procesov za účelom dosiahnutia naplánovaného budúceho stavu business process modelu.

***Collective Intelligence*** - je prístup k získavaniu intelektuálneho obsahu, ako napríklad programového kódu alebo dokumentácie, vďaka nezávisle pracujúcim jednotlivcom bez centrálneho riadenia. Táto forma spolupráce môže byť nielen finančne výhodná, ale vďaka „skupinovému IQ“ a „skupinovému poznaniu“ môže prispieť k lepšiemu vnímaniu, pochopeniu, koordinácii a spolupráci. Hlavným princípom kolektívnej inteligencie je konsenzuálne rozhodovanie.

***Content Management System (CMS)*** - je systém pre správu obsahu a týmto termínom označujeme softvér, ktorý umožňuje pohodlnú a efektívnu správu dokumentov. Spravovanými dokumentmi je veľmi často webový obsah a CMS je teda väčšinou chápaný ako webová aplikácia. Niekedy sa v rovnakom význame používa aj termín redakčný či publikačný systém. CMS poskytuje najmä tieto funkcie: užívateľsky prispôsobivá tvorba a publikácia dokumentov, riadenie prístupu k dokumentom (workflow, groupware), správa súborov a multimediálneho obsahu, štatistika prístupov, diskusia a iné. Existujú doslova stovky najrôznejších CMS systémov, komerčných, ale aj Open-source. Systémy sa líšia zameraním i množstvom a prepracovanosťou jednotlivých funkcií. Technológiami najčastejšie používanými pre tvorbu CMS sú PHP (menšie systémy) a Java (podnikové systémy). Hitom poslednej doby je Java štandard JSR-170, ktorý definuje nezávislé API

pre prístup k obsahu, čo umožňuje podnikom vytvárať JSR-170 repozitára obsahu, ktoré sú nezávislé na aplikáciách, ktoré s obsahom pracujú.

***Customer Relationship Management (CRM)*** - znamená riadenie vzťahu so zákazníkom. CRM systémy sa sústreďujú na rozpoznanie konkrétnych požiadaviek zákazníka, poskytujú organizácii kompletné informácie o súčasných aj potenciálnych klientoch. Tým sa stáva CRM jednou z kľúčových súčastí každej organizácie, rastie lojalita voči zákazníkovi, znižujú sa náklady na marketing, dochádza k výraznému zvýšeniu predaja.

***Design Patterns*** - v oblasti softvérového inžinierstva sú chápané návrhové vzory ako programové kusy kódu, ktoré riešia určitý všeobecný čiastkový problém návrhu konkrétnej aplikácie. Zvyšujú čitateľnosť zdrojového kódu a prispievajú k jeho lepšej prenosnosti a pochopeniu, najmä ak so zdrojovým kódom pracuje viac programátorov. Návrhové vzory sú nezávislé na konkrétnom programovacom jazyku. Medzi najznámejšie návrhové vzory patrí tzv. Single ton, od ktorého existuje len jedna inštancia pri behu programu. Návrhové vzory sú známkou profesionálnej práce programátorov. Síce nie sú všeliekom pre kvalitný kód, výrazne však zlepšujú čitateľnosť, nájdenie prípadných chýb a uľahčujú údržbu zdrojového kódu.

***Enterprise Service Bus (ESB)*** - je technológia umožňujúca integráciu heterogénnych aplikácií pri budovaní a nasadzovaní SOA v podnikovom prostredí. Prináša efektívne riešenie bežných problémov, ako je synchronizácia ponúkaných služieb, synchronizácia dát a monitorovanie prebiehajúcich procesov. Architektúra ESB využíva princíp zbernice, ktorá je zodpovedná za doručenie správ založených na štandardoch ako SOAP, HTTP a Java Messaging Service (JMS) a je väčšinou zameraná na vysokú priepustnosť, garantované doručenie správ rôznym poskytovateľom a konzumentom služieb. Väčšina ESB podporuje XML ako prirodzený formát správ a zároveň ponúka podporu pre ďalšie alternatívne dátové typy. ESB prináša nový pohľad na spoluprácu webových služieb a servisne orientovaných architektúr do robustného celku. ESB umožňuje okamžité použitie webových služieb a ostatných integračných technológií spolu s najmodernejšími technológiami súčasnej doby a je to platformne nezávislý koncept. V kontexte EBS sa často spomína termín JBI (Java Business Integration) - Java štandard, ktorý predstavuje API pre komunikáciu so zbernicou a manipuláciu so správami.

***e-Learning / LMS (Learning Management System)*** - je značenie pre rôzne formy vzdelávania „cez počítač“. Hlavnými prvkami systému sú multimediálne učebné texty v elektronickej podobe a podporný systém pre evidenciu študentov, inštruktorov, tried, skladanie testov a pod. Zvyčajne sa jedná o vzdialenú výučbu, pri ktorej študenti aj

inštruktori pracujú a komunikujú prostredníctvom LMS, často s využitím internetu. LMS je softvérový balík, ktorý umožňuje správu a prístup k vzdelávaciemu obsahu študentom a inštruktorom. Najčastejšie sa jedná o webovú aplikáciu. Systém umožňuje študentom registráciu, prehľadávanie ponuky, prihlasovanie ku kurzom, ich absolvovanie a často aj otestovanie získaných poznatkov. Kurzy môžu byť tvorené sadou statických stránok, ale aj komplexnými dynamickými aplikáciami či vytvorenou virtuálnou triedou riadenou reálnym inštruktorom.

***Event-Driven Architecture (EDA)*** - táto udalosťami riadená architektúra definuje metodiku ako navrhovať a implementovať aplikácie a systémy, kde sú prenášané udalosti medzi nezávislými softvérovými komponentmi alebo službami. Udalosťami riadený systém je typický zložený z konzumentov udalostí a producentov udalostí. Konzument udalostí sa prihlasuje ku sprostredkovateľovi - manažérovi udalostí, ku ktorému producent udalosti zverejňuje. Akonáhle manažér udalostí obdrží od producenta správu, presmeruje ju konzumentovi. V prípade, že je konzument nedostupný, manažér správu uchová a pokúsi sa ju doručiť neskôr. Táto metóda prenosu udalostí je uvádzaná v systémoch založených na prenose správ ako „ulož a pošli“.

***Model-Driven Architecture (MDA)*** - je štandard pre vývoj softvérových aplikácií definovaný ešte v roku 2001 skupinou OMG. Pri vývoji softvéru pomocou MDA sú používané modely systému s rôznou úrovňou abstrakcie. Východiskovým bodom je model požiadaviek CIM (computation independent model), ktorý definuje prostredie vyvíjanej aplikácie, funkcionalitu a interakciu s vonkajšími systémami. Tento model je pomocou automatizovaných transformácií prevedený do detailnejších modelov PIM (platform independent model) a tie ďalej do modelov PSM (platform specific model). Z nich je nakoniec vygenerovaná prevažná časť vyvíjanej aplikácie. Hlavnými cieľmi technológie MDA sú prenosnosť softvéru medzi rôznymi platformami, zaistená nezávislosťou PIM modelov a zachovanie väzby medzi funkčnými požiadavkami na systém a jeho implementáciou. Nemenej dôležité je aj zvýšenie produktivity vďaka automatizácii určitých krokov vývoja a generovanie zdrojového kódu.

***Object-Relational Mapping (ORM)*** - sú to rôzne techniky automatizovanej perzistencie objektov do tabuliek relačnej databázy s použitím metadát, ktoré popisujú mapovanie medzi objektmi a databázou. Táto transformácia je obojstranná. Termín ORM zahŕňa perzistenciu objektov v situácii, keď sú potrebné SQL príkazy generované automaticky, nie vytvorené ručne programátorom. ORM rieši rozdielnosť medzi objektovou a relačnou paradigmou a vytvára medzi nimi most, ktorý umožňuje v OOP produkovať plne

objektový kód bez nutnosti písania SQL príkazov. Medzi najznámejšie ORM frameworky patrí Hibernate a TopLink.

***On-demand Business*** - je stav, v ktorom dokáže firma, resp. jej IT zložky, flexibilne a prakticky okamžite reagovať na vonkajšie alebo vnútorné zmeny. Jedná sa o cieľ a finálny stupeň osvojenia SOA, keď je celé IT transformované na služby a založené na optimalizovaných business procesoch, ktoré sú integrované „end-to-end“ skrz celú organizáciu za účelom umožniť rýchlu transformáciu.

***Outsourcing*** - v najširšom poňatí sa jedná o subdodávateľskú činnosť pre inú spoločnosť, ktorá je najmä v segmente IT stále viac využívaná. Dnes sa využíva ako nástroj strategického riadenia organizácie, ktorá sa špecializuje len na svoju hlavnú činnosť a ostatné podružné činnosti, ktoré však môžu byť vysoko špecializované alebo sofistikované. Medzi jeho hlavné výhody patria úspory na investíciách do zariadení, na školeniach zamestnancov, alebo zvýšenie dostupnosti a kvality služieb. Outsourcing umožňuje zákazníkovi jednoduché plánovanie prevádzkových nákladov a prenáša veľkú časť rizika zo zákazníka na dodávateľa aj napríklad formou penalizácie. Rozlišujeme niekoľko základných typov na základe geografického umiestnenia zákazníka a dodávateľa: onshore - rovnaká krajina, nearshore - blízka, susediaca krajina, zvyčajne v rovnakej časovej a geopolitickej zóne, offshore - iná krajina, iný kontinent.

***Portál*** - podnikový portál je najvyšším stupňom integrácie webových aplikácií. Hlavnou schopnosťou portálu je agregácia aplikácií a ďalšieho obsahu do jedného miesta s jednotným užívateľským rozhraním. Dôležitými aspektmi sú ďalej bezpečnosť a personalizácia obsahu na základe konkrétnej roly používateľa v organizácii. Portálom môže byť označovaný rozsiahly integrovaný podnikový systém s webovým rozhraním, ale v poslednej dobe je portál čoraz viac chápaný ako infraštrukturálne komponenty, ktoré ponúkajú služby pre jednotnú prezentáciu, personalizáciu, vyhľadávanie, on-line spoluprácu, správu obsahu, integráciu, single sign on, zabezpečenia a pre rýchly vývoj aplikácií. Portály bývajú často podnetom alebo prvým krokom k implementácii SOA. Až na výnimky sú enterprise portály implementované v technológii Java EE.

***Portlet*** - je štandardizovaný aplikačný komponent, ktorý rozširuje funkciu portálu. Z užívateľského pohľadu sú portlety malé, desktopové aplikácie pripomínajúce okienka, ktoré osídľujú portálové stránky. Portlety typicky zapuzdrujú určitú funkcionalitu a slúžia ako „okienko“ do backend systémov alebo ako rozhranie nejakého business procesu. Portlety umožňujú oveľa dynamickejšiu správu ako webové aplikácie, možno ich pridávať na stránky pretiahnutím myšou a nezávisle nastavovať parametre pre jednotlivé inštancie

alebo individuálne pre každého užívateľa. Hoci sú portlety navzájom nezávislé, môžu spolu navzájom komunikovať a vytvárať tzv. distribuované aplikácie. Z technického pohľadu sú portlety špeciálnym typom J2EE webovej aplikácie, a vďaka štandardu JSR-168 a ďalším sú portlety prenositeľné medzi portálovými servermi rôznych výrobcov.

***Representational State Transfer (REST)*** - je architektonický štýl vhodný pre tvorbu dobre škálovateľných distribuovaných aplikácií, ktorý kladie dôraz na jednotné rozhranie medzi všetkými komponentmi. Jedná sa o jednoduché rozhranie, ktoré prenáša doménové špecifické dáta cez HTTP bez ďalších nadbytočných protokolov. REST bol navrhnutý pre efektívny prenos dát hypermediálne webových aplikácií, čo obmedzuje jeho využitie v iných aplikáciách a je postavený na princípe, že stav aplikácie a jej funkcionalita je rozdelená do zdrojov, pričom každý zdroj je unikátne adresovateľný univerzálnou internetovou syntaxou (URI). Všetky zdroje zdieľa jednotné rozhranie pre prenos stavov medzi zdrojom a klientom, ktoré sa skladá sa z operácií a typov. Komunikačný protokol je bezstavový, viacvrstvový a podporuje vyrovnávaciu pamäť.

***Search Engine Optimization (SEO)*** - optimalizácia pre vyhľadávače - je to pojem označujúci postupy smerujúce k efektívnejšiemu indexovaniu webovej stránky vyhľadávačmi (ako je napríklad Google) s cieľom lepšieho umiestneniu stránky vo výsledkoch vyhľadávania. Týchto postupov je celá rada, niektoré z nich sú prínosné aj pre bežných užívateľov a sú všeobecne vnímané veľmi pozitívne. Existujú však aj postupy, ktorých cieľom je vyhľadávač oklamať; takéto postupy bývajú považované za podvodné a ak sú odhalené, môžu viesť aj k penalizácii zo strany vyhľadávača.

***Service-Oriented Architecture (SOA)*** - tento servisne orientovaný prístup je dlho známy a používaný architektonický koncept, ktorý modeluje informačný systém ako súbor komponentov poskytujúcich, alebo aj konzumujúcich služby. Službou sa myslí dobre zadefinovaná vnorená funkcionalita, ktorá je nezávislá na okolí. Skratka SOA sa presadila až po roku 2000, kedy došlo v tejto oblasti k štandardizácii vďaka masívnemu rozšíreniu web services. V súčasnej dobe termín SOA tiež zastrešuje súbor mnohých technológií, ktoré pokrývajú širokú škálu IT od samotnej komunikácie, cez integráciu až po riadenie organizačných procesov. Osvojenie SOA v organizácii prebieha postupne, konečným cieľom je flexibilita a on - demand business.

***Servisce-Oriented Infrastructure (SOI)*** - jedná sa o značne modulárnu, flexibilnú IT štruktúru, ktorá je založená na štandardných prvkoch. Tieto prvky sú vysoko konfigurovateľné, aby splnili rapídne rastúce požiadavky dnešnej doby. Pozeráme sa na infraštruktúru systému ako na viacvrstvovú, SOI predstavuje vrstvu, ktorá sa zameriava na

riadenie a virtualizáciu výpočtov, na sieťovú komunikáciu a prístup ku zdrojom. SOI zabezpečuje zdroje tak, aby boli dostupné v požadovanom množstve a umiestnené v konkrétnej požadovanej vrstve. So SOI abstrakciou sú detaily o fyzickej štruktúre zariadenia pokryté softvérom. Zariadenia sa stávajú ovládateľné cez abstraktné softvérové rozhrania, ktoré sú v systéme definované ako služby.

***Social network analysis*** - sú to techniky získavania organizačne významných informácií a vedomostí pomocou agregácie dát z osobných sietí jednotlivcov. Môže sa jednať o objavenie nových príležitostí, zostavenie projektového tímu, alebo zvyšovanie efektivity nerutinných organizačných postupov ako sú objavovanie, inovácie, spolupráca, vedenie a učenie. Mnohé systémy už dávno podporujú rôzne formy agregácie a zdieľanie obsahu, ale nové nástroje umožnia lepšie analyzovať prebiehajúce sociálne procesy a tak umožní ich lepšie pochopenie a následné využitie.

***Software as a Service - softvér ako služba (SaaS)*** - je to nový spôsob fungovania podnikových a kancelárskych aplikácií, ktoré sú prevádzkované kompletne externým internetovým dodávateľom. O vlastnú prevádzku aplikácie sa teda kompletne stará niekto iný, čo môže znížiť celkové náklady na prevádzku aplikácie (správa, bezpečnosť, zálohovanie, servery, licencie atď.) Používatelia majú k dispozícii samoobslužné webové rozhranie, často formou portálu. Väčšina dodávateľov SaaS buduje svoje systémy na základe SOA, vďaka čomu je možné rozumne integrovať s existujúcimi on-site systémami. Bezpečnostné dopady využívania SaaS síce ešte stále pretrvávajú, ale pre podniky už dnes nepredstavujú v porovnaní s minulosťou natoľko zásadný problém. Formou SaaS sú ponúkané CRM systémy, ERP systémy, procurement e-business aplikácie a ďalšie. Najmä pre malé a stredné podniky SaaS predstavuje zaujímavú možnosť, ako finančne dosiahnuť na tradične nákladné podnikové aplikácie.

***Supply Chain Management (SCM)*** - čiže riadenie dodávateľských reťazcov, predstavuje komplex aplikácií informačných a komunikačných technológií, ktoré slúžia na koordináciu tokov tovarov, služieb, informácií a financií medzi dodávateľmi surovín, ich spracovateľmi, medzi výrobcami komponentov aj finálnych výrobkov, obchodníkmi, predajcami s cieľom efektívneho uspokojenia požiadaviek konečných spotrebiteľov čo do rýchlosti, kvality a ceny produktu či služby. Súčasné podnikateľské prostredie sa vyznačuje tým, že sa orientuje na hlavné aktivity daného podniku, v ktorých daný podnik vyniká. Všetky ďalšie aktivity sa potom podnik snaží zabezpečiť si formou subdodávok produktov, alebo ako služby -outsourcing. Tak vzniká hodnotový reťazec (miesto, výroba, zásoba, doprava), ktorého hlavným poslaním je uspokojiť požiadavky konečného

zákazníka čo najrýchlejšie a za čo možno najnižšiu cenu.

***Unified Messaging (UM)*** - pod týmto pojmom sa rozumie integrácia rozdielnych komunikačných kanálov. Užívateľ je schopný posielať a prijímať emaily, hlasové správy, SMS alebo faxy pomocou jedného rozhrania, či už používa mobilný telefón, pevnú linku alebo počítač pripojený k internetu. Ak má užívateľ k dispozícii aspoň jeden z podporovaných komunikačných kanálov, sú mu prístupné aj všetky formy komunikácie. Ďalšou výhodou je notifikácia správ prijatých centrom správ, to znamená, že ak je užívateľ na určitom kanáli nedostupný, možno ho kontaktovať cez iný komunikačný kanál. To je umožnené najmä vďaka automatizovanému prevodu hovoreného slova na text a opačne. Systémy UM umožňujú široké možnosti nastavenia - napr. užívateľ si nepraje byť rušený medzi 21. až 8. hodinou a podobne.

***Virtualizácia*** - v oblasti výpočtovej techniky je virtualizácia postup, ktorý pomocou skrývania technických detailov umožňuje pristupovať k zdrojom iným spôsobom, než ako k fyzicky existujúcim. Vo výsledku sa potom chová jeden fyzický prvok (napr. počítač, operačný systém alebo úložné zariadenia) ako viac logických prvkov, alebo naopak viac fyzických prvkov vyzerá ako jeden logický (typicky diskové pole). Virtualizácia zdrojov (pamäť, disky) je hojne využívaná technika, ktorá je dnes samozrejmou súčasťou operačných systémov. V súčasnosti však zažíva veľký rozmach virtualizácia behu operačných systémov. Vďaka tejto technike je možné mať na jednom fyzickom stroji k dispozícii niekoľko súbežne pracujúcich na sebe nezávislých operačných systémov, čím sa dosiahne jednoduchšia správa, vyššia bezpečnosť a prenosnosť. Komplikáciou však predstavuje licencovanie komerčných operačných systémov pre „virtuálne“ počítače.

***VoIP (Voice over IP)*** - je všeobecné označenie hlasových prenosov cez internetový protokol namiesto klasického spájania telefónnych okruhov, alebo prenos špeciálnymi sieťami telefónneho operátora. VoIP tak otvára priestor pre virtuálnych telekomunikačných operátorov, ktorých služby nie sú viazané na vlastnú sieť. Výhodou je využitie už vybudovaných dátových sietí, škálovateľnosť služieb a celkové finančné úspory na všetkých stranách. VoIP môže byť riešené ako softvérovo (Skype, GoogleTalk / XMPP Jingle, a pod.) tak aj hardvérovo, IP telefónom. Kým hardvérové VoIP je pomerne dobre pokryté medzinárodnými štandardmi, softvérové riešenia sú často navzájom nekompatibilné. VoIP dnes bežne podporujú telefónne ústredne a aj prvky počítačových sietí, preto sa často hovorí o konvergencii hlasu a dát.

***Web 2.0*** - tento nejasne definovaný pojem, sa používa v súvislosti s novými vlastnosťami internetových služieb, najmä ich posunu od statických foriem k dynamickým

a komunitným, kde sami užívatelia spoluvytvárajú obsah. Typickým príkladom sú video, foto, audio blogy. Ďalším aspektom Web 2.0 je prístupnosť bežným užívateľom. Toto môže byť dosiahnuté napríklad pomocou personalizácie v prostredí portálu alebo vhodným použitím technológie AJAX, ktorá približuje webové aplikácie desktopovým zvyklostiam čo do spôsobu ovládania a funkčných možností. Web 2.0 možno chápať ako netechnický pojem, ktorým sa označuje vidina nového a lepšieho internetu, či už to znamená čokoľvek.

***Web Services*** - je otvorený štandard pre výmenu informácií medzi jednotlivými prvkami systému. Je sieťovo transparentný a nezávislý na platforme. Silne využíva XML formát. Na vlastný prenos sa v drvivej väčšine prípadov používa protokol SOAP transportovaný pomocou HTTP, ale toto nie je jediný možný spôsob, možno použiť aj napríklad SMTP a ďalšie protokoly. Web services sú základným stavebným kameňom servisne orientovanej architektúry SOA. Po praktickej stránke je najväčším prínosom nezávislosť na platforme. Napríklad aplikácie bežiace na Windows PC môžu štandardizovane a transparentne pristupovať k službám bežiacim na vzdialenom unixovom mainframe s úplne odlišnou architektúrou hardvéru, softvéru i danej aplikácie. Cenou za to je veľká réžia XML, pretože je prenášané množstvo pomocných dát, čím dochádza k poklesu výkonu.

***Web Services for Remote Portlets (WSRP)*** - je štandard, ktorý definuje spôsob interakcie s aplikačnými službami, ktoré poskytujú priamo markup užívateľského rozhrania - napr. fragmenty HTML. Vďaka tomu môžu aplikácie prostredníctvom štandardu web services publikovať alebo používať časti používateľského rozhrania bez toho, aby bolo nutné programovať špeciálne rozhranie pre každý komponent. V prípade portálov WSRP umožňuje jednoduchým konfiguračným nastavením zobraziť v jednom portáli portlet fyzicky bežiaci na inom portáli a tým dosiahnuť ďalšieho stupňa integrácie a flexibility.

***Wiki*** - je to technológia, ktorá umožňuje veľmi jednoduché pridávanie alebo úpravu obsahu webovej stránky ľubovoľným návštevníkom. Obsah stránky je popísaný pomocou jednoduchého značkovacieho jazyka, ktorého formát nie je ustáleným a u rôznych implementáciách wiki sa líši. Vďaka tomu nemusí užívateľ poznať jazyk HTML a úprava stránok je jednoduchá aj pre bežných návštevníkov. Princíp wiki umožňuje tvorbu obsahu a opravu chýb komunitou užívateľov a asi najznámejším projektom je otvorená internetová encyklopédia Wikipédia (http://sk.wikipedia.org). Wiki môže zvýšiť kvalitu obsahu vďaka maximalizácii okruhu možných prispievateľov, je nutné sa však vysporiadať s problémom (ne)úmyselného poškodzovania obsahu stránok. Pre svoju obľúbenosť sa princípy wiki dostávajú aj do podnikových systémov, kde koexistujú vedľa klasických schvaľovacích procesov získavania obsahu. V podnikoch sa wiki uplatní napríklad pri tvorbe

dokumentácie alebo ako prostriedok komunikácie projektových tímov.

**Expertné a Znalostné informačné systémy** - Znalostné informačné systémy (KMS – Knowledge Management System) slúžia v organizáciách hlavne na uchovávanie, zdieľanie a efektívnejšie využitie znalostí. Prvoradou úlohou a základnou myšlienkou znalostných systémov je umožniť zamestnancom prístup ku všetkým oficiálnym znalostiam, faktom, dokumentom, zdrojom informácií a následných riešení. Pomocou KMS je možné efektívne zdieľať inak ťažko dostupné informácie, predísť zbytočnému redundantnému úsiliu pri riešení známych problémov, zredukovať čas na zaškoľovanie nových zamestnancov a hlavne predchádzať strate znalostí pri odchode zamestnanca z organizácie. KMS môžu mať rôzne stupne spracovania dát - od jednoduchého dátového skladu vo forme skladu dokumentov až po sofistikované Expertné informačné systémy, ktoré ukladajú dáta a dokumenty vo forme ontológií a klasifikujú ich (Ontology and taxonomy based systems).

**Špeciálne informačné systémy** - sú všetky ostatné IS, ktoré slúžia rôznym špecifickým účelom v organizácii, typicky v závislosti na konkrétnom zameraní podniku. Príkladom takýchto systémov môžu byť napríklad Geografické informačné systémy, ktoré okrem typických úloh IS ako je získavanie, ukladanie a analýza dát tieto dáta vizualizujú väčšinou v podobe konkrétnych zobrazovacích máp. Tieto dáta - často nazývané aj geodáta - majú teda priestorový vzťah ku konkrétnemu pozemku, k povrchu Zeme, a sú definované geometriou, typológiou, atribútmi a dynamikou.

## 1.3 Tradičné modely implementácií komunikačných systémov

V praxi existujú 3 spôsoby komplexnej implementácie komunikačného systému:

1. **vlastné riešenie**, vytvorené prostredníctvom vlastných odborníkov vo vnútri organizácie,
2. **zakúpené univerzálne hotové „krabicové“ riešenie**,
3. **kombinácia týchto dvoch predchádzajúcich**, s odporúčaním: „Buy the best and build the rest“ – to znamená dotvoriť niektoré oblasti systému vlastným riešením, ak sme zistili, že nevyhovujú našim požiadavkám.

Nesprávnym prístupom je výber komunikačného systému zredukovaný len na

otázku ceny. Správny postup pri vytváraní akéhokoľvek komunikačného systému obsahuje tieto etapy v tomto slede:

**A. ORGWARE**

**B. SOFTWARE**

**C. HARDWARE**

**D. POČÍTAČOVÁ SIEŤ**

ORGWARE – predstavuje vytvorenie organizačnej štruktúry s vymedzením právomocí a zodpovedností, až potom môžu prísť na rad ďalšie etapy ako výber alebo vytvorenie vhodného SOFTWARE, potom podľa náročnosti a požiadaviek softvéru sa vyberie vhodný HARDWARE a až na základe existujúcej organizačnej štruktúry, softvéru a hardvéru sa môže vytvoriť vhodná POČÍTAČOVÁ SIEŤ.

Postup pri výbere a nákupe by mali tvoriť tieto kroky:

1. formulácia - ujasnenie informačných požiadaviek na komunikačný systém,
2. zistenie - určenie vlastných finančných možností na nákup,
3. získanie informácií o dostupných komunikačných systémoch na trhu,
4. analýza a porovnanie jednotlivých ponúk komunikačných systémov, porovnanie základných atribútov,
5. samotný výber a nákup vhodného komunikačného systému,
6. zaškolenie pracovníkov a postupné zavádzanie a využívanie zakúpeného systému

Dôležitými kritériami pri výbere komunikačného systému sú:

- prípadná kompatibilita s doterajším systémom v organizácii,
- cenové hľadisko,
- posúdenie časového harmonogramu samotného zaobstarania a implementácie systému,
- zvážiť rozsah budúcich „služieb“ po implementácii,
- ďalšie kritériá ako sú image dodávateľskej firmy, počet referencií - úspešných aplikácií a spokojnosť užívateľov, aká je podpora dodávateľskej firmy pri zavádzaní systému, aké ponúka doplnkové služby napríklad hotline, aká je časová telefonická komunikácia pre prípad vzniknutých problémov, aké sú možnosti Update - teda možnosti aktualizácie na vyššiu verziu, aké budú možné nepredvídané riešenia z hľadiska legislatívy, za akých podmienok je možný prípadný Upgrade - možné ďalšie inštalované vylepšené riešenia prostredníctvom najnovších prvkov informačných

technológií, akú úroveň poskytuje dodávateľ v balíku dodanej dokumentácie a ďalšie.

Očakávania spojené s implementáciou a využívaním IS sa mnohokrát nenapĺňajú, aj keď organizácie investovali a investujú do tejto oblasti veľké prostriedky. Na trhu síce existuje množstvo progresívnych a podľa názoru ich výrobcov aj komplexných softvérových balíkov, ktoré sa implementujú v organizáciách rôzneho zamerania a nahradzujú pôvodné informačné systémy, avšak na zlepšenom a efektívnejšom fungovaní organizácií to veľmi necítiť. Manažéri týchto subjektov si často kladú otázky typu: „Investovali sme správne?“, „Bolo nevyhnutné nahradzovať existujúci informačný systém, ktorý už bol adaptovaný na naše podmienky?“, „Nestačilo rozšírenie existujúceho systému o výkonnejší hardvér a systémový respektíve databázový softvér?“. Skúsenosti z niektorých amerických a západoeurópskych inštitúcií ukazujú, a manažéri týchto firiem to tvrdia, že nasadením súčasných štandardných softvérových balíkov, hlavne ERP systémov, by nedosiahli požadovanú funkcionalitu, ba v mnohých prípadoch by boli nútení prerábať aj tie časti informačného systému, ktoré fungujú dobre a spoľahlivo (http://www.apress.com). Prečo je to vlastne tak? Štandardne vyvinuté podnikové informačné systémy ako ERP, MIS, EIS, sami o sebe pravdepodobne nedokážu v plnej miere zabezpečiť zlepšené a efektívne fungovanie organizácií. Ich implementácia a nasadzovanie predstavuje pomerne veľký problém, ktorý sa ešte viac prehĺbi pri zavedení systému do praxe.

**Ako teda implementovať nový informačný komunikačný systém?**

Musíme predovšetkým vychádzať z podstaty informačného systému, čo je informačný systém a čo je vlastne jeho úlohou. Na úvod niekoľko definícií (http://sk.wikipedia.org):

**Informačný systém** je systém na zber, udržiavanie, spracovanie a poskytovanie informácií.

**ERP systém** je informačný systém, ktorý integruje dáta a procesy organizácie do jednotného systému.

**Manažérsky informačný systém** možno definovať ako informačný systém na

manažérskom stupni riadenia organizácie, ktorý slúži funkciám plánovania, kontroly a rozhodovania poskytovaním pravidelných prehľadov a hlásení o výnimočných situáciách.

S uvedenými definíciami, aj keď jednoduchými, je možno súhlasiť. Odrážajú to, čo je úlohou týchto komunikačných systémov. Všeobecne chápeme informačný systém ako systém pre spracovanie dát, ktorý má ciele strategické (rozhodovanie, plánovanie investícií, plánovanie zdrojov a pod.), taktické (riadenie, kontroling) a operatívne (každodenná rutina).

Vyššie uvedená definícia informačného systému je všeobecne známa. Úlohou informačného systému, hlavne ERP časti, je integrovať dáta a procesy, teda informačne podporovať procesy a jednotlivé procesné kroky organizácie. ERP, vyplýva to aj z jeho anglického názvu, je softvér na riadenie (plánovanie) zdrojov organizácie, ktoré zabezpečujú vykonávanie procesov. Manažérsky informačný systém pôsobí na manažérskej úrovni a slúži predovšetkým na poskytovanie informácií pre manažérov a ich rozhodovanie.

Z uvedeného vyplýva, že musíme najprv poznať procesy organizácie, bez ktorých organizácia nemôže fungovať, a až potom sa zaoberať myšlienkou ako tieto procesy sú, alebo budú podporované naimplementovaným informačným systémom, v našom prípade sa jedná o ERP systém a na manažérskej úrovni o MIS. Spomínali sme, že MIS by mal slúžiť predovšetkým na rozhodovanie manažérov. Najdôležitejšie, a najťažšie, je pre organizáciu strategické rozhodovanie. Poskytujú súčasné informačné systémy dostatok informácií pre strategické rozhodovanie? Podľa názoru väčšiny manažérov našich významných organizácií nie.

Prečo je to tak? V prvom rade je to nevyjasnená stratégia organizácií. Víziu, čo chce dosiahnuť, má pravdepodobne každá organizácia, ale stratégia, ako to dosiahnuť, pomerne často absentuje. V čom je teda príčina? Nie je úlohou informačného systému poskytovať informácie aj pre strategické rozhodovanie? Určite áno, avšak realita je bohužiaľ úplne iná. Pri implementácii Business Intelligence riešení, ktoré poskytujú alebo lepšie povedané mali by poskytovať, komplexný pohľad na strategické činnosti organizácie, veľmi často riešiteľský tím narazí na problém neexistencie potrebných dát. Proste existujúce, aj nové, informačné systémy ich neposkytujú. Manažéri sa divia, riešiteľský tím myká plecami a dodávateľ informačného systému už svoju prácu dávno

odovzdal a zaoberá sa už len podporou toho, čo dodal a naimplementoval. Zrazu niet vinníka.

V čom je jadro problému? Zaoberal sa niekto myšlienkou, ako budeme hodnotiť výkonnosť organizácie z pohľadu naplňovania stratégie, strategických cieľov a aké informácie budeme k tomu potrebovať? Ako sa dostať z tejto nepríjemnej situácie? Je možné, že bez BPM (Business Process Management) prístupu pri implementácii komunikačných systémov sa nepohneme. Čo by mohlo byť riešením existujúcich problémov?

Jedným z návodov je nasledovný postup:

- Vytvorenie mapy procesov organizácie a stanovenie ich priorít na základe Paretovho princípu (80:20) a teda stanoviť úrovne detailizácie jednotlivých procesných krokov.
- Striktné rozdelenie procesov na kľúčové (prinášajúce pridanú hodnotu) a podporné (vrátane riadiacich) procesy.
- Prvotná optimalizácia procesov (odstránenie nepotrebných, resp. zbytočných procesov a činností) v súlade so stratégiou a cieľmi organizácie a vytvorenie optimalizovaného procesného modelu (mapy) organizácie na báze princípov a metodík procesného riadenia (BPM).
- Vypracovanie prvotného modelu strategického hodnotenia výkonnosti organizácie, postupujúc cez rôzne uhly pohľadu na organizáciu, kritické faktory úspešnosti dosahovania cieľov pre jednotlivé bunky inštitúcie, kľúčové výkonnostné indikátory a cez merania, do ktorých budú prúdiť dáta z komunikačného systému.
- Vypracovanie cieľového konceptu (Business Blueprint) implementácie komunikačného systému na základe prvotného modelu strategického hodnotenia výkonnosti organizácie, procesov z vytvoreného optimalizovaného procesného modelu a ich požadovanej informačnej podpory.
- Podpora po zavedení nového komunikačného systému.
- Nepretržité zlepšovanie procesov už aj za pomoci nového komunikačného systému.
- Knowledge management, teda vytvorenie databázy znalostí o procesoch a ich informačnej podpore.

Teda cieľom zavedenia komunikačného systému musí byť podpora stratégie a procesov, ktoré organizácia nevyhnutne potrebuje. Budovanie komunikačného systému

je záležitosť tímovej práce, na ktorej sa podieľajú celé skupiny pracovníkov najrôznejšieho zamerania a kvalifikácie. Títo pracovníci sú zamestnancami rôznych buniek inštitúcie a preto je jednou z najťažších úloh zladenie ich práce, aby bola zmysluplná a produktívna a aby všetci išli za spoločným cieľom, za úspešným zavedením komunikačného systému. Pri vyššie uvedenom postupe a kvalitnom riadení projektu implementácie komunikačného systému bude určite eliminovaná väčšina uvedených problémov.

**Kvalita IS** - pre určenie kvality a úžitkovosti IS možno použiť 12 pravidiel a 18 znakov formulovaných podľa EF Codd & Associate, alebo kratšie hodnotenie FASMI1 (Fast Analysis of Shared Multidimensional Information) so šiestimi pravidlami (http://www.olapreport.com):

1. ***Fast*** - rýchlosť odpovede systému na analytický dotaz má byť do 5 sekúnd, pri jednoduchších dotazoch potom do jednej sekundy. V prípade zložitých dotazov do 20 sekúnd. V prípade prekročenia 30 sekúnd možno očakávať, že používateľ toto vyhodnotí ako chybu systému.
2. ***Analysis*** - IS by mal ponúkať možnosť jednoduchej tvorby analýz a reportov na základe potrieb manažérov bez nutnosti doprogramovania či nastavenia danej oblasti. Tiež by nemal vyžadovať znalosti štatistických metód alebo metód k nim vedúcich. Výstupy týchto analýz musia byť ľahko zrozumiteľné.
3. ***Shared*** - systém má ponúkať zdieľania dát v spoločnosti. Každý, kto má v systéme náležité oprávnenie, má mať možnosť zdieľať s ostatnými v spoločnosti svoje analýzy.
4. ***Multidimensional*** - manažérske systémy musia ponúkať pohľad na dáta z viacerých uhlov pohľadov (dimenzií). Multidimenzionalita je základom všetkých OLAP nástrojov.
5. ***Information*** - výstupom komunikačných systémov sú kvalitné, relevantné a správne informácie.
6. ***Data mining*** - tento pojem predstavuje proces hľadania neznámych a potenciálne užitočných informácií v databázach a dátových skladoch. Jeho výstupom býva často konkrétny model, ktorý na základe známych vstupných informáciách o procesoch je schopný „predpovedať“ ich budúce správanie. Po technickej stránke máva model väčšinou podobu matematickej rovnice alebo sady pravidiel. Na modelovanie sa využíva celá rada data miningových metód, medzi najznámejšie patria logistická regresia, neurónové siete či rozhodovacie stromy.

Bolo by chybou myslieť si, že proces data miningu končí iba vývojom modelu a jeho implementáciou. Každé akékoľvek prostredie sa totiž vyvíja veľmi rýchlo a model, ktorý bol práve dnes vyvinutý na najaktuálnejších dátach, už zajtra nemusí byť aktuálny. Aby model slúžil čo najdlhšie, je potrebné ho udržiavať.

Pozrime sa teda, v čom taká údržba spočíva, ako často je potrebné sa jej venovať, ktoré kroky je možné delegovať na správcu dátového skladu a ktoré je naopak nevyhnutné prenechať dátovým analytikom.

Údržbu modelu je možné podľa stupňa náročnosti rozdeliť do štyroch úrovní:

- **Kontrola konzistencie**
- **Kalibrácia modelu**
- **Obnova modelu**
- **Vývoj nového modelu.**

***Kontrola konzistencie*** - je najjednoduchšia, ale zároveň najpodstatnejšia súčasť údržby modelu spočívajúca v kontrole, či je model konzistentný, čiže či sa nemenia jeho vlastnosti, a to oproti dlhodobému trendu. Ku kontrole konzistencie modelu stačí mnohokrát obyčajné porovnanie s minulými hodnotami, od ktorých by sa nemal výrazne odlišovať. Pre úplnosť je možné túto informáciu doplniť o počet chýbajúcich pozorovaní (NULL) či o smerodajnú odchýlku. Ak sa zistí, že došlo k porušeniu konzistencie, môže to signalizovať:

- technickú závadu,
- zmenu vstupných údajov,
- zmenu správania modelu.

Prvý prípad nastáva predovšetkým vtedy, ak dôjde k nejakému zásahu do štruktúr dátového skladu - pridali sa nové tabuľky, zmenili sa ich názvy, rozšírili sa či boli zrušené niektoré číselníky, optimalizovali sa skripty a podobne. Potom môžu nastať dve situácie - model úplne zlyhá, čo si väčšinou všimneme veľmi rýchlo, ale oveľa častejšie proces prebehne bez jedinej chybovej hlášky, avšak výstupom sú nezmyselné „hodnoty“. A tak sa stáva, že všetky výstupy majú rovnaké údaje, alebo tieto údaje obsahujú iba prázdne hodnoty typu NULL.

Druhá situácia nastáva, ak sa zmení portfólio údajov, ktoré sú spracovávané.

K tomu dochádza väčšinou rozhodnutím manažmentu, že k niektorým dátam je potrebné priradiť ďalšie, alebo sa viac zameriavajú na špecifickú skupinu dát. Potom sa však manažment nesmie čudovať, že sa tým zmení aj štruktúra dát v databáze. V tejto situácii nie je dôvod model nijako upravovať a ponecháva sa v takom stave, v akom momentálne je.

Oveľa horšia situácia ale nastane, keď je zmena dát spôsobená náhlou zmenou správania modelu. Predstavte si situáciu, kedy bol vyvinutý model na predikciu využívania dát, ktorého hlavným prediktorom je napríklad miera využívania poskytovaných tabuliek a zrazu sa pôvodný model tak stane nepoužiteľným.

Ako ale zistiť, čo je príčinou nekonzistencie modelu? Často stačí preveriť konzistenciu všetkých premenných, ktoré vstupujú do tohto modelu. Potom sa ľahko príde na podstatu problému, t. j. ktoré premenné sú zodpovedné za to, že došlo k zmene vlastností celého modelu. Kontrolu konzistencie jednotlivých premenných možno vykonať opäť prostredníctvom jednoduchých premenných a ich porovnaním s minulými hodnotami, u kategoriálnych premenných potom prostredníctvom napríklad frekvenčných tabuliek.

***Kalibrácia modelov*** - najmä vplyvom „sezónnych“ faktorov dochádza často k tomu, že sa predpovedané hodnoty modelu rozchádzajú so skutočnými hodnotami. Môže napríklad nastať situácia, keď je zrazu veľký záujem len o konkrétne dáta z konkrétnej oblasti, na čo zrazu náš model naučený na dátach z iného obdobia na to vôbec nie je pripravený. Na základe dekalibrovaného modelu je síce možné určiť, ktoré dáta sú z hľadiska záujmu horšie a ktoré lepšie dostupné - v relatívnom porovnaní model funguje stále dobre, ponúka však neúplné a často aj skreslené informácie. Dekalibrovaný model má za následok, že model podceňuje alebo preceňuje reálnu situáciu, to znamená, že si používateľ myslí, že údaje sú relevantnejšie, než v skutočnosti sú. Či je model dekalibrovaný možno zistiť ľahko aj tým, že sa porovná priemerné skóre s priemernou hodnotou predikovanej premennej. Ak sú obe veličiny výrazne odlišné, je nutné model kalibrovať.

Kalibrácia pripadá do úvahy iba u lineárnej alebo logickej regresie, keď sú výstupné informácie porovnávané konkrétnym modelovým riešením väčšinou ako vážený súčet prediktorov alebo porovnávanie absolútnych členov (Bébr - Doucek, 2005). Kalibrácia modelu potom spočíva len v úprave absolútneho člena bez nutnosti použitia nejakého špeciálneho softvéru alebo preprogramovávania celého modelu, t. j. ostatných

koeficientov. V kóde, ktorý zaisťuje programový chod modelu, potom často stačí len zmeniť jedno jediné číslo. Kalibrovať možno teoreticky aj modely založené na neurónových sieťach či rozhodovacích stromoch, nie je to však také jednoduché ako pri lineárnej alebo logickej regresii. Oveľa jednoduchšie je potom celý model preprogramovať.

Nutné je však tiež dodať, že dekalibrácia často vôbec nie je potrebná a teda nemá zmysel s ňou strácať čas. Táto situácia nastáva najmä v cielenej komunikácii, kedy vzniká potreba osloviť len určité percento účastníkov v komunikačnom procese, bez ohľadu na jeho absolútnu úroveň.

***Obnova modelu*** - často v dlhšom období používania toho istého modelu dochádza nielen k dekalibrácii tohto modelu, ale aj k postupnej strate jeho informačnej schopnosti. To znamená, že model už nie je schopný dobre rozoznávať medzi dobrými a zlými informáciami. Ak používa model v cielenej komunikácií od začiatku stále nemenné premenné, časom sa prejaví znížená predpovedacia schopnosť a vo výsledných výstupoch potom užívateľ zaznamená zvýšenú delikvenciu na poskytnutých informáciách, ktoré boli prostredníctvom modelu schválené. Obnova modelu spočíva v novom nastavení všetkých váh, ktoré potom lepšie vyjadrujú význam premenných v modeli. Na rozdiel od kalibrácie už obnova modelu vyžaduje uplatnenie zdrojového softvéru a preskúšavanie modelu na novej vzorke dát. Pri obnove sa ale našťastie nemení zoznam atribútov, iba ich váhy a implementácia obnoveného modelu nevyžaduje žiadny väčší zásah do fungujúcich skriptov - stačí zmeniť iba niekoľko parametrických hodnôt.

***Vývoj nového modelu*** - pri obnove modelu sa väčšinou nepodarí dosiahnuť pôvodnú výkonnosť modelu, dôjde len k jeho oživeniu, často k miernemu zvýšeniu jeho kvality. Zároveň sa tým aj model kalibruje. Tento prístup je však z dlhodobého hľadiska neudržateľný. Po niekoľkých rokoch totiž niektoré atribúty úplne stratia svoju prediktívnu silu a vypovedaciu schopnosť a potom je jediným riešením vývoj úplne nového modelu. Ten je tiež potrebný v prípade náhlej zmeny správania účastníkov v komunikačnom procese. Vývoj nového modelu potom znamená značný zásah do IT infraštruktúry - je potrebné zmeniť nielen samotné parametre modelu, ale aj pripraviť výpočet nových atribútov, a potom musíme model určitú dobu aj testovať. Ako často teda máme vykonávať údržbu modelu?

Frekvencia údržby závisí najmä na tom, ako veľmi je pre nás kritický nefunkčný

model potrebný, na akú dlhú dobu by asi bolo tragické ho mať dlhodobo nefunkčný. Spravidla sa dá povedať, že kontrola konzistencie by sa mala vykonávať bezprostredne po každom zadefinovanom časovom cykle. Jedná sa totiž o veľmi jednoduchú kontrolu, ktorá môže organizáciu ochrániť pred obrovskými stratami. Naviac vzhľadom k svojej jednoduchosti nevyžaduje táto kontrola žiadne znalosti napríklad z oblasti data miningu a je možné ju bez problémov delegovať aj na oddelenie IT, ktoré môže kedykoľvek operatívne zasiahnuť.

Ak potrebuje užívateľ kalibrovaný model, je ideálne zvoliť si ideálnu časovú periodicitu a teda čím rýchlejšie sa bude na „sezónne“ výkyvy reagovať, tým lepšie, ale zase na druhej strane dekalibrácia modelu neznamená až tak vysoké riziko, aby ju bolo potrebné vykonávať každý deň. Každých šesť až dvanásť mesiacov (možno po každom semestri) stojí za úvahu model obnoviť, to znamená prepočítať jeho parametre. Vývoj úplne nového modelu sa spravidla odporúča po dvoch až štyroch rokoch. Životný cyklus systému je časové obdobie životnosti a aj vývojových zmien systému od jeho zadania cez projektovanie, realizáciu, zavedenie a prevádzku vrátane údržby a rozvoja až k jeho zániku, resp. nahradenie iným systémom. Je to taktiež označenie pre etapizáciu života projektu a metodiku vývoja systému, je to aj model vývoja samotného systému a je to tiež špeciálna metodika pre postup práce na projekte - jedna zo špeciálnych systémových metodík.

***Kľúčové etapy životného cyklu KS:***

- *Predanalytická fáza* - je modelovanie, špecifikácia požiadaviek na činnosť, funkcionalitu systému, obsahujúca štúdia uskutočniteľnosti (projektová úloha, úvodný projekt); sú to štúdie určené k identifikácii a analýze problémov spojených s predbežným návrhom na vývoj systému s cieľom ukázať jeho realizovateľnosť a účelnosť špecifikácie požiadaviek; je to podrobná formulácia v tvare dokladu poskytujúca definitívny opis systému z pohľadu užívateľa pre potreby jeho vývoja a ocenenia jeho správnosti.
- *Analýza (system analysis)* - je modelovanie budúceho systému na konceptuálnej úrovni.
- *Návrh (system design)* - je modelovanie budúceho systému na technologickej úrovni; je to prevedenie špecifikácií na hierarchii stále detailnejších schém, ktoré definujú

požadované dáta a rozkladajú procesy, ktoré sa majú s dátami vykonávať, až na úroveň, kedy môžu byť vyjadrené v podobe inštrukcií pre konkrétny počítačový program.

- *Vývoj systému (system development)* - je písanie a testovanie počítačového softvéru a teda programovanie, je to vývoj vstupných a výstupných formulárov a konvencií.
- *Implementácia systému* - je uvedenie systému ako hardvéru tak aj softvéru do prevádzky; je to inštalácia fyzického systému a aktivít s tým súvisiacich, ako je školenie správcov, operátorov a užívateľov.
- *Správa systému* - je to ďalší vývoj funkcií a štruktúry systému, ktoré vyplývajú zo zmenených požiadaviek a technológií, skúseností s užívaním systému a dolaďovanie jeho výkonu.
- *Údržba systému* - je úprava systému pri jeho prevádzkovaní, podľa vzniknutých požiadaviek užívateľa, prípadne po odstránení vzniknutých závad.

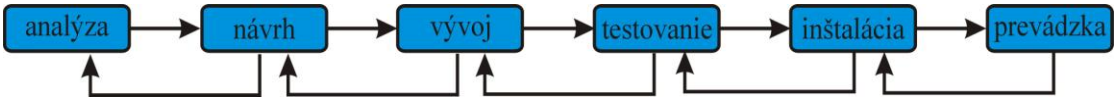
**Typy životných cyklov vývoja systému:**

**1. vývojový cyklus „vodopád“**

**výhody:** - prehľadnosť, jednoduchosť, jasná postupnosť etáp

**nevýhody:**
- užívateľ často nedokáže v počiatočných etapách formulovať presné požiadavky na systém,
- neráta sa s dodatočnými zmenami v priebehu riešenia - spätná väzba od užívateľa je k dispozícii až na záver a teda hrozí neskoré odhalenie nedostatkov,
- systém je dodaný celý naraz a užívateľ má dlho pocit, že sa nič vlastne nedeje.

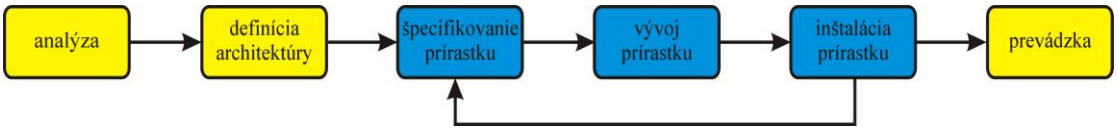
**2. fontánový typ**




**výhody:** - po dosiahnutí vyššej etapy sa vraciame k predchádzajúcej etape

**nevýhody:** - vraciame sa vždy len o jeden krok a preto najzávažnejšie chyby sa opäť môžu prejaviť až na záver pri prevádzke

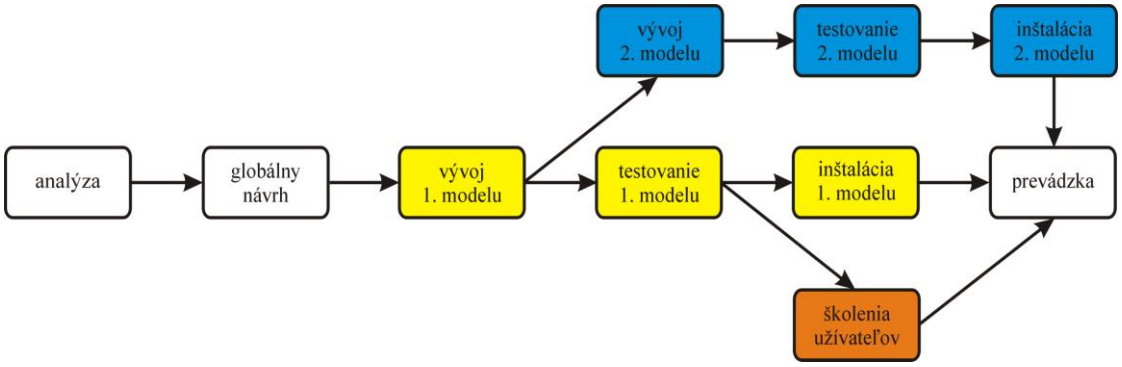
**3. prírastkový - inkrementálny vývojový cyklus**




**výhody:** - systém je celý špecifikovaný a navrhnutý na začiatku, ale zákazníkovi je dodávaný v sérii prírastkov, teda spätná väzba od užívateľa je po každom odovzdaní prírastku

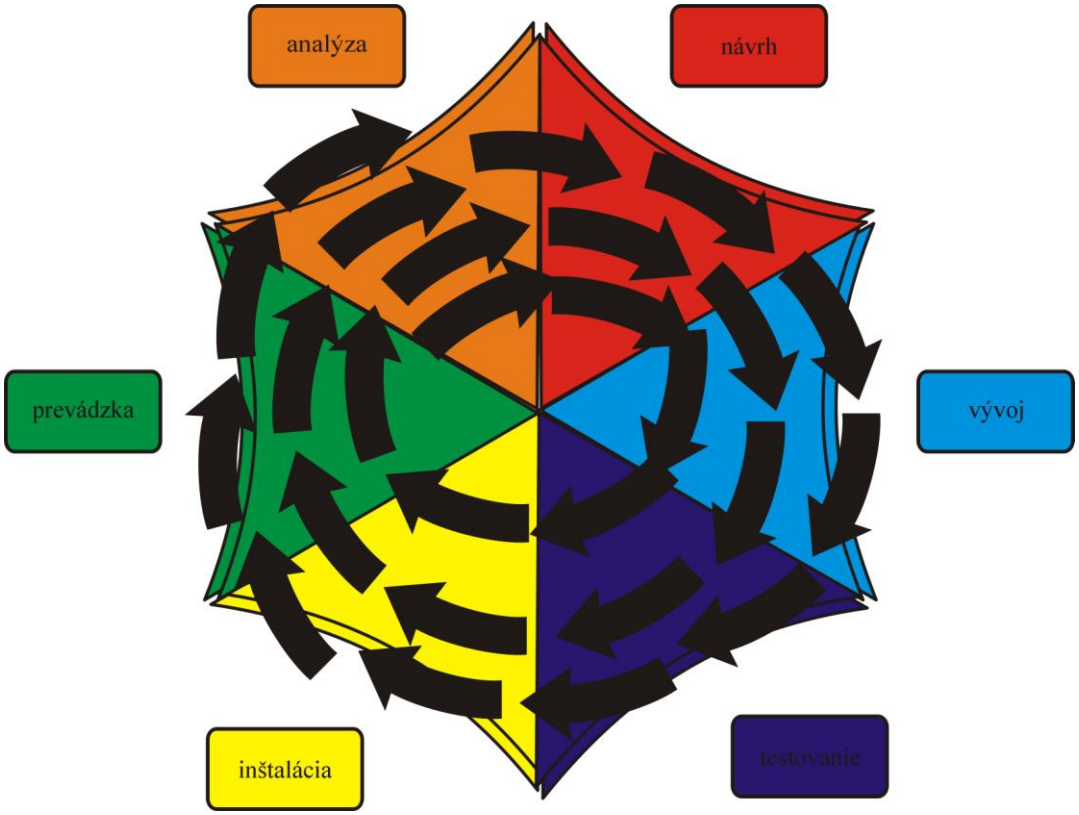
**nevýhody:** - obtiažne sledovanie a kontrolovanie postupu prác na vývoji systému

**4. sieťový typ**




**výhody:** - dosahovanie časových úspor umožnením súčasného teda paralelného riešenia niektorých etáp

5. **špirálový typ**




**výhody:**
- postupné zdokonaľovanie systému ďalšími verziami - etapy sa opakujú v rovnakú postupnosť,
- kombinácia prehľadnosti vodopádového modelu s prírastkovým a iteratívnym vývojom.

**Nedostatky metodológie životného cyklu systému:**

- značná časová a finančná náročnosť,
- veľa krokov, po každom nasleduje množstvo dokumentácie,
- neberie do úvahy užívateľa po celej dobe vývoja,
- vysoký podiel činností programovania a testovania,
- reálne projekty spravidla nerešpektujú jednotlivé kroky v predpísanom poradí.

**Alternatívy tradičného životného cyklu systému:**

1. **Softvérové balíky** (OpenOffice.org, 602Pro PC Suite, Microsoft Office atď.) - sú to

sady programov (software package) a teda úplné a dokumentované množiny programov, dodávané viacerým užívateľom pre generickú aplikáciu alebo funkciu.

2. **Knižnice programov a komponentov (reusable code)**.
3. **Prototypovanie, prototyp** je modelová alebo predbežná implementácia, vhodná na vyhodnotenie projektu systému, jeho výkonnosti a možností, alebo pre lepšie porozumenie požiadavkám, prípadne stanovenie požiadaviek na systém.
4. **CASE** - sú to programy na podporu vývoja systému s vlastnou metodikou.
5. **Reverse engineering** - z implementovaného systému spätne odvodzujeme jeho model.
6. **Objektovo orientovaná a komponentová technológia**.
7. **Vývoj systému koncovým užívateľom**.
8. **Extrémne programovanie, agilné programovanie**.

**Smery v životnom cykle:**

- forward engineering (dopredné inžinierstvo, top-down)

**MODEL ⟶ KÓD**

- reverse engineering (spätné inžinierstvo, bottom-up)

**KÓD ⟶ MODEL**

**Trendy:**

- iterácia (opakovanie procesov);
- inkrementácia (prírastkový postup);
- komponenty (modulárny postup).

**IID - iteratívny a inkrementálny vývoj** - je prístup integrujúci pozitívne skúsenosti z vodopádového, prírastkového a prototypového modelu

**Špecifikácia požiadaviek na komunikačný systém - odpovedá na otázky:**

- Čo má komunikačný systém robiť? (funkcionalita).
- Pre koho? (používatelia).
- Za akých podmienok? (obmedzenia).

**Požiadavka -** je zdokumentovaná reprezentácia:

- vlastností alebo podmienok umožňujúcich riešiť problém alebo dosiahnuť ciele projektu;
- vlastností, ktorými by mal disponovať navrhovaný komunikačný systém;
- služieb, ktoré má systém poskytovať;
- obmedzení, za ktorých má systém pracovať.

**Špecifikácia požiadaviek -** je technický dokument, ktorý:

- ustanoví požiadavky na funkcionalitu budúceho systému z pohľadu užívateľa (používame aj jeho terminológiu);
- neobsahuje návrh riešenia;
- nerieši obchodné vzťahy medzi zadávateľom a riešiteľom (tie stanovuje obchodná zmluva).

**Vlastnosti kvalitnej špecifikácie:**

- kompletnosť (má obsahovať všetky požiadavky);
- konzistencia (nerozpornosť - jednotlivé požiadavky si nesmú navzájom odporovať);
- parametrizácia požiadaviek - požiadavkám sú priradené kvantitatívne a kvalitatívne atribúty: dôležitosť, technická náročnosť riešenia, analýza rizík, vzťah k iným požiadavkám, užívatelia;
- kategorizácia užívateľov;
- rovnaká úroveň detailu každej požiadavky (v prípade potreby je možné dokument štruktúrovať do ďalších úrovní);
- kontrolovateľnosť.

**Tip pre stanovenie požiadaviek:**

1. sformulujte otázky, na ktoré má komunikačný systém poskytovať odpovede,
2. pomenujte problémy, ktoré má komunikačný systém pomôcť vyriešiť.

**Základné otázky, ktoré si kladieme pred začatím projektovania komunikačného systému (KS):**

1. Čo chceme riešiť (projektovať) a prečo?
2. Komu to bude slúžiť?
3. Aký úžitok z toho bude mať užívateľ KS a aký prevádzkovateľ KS:

   aký čas sa ušetrí?

   aké prostriedky sa ušetria?

   aké nové služby bude možné realizovať?

4. Koľko to bude stáť a kto to zaplatí:

   aké budú investície?

   aká bude prevádzka systému?

5. Dá sa prevziať alebo modifikovať už existujúce riešenie?
6. Aká bude kooperácia a interoperabilita, prípadne kompatibilita s inými systémami?
7. Aké požiadavky budú na systém kladené v budúcnosti?

**Štruktúrovaný popis problému, špecifikácia požiadaviek:**

- cieľ - účel projektovaného systému,
- prvky systému - podstatné znaky, odlíšenie od okolia, vzťah k už existujúcim systémom,
- funkcie - procesy systému - typy riešených úloh: napr. vyhľadávanie uložených dát, otázky, analýza údajov - vzorce, prezentácie spracúvaných údajov - sumarizácia, grafy a podobne,
- používatelia systému, dodávatelia a odberatelia informácií,
- návrh možných alternatív riešení a výber tej alternatívy, ktorá sa bude realizovať, prípadné vymedzenie hraníc projektu.

**Odpoveď na veľmi dôležité otázky:**

- Čo má projektovaný komunikačný systém dosiahnuť?
- Pre koho bude komunikačný systém určený, ako a do kedy?
- Čo je dôvodom návrhu komunikačného systému a čo nové prinesie?

**Analýza a návrh KS:**

- Definovanie výstupov - požiadavky na cieľové správanie KS:
  - obsah,
  - forma (napr. tabuľka, dokument, formulár, recenzia, správa a pod.),
  - kvantifikácia (napr. 3 tabuľky a 5 dokumentov za obdobie, počet vyriešených úloh za obdobie atď.),
  - periodicita (napr. denne, raz týždenne, na konci semestra a pod.).

  *Odpoveď na otázky:*
  - Aké sú požiadavky na cieľové správanie KS?
  - Aké informačné produkty (výstupy) by mali byť vytvorené?
  - Prečo, resp. načo (k čomu) potrebujeme určité dáta?
- Definovanie vstupov - požiadavky na zdroje systému:
  - obsah,
  - forma a technológie zberu dát,
  - kvantifikácia,
  - periodicita.

  *Odpoveď na otázky:*
  - Aké zdroje dát sú k dispozícii - odkiaľ dáta získame?
  - Ako sa majú zhromažďovať (získavať) dáta?
  - Ako sa budú dáta pripravovať na spracovanie?
- Spracovanie dát:
  - použité triedenie (číselníky) - definovanie skupín údajov,
  - definovanie sumarizačných a vypočítaných funkcií.

  *Odpoveď na otázku:*
  - Ako sa bude s dátami manipulovať a ako budú premieňané na informácie?
- Uloženie a sprístupnenie dát:
  - spôsob archivácie,
  - spôsob vyhľadávania,
  - spôsob sprístupnenia údajov,
  - spôsob ochrany dát.

  *Odpoveď na otázky:*
  - Ako budú zdroje dát organizované, uchovávané, aktualizované a vyhľadávané?
  - Ako budú informácie doručované, sprístupnené užívateľom?

- Definovanie dátovej základne:
  - konceptuálny model:

    dátový model (ERA diagram)

    diagram tried

    diagram dátových tokov
  - logický model (logická - fyzická štruktúra dát pre konkrétny softvér a hardvér):

    slovník dát (data dictionary)

    rôzne schémy relačnej databázy a definície integritných obmedzení

**Návrh riešenia implementácie KS:**

- návrh a odhad finančnej náročnosti obstarania potrebného hardvéru a softvéru,
- návrh finančného a organizačného zabezpečenia vstupu dát,
- odhad veľkosti (objemu) dátovej základne (počet záznamov, veľkosť napr. v GB),
- návrh metodiky zberu dát,
- návrh typológie dátovej základne a spracovateľských procesov (centralizácia, decentralizácia, distribuované spracovanie atď.),
- návrh médií na ukladanie dát,
- návrh spôsobu prenosu informácií medzi funkčnými prvkami systému (lokálna sieť, internet, e-mail, USB kľúče, CD-ROM, telefón a pod.),
- návrh zaistenia bezpečnosti a archivácie dát,
- návrh personálneho zabezpečenia systému.

**Computer Aided Software (System) Engineering - CASE -** je nástroj na podporu práce analytikov a programátorov pri vývoji informačných a komunikačných systémov, najmä vo fáze analýzy a návrhu - je to tvorba modelov, je to medzistupeň medzi analýzou problémov a programovaním, je to označenie pre integrovanú tvorbu programových aplikácií pomocou programových prostriedkov s minimálnou potrebou manuálneho písania zdrojového kódu programu (Vrana, 1997). Obsahom CASE sú databázy (repository, systémová encyklopédia) navrhovaných prvkov KS. Funkciou je podpora realizácie projektu KS a základ tvorí metodika, t. j. návod na vytváranie modelov a určenie závislostí medzi nimi. Medzi základné funkcie CASE patrí automatizovaná evidencia vytvorených objektov a špecifikácií, dokumentácia vývoja systému, grafická podpora modelovania

(notácia), kontrola správnosti modelov podľa zvolenej metodiky, zabezpečenie integrity a konzistencie návrhu (preddefinované integritné obmedzenia a ich kontrola, automatické uplatnenie zmien vytvorených v jednej časti vo všetkých súvisiacich častiach návrhu), podpora tímovej práce, t. j. identifikácia osôb a tímov zodpovedných za jednotlivé modely, tvorba viacerých modelov súčasne, súčasná práca viacerých osôb na jednom modeli, správa verzií, automatický prevod definovaných modelov do konkrétneho logického a fyzického návrhu, generovanie programu, popis databázy prípadne prototypu, reverse engineering - spätné generovanie konceptuálneho modelu komunikačného systému z existujúcej aplikácie.

## 1.4 Porovnanie tradičných modelov implementácií komunikačných systémov

Ako sme už spomenuli v predchádzajúcej kapitole, v praxi teda existujú 3 spôsoby komplexnej implementácie KS: **vlastné riešenie**, **zakúpené hotové „krabicové“ riešenie** a **kombinácia týchto dvoch spôsobov**. Veľkými výhodami komunikačného systému implementovaného „na kľúč“ sú hlavne jeho odladenia na konkrétnu potrebu organizácie a rozvoj samotnej aplikácie zároveň s rozvojom tejto organizácie. Krabicové riešenie je ako krabicové víno predávané v krásnej farebnej krabici - dá sa použiť, ale veľmi nás neuspokojí. Obsahuje často veľa funkcií, ktoré vlastne ani nikdy možno nevyužijeme a na druhej strane funkcie, ktoré nebudeme využívať úplne podľa našich predstáv.

Postupy pri nákupe softvéru sa zvyčajne líšia podľa toho, k akému účelu je softvér vyhotovený. Na trhu je obrovské množstvo hotových softvérových riešení, od tých jednoduchých, ako je napríklad všetkým dobre známa sada kancelárskych nástrojov Microsoft Office, až po zložité podnikové informačné systémy. Sú však odbory, v ktorých je činnosť tak špecifická, že krabicový softvér nie je vhodný, a je nutné nechať si riešenie vytvoriť podľa špecifických požiadaviek. Čím komplikovanejšie a individuálnejšie sú procesy vo firme, čím unikátnejšie je jej know-how, tým naliehavejšie firma potrebuje softvér na zákazku.

Krabicovým softvérom zvyčajne označujeme softvérový produkt balený v krabici vrátane všetkých potrebných komponentov, ako sú inštalačné médiá, licenčné dojednania a podobne. Pre naše porovnanie však bude vhodnejšie označovať termínom „krabicový

softvér“ všetky softvérové produkty vyvíjané na základe všeobecných požiadaviek širokej verejnosti. Ideálnym príkladom takéhoto krabicového softvéru je už spomenutá sada nástrojov Microsoft Office. Takýto produkt je charakterizovaný najmä veľmi širokou cieľovou skupinou zákazníkov, plynúcou z jeho cenovej dostupnosti, možnej viacjazyčnosti a špecializáciou na činnosti obvyklé u väčšiny používateľov. Obvyklou nevýhodou oproti na mieru navrhnutým aplikáciám je skutočnosť, že nie je prispôsobený konkrétnym potrebám toho ktorého užívateľa, ale užívateľ musí svoje potreby do istej miery prispôsobiť tomuto softvéru. Výhodami krabicových riešení je potom už spomínaná nižšia cena vyplývajúca práve z jeho masovosti a fakt, že spolu s právom používať softvér si užívateľ často kupuje zároveň aj službu technickej podpory.

Najväčšou výhodou softvéru na zákazku je schopnosť pokryť jedinečné požiadavky užívateľa, rýchlosť, flexibilita a schopnosť reflektovať na akékoľvek potreby vrátane tých, ktoré sa výrobcom štandardného softvéru neoplatí pokrývať. Hotový softvér úspešne pokrýva bežné štandardné procesy, avšak väčšinou nestačí podporiť nové, aktuálne sa vynárajúce potreby zákazníka. Softvér na zákazku potom obvykle vykrýva priestor medzi štandardnými softvérovými riešeniami. Všeobecne sa riešenie softvéru „na mieru“ používa vtedy, ak softvér, ktorý používateľ potrebuje, buď nie je na trhu k dispozícii, alebo ho nie je možné modifikovať, alebo je naopak tento softvér príliš komplexný a zákazník by využíval len jeho časť. Softvér na zákazku si často nechávajú vytvoriť zákazníci so špecializovanými činnosťami, napríklad v zdravotníctve, účtovníctve, alebo pre výučbové programy v školstve.

Výber medzi hotovým, krabicovým softvérom a softvérom na zákazku závisí na konkrétnych potrebách zákazníka, množstve softvéru a jeho celkovej IT stratégii. Nespornou výhodou štandardného softvéru je pevná cena, harmonogram implementácie a vopred už známe vlastnosti produktu. Väčším organizáciám so zložitou štruktúrou IT prostredia však toto riešenie nemusí vyhovovať, predovšetkým preto, že akokoľvek dobrý štandardný softvér v priebehu času nestačí na nové požiadavky. Zákazkový vývoj softvéru je totiž potrebné chápať ako komplexnú službu poskytovanú jedným dodávateľom. Mala by zastrešovať jednotlivé úkony v priebehu celého životného cyklu projektu komunikačného systému - od jeho návrhu a vývoja, cez implementáciu a dodávku hardvérových a softvérových komponentov, až po podporu prevádzky finálneho produktu a jeho ďalší rozvoj. Rozhodne sa teda nejedná iba o jednorazový predaj a následnú kúpu zákazkového KS. Je preto potrebné venovať viac úsilia pri výbere firmy, ktorá softvér na

zákazku implementuje, brať do úvahy overené referencie, históriu dodávateľa, jeho odbornosť a získané certifikácie.

Všeobecne teda platí, že zákazkový vývoj je pre zákazníka efektívnejší. Spôsob práce uplatňovaný pri vývoji softvéru na kľúč nielen umožňuje, ale dokonca vyžaduje spoluúčasť klienta, ktorý sa tak stáva spolutvorcom celého riešenia. Výsledné riešenie je teda presne prispôsobené jeho požiadavkám, vyhovuje organizačným procesom i firemnej kultúre a môže aj veľmi dôsledne podporovať kľúčové odlišnosti od konkurencie.

Je dôležité si uvedomiť, že existuje veľký rozdiel pri nasadení finálneho krabicového softvéru oproti riešeniu, ktoré je tvorené na mieru podľa požiadaviek zadávateľa. Krabicový softvér je testovaný veľkým počtom užívateľov, ktorí vytvárajú dlhodobú spätnú väzbu. Vďaka tejto spätnej väzbe výrobca postupne produkt vylepšuje. Takýto produkt je otestovaný časom. Oproti tomu zákazkový softvér býva krokom do neznáma. Aj keď sa obvykle riešia podobné problémy, nakoniec nikdy neexistuje rovnaké riešenie. Rôznorodé požiadavky zadávateľov prinášajú aj rôznorodé problémy. Je to podobné, ako keď sa podľa želania investora navrhuje a stavia dom. Prvý dôležitý krok je stanovenie účelu stavby. Príde na rad architekt, ktorý musí správne pochopiť potreby obstarávateľa a pretransformovať ich na konkrétne požiadavky staviteľa. Ďalej je nutné vhodne zvoliť stavebný materiál, dobre položiť základy, postaviť správne hrubé obvodové múry a zabezpečiť kvalitnú strechu. Každá jednotlivá činnosť je dôležitá, každé opomenutie v kvalite prevedenia jednotlivých častí komplikuje celkovú stavbu. Nielen že jednotlivé zakopnutia ohrozujú zvyčajne napätý rozpočet, ale v prípade kumulácie problémov môže dôjsť k tomu, že výsledok vôbec nezodpovedá tomu, čo sme pôvodne chceli postaviť.

Pri vývoji softvéru dochádza takisto k podobným rizikám. Najdôležitejšou časťou projektu je stanovenie požiadaviek na KS, a to predovšetkým schopnosť odlíšiť dôležité požiadavky od tých menej významných, alebo dokonca úplne nepodstatných. Zdá sa, že rovnako ako v ekonómii, tak aj tu platí už spomenuté Paretovo pravidlo 80:20 - v tomto prípade dvadsať percent požiadaviek, ktoré definuje obstarávateľ, je tak dôležitých, že pokryjú osemdesiat percent skutočných potrieb. Ide teda predovšetkým o nájdenie týchto kritických požiadaviek a priradenie zodpovedajúcich priorít. V prípade existencie problémov, či už časových alebo rozpočtových, je nevyhnutné sa zamerať predovšetkým na prioritné požiadavky a nevyčerpávať energiu na niečo, čo v skutočnosti nie je vôbec dôležité. Inak hrozí zvýšenie nákladov kvôli realizácii nepotrebných funkcionalít, alebo

zníženie kvality v dôležitých častiach systému. V najhoršom prípade oboje zároveň.

Zložitou časťou implementácie KS je prevedenie problému z reálneho sveta do domény informačných technológií. Častou chybou je snaha o otrockú migráciu existujúcich postupov a procesov do nového KS. Tento postup obyčajne vedie k zbytočnej zložiteľnosti a byrokratizácii procesu. Čo dobre funguje v reálnom svete, nemusí dobre fungovať (a obvykle tiež nefunguje) vo svete IT. V reálnom svete neexistujú zásadné obmedzenia, a ak proces definovaný smernicou nefunguje, je možné ho modifikovať ad hoc rozhodnutím konkrétneho pracovníka. Vo svete IT ale tieto ad hoc riešenia nemusia byť vôbec k dispozícii. Napríklad poznámka, ktorá sa zvyčajne píše na jednoduchom formulári pod čiarou, teraz v novom systéme sa nedá do žiadnej kolónky zapísať, alebo dokument omylom poslaný kolegovi, ktorý je teraz na dovolenke, už nemožno získať späť pre dôležitú korektúru a podobne. Je teda nutné prejsť všetky procesy a rozhodnúť sa, ktoré ponechať, ktoré od základu zmeniť a ktoré do KS vôbec neprevádzať. V prípade nesprávnej migrácie procesov môže dôjsť k tomu, že miera byrokratizácie systému prekročí únosnú hranicu a vysnívaný systém nakoniec prinesie viac nových problémov, než koľko má pôvodne riešiť.

Ďalšou kapitolou je nájdenie vhodnej platformy a technológie. Tu dodávateľ často pretláča svoje riešenie, ktoré môže, ale nemusí byť v súlade so skutočnými potrebami zákazníka. Výber technológie je dôležitý pre budúce rozšírenie a kompatibilitu s aplikáciami tretích strán. S týmto tiež súvisí požiadavka na určitú nezávislosť na dodávateľovi systému. Neexistencia aspoň hypotetickej možnosti odísť v budúcnosti ku konkurencii týmto zvyšuje prevádzkové náklady a zdražuje prípadné budúce požiadavky na rozšírenie.

Dôležitou časťou je testovanie výslednej aplikácie. Neide len o bežné vyskúšanie základnej požadovanej funkcionality, ale o širšie zodpovedanie otázok typu „čo sa stane, keď...“. Napríklad čo sa stane pri väčšej záťaži (so systémom pracuje väčšie množstvo používateľov ako obvykle), pri neoprávnenom prístupe, pri neštandardnom postupe užívateľa a podobne. Pri nedostatočnom otestovaní aplikácie hrozí akceptácia nekvalitného riešenia. Následne skryté kritické chyby môžu neskôr dokonca poškodiť prevádzkové dáta a problém nakoniec môže eskalovať aj mimo prostredia organizácie.

Vyššie spomínané riziká môžu neplánovite navýšiť celkové náklady na KS. Okrem nákladov, ktoré sú vidieť - základný vývoj a implementácia systému - existujú tiež

náklady, ktoré bežne nie sú viditeľné v tomto kontexte. Sú to náklady súvisiace s:

- neefektivitou pracovných postupov užívateľov systému,
- administráciou systému,
- riešením havarijných situácií systému,
- budúcim rozšírením a teda novými požiadavkami na systém,
- integráciou s inými systémami.

Opis toho, čo organizácia potrebuje, a opis špecifikácie budúcich výsledkov, ktoré si predstavuje, je práca obstarávateľa - budúceho užívateľa komunikačného systému. Bez predchádzajúcich skúseností je to však pomerne zložitá úloha. Dodávateľ KS by mal byť schopný pomôcť a následne prísť s vlastnou sumarizáciou a návrhom možných riešení. V prvom kroku je potrebné popísať deje, ktoré vo firme prebiehajú. Zistiť, aké informácie evidujú jednotlivé oddelenia, ako prebieha ich prenos a kde sú úzke miesta. Vytvorí sa pracovný model, do ktorého KS zapadne.

Tvorba KS na mieru je väčšinou smerovaná predovšetkým do špecifických oblastí obchodu, priemyslu a služieb. Týchto systémov je oproti klasickým ekonomickým oveľa menej. Navyše ekonomické systémy majú úzku väzbu na legislatívu a tou sú nútené sa riadiť všetky podnikateľské subjekty. A práve softvér pre vyššie uvedené oblasti je v mnohých firmách natoľko špecifický, že len niekedy možno aplikovať hotové krabicové systémy. Nielen že to vyžaduje veľmi často meniť organizačnú štruktúru firmy, ale je nutné prepracovať aj jednotlivé kroky a činnosti. A pritom môže ísť o veľmi dobre fungujúce činnosti. Preto je praktickejšie nechať si naimplementovať systém, ktorý bude organizácii naprogramovaný na mieru.

Kvalitný KS nemusí byť doménou len veľkých a ekonomicky silných podnikov. Aj stredné a menšie firmy, ba aj rôzne inštitúcie potrebujú na to, aby mohli uspieť v dnešnej konkurencii dobrý manažérsky nástroj. Je prinajmenšom jasné, že jednou z mála ciest k prosperite firmy je nielen jej znižovanie nákladov, ale aj zdokonalenie organizačných procesov. A to je možné bez kvalitného komunikačného systému dosiahnuť len veľmi ťažko.

Je vidieť, že vývoj, implementácia a prevádzka komunikačných systémov je komplexný proces, ktorý prináša určité riziká, či už projektové, technické alebo obchodné.

Skúsený dodávateľ môže týmto rizikám predchádzať, avšak nie vždy sa záujmy dodávateľa kryjú so záujmami zadávateľa. Navyše môže byť ťažké kvalitného a skúseného dodávateľa vôbec nájsť. Východiskom z tejto situácie môže byť existencia nezávislého pohľadu tretej strany. Podobne ako môže byť ohrozená stavba bez kvalitného stavebného dozoru, implementácia KS môže byť veľmi riziková bez nezávislého softvérového dozoru.

Softvérový dozor stojí na strane zadávateľa a jeho úlohou je predovšetkým:

- analýza a detekcia dôležitých prioritných požiadaviek,
- korekcia nereálnych prianí a vízií,
- rozoznanie, analýza, plánovanie a monitorovanie rizík,
- kontrola technickej realizácie,
- kontrola testovania,
- kontrola produkčného prostredia.

Softvérový dohľad by mal vykonávať odborník s kvalitným vzdelaním a dlhoročnou praktickou skúsenosťou v odbore implementácií komunikačných systémov. Či už obstarávateľ využíva služby softvérového dozoru, alebo nie, je nutné venovať dostatočný čas predovšetkým prípravnej analytickej fáze, kedy sa dávajú dohromady kľúčové požiadavky a technológie. Aj na drobné chyby v tejto fáze sa môžu postupne nabaľovať ďalšie problémy a v konečnom dôsledku degradovať výsledný produkt, keď vynaložená energia a hlavne náklady nezodpovedajú očakávaným výsledkom.

## 1.5 Alternatívne modely implementácií komunikačných systémov

Skratku **SaaS** (Softvér ako služba) nie je treba nijako zvlášť predstavovať. Je to nový model dodávky softvéru v podobe služby pomocou prostredia Cloud Computing. Ten je ďalšou vývojovou etapou evolúcie softvéru v čase. Cloud Computing (skrátene „Clouding“) sa podľa (http://xprogramming.com) dostal medzi top 10 strategických technológií pre rok 2010. Do Cloudingu sa ďalej prirodzene radia modely IaaS (Infraštruktúra ako služba) a PaaS (Platforma ako služba).

Cloud computing je na internete založený model vývoja a používania počítačových technológií (http://www.itnews.sk). Možno ho tiež charakterizovať ako poskytovanie služieb alebo programov uložených na serveroch na internete s tým, že používatelia k nim môžu pristupovať napríklad pomocou webového prehliadača alebo klienta danej aplikácie a používať prakticky odkiaľkoľvek. Používatelia neplatia (za predpokladu, že je služba platená) za vlastný softvér, ale za jeho použitie. Ponuka aplikácií sa pohybuje od kancelárskych aplikácií, cez systémy pre distribuované výpočty, až po operačné systémy prevádzkované v prehliadačoch, ako je napríklad eyeOS, Cloud alebo iCloud.

Stále je veľmi často možné sa stretnúť s názormi, že SaaS - softvér ako služba - je len momentálna záležitosť, sezónny fenomén, ktorý spľasne ako bublina. Na druhej strane spoločnosť Gartner (http://www.itnews.sk) už v roku 2008 predpovedala, že v roku 2012 by až tretina výdavkov na IS/IKT v rámci organizácie mohla byť vynakladaná na aplikácie v modeli dodávky SaaS. Táto skutočnosť je dostatočne pádny argument pre bližší záujem o model. Kde sa teda vzala potreba a ochota organizácií neinvestovať do vlastného riešenia, ale nakúpiť „len“ službu?

Ešte začiatkom osemdesiatych rokov bola v organizáciách bežná vertikálna integrácia, kedy podnikateľský subjekt riadi a vlastní všetky súčasti podnikateľského reťazca. Postupne si organizácie začali uvedomovať, že správa celého podnikateľského reťazca a všetkých jeho súčastí môže byť neefektívna a neumožňuje im sústrediť sa na kľúčové časti ich podnikania. Prečo vlastniť procesy, ktoré sú pre hlavný cieľ nášho podnikania len podporné? Pracovníkov „hviezdy“, najlepších z najlepších, budú organizácie vždy primárne potrebovať v rámci svojho hlavného procesu. Takmer vždy je možné nájsť niekoho, pre koho bude doplnkový proces kľúčový a bude na jeho podporu prideľovať tých najlepších pracovníkov. Tak prečo zbytočne trieštiť sily, keď je možné výsledky procesu nakúpiť ako službu? Väčšina aplikácií IS/IKT má pre organizácie význam len ako vedľajší proces, teda majú aj len obmedzenú pridanú hodnotu.

Je skutočne možné, že sa jedného dňa dostaneme do situácie, kedy si budeme musieť priznať, že všetky IT sú skutočne len komoditou. Zatiaľ stále existujú kľúčové aplikácie, ktoré organizácie nie sú ochotné nakupovať iným modelom než klasickým. Všetky ostatné aplikácie sú vhodné pre nákup alternatívnymi modelmi, napríklad práve model softvér ako služba. Je totiž veľmi pravdepodobné, že pre procesy, ktoré sú pre našu organizáciu len podporné, nájdeme dodávateľa. Pre neho budú tieto procesy kľúčovou činnosťou a bude schopný alokovať na ich výkon a dodávku svojich najlepších ľudí.

V priebehu historického vývoja informačných technológií sa v rámci organizácií model vývoja a prevádzky IS/IKT významne stabilizoval. Pod tradičným vývojom je chápaný nasledujúci postup implementácie novej aplikácie: spoločnosť nakúpi licencie, základný hardvér a softvér od externých dodávateľov, komunikačný systém je naimplementovaný a integrovaný v rámci organizácie s pomocou externých poradcov - systémových integrátorov. Potom je systém prevádzkovaný vlastnými silami. Tento model vývoja a prevádzky IS/IKT sa postupne stal veľmi nákladný najmä na prvotné, vstupné investície. Organizácia ešte nevyužije softvér pre podporu procesov, ale už musí zaplatiť minimálne polovicu z celkových nákladov na budúci IS.

Princíp nákupu softvéru v klasickom modeli by sme mohli prirovnať napríklad k podnikaniu ropných spoločností - kúpia si určitú parcelu, na vlastné náklady vykonajú výskum a vrty, ak majú šťastie a nájdu použiteľné ložiská, preinvestujú pri budovaní plošín a inej infraštruktúry 90 % celkových nákladov a až približne za 30 rokov sa dostávajú z červených čísel. Skutočnosť, že nie je možné dať jasnú odpoveď na otázky o skutočných nákladoch na IS/IKT, organizáciám veľmi sťažuje plánovanie a škálovanie nákladov. Vedúci IT oddelenia nie sú schopní presne vyčísliť pridanú hodnotu, ktorou prispievajú k hlavnému predmetu podnikania organizácie. Podľa už spomenutej spoločnosti Gartner je 80 % nákladov vynaložených na IS/IKT použité na udržanie súčasného stavu a len 20 % na investície do vylepšenia. Nielen u už implementovaných aplikácií je problémom vyčísliť náklady na ich vlastníctvo a prevádzku, ale už v priebehu implementácie sa často nedarí náklady plánovať a riadiť.

Klasický model implementácie a prevádzky aplikácií prináša riziká, ktoré ani prepracované a mnohokrát aplikované metodiky implementácie neodstránili. Pretože v mnohých organizáciách je stále IT oddelenie považované za akési „nevyhnutné zlo“ a neprispieva vlastne ničím k hlavnej činnosti, snaha organizácií ušetriť náklady sa markantne premieta do znižovania rozpočtov IS/IKT. A hoci oddelenia IT sú často schopné vysvetliť svoj prínos k výsledkom hlavných procesov organizácií, mnohokrát pri akomkoľvek neúspechu leží vina na samotnom oddelení IT, niekedy je to dané historickým vývojom tohto oddelenia v konkrétnej organizácii.

Pokúsme sa teraz pozrieť na dôvody, prečo zákazníci nakupujú aplikácie v modeli softvér ako služba, aké sú úskalia nákupu v tomto modeli a na čo by sa mali organizácie pri výbere zamerať. Medzi najčastejšími dôvodmi nákupu v modeli softvér ako služba - ak teda porovnáme rôzne prieskumy, ktoré skúmajú dôvody na zakúpenie programu v modeli

softvér ako služba - sa takmer vždy na prvom mieste umiestňuje rýchlosť a jednoduchosť implementácie. V tesnom závese potom nasledujú nízke celkové náklady na vlastníctvo, predikovatelnosť nákladov a škálovateľnosť aplikácie podľa aktuálnych potrieb. Dôvody sa pritom nelíšia od veľkosti organizácie ani typu odvetvia, v ktorom organizácia pôsobí. Úskalia výberu riešenia existujú všade.

Pred samotným výberom konkrétneho riešenia a dodávateľa musí organizácia zistiť, či jej vnútorné prostredie je pripravené na nákup aplikácie ako služby. Vnútorné procesy, pri ktorých sa uvažuje o podpore aplikácií nakúpených ako služba, musia byť popísané, musia byť definovaní ich vlastníci a zadefinovaná opakovateľnosť. Organizácie nemôžu nakúpiť aplikáciu v modeli softvér ako služba a očakávať, že vďaka vnútorným mechanizmom aplikácie sa začnú správne vykonávať procesy v organizácii. Pred tým, ako sa organizácia rozhodne nahradiť existujúce alebo nakúpiť nové aplikácie v modeli softvér ako služba, musia poznať odpovede aj na tieto otázky:

- Podporuje budúca aplikácia hlavné alebo vedľajšie procesy spoločnosti?
- Je podporovaný proces dostatočne štandardizovaný v rámci spoločnosti, aby bolo možné nakúpiť uniformované riešenie?
- Bude potrebná integrácia medzi novým systémom a existujúcimi systémami v tradičnom modeli?
- Máme dostatok skúseností, aby sme riešenie implementovali sami, alebo budeme potrebovať externých konzultantov?
- Kto bude zodpovedný za vzťah s dodávateľom?
- Existuje pre požadovaný systém vhodné riešenie v modeli softvér ako služba?
- Je naše pripojenie dostatočne spoľahlivé, aby sme mohli k aplikácii pristupovať cez internet?
- Sme ochotní umiestniť naše citlivé dáta mimo sídla organizácie?
- Je pre nás potencionálne vybraný dodávateľ dostatočne spoľahlivý a dôveryhodný?

Priemerná doba, za ktorú začnú aplikácie prinášať užívateľom úžitok, je v tradičnom modeli riešenia 27 mesiacov a v prípade modelu softvér ako služba sa jedná o 6 mesiacov, výpovedná lehota sa pohybuje v rádoch mesiacov pri SaaS a u klasického modelu vlastne nič také neexistuje (http://www.itnews.sk). Kratšia ako predpokladaná doba použitia aplikácie v tradičnom modeli môže zvyšovať jej celkové náklady na

vlastníctvo. Model SaaS umožňuje zákazníkom otestovať riešenia v ostrej prevádzke len s minimálnymi nákladmi. Rôzne spoločnosti ponúkajú mesačné použitie svojich systémov zadarmo, avšak pre obmedzený počet užívateľov; a ak sa organizácia rozhodne nakúpiť aplikáciu v modeli softvér ako služba, môže tak urobiť len napríklad pre jednu jednotku a až po získaní skúseností môže toto riešenie rozšíriť aj do celej spoločnosti.

Tým, že organizácia nakúpi riešenie v modeli softvér ako služba, sa týmto vôbec nezbavuje všetkej zodpovednosti za kvalitu podpory procesu. Softvér ako služba organizáciám umožňuje prístup k technológiám a nástrojom, na ktoré by v tradičnom modeli dodávky nemali dostatok prostriedkov. Vďaka rozloženiu nákladov na strane dodávateľa, ktorý je zároveň nositeľom rizika, môžu organizácie v rámci poplatku za užívanie zaplatiť ako prevádzku týchto inak nákladných aplikácií, tak aj ich ďalší vývoj.

Dávno však neplatí pravidlo, že aplikácie v modeli softvér ako služba nakupujú len malé a stredné podniky, ale už aj veľa nadnárodných spoločností. Veľké spoločnosti využívajú aplikácie v modeli softvér ako služba pre nákup riešenia, ktorá používa len určitá skupina zamestnancov a neoplatilo by sa pre nich v danej kvalite implementovať robustné riešenie. Vhodné sú tiež riešenia, kde je proces vo všetkých jednotkách organizácie štandardizovaný na rovnakej úrovni a vykonávaný identickým spôsobom. Na druhej strane je pravdou, že model softvér ako služba čiastočne zväzuje organizácie do uniformných riešení a nie je tu priestor pre kustomizácie, čo predovšetkým pre veľké spoločnosti môže byť nevýhodou.

Ale predovšetkým tradiční dodávatelia softvéru ponúkajú riešenie modelu softvér ako služba postavené na single Tenant technológii, ktorá umožňuje malé úpravy aplikácií podľa požiadaviek zákazníkov. Tieto kustomizácie ale nikdy nie sú porovnateľné s kustomizáciou aplikácií dodaných v tradičnom modeli. Ak nakupujeme službu, je Service level agreement (ďalej len SLA) kritickým bodom dodávky. Dodávateľom napomáha SLA nastaviť realistické očakávania zákazníkov (http://www.wallstreetandtech.com). Mnoho nových projektov softvér ako služba nenaplnia očakávania zákazníkov preto, že predstavy zákazníkov sú čiastočne nerealistické. SaaS zákazníka nezbaví zodpovednosti za proces a za jeho vykonávanie, dodá mu iba nástroje na podporu jeho procesu.

Bezpečnosť a ochrana dát nepatrí medzi dôvody, ktoré zákazníka presviedčajú o vhodnosti použitia modelu softvér ako služba, práve naopak, pri rozhodovaní sa často objavujú v kolónkach v neprospech modelu. Ešte z doby pred zavedením informačných

technológií do podnikateľského prostredia sú organizácie zvyknuté mať všetky dokumenty, teda svoje duševné vlastníctvo, uložené v niektorej zo svojich budov. Koncept umiestnenia dát mimo fyzický dosah organizácií je proti ich zvyklostiam a odtiaľ pramení strach zo zneužitia. Ak porovnáme kvalitu serverovne malých organizácií s priestormi dátových centier, kde je fyzicky umiestnený hardvér poskytovateľov aplikácií v modeli softvér ako služba, kvalita fyzického zabezpečenia jasne hovorí v prospech poskytovateľov. Zabezpečenie aplikácií v tradičnom modeli je pre organizácie len podporný proces a tiež v porovnaní so zabezpečením aplikácií v modeli SaaS svoj boj často prehrávajú. Pre poskytovateľov softvéru ako služba je totiž bezpečnosť, či už fyzická alebo psychická, jedným z kľúčových procesov. Umiestniť dáta mimo firmu a zamedziť prístupu k nim databázovým alebo doménovým administrátorom je pre mnohé organizácie a pre niektorý druh dát veľmi zaujímavé. Zastávame názor, že ak sa podarí dodávateľom vytvoriť nový koncept business kontinuity plánu a nanovo definovať SLA podľa špecifických vlastností softvéru ako služba, potom môžu prilákať ďalších zákazníkov, ktorí doteraz z dôvodu obáv o svoje dáta tento model odmietali.

Získanie audítorských certifikátov, napr. SAS70, môže tiež zvýšiť dôveryhodnosť poskytovateľa. Zo svojej podstaty je model vhodný na on-line sledovanie kvality dodávky služieb, ale bohužiaľ táto oblasť je zatiaľ nielen na Slovensku jednou z najmenej rozvinutých. Jedine tlak zákazníkov na dodávateľa môže tento stav zmeniť.

Zásadné odlišnosti SLA tradičného modelu od SLA modelu softvér ako služba sú definície postupu pri ukončení a teda vypovedaní zmluvy o poskytovaní, kvalita a formát dát, spôsoby ich exportu zo systémov pri ukončení zmluvy. Už pri plánovaní nákupu služby musí organizácia počítať so skutočnosťou, že služba môže kedykoľvek skončiť neúspechom a musí mať pripravenú pre túto situáciu stratégiu náhradného riešenia. Vzhľadom k faktu, že dáta sú uložené mimo jej sídlo, organizácia musí byť pripravená na situáciu, keď aplikácia, na ktorej bola doteraz závislá, nebude naďalej v prevádzke bez predchádzajúceho upozornenia. Organizácia by mala mať pripravené alternatívne - náhradné riešenie, najlepšie opäť formou softvéru ako služba. Zároveň by mala mať vo svojom pláne obnovy činností definované postupy náhrady aplikácie, aké je k tomu potrebné vykonať kroky a hlavne spôsob prevodu doposiaľ vytvorených dát k novému poskytovateľovi, poprípade do iného riešenia. Súčasťou stratégie ukončenia služby by malo byť určenie vlastníctva a ohodnotenie dát. Starý a budúci systém pravdepodobne nebudú úplne totožné, organizácie by preto mali mať stanovené, ktoré dáta sú pre nich

kľúčové, ktoré chcú zachovať a prípadne naďalej používať v novej aplikácii.

Americký softvérový gigant **Microsoft** plánuje zaviesť novú softvérovú službu pre firemných zákazníkov za mesačné predplatné namiesto licenčných poplatkov (http://www.itnews.sk). Firma oznámila, že organizáciám s určitým počtom zamestnancov začne ponúkať možnosť, že im bude určitý softvér ako službu posielať po internete. Tí, ktorí túto možnosť odmietnu, budú ďalej platiť za licencie ako doteraz. Ide o prvý pokus firmy dodávať programy touto cestou. Microsoft začne ponúkať internetovou cestou programy pre elektronickú poštu, zasielanie odkazov a ďalší softvér pre spoluprácu organizácií. Nová stratégia predstavuje odklon od súčasného modelu, keď Microsoft predáva licencie k programom, ktoré bežia lokálne v počítačoch patriacich zákazníkom. Americký gigant k tomuto kroku prinútila konkurencia spoločností **Salesforce.com** a **Google**.

Ako príklad novej stratégie Microsoft uviedol Office Live Workspace, ktorý umožňuje zamestnancom deliť sa cez internet zadarmo o dokumenty vo Worde, programy v Exceli a prezentácie v PowerPointe. Vo Workspace môže užívateľ prizvať ostatných, aby sa na dokumenty pozreli a vyjadrili sa k nim, aj keď nemajú Microsoft Office. Užívatelia si na Workspace môžu uložiť vyše 1 000 dokumentov Office a dostať sa k nim cez akýkoľvek internetový prehliadač. Do mamutích dátových centier, plných počítačov a úložných médií, investoval Microsoft miliardy dolárov, aby mohol ponúkať služby bežným zákazníkom aj väčším organizáciám.

V súčasnosti už väčšina vlastníkov malých firiem s dobrým prehľadom o internete minimálne počula o spoločnosti Salesforce.com (Buchalcevová - Kučera, 2008). Uvedená korporácia bola založená v roku 1999 a stala sa vďaka poskytovaniu svojho systému CRM umiestneného v cloude priekopníkom. Produkt bol určený pre malé a stredne veľké firmy a dnes je táto spoločnosť lídrom v oblasti cloudových systémov CRM. Pretože existuje mnoho ďalších cloudových služieb od iných dodávateľov, ktoré umožňujú integráciu s riešením Salesforce.com, majú používatelia k dispozícii viac možností a funkcií. Salesforce.com navyše ponúka nástroje pre firmy všetkých veľkostí. Napríklad služba Salesforce Contact Manager stojí len päť dolárov mesačne za používateľa.

Mnohým sa môže teda zdať, že cloud computing je určený len pre väčšie firmy, ktoré si to môžu dovoliť. Tieto služby však prešli za posledné obdobie obrovským skokom a dodávatelia neustále rozširujú svoju ponuku. Dnes kombinácie služieb IaaS, PaaS a SaaS

poskytujú ucelený súbor možností pre každý typ spoločnosti (International Software testing Qualification Board, 2007). V nasledujúcich riadkoch sú uvedené výhody využitia aktuálne dostupných služieb cloud computingu pre začínajúcu, zabehnutú aj expandujúcu firmu. Do samostatnej kategórie je vyčlenená IT firma, ktorá má v cloudových službách zaujímavý potenciál na úsporu nákladov i priestor na rozšírenie a zvýšenie kvality služieb svojim zákazníkom.

Asi najstaršia služba, ktorú môžeme kategorizovať ako SaaS, je webmail a taktiež rozšírený webhosting, ktorý dal základ službám typu PaaS a IaaS. Potrebu elektronickej komunikácie a prezentácie rieši dnes každá firma, sú to základné kroky v jej začiatkoch. Po ére freemailových schránok, ako napríklad Gmail, Zoznam či Centrum, dnes môže mať každý firemný e-mail, kontakty, kalendár i úlohy s vlastnou doménou a so všetkými benefitmi vrátane zdieľania spoločných údajov bez toho, aby si zakúpil celý systém aj so serverom. Stačí, aby si aktivoval jednotlivé e-mailové účty ako službu SaaS. A ak si chce elektronickú komunikáciu spravovať vo vlastnej aplikácii a vo vlastnej réžii, jednoducho si prenajme virtuálny server podľa potrieb ako službu IaaS.

Druhú podstatnú časť fungovania firmy tvoria firemné aplikácie a zdieľanie údajov. Aj v tejto oblasti majú cloudové služby čo ponúknuť – či už si vyberie firma len server podľa vlastných parametrov a bude si ho vo vlastnej réžii spravovať na báze IaaS, alebo rovno server vrátane operačného systému a ďalších súvisiacich softvérov vo forme PaaS. Ak bude prevádzkovať niektorý z ekonomických IS, napríklad ERP, systém na riadenie vzťahov so zákazníkmi - CRM, alebo iné potrebné aplikácie, a či bude len zdieľať dokumenty medzi zamestnancami, má tento spôsob fungovania veľa predností a je nielen ekonomicky výhodný. Server podľa požadovanej konfigurácie je spravidla aktivovaný v priebehu minút, vo výnimočných alebo zložitejších prípadoch do niekoľkých hodín bez potreby čakania na dodávku. Cloudová infraštruktúra je prevádzkovaná v zabezpečených dátových centrách so zálohovaním v prípade neočakávaných udalostí, prístup je chránený tak pred fyzickým, ako aj pred elektronickým vniknutím.

V prípade, že firma využije služby SaaS alebo PaaS, má k dispozícii posledné verzie aplikácií a dodávatelia garantujú, že sa k vašim údajom či aplikáciám vždy dostanete. Všetky náklady na energie, dátové pripojenie, klimatizáciu v serverovni, údržbu, zabezpečenie, náhradné hardvérové diely a servis znášajú oni. Takisto v prípade služieb SaaS ako je uvedený e-mailový systém či akýkoľvek ponúkaný IS sa aktivácia konta realizuje rádovo do niekoľkých minút. Takýmto spôsobom na štart firmy potrebujeme len

akýkoľvek notebook, alebo jednoduché PC, pripojenie k internetu, a môžeme tak spolu s ďalšími kolegami pracovať úplne kdekoľvek. A ako bude naša organizácia rásť, stačí len upravovať konfiguráciu serverov, pridávať používateľov či aktivovať ďalšie potrebné služby. A tak sa môžeme sústrediť na svoj „biznis“ a nie na IT. A ak máme tieto služby od jedného dodávateľa, máme len jedného partnera, ktorý garantuje funkčnosť a nevyhovára sa na iných, ďalších dodávateľov, že je to prípadný problém hardvéru, sieťového pripojenia alebo konkrétnej aplikácie.

Zabehnutá firma, ale aj dlhodobo fungujúca organizácia, má priestor na ďalšie rozširovanie alebo zmeny fungovania. V praxi neexistuje organizácia, ktorá by mala všetko vyriešené a fungujúce podľa požiadaviek či posledných trendov v oblasti IT. Samotný vývoj ekonomiky a hospodárska kríza posledných rokov ukázali jasnú potrebu flexibilnejšieho prispôsobovania sa organizácií situácii v oblasti nákladov a fungovania. Cloud computing má kľúčovú výhodu práve vtedy, keď organizácia má kolísajúci počet zamestnancov – či už napríklad zo sezónnych, alebo projektových dôvodov. Jednoducho jeden mesiac platí organizácia za 15 používateľov, nasledujúci napríklad aj za 60 a ďalší mesiac opäť len za 15.

Keďže väčšina organizácií buduje infraštruktúru postupne, ako sa zvyšujú požiadavky, aj končiaca sa životnosť hardvéru je takisto dôvod na presun fungovania do cloudového prostredia. Pritom to nemusí byť naraz, organizácia môže napríklad fungovať s časťou služieb na vlastných serveroch a časťou v cloudovom prostredí takým spôsobom, že sú navzájom prepojené a jednotlivý používatelia ani nerozoznajú, ktorá časť sa kde nachádza. Veľkou výhodou je, že organizácie, ktoré majú aj svoje ďalšie pobočky mimo svojho sídla, alebo ak chcú sprístupňovať časti systémov nejakým partnerom cez internet alebo mobil, majú cloudové služby pomocou neustáleho zabezpečeného prístupu k firemným aplikáciám.

Vybudovať vlastné a pritom skutočne bezpečné pripojenie z internetu do firmy vyžaduje nemalé náklady, skúsenosti a hlavne odbornosť zodpovedných pracovníkov. V každej organizácii prebieha neustály proces hľadania, skúmania a nasadzovania nových služieb a aplikácií. Ak sa interné IT oddelenie práve podieľa na testovaní nového IS, alebo len jeho nejakej rozšírenej funkčnosti, môže si na toto krátke obdobie prenajať server alebo platformu, na ktorej prebehne skúška bez potreby kupovania fyzického servera a ohrozenia fungovania svojej „ostrej“ inštalácie. A ak bude aplikácia či celé prostredie vyhovovať, môže byť okamžite sprístupnená pre všetkých používateľov v cloudovom prostredí alebo

prenesená na nový, zakúpený fyzický server.

Cloudové služby dnes ponúkajú nielen možnosti, ktoré v minulosti patrili výhradne fyzickým zariadeniam, ale aj jednoduchý spôsob, ako využiť ďalšie doplnkové služby. Časť dodávateľov ponúka pravidelné zálohovanie alebo komplexnú replikáciu celých serverov, vybudovanie záložných štruktúr a pracovísk, virtuálne telefónne ale aj videotelefónne ústredne a ďalšie. Bezpečnosť firemných údajov sa v cloudovom prostredí aj ľahšie riadi. Zamestnanec nenosí domov údaje na kľúčoch USB, možno zakázať používanie USB a DVD vo firme, a hlavne v ukradnutých a stratených notebookoch a smartfónoch nie sú žiadne firemné údaje. Potrebné dáta má zamestnanec kedykoľvek a odkiaľkoľvek dostupné zabezpečeným pripojením vo firemných serveroch, navyše sú aj vždy aktuálne.

Veľký potenciál prináša cloud computing samotným IT firmám a to tak pri fungovaní, ako aj pri ponuke služieb zákazníkom. Dokážu doslova na počkanie získať kedykoľvek platformu na plánovaný vývoj svojich aplikácií alebo na testovanie úprav pre konkrétneho zákazníka. Vo fáze získavania zákazníka si bez námahy môžu pripraviť, predviesť a zadefinovať čas na odskúšanie demoaplikácie, a to bez potreby inštalácie na zákazníkov počítač. Po podpise zmluvy môže pristúpiť okamžite k realizácii bez potreby čakania na fyzický server a hlavne môže využiť cloudové prostredie dodávateľa na poskytovanie svojich aplikácií formou SaaS a tým získať ďalšiu skupinu zákazníkov, ktorá by si napríklad aplikáciu v štandardných podmienkach nemohla dovoliť.

Cloud computing dnes skutočne prináša výhody a vhodné služby pre každý typ firmy či organizácie. K tomu získava každá organizácia aj množstvo doplňujúcich služieb s potenciálom ďalšieho rastu presne podľa aktuálnych potrieb.

## 1.6 Teória hodnotenia zo zavádzania komunikačných systémov

Aby sme mohli porovnať, ako jednotlivé metodiky pristupujú k hodnoteniu zavádzania komunikačných systémov, musíme zadefinovať nasledujúce sady kritérií (http://objekty.vse.cz):

- **zameranie testovania,**

- **proces testovania,**
- **úloha,**
- **úrovne testovania,**
- **trasovateľnosť,**
- **životný cyklus chyby,**
- **typy testov a testovacie techniky,**
- **artefakty testovania,**
- **použitie nástrojov a automatizácie.**

***Zameranie testovania*** - Činnosti spojené s testovaním vykonávajú väčšinou špecializované testovacie roly, ale môžu sa na nich podieľať aj ďalšie roly. Z tohto pohľadu rozlišujeme testovanie závislé a nezávislé. *Závislé* testovanie realizujú úlohy, ktoré majú inú špecializáciu ako testovanie. Príkladom môže byť vývojárske testovanie, kedy napríklad programátori vykonávajú jednotkové testovanie. *Nezávislé* testovanie vykonávajú úlohy, ktoré testujú výsledok práce iných rolí. Jedná sa najmä o úlohy, ktorých špecializáciou je testovanie, ale môže ísť aj o zástupcu zákazníka vykonávajúce akceptačné alebo beta testovania.

***Proces testovania*** - Procesom testovania rozumieme činnosti súvisiace s testovaním, ich sled a vzájomné vzťahy. Toto kritérium hodnotí fázy životného cyklu projektu, do ktorých je testovanie zapojené, a prístup k tvorbe testov (tvorba testov súbežne s implementáciou či Test-first Design).

***Úloha*** - V rámci tohto kritéria sa hodnotia úlohy, ktoré sa na testovaní podieľajú. Ide jednak o úlohy špecializované na testovanie a jednak úlohy, ktoré sa na testovaní len podieľajú.

***Úrovne testovania*** - Úroveň testovania je definovaná ako sada testov, ktorými je softvérový produkt testovaný na zadefinovanej úrovni podrobnosti. Rozoznávame *jednotkové* testovanie, ktoré testuje najmenšie časti systému, *integračné* testovanie, ktoré sa zameriava na testovanie väzieb a prepojení častí systému, *systémové* testovanie, teda testovanie systému ako celku, a *akceptačné* testovanie, ktorého cieľom je preukázať splnenie definovaných akceptačných kritérií. V rámci tohto kritéria sa zisťuje, ktoré úrovne testovania metodika pokrýva, aké roly sa podieľajú na testovaní na danej úrovni a aká je intenzita testovania.

***Trasovateľnosť*** - Trasovateľnosť umožňuje identifikovať vzájomné vzťahy medzi položkami dokumentácie a softvéru. Toto kritérium obmedzuje trasovateľnosť na vzťah medzi požiadavkami a testami. Ak testy vznikajú v nadväznosti na definované požiadavky, potom je metodika hodnotená tak, že trasovateľnosť zabezpečuje.

***Životný cyklus chyby*** - Životný cyklus chyby začína jej odhalením a končí jej vyriešením, odstránením či zdokumentovaním postupu, ako sa vyhnúť zlyhaniu. Toto kritérium hodnotí, či v metodike existuje definovaný životný cyklus chyby a akým spôsobom sa rozhoduje o riešení nájdených chýb.

***Typy testov a testovacie techniky*** - Testy kategorizujeme spravidla podľa oblastí kvalitatívnych ukazovateľov, ktoré testujú. Rozoznávame testy funkčnosti, použiteľnosti, výkonu, spoľahlivosti a podpory. Medzi testy *funkčnosti* zahŕňame testy overujúce správnu implementáciu požadovaných funkcií systému, alebo bezpečnosť systému. Testy *použiteľnosti* sa zameriavajú na užívateľské rozhranie a jeho prívetivosť, ďalej na oblasť Pomocníka (HELP), dokumentácie a kvalitu výukových materiálov. *Výkonnostné* testy overujú správny a stabilný priebeh funkcií systému v situáciách, kedy systém musí čeliť veľkej záťaži. Testy *spoľahlivosti* testujú schopnosti systému zvládať neštandardné správanie, nesprávne vstupy alebo možnosti obnovy po havárii systému. Testy *podpory* pokrývajú oblasti ako je možnosť nasadenia na rôznych platformách, možnosti údržby a rozšírenia systému, alebo možnosti konfigurácie systému. Zvláštnym typom testu je *smoke test,* kde sa jedná o realizáciu vybranej podmnožiny vytvorených alebo plánovaných testov s cieľom overiť správnu funkčnosť všetkých kritických častí systému. Medzi testovacie techniky patrí napríklad *prieskumné* testovanie, alebo *beta* testovanie.

***Artefakty testovania*** - Súčasťou tohto kritéria je zhodnotenie, aké dokumenty v porovnaní so štandardom IEEE 829 metodika pokrýva. Štandard IEEE 829 definujú nasledovné dokumenty (Jayant, 2009):

- *Test Plan* - dokument opisujúci, čo bude testované, aké k tomu budú použité testy, techniky a nástroje, akým spôsobom bude testovanie prebiehať, aké sú požiadavky na kvalitu a aký je harmonogram testovania.
- *Test Design Specification* - dokument, ktorý na základe ďalších podkladov, ako je napríklad návrh systému alebo špecifikácia produktu, sumarizuje a bližšie vymedzuje čo má byť otestované a ako bude posudzovaná úspešnosť jednotlivých testov.
- *Test Case Specification* - opis testovacieho prípadu zahŕňajúci vstupné a výstupné dáta,

vstupné a výstupné hodnoty a jednotlivé kroky testovacieho prípadu.

- *Test Procedure Specification* - dokument opisujúci jednotlivé kroky pri implementácii testovacieho prípadu.
- *Test Item Transmittal Report* - dokument, ktorý sprevádza prenos novej verzie produktu na testovanie. Obsahuje popis zmien oproti minulej verzii a je tu definované, čo má byť otestované v súčasnej verzii.
- *Test Log* - dokument, v ktorom je zachytený priebeh testovacích prípadov a ich výsledkov.
- *Test Incident Report* - dokument, ktorý obsahuje opísané pozorované nezhody, kroky a okolnosti, za ktorých k nim došlo, a ďalšie doplňujúce informácie.
- *Test Summary Report* - súhrnná správa o priebehu celého testovania a zistené úrovne kvality testovaného systému.

***Použitie nástrojov a automatizácie*** - Automatizáciou testovania rozumieme použitie softvéru na podporu alebo vykonávanie činností spojených s testovaním, ako je riadenie testovania, vykonávanie a vyhodnocovanie testovacích prípadov a scenárov. Z tohto hľadiska sa u metodiky hodnotí, či existuje definovaný prístup k automatizácii testovania a aké oblasti testovania metodika odporúča automatizovať. Praktiky niektorých metodík priamo vyžadujú, aby testovanie bolo automatizované, inak je ich realizácia zložitá.

**Extrémne programovanie (XP)** - je metodika určená najmä pre malé až stredne veľké tímy (2 - 10 programátorov), ktoré vyvíjajú softvér, ktorého zadanie nie je jasné alebo sa mení. Autormi metodiky sú Kent Beck, Ward Cunningham a Ron Jeffries (http://objekty.vse.cz).

***Zameranie testovania*** - XP sa zaoberá najmä závislým testovaním, ktoré je reprezentované testovaním programátorov. Do kategórie nezávislého testovania patrí akceptačné testovanie, ktoré je v tejto metodike primárnou zodpovednosťou samotných užívateľov, ktorí by sami mali navrhovať akceptačné testy.

***Proces testovania*** - Testovanie je súčasťou všetkých fáz životného cyklu projektu. Testeri sa podieľajú na odhade prácnosti definovaných používateľských „príbehov“. V rámci fáz analýzy, návrhu a implementácie prebieha testovanie priebežne a veľmi často.

Vo fáze implementácie sú vykonané akceptačné testy. Metodika si vyžaduje prístup Test-first Design, testy pre každú požiadavku musia byť vytvorené pred tým, než sú implementované. Týmto spôsobom sú tvorené ako unit testy, tak aj testy funkčnosti. Pre testovanie programátorov je prístup TFD podrobne popísaný. Každá možná cesta programom by mala mať v ideálnej prípade svoj test. Programátor by mal postupovať tak, že implementuje jednoduchý test, ktorý musí zlyhať, pretože ešte neexistuje kód programu, ktorý je testovaný. Potom je implementovaný čo najjednoduchšie kód programu, ktorý úspešne prejde testom. Nasledujúcim krokom je refaktorizácia, ktorej cieľom je čitateľný a znovu použiteľný program. Po každom kroku refaktorizácie je test znovu spustený.

***Roly*** - Testovanie a kvalita je podľa XP zodpovednosťou každého člena tímu, teda aj všetkých rolí. Avšak hlavnými rolami, ktorých sa testovanie priamo týka, sú roly zákazník, vývojár a tester. Zákazníci by mali tvoriť testy funkčnosti na overenie nimi definovaných používateľských príbehov. Vývojári tvoria, realizujú a vyhodnocujú unit testy. XP definuje len úlohu tester, nerozširuje ju do ďalších špecializovaných rolí. Testeri plnia úlohu poradcov, ktorí oboma vyššie spomenutým rolám pomáhajú so zostavovaním testov, ich automatizáciou a tvorbou testovacieho prostredia. Pomáhajú s odhadom rizík, preverujú užívateľské príbehy, napomáhajú definovať požiadavky na kvalitu. Dohliadajú aj na to, že testy funkčnosti skutočne overujú definované užívateľské príbehy.

***Úrovne testovania*** - Metodika XP špecifikuje dve skupiny testov: unit testy a testy funkčnosti.

Unit testy slúžia na testovanie jednotiek. Pri ich vytváraní je potrebné brať do úvahy aj interakciu jednotky s okolím. Unit testy definované podľa metodiky XP plnia úlohu jednotkových testov a integračných testov podľa úrovní testovania. Tieto testy sú prevažne tvorené a spúšťané vývojármi. Metodika odporúča vykonávať toto testovanie veľmi často, pokiaľ možno neustále.

Testy funkčnosti slúžia na overenie splnenia používateľského príbehu. Testovaný je systém od vstupu po výstup. V rámci iterácie by sada testov funkčnosti alebo jej podmnožina mala byť spustená aspoň raz denne, tu tieto testy plnia úlohu systémových testov. Na konci iterácie pri odovzdávaní produktu zákazníkovi slúžia tieto testy na preukázanie, že všetok dohodnutý rozsah bol implementovaný. Plní teda úlohu akceptačných testov. Tvorba testov funkčnosti je spoločnou zodpovednosťou zákazníkov a testera. Tester potom vykonáva vytvorené testy a vyhodnocuje ich priebeh. Ku každému

užívateľskému príbehu musí existovať jeden alebo viac testov funkčnosti na overenie jeho splnenia.

Testy vznikajú ešte pred napísaním programového kódu. Vývojár pri rozhovore so zákazníkom získa podrobnosti k jednotlivým užívateľským príbehom a rozhodne o spôsobe ich realizácie. Potom nasleduje tvorba príslušných unit testov. Väzby medzi požiadavkami a testami vznikajú v priebehu vývoja a ich vznik je daný navrhnutým procesom. Tvorba dokumentu, ktorý by trasovateľnosť zachytil, ale nie je spomínaná. Nakladanie s odhalenou chybou je popísané iba v súvislosti s tvorbou testov. Po objavení chyby by mal byť vytvorený test, ktorý overí jej odstránenie. Potom by chyba mala byť odstránená.

Spôsob rozhodovania o odstraňovaní chýb je iba načrtnutý v súvislosti s výhodami častého testovania. Jednou z výhod častého testovania je, že chyby sú objavené skoro po tom, čo je dokončená nová časť aplikácie alebo je vykonaná zmena. Pre programátorov sa jedná o čerstvú záležitosť, lokalizovať a odstrániť chybu by teda malo byť jednoduchšie.

XP ako agilná metodika sa príliš nezameriava na dokumentáciu. Dôsledkom toho je, že nedefinuje žiadne povinné artefakty testovania okrem samotných testov a ich výsledkov. Dokumentácia testovania nezodpovedá štandardu IEEE 829. Veľmi krátke iterácie, neustále spúšťanie jednotkových testov a integrácie, časté spúšťanie sady testov funkcionality sú faktory, ktoré vedú k nutnosti testovanie automatizovať.

Extrémne programovanie nie je priamo spojené so žiadnym robustným komerčným testovačom či vývojovým nástrojom, často sa ale používajú nástroje voľne distribuované, medzi nimi možno nájsť dva, na ktorých tvorbe sa podieľali autori tejto metodiky. Ide o nástroje xUnit a Fit. xUnit je rodina testovacích nástrojov pre vykonávanie unit testov tak, ako ich definuje Extrémne programovanie. X v názve nástroja značí programovací jazyk alebo platformu, pre ktorú je určená daná implementácia tohto prístupu k testovaniu - Sunita pre Smalltalk, JUnit pre jazyk Java alebo CUnit pre jazyk C. Fit je jednoduchý nástroj na podporu systémových alebo akceptačných testov, ktorého autorom je Ward Cunningham (http://www.ibm.com). Existuje v niekoľkých verziách pre rôzne vývojové platformy. K tomuto nástroju existujú tiež nadstavby, ktoré rozširujú jeho možnosti a uľahčujú jeho používanie. Jedným z týchto rozšírení je Fitness. Ide o implementáciu webového servera, kam môžu členovia tímu pridávať nové stránky s testami a vytvárať tak sady testov a testovacie scenáre.

**Rational Unified Process** - teda presne Metodika Rational Unified Process **(RUP)** je rámec pre definovanie metodiky pre celé spektrum projektov od malých až po veľmi veľké. Predtým bola radená medzi rigorózne metodiky, ale v súčasnej dobe zahŕňa agilné princípy a praktiky a je možné ju prispôsobiť aj agilnému vývoju (http://technet.microsoft.com). Metodika RUP je šírená na komerčnej báze a je podporovaná nástrojmi IBM Rational.

Táto metodika sa zaoberá závislým aj nezávislým testovaním. Závislé testovanie vykonávajú vývojári. Nezávislé testovanie má na starosti testovací tím alebo v prípade beta testovania samotní užívatelia. Pre závislé i nezávislé testovanie sú spracované koncepty, ktoré popisujú všeobecné odporúčania a upozorňujú na možné problémy. Testovanie prechádza všetkými fázami životného cyklu projektu, pričom množstvo práce a prostriedkov venovaných testovaniu spravidla postupne narastá, maximum je dosiahnuté vo fáze konštrukcie.

Celý proces testovania je rozdelený do niekoľkých krokov, z ktorých niektoré prebiehajú paralelne a opakujú sa v každej iterácii. Prvým krokom testovacieho cyklu je určenie cieľov testovania. Nasleduje overenie, že aktuálna zostava je stabilná. Ak nie je, o čom svedčí neprimeraný výskyt chýb, hlavne tých veľmi závažných, je táto zostava vrátená späť do vývoja. Ak je zostava stabilná, je podrobená testovaniu a ohodnoteniu. Paralelne s touto činnosťou prebieha príprava správy o priebehu a výsledkoch testovania určená manažérom projektu a ďalším zainteresovaným osobám.

Po dokončení testovania a vytvorení výkazov nasleduje proces údržby a skvalitnenie testovacej dokumentácie, testovacích dát, vytvorených testov a ďalších náležitostí testovania - v metodike súhrnne označených ako Test Assets. Cieľom je udržiavať náležitosti aktuálne a tvoriť z nich základňu, ktorej prvky možno znovu použiť.

Činnosti od overenia stability zostavy po skvalitnenie náležitostí testovania môžu v rámci iterácie prebehnúť viackrát, jedná sa o tzv. testovací cyklus. Paralelne s testovacím cyklom beží proces, ktorého úlohou je overovať, že zvolený prístup k testovaniu je vhodný, testy sú kvalitné a podávajú relevantné výsledky. Metodika odporúča užiť prístup TFD, je ale potrebné zvážiť, kedy je vhodný čas testy navrhnúť. Nie je účelom navrhnúť všetky testy čo najskôr, je potrebné postupovať v súlade s cieľmi iterácie a s fázami projektu.

V disciplíne testovania vymedzuje metodika RUP štyri roly:

- Manažér testovania,
- Test analytik,
- Návrhár testov,
- Tester.

Pre každú úlohu metodika vymedzuje činnosti, ktoré vykonáva a artefakty, s ktorými pracuje. Ďalej tiež uvádza odporúčania pre personálne obsadenie jednotlivých rolí.

***Manažér testovania*** zastrešuje prácu celého testovacieho tímu. Jeho úlohou je riadiť proces testovania a riešiť problémy s ním spojené. Je zodpovedný za plánovanie zdrojov a mal by v celom riešiteľskom tíme vystupovať ako advokát kvality.

Úlohou ***Test analytika*** je identifikovať a popísať potrebné testy, sledovať priebeh procesu testovania a hodnotiť dosiahnutú úroveň kvality zistenú počas testovania. Je tiež zodpovedný za vhodnú voľbu reprezentácie výsledkov testovania.

***Návrhár testov*** je zodpovedný za definíciu prístupu k testovaniu a jeho úspešnú realizáciu, čo zahŕňa identifikáciu vhodných postupov, nástrojov a odporúčaní pre vykonávanie testov a pre odhad zdrojovej hospodárnosti testovania.

***Tester*** je zodpovedný za tvorbu testov, ich implementáciu, zaznamenanie priebehu a ich vyhodnotenie.

Testovanie nie je zodpovednosťou len týchto rolí, ale prebieha aj v rámci vývojárskeho tímu. Vývojárske testy sú zodpovednosťou roly autor alebo implementátorov. Za integráciu systému a implementácie integračných testov je zodpovedná úloha integrátora. Vo fáze odovzdávania riadi realizáciu akceptačného a beta testovania manažér nasadenia - ***Deployment Manager***. V tejto fáze sú do testovania zapojení aj zástupcovia zákazníka. U akceptačných testov rozhodujú o prijatí či neprijatí vyvinutého produktu. Počas beta testovania systém testujú priamo.

Metodika RUP pokrýva všetky úrovne testovania. Jednotkové a integračné testy by mali vykonávať vývojári, ktorí sa tiež môžu podieľať na systémových testoch. Testovací tím by mal vykonávať hlavne testy systémové a akceptačné. Jednotkové testy sa vykonávajú podľa potreby, typicky niekoľkokrát počas iterácie.

Integračné testy sú rozdelené na testy podsystémov a testy systému ako celku.

Testy podsystémov by mali prebehnúť aspoň raz počas iterácie, testy celého systému sú viazané na tvorbu zostavy. Pre tvorbu zostáv metodika neurčuje žiadne špecifické načasovanie. Systémové testy by mali prebehnúť niekoľkokrát počas iterácie, akceptačné testy sú potom vykonávané v záverečnej fáze projektu, kedy dochádza k odovzdávaniu hotového riešenia zákazníkovi.

Testovacie prípady a scenáre sú v metodike RUP odvodené od prípadov použitia systému. Model prípadov použitia slúži na zachytenie požiadaviek na systém a je odsúhlasený zákazníkom. Definícia konkrétneho prístupu k zabezpečeniu trasovateľnosti je súčasťou testovacieho plánu. Popis testovacích prípadov a scenárov je doplnený o odkazy na požiadavky, ktoré sú nimi testované. Trasovateľnosť je v tejto metodike plne zaistená a aj zdokumentovaná.

Riešenie nájdených chýb je predmetom zmenového konania, kedy je požiadavka na odstránenie chyby jedným z typov požiadaviek na zmenu. Životný cyklus zmenovej požiadavky je popísaný. Chyba je označená prioritou a sú odhadnuté jej dopady. V rámci zmenového riadenia je rozhodnuté o spôsobe jej riešenia, priradenie príslušných kapacít a spôsobe, akým bude odstránenie chyby overené.

Ak uvažujeme nad typmi testov a testovacej techniky, tak metodika RUP sa zmieňuje o použití smoke testu a beta testovania. Smoke test slúži na overenie stability aktuálnej zostavy produktu. Beta testovanie je doporučované vykonať vo fáze zavádzania, keď je produkt nasadený do skúšobnej prevádzky. Zistené problémy tak môžu byť odstránené alebo aspoň zdokumentované pred tým, ako je produkt nasadený do bežnej prevádzky.

Typy testov sú popísané a u niektorých z nich sú spracované odporúčania, ako ich vykonávať, kedy pristúpiť k ich automatizácii a ktoré nástroje je na to vhodné využiť. Ich voľba závisí na konkrétnom projekte. Metodika RUP definuje celú radu artefaktov testovania. U každého vymedzuje, jeho účel, náplň a zodpovednú úlohu. Najvýznamnejšie artefakty testovania sú Test Plan, Test Evaluation Summary (ekvivalent dokumentu Test Summary Report), Test Script (zodpovedá dokumentu skúšobný postup Specification z IEEE 829), Test Log, Test Case (zodpovedá dokumentu Test Case Specification z IEEE 829), Test Results, Test Strategy (ekvivalentné k dokumentu Test Design Specification). Nie sú pokryté dokumenty Test Item Transmittal Report a Test Incident Report (http://objekty.vse.cz). Metodika uchováva a spravuje výsledky testov, samotnú správu

o nezhodách však nevypracováva.

Pre automatizáciu testovania je v metodike RUP obsiahnutý všeobecný koncept, ktorý opisuje výhody a nevýhody automatizácie a možné problémy a riziká jeho použitia. Metodika zdôrazňuje potrebu ľudského úsudku pri testovaní. Hlavnou oblasťou pre využitie automatizácie testovania sú vývojárske testy, hlavne testov jednotiek. Ďalšie odporúčania pre využitie automatizácie možno nájsť u popise typov testov. Automatizácia je teda hodnotená ako odporúčaná.

Nástroje IBM Rational tvoria široké portfólio, ktoré svojim rozsahom pokrýva celý cyklus vývoja softvéru. Metodika RUP obsahuje poradcu pre prácu s nástrojmi (Tool Mentors), ktoré popisujú použitie jednotlivých nástrojov pri konkrétnych činnostiach opísaných v RUP.

- ***Nástroj IBM Rational PurifyPlus*** pre analýzu chodu aplikácií - skôr ako samotní testeri využijú ho vývojári pri riešení problémov spojených so spoľahlivosťou a výkonom aplikácií.
- ***IBM Rational ClearQuest*** je komplexný nástroj pre podporu, kontrolu a automatizáciu vývojového cyklu. Umožňuje definíciu a automatizáciu pracovného toku (workflow), tvorbu hlásenia, kalkuláciu metrík, automatizáciu tvorby zostavy aplikácie a jej nasadenie. Tento nástroj slúži tiež ako nástroj pre riadenie testovania - jeho plánovanie, prípravu, realizáciu a oznamovanie výsledkov.
- ***IBM Rational Manual Tester*** je nástroj na podporu manuálneho testovania (http://www.ibm.com). Umožňuje vytvárať a spravovať testovacie prípady a scenáre pre manuálne testy.
- ***Nástroj IBM Rational Functional Tester*** je určený pre automatizáciu testovania funkčnosti. Testovacie scripty pre automatizované vykonávanie testov je možné vytvoriť záznamom interakcie s testovaným systémom alebo ho vytvárať a upravovať v jednom z podporovaných programovacích jazykov. IBM Rational Performance Tester je špecializovaným nástrojom pre realizáciu záťažových a výkonnostných testov.
- ***Nástroj IBM Rational AppScan*** je určený pre testovanie zabezpečenia webových aplikácií.
- ***IBM Rational Policy Tester*** je nástroj na testovanie webových aplikácií a existuje

v troch verziách. Verzia *Accessibility Edition* sa špecializuje na testovanie dostupnosti, predovšetkým na plnenie štandardov, zákonných nariadení a odporúčaní pre tvorbu webových aplikácií. *Privacy Edition* je špecializovaná na ochranu osobných údajov používateľov. *Quality Edition* sa zameriava na problémy ako sú nefunkčné odkazy medzi stránkami, gramatické chyby v obsahu stránok alebo ich pomalé načítanie a ďalšie. Všetky tieto nástroje fungujú tak, že prechádzajú stránky testovanej webovej aplikácie a vykonávajú definované testy. Zo získaných výsledkov je potom zostavená správa.

- ***IBM Rational Robot*** je nástroj pre automatizáciu testov pre klient-server aplikácie. Tento nástroj umožňuje tvorbu testovacích scriptov záznamom interakcie s grafickým užívateľským rozhraním aplikácie.

**Microsoft Solutions Framework (MSF)** je všeobecný rámec, na základe ktorého možno vytvoriť osobitnú metodiku, ktorá zodpovedá potrebám konkrétneho projektu alebo skupiny projektov. Zameriava sa tak na oblasť softvérovo-inžiniersku, tak na riadenie projektu a je podporovaný vývojovou platformou Microsoft Visual Studio Team System. MSF ponúka dve preddefinované metodiky - MSF for Agile Software Development a MSF for CMMI Process Improvement (http://online.gerrardconsulting.com).

***MSF for Agile Software Development*** sa zaoberá oblasťou závislého aj nezávislého testovania. U každej roly sú definované činnosti spojené s testovaním, ktoré sú detailnejšie rozpracované do jednotlivých krokov, pri ktorých sú k dispozícii odporúčania a rady pre ich vykonávanie. Testovacie aktivity a kroky sú u oboch kategórií testovania opísané rovnako podrobne a testovanie sa zapája do životného cyklu vo fáze analýzy a návrhu. Testy sú tvorené pred kódovaním alebo súbežne s ním. Metodika odporúča vyčleniť samostatné iterácie na riešenie a odstraňovanie chýb, ak pribúdajú podstatne vyšším tempom, ako sa ich darí odstraňovať v rámci bežného behu iterácie.

Zo skupiny testovacích rolí je v tejto metodike definovaná iba rola testera. Úlohou testera je objaviť chyby v produkte a ďalšie problémy, ktoré môžu ovplyvniť výslednú kvalitu a hodnotu pre zákazníka, určiť ich vplyvy a navrhnúť postupy, ako tieto vplyvy zmierniť. Podieľa sa tiež na plánovaní iterácie, kde plní konzultačnú úlohu.

Na testovaní sa podieľajú aj ďalšie roly. Vývojári tvoria, realizujú a vyhodnocujú testy jednotiek a integračné testy. Databázoví vývojári tiež testujú pomocou testov

jednotiek. Vo fáze nasadenia preberá zodpovednosť za testovanie manažér nasadenia - release manager, tester prechádza do roly konzultanta. Vývojári sú tiež zodpovední za realizáciu testov pri overovaní, či chceme aktuálnu zostavu produktu, či spĺňa minimálne stanovené požiadavky na kvalitu.

Jednotkové testy metodiku delia do dvoch skupín. Prvou skupinou sú jednotkové testy aplikácie, za ktorých tvorbu, realizáciu a vyhodnotenie sú zodpovední vývojári. Druhú skupinu tvoria testy jednotiek databázy, za ktoré sú zodpovední databázoví vývojári. Oba druhy jednotkových testov sa vykonávajú priebežne vždy, keď vývojári dokončia im pridelené úlohy alebo ak dôjde k zmene vo vyvíjanom systéme. Po dokončení úloh vývojárov nasledujú integrácie a integračné testovanie.

Ako systémové testy môžeme chápať testy scenárov a testy kvality služby, obe skupiny testujú celý systém od vstupov až po výstupy. Testy scenárov overujú splnenie funkčných požiadaviek, testy kvality služby testujú nefunkčné požiadavky, respektíve výkonnosť, bezpečnosť, záťaž a výkon pri obmedzených zdrojoch. Akceptačné testy ako také nie sú metodikou definované.

Pred nasadením produktu v organizácii je vytvorený produkt otestovaný, aby sa overilo, že sú splnené všetky špecifikované požiadavky a všetky známe problémy sú riešené - buď boli odstránené, alebo aspoň existuje popísaný návod, ako sa im vyhnúť. Zákazník ale nie je prítomný, po vykonaní vyššie opísaného testovania je produkt nasadený. Vzťah medzi požiadavkami a testami nie je jasný a nie je zachytený ani v rámci popisov jednotlivých činností ani v dokumentácii testov, alebo v požiadavkách nie je vyznačená ich súvislosť.

Životný cyklus chyby je podrobne popísaný. Nové chyby ohlasujú testeri a vývojári, chyba sa tak dostáva do stavu aktívnej. Manažér projektu priraďuje chybám prioritu z hľadiska ich riešenia, o ktoré sa starajú vývojári. Chyba prechádza do stavu vyriešená. Odstránenie chyby je overené testermi a životný cyklus chyby je uzavretý.

V rámci testovania požiadaviek na kvalitu služby sú vykonávané nasledujúce typy testov: výkonnostné testy, bezpečnostné testy, záťažové testy a stress test (testy správania aplikácie s minimom alebo nedostatkom výpočtových zdrojov). Zvlášť sú vymedzené webové testy, ktoré overujú webové stránky a komunikáciu prostredníctvom internetu. Na prieskumné testovanie je odporúčaná technika testovania ako na testovanie funkčných požiadaviek tak aj požiadaviek nefunkčných. Základom pre implementáciu tohto

testovania je definícia vzorových užívateľov, tzv. personas. Ich správanie je počas prieskumného testovania simulované. Po integrácii produktu je doporučené vykonať smoke test.

Metodika nepodporuje vytváranie rozsiahlej testovacej dokumentácie. Zo štandardu IEEE 829 pokrýva iba dokument Test Plan, ktorému zodpovedá dokument nazývaný Test Approach - prístup k testovaniu (http://online.gerrardconsulting.com). Metodika odporúča, aby stanovený prístup k testovaniu obsahoval ako testy vykonávané manuálne, tak testy automatizované. Automatizované by mali byť tie testy, ktoré sú podľa situácie vhodné a možné. Bližšie určenie prístupu k automatizácii metodika neobsahuje. Automatizácia je teda hodnotená ako odporúčaná.

Prepojenie medzi metodikami založenými na **Microsoft Solutions Framework** a sadou nástrojov **Microsoft Visual Studio Team System (VSTS)** je veľmi silné. Aplikovať princípy MSF vo vývoji je možné aj bez VSTS, však už spôsob distribúcie týchto metodík napovedá, že najväčší efekt prináša ich použitie v spojení práve s týmito nástrojmi. Metodiky MSF Agile a MSF CMMI sú distribuované ako procesné šablóny pre VSTS. To znamená, že definujú, aké činnosti sú jednotlivými členmi tímu vykonávané, akú podobu majú vstupy a výstupy týchto činností, aké pracovné položky budú používať a aké správy a reporty bude mať vedenie projektu k dispozícii. Procesné šablóny možno podľa potreby upravovať a definovať ako aj vlastný proces. Okrem procesných šablón pre MSF sú k dispozícii aj šablóny pre procesy a metodiky tretích strán, napríklad pre agilné metodiky **SCRUM**.

Testerom je určená samostatná verzia ***Visual Studia***, ktorá obsahuje nástroje pre tvorbu, spúšťanie a správu testov. Testovanie nie je v MSF záležitosťou iba testerov, preto nástroje pre jednotkové testovanie sú obsiahnuté aj vo verziách pre vývojárov a databázových špecialistov. Nástroje pre zisťovanie pokrytia kódu testami sú spoločné verziám pre vývojárov a pre testerov. Verzia pre testerov podporuje jednotkové testovanie, záťažové testovanie a testovanie webových aplikácií.

Opísané testové prípady, ktoré majú byť vykonávané manuálne, je možné uložiť ako manuálny test. Manuálne testy nemajú žiadnu osobitnú podporu, jedná sa len o textové dokumenty s opísanými krokmi testovacieho scenára, ktoré sú uložené v spoločnom úložisku a ich obsahu možno pristupovať z prostredia Visual Studia.

## 1.7 Fakultná komunikácia

**Aktívna fakultná komunikácia by mala plniť dvojakú úlohu:**

- musí zabezpečiť správny komunikačný tok vo vnútri inštitúcie,
- musí zabezpečiť správny komunikačný tok s vonkajším prostredím, ktorý je determinovaný stanovenou univerzitnou kultúrou.

**Vnútrofakultná komunikácia:**

- Indiferencie vnútrofakultnej komunikácie oproti komunikácii všeobecnej spočívajú v tom, že komunikačný tok je determinovaný špecifikami konkrétneho univerzitného prostredia.

**V rámci vnútrofakultnej komunikácie je potrebné venovať pozornosť:**

- komunikačným nástrojom,
- motivácii a správaniu sa účastníkov v komunikačnom procese,
- sociálno-kultúrnemu rámcu komunikácie.

**Medzi základné funkcie komunikácie patrí (Carda - Kunstová, 2003):**

- ***Informatívna*** - u vedúcich „manažérov“ sa predpokladá, že majú schopnosť jasne, jednoducho, zrozumiteľne sformulovať informáciu a že ju vedia efektívne aj ďalej posunúť.
- ***Inštruktážna*** - každý vedúci „manažér“ musí byť schopný podať informáciu tak, aby ľudia, ktorým je určená, dokázali túto informáciu absorbovať, správne vyhodnotiť, osvojiť si ju a využiť ju pri výkone svojej pracovnej náplne.
- ***Personálna*** - schopný „manažér“ musí vedieť presvedčiť svojich podriadených o zmysluplnosti a relevantnosti svojich požiadaviek a získať ich pre efektívnu spoluprácu. Tento „manažér“ by mal byť schopný vytvoriť takú atmosféru, aby si zamestnanci vybudovali vlastný vnútorný súhlas s jeho rozhodnutím aj požiadavkami a aby aktívne, dobrovoľne spolupracovali. Mal by vedieť apelovať na prestavbu názorov, postojov a záujmov zúčastnených strán komunikácie.

- ***Motivačná*** - „manažér“ musí byť zručný v motivovaní zamestnancov, musí vytvoriť také prostredie a pracovnú atmosféru, ktorá napomôže uspokojovať vnútorné ašpirácie, potreby a záujmy zamestnancov, a ktoré ich budú stimulovať konať žiaducim spôsobom a odvádzať efektívnu prácu.
- ***Kontaktná*** - schopnosť „manažéra“ vytvárať a udržiavať komunikačný kontakt s partnerom počas celého komunikačného aktu, aj ak sa nevyvíja podľa jeho predstáv či je v rozpore so zámerom komunikácie s dôrazom, že táto situácia sa aj môže začať vyvíjať konfliktne.
- ***Kontrolná*** - „manažér“ musí vedieť prostredníctvom komunikácie efektívne podať zamestnancovi spätnú väzbu o účelnosti jeho vlastného príspevku k tvorbe univerzitnej hodnoty.
- ***Emotívna*** - zručnosť „manažéra“ prostredníctvom komunikácie vytvoriť priestor na saturovanie potrieb sociálneho kontaktu (t. j. možnosť vyjadrenia určitých myšlienok, názorov, pocitov na základe, ktorých sa ľudia na škole poznávajú, porovnávajú a rozvíjajú si svoje komunikačné zručnosti).
- ***Sebapoznávacia*** - schopnosť sebareflexie každého pedagóga a zamestnanca, pretože každý účastník komunikácie prostredníctvom vlastného príspevku do dialógu odráža svoju vlastnú sebadôveru, seba vnímanie.

Tabuľka 1 Fakultná komunikácia

| | Oblasti komunikácie | Ciele |
|---|---|---|
| **Fakultná komunikácia** | 1. *komunikácia smerom dolu a do strán*<br>• informácie o celouniverzitných, fakultných cieľoch, o plánoch, rozpočtoch... | • zaistiť, aby pedagógovia a radoví zamestnanci obdržali presné a včasné informácie o úlohách, ktoré je potrebné vykonať pre dosiahnutie vytýčených cieľov |
| | 2. *komunikácia smerom dolu*<br>• priame inštrukcie podriadeným, čo majú vykonať | • zaistiť, aby inštrukcie boli jasné a presné<br>• poskytnúť potrebnú motiváciu ľuďom k práci |
| | 3. *komunikácia smerom hore*<br>• návrhy, námety a pripomienky k fakultným cieľom, plánom, rozpočtom... | • zabezpečiť dostatočný priestor pre všetkých zamestnancov, ktorý by im umožnil ovplyvniť školské rozhodnutia v oblastiach, v ktorých získali určitú odbornosť a skúsenosť |
| | 4. *komunikácia smerom hore a do strán*<br>• informácie o vykonávanej práci a výsledkoch | • umožniť vedeniu monitorovať a riadiť výkon<br>• v prípade potreby uskutočniť nápravné opatrenia |

*Zdroj: vlastný návrh*

# 2 Cieľ práce

Vo všetkých firmách, organizáciách, a hlavne inštitúciách zavádzajúcich model svojho komunikačného, komplexného, manažérskeho informačného systému, je tento model zároveň postupom implementácie systému, teda nástrojom zlepšovania tvorby pridanej hodnoty, akejkoľvek konkurenčnej výhody, je cestou k inováciám, a v neposlednom rade je zdrojom zmeny aj samotnej podnikovej kultúry, kde prevláda tvorivosť ale je aj zadefinovaná disciplína rozhodovania sa na základe informácií získaných z naimplementovaného komunikačného systému.

Postupy implementácie komunikačného systému však so sebou prinášajú aj neistotu. Manažment často dnes rozhoduje o umiestnení obmedzených zdrojov do konkrétnych projektov zlepšovania a ďalších rôznych svojich aktivít, zároveň je návratnosť z týchto projektov neistá, avšak existencia takýchto projektov zlepšovania v akomkoľvek konkurenčnom prostredí je nevyhnutná. Táto neistota návratnosti je typická nielen pre rôzne malé a stredné podniky, ktoré nemajú možnosť diverzifikácie svojich aktivít do veľkého množstva projektov zlepšovania, ale hlavne pre inštitúcie a obzvlášť pre školské inštitúcie. Práve z tohto dôvodu považujeme za veľmi aktuálne, najmä v rámci vysokoškolského prostredia na Slovensku, vytváranie modelov komunikačných systémov na univerzitách.

**Hlavným cieľom** dizertačnej práce je vytvoriť model fakultného komunikačného systému na univerzite ekonomického zamerania. Zo stanoveného hlavného cieľa sme odvodili **čiastkové ciele** dizertačnej práce:

- vytvoriť prehľad, ktorý by zosumarizoval existujúce modely komunikačných systémov z pohľadu vedeckého a praktického;
- identifikovať existujúce modely kompatibilných komunikačných systémov využiteľných na akademickej pôde;
- navrhnúť funkčný model a vhodné IKT prostredie;
- aplikovať navrhnutý model komunikačného systému v konkrétnych podmienkach našej fakulty;
- určiť výkonnostné parametre na posúdenie vhodnosti naimplementovaného modelu komunikačného systému.

Na základe takto nami definovaných cieľov je očakávaný aj prínos dizertačnej práce, ktorým má byť vytvorenie postupu, ktorý by mohol viesť k úspešnej implementácii komunikačného systému. Vychádzajúc z dostupnej literatúry, je spoločnosť akademikov a praktikov sústredená v oblasti využitia manažmentu znalostí pri implementácii akéhokoľvek komunikačného systému zameraná väčšinou na manažérsku oblasť rôznych malých či stredných podnikov, ale čo sa týka akademickej pôdy, je to väčšinou akademický informačný systém okliešteny na správu študentskej agendy a evidenciu výsledkov štúdia. Prínosom, ktorý sa dosiahne splnením cieľov dizertačnej práce, bude práve model implementácie komunikačného systému pomocou nami zadefinovaných konkrétnych inštalačných postupov.

Dizertačná práca významovo bude vychádzať zo skúšobnej paralelnej implementácie komunikačného systému na našej fakulte so zreteľom na prispôsobenie sa všetkých katedier a oddelení, ktoré budú so systémom pracovať. Vzhľadom k tímovej spolupráci so všetkými pracoviskami fakulty je pravdepodobné, že predmetom prípadovej štúdie a experimentálnej implementácie bude jedna konečná výsledná podoba modelu komunikačného systému. Na túto implementáciu bude aplikovaná navrhnutá metodológia spolu s odporúčaniami, čím sa do budúcnosti vytvorí možnosť verifikovať navrhované znalostné postupy na zdokonaľovaní už naimplementovaného systému, čím sa zároveň vytvorí možnosť komparácie s rôznymi odskúšanými tradičnými metódami implementácií.

V dizertačnej práci je kladený dôraz na vytvorenie nového unikátneho modelu komunikačného systému. To z objektívnych dôvodov vylučuje overovanie jeho správnej funkčnosti prostredníctvom rôznych induktívnych metód. V prípade pilotných implementačných modelov sú preto vhodnejšie kvalitatívne, logické metódy hodnotenia implementácie, podporované aj kvantitatívnymi metódami, a to znamená, že v tomto prípade ide o odlišný prístup ako napríklad v prípade výskumu nejakých hromadných implementácií firemných komunikačných systémov, kde kvalitatívne a logické metódy hodnotenia následných inštalácii podporujú aj kvantitatívny výskum. Tomu sú podriadené aj tézy dizertačnej práce:

1. Analýza potreby zavedenia modelu komunikačného systému na Podnikovohospodárskej fakulte Ekonomickej univerzity v Bratislave so sídlom v Košiciach.
2. Charakteristika existujúcich postupov implementácie modelov komunikačných systémov.

3. Rozbor kritických faktorov, premenných, predpokladov a podmienok, ktoré by mali vstupovať do navrhovaného modelu implementácie komunikačného systému.
4. Návrh modifikácií, úprav existujúcich modelov a samotný návrh nového modelu implementácie komunikačného systému.
5. Implementácia modelu do univerzitnej praxe, jeho revízia a stanovenie nových kritických parametrov pre budúcu vedeckú prácu s týmto systémom.

# 3 Metodika práce a metódy skúmania

Kapitola špecifikuje popis objektu skúmania, popis kritérií pri výbere fakultného komunikačného systému, obsahuje zadefinovanie možných bariér komunikácie, metodiku práce a podrobný postup riešenia dizertačnej práce.

## 3.1 Popis objektu skúmania

**Fakultný komunikačný systém by mal tvoriť niekoľko zložiek:**

- ***subjekty komunikácie:***
  - vedenie školy,
  - pedagógovia,
  - študenti,
  - zamestnanci,
  - subjekty vonkajšieho prostredia (záujemcovia o štúdium, široká verejnosť, inštitúcie..),
- ***nástroje komunikácie,***
- ***prostredie komunikácie (kanály, technické a organizačné zabezpečenie).***

**Môžeme teda hovoriť o fakultnom komunikačnom systéme a o komunikácii:**

- ***formálnej*** – toky informácií, ktoré sú inštitúciou oficiálne uznané a aj vyžadované subjektom komunikácie, zároveň upravené formálnymi predpismi,
- ***neformálnej*** – toky informácií, ktoré nie sú oficiálne uznané a nie sú žiadnym spôsobom korigované a vznikajú na základe spontánnych medziľudských vzťahov za účelom uspokojenia potrieb sociálneho kontaktu.

Základným rozhodnutím pri prenose informácií je identifikácia optimálneho komunikačného nástroja - každý fakultný komunikačný nástroj by mal byť navonok integrovaný a vnútorne ekonomicky a psychologicky štruktúrovaný prostriedok prenosu

a spätnej odozvy informácií, správ, odkazov, príkazov a nariadení v komunikačnom procese, ktorého cieľom je zabezpečiť na fakulte viacstupňový tok informácií potrebných na plnenie stanovených pracovných úloh a dosiahnutie efektívneho vyučovacieho procesu.

Každá fakulta už má dnes naimplementovaný akademický informačný systém, ktorý umožňuje elektronické spracovanie celej študijnej agendy konkrétneho študenta od jeho prijatia až po zapísanie výsledkov zo štátnych záverečných skúšok a obhajobu záverečnej práce. Tieto systémy, ako napríklad AIS, MAIS, IS MUNI a ďalšie, však nedokážu plnohodnotne nahradiť celofakultný komunikačný nástroj.

**Pri výbere vhodného fakultného komunikačného nástroja by sa mali uplatniť nasledujúce kritériá:**

- potrebná prenosová rýchlosť informácií s požiadavkou na aktuálnosť,
- finančná náročnosť prenosu,
- dostupnosť,
- potrebné zručnosti na využívanie komunikačného nástroja,
- kvantita a forma výstupu informácií za použitia jednotlivých komunikačných nástrojov,
- miera spoľahlivosti,
- hodnota komunikovanej informácie,
- možnosť interakcie,
- účel a cieľ komunikácie,
- zohľadnenie štádia vývoja komunikácie, v akom sa fakulta nachádza,
- veľkosť fakulty - počet zamestnancov,
- kultúra vzdelávacej inštitúcie.

Tabuľka 2 Možné bariéry komunikácie

| MOŽNÉ BARIÉRY KOMUNIKÁCIE | | |
|---|---|---|
| ***na strane prostredia fakulty - vedenia*** | ***na strane účastníkov komunikácie - pedagógov, zamestnancov*** | ***vyplývajúce zo vzťahu nadriadenosti a podriadenosti*** |
| interferencie, šumy | limitované schopnosti adresáta, sémantické problémy | nedôvera, hrozby, obavy, osobné komplexy, stres, nedostatok pocitu bezpečnosti |
| časová tieseň | neurčitosť výpovede | podvádzanie, klamanie, filtrovanie, informačná blokáda |
| komunikačné preťaženie, informačná presýtenosť | nevhodná forma prezentácie | odlišné postavenie, staus, moc |
| nedostatok komunikačných kanálov | selektívne vnímanie, abstrahovanie | dôveryhodnosť zdroja |
| nedostatočné podmienky pre otvorený komunikačný tok | neschopnosť koncentrácie | |
| nedostatočné riadenie komunikácie | zlá schopnosť aktívne počúvať | |
| kultúrne, jazykové, etické odlišnosti | odlišné postoje, názory, skúsenosti | |
| | zamlčané predpoklady | |
| | motivácia | |

*Zdroj: Vlastný návrh*

**Na strane prostredia, teda fakulty, môžu vzniknúť nasledovné objektívne a subjektívne bariéry:**

- ***Časová tieseň***
  - vedúci pracovníci nemajú čas komunikovať častejšie s každým svojím podriadeným. Z tohto dôvodu sú pedagógovia a zamestnanci často oboznámení s úlohami alebo prípadnými zmenami na fakulte pomerne neskoro a musia sa veľmi rýchlo adaptovať na novú situáciu,
  - časová tieseň významným spôsobom tlačí na zvyšovanie rozhodovacích právomocí jednotlivých zamestnancov. Efektívnym spôsobom prekonávania tejto bariéry je zo strany vedúceho pracovníka delegovanie častí právomocí na podriadených zamestnancov, aby v časovom strese mohli sami rozhodnúť bez toho, aby museli žiadať súhlas nadriadeného.

- ***Komunikačné preťaženie***
  - v súvislosti s výrazným progresom vo využívaní IKT pri práci zamestnancom neustále prichádza príliš veľké množstvo informácií. Pri takomto informačnom preťažení, presýtení, nie sú schopní ich adekvátne absorbovať a zároveň na ne primerane reagovať. Správne nastavený komunikačný tok na fakulte a riadenie komunikácie dokážu usmerňovať toky informácií naozaj tým, ktorí ich potrebujú, a zároveň obmedzuje toky informácií tým, ktorí ich nepotrebujú.
- ***Nedostatok komunikačných tokov***
  - v súvislosti s neefektívnym riadením fakultnej komunikácie nastávajú situácie, kedy by mohli zamestnanci fakulty komunikovať efektívne, ale nemajú k dispozícii potrebné komunikačné kanály alebo komunikačné nástroje. Z tohto dôvodu vzniká veľké množstvo fakultných problémov, pretože zamestnanci sa často dôležité skutočnosti nedozvedia alebo sa ich dozvedia príliš neskoro.

**Zdokonaľovanie komunikačného toku** - ak sa na fakulte identifikujú možné bariéry komunikácie, ktoré môžu ohrozovať efektívnu komunikáciu, potom by mala fakulta hľadať možnosti a spôsoby ich odstránenia a zároveň nastaviť také podmienky pre komunikáciu, aby sa tieto nedostatky už neopakovali. Zároveň by sa mala fakulta snažiť o neustále zlepšovanie komunikačného procesu. Cieľom je teda vytvoriť celofakultný portál na zefektívnenie práce zamestnancov. Jeho úlohou je zhromažďovať informácie z rôznych zdrojov a následne ich zobrazovať na jednotnú obrazovku podľa nastavenia konkrétneho užívateľa. Hlavnou myšlienkou portálu je vyriešiť v rámci jedného komunikačného systému všetky svoje úlohy. Pomocou tohto riešenia sa zamestnancom uľahčí práca a zvýši sa ich produktivita, pretože už nebudú musieť neustále prechádzať z jednej aplikácie do druhej.

## 3.2 Metodika práce

Každá systémová analýza a projektovanie komunikačných systémov sa skladá z dvoch podstatných komponentov:

- ***podstata*: riešenie problémov**
- ***problém*: odchýlka od žiaduceho stavu, vyžadujúce riešenie, ktoré hľadáme**

Aby sme dosiahli konkrétny cieľ, musíme zakomponovať všeobecne používané metódy: **systémový prístup, modelovanie, inžinierstvo** a **projektovanie**. Pritom tieto metódy majú aj svoje obmedzenia, pretože nejde o metódy univerzálne a teda použiteľné pri riešení akýchkoľvek problémov, sú však účinné v prípade návrhu komunikačných systémov.

**Systémový prístup** je všeobecná metóda vedeckého myslenia, ktorej podstatou je analýza fungovania zložitých celkov v dôsledku ich štruktúry. Rozoznávame ale *klasický newtonovský* alebo taktiež označovaný ako *mechanistický prístup*, ktorý je význačný poznávaním celku jeho rozdelením na časti a štúdiom ich vlastností a s neuvažovaním vzťahov medzi časťami. Mechanistický prístup je konkrétny - zaujímajú nás fyzické komponenty - prvky.

*Systémový prístup* je naopak význačný poznávaním celku prostredníctvom vzťahov medzi jeho časťami a tým, že celok môže mať vlastnosti nevyplývajúce priamo z vlastností jeho jednotlivých častí. Systémový prístup je abstraktný - zaujímajú nás konkrétne logické komponenty - prvky:

- vymedzenie hľadiska skúmania, stanovenie cieľa - odlíšenie daného systému od iných systémov, ktoré možno na objekte definovať,
- vymedzenie hraníc systému, zahrnutie prvkov a procesov - odlíšenie daného systému od okolia, určenie hranice systému rozhodnutí, ktoré entity zahrnúť alebo nezahrnúť do systému a zoznam prvkov aj procesov systému,
- proces štruktúrovania - sú to rozhodnutia, ako usporiadať vlastnosti množiny zvolených entít, a teda prvkov systému, definovanie vzťahov prvkov a procesov, rozlíšenie podstatných a nepodstatných prvkov a vzťahov.

Základné kroky pri využití systémového prístupu k riešeniu problémov sú:

- Rozložiť pôvodný problém na niekoľko menších problémov; pokračovať v procese rozkladu, až bude každý podproblém dostatočne malý, aby sa dal rýchle vyriešiť.

- Sformulovať riešenie pre každý jednotlivý podproblém.
- Spojiť riešenia podproblémov v jeden celok.
- Aplikovať kompletné riešenie na pôvodný problém.
- Preveriť, či riešenie je správne - otestovanie navrhnutej štruktúry a funkcie.

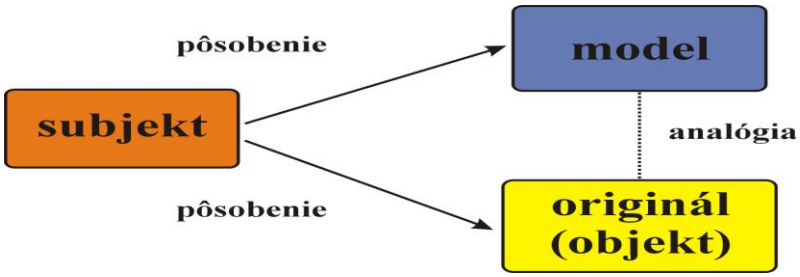
Riešenie problémov prostredníctvom komunikačných systémov:

- Analýza problému (systémová analýza - systems analysis) a teda aký je problém? Aké je najlepšie riešenie problému? Výsledok: návrh riešenia, jeho časového horizontu, požiadavky na financie.
- Návrh riešenia problému (projekt systému - systems design), a to ako transformovať zvolené riešenie do navrhovaného informačného systému? Výsledok: voľba hardvéru, návrh spôsobu reprezentácie dát v počítači, konkrétna voľba resp. vlastný návrh softvéru, riešenie komunikácie užívateľa so systémom - vytvorenie užívateľského rozhrania.

**Modelovanie (modelová tvorba)** vychádza zo všeobecných zákonitostí teórie podobnosti, najmä z princípu analógie. Je to odbor študujúci otázky racionálnej konštrukcie modelov, ktoré odrážajú vlastnosti, vzťahy a správanie reality, a je to aj nástroj umožňujúci namiesto teoretického skúmania použiť empirické metódy (napr. experiment) tým, že empirické výsledky sa prenesú z modelu na skúmaný objekt alebo na konkrétny navrhovaný artefakt.

**Model** je zjednodušené, alebo zovšeobecnené zobrazenie systému zavedeného na objekte, ktoré sa s týmto objektom zhoduje v podstatných vlastnostiach, ale aj analogicky určenej časti prírodnej alebo sociálnej reality (schéma, štruktúra, znakový systém) ako originálu; tento model (označovaný aj ako analóg), slúži k hlbšiemu poznaniu originálu, jeho konštrukcii a organizácie, ale aj rôznych premien a ich podmienok.

Vzťah *subjekt - model - originál* vieme znázorniť aj graficky (obrázok 4).




Obrázok 4 Vzťah subject - model - original (vlastný návrh)

Funkcie (využitie modelu) - model nahrádza subjektu pochopenie originálu alebo manipuláciu s originálom:

a) poznávacia - štúdium štruktúry pôvodného objektu napr. prostredníctvom verbálneho opisu inak nedostupného objektu; vzor/plán/návrh budúceho objektu alebo procesu,
b) manipulačná, overovacia - myšlienkové alebo materiálne experimenty, simulácia a overenie vlastností skutočných objektov napr. stability,
c) komunikačná, dokumentačná.

Základné typy modelov:

a) statický - dynamický
   - statický model zobrazuje spravidla štruktúru a teda vnútorné usporiadanie javov a procesov,
   - dynamický model zobrazuje prevažne správanie, funkciu systémov,
b) konkrétny - abstraktný
   - konkrétny model, alebo tiež názorný model - môže to byť maketa, prototyp - sú to obrazy, reálne modely, priestorové (3D) modely, animácie, video (4D tu je zahrnutý aj čas),
   - abstraktný model: napríklad matematický vzorec,
c) individuálne vytváraný - štandardizovaný
   - individuálne vytváraný model umožňuje vystihnúť špecifiká originálu,
   - štandardizovaný model je všeobecne zrozumiteľný,
d) konceptuálny (pojmový, esenciálny) - technologický
   - konceptuálny model je implementačne nezávislý,
   - technologický model je napríklad návrh systému pre konkrétne implementačné prostredie (softvér, hardvér),

e) AS-IS - TO-BE
   - AS-IS je model súčasného stavu,
   - TO-BE je model cieľového stavu (plán, projekt),

f) podľa formy vyjadrenia delíme modely na:
   - textový (verbálny alebo slovný) - hľadaná informácia sa získa pochopením (interpretáciou) textu, používa prostriedky prirodzeného jazyka, môže vyjadriť aj to, čo je počuť (písmo je modelom reči, noty sú modelom hudby), môže vyjadriť aj abstrakciu alebo idey (myšlienky), je lineárny (neštruktúrovaný) - štruktúru popisovaného problému musíme vytvoriť vo svojej mysli,
   - matematický (vzorec) - hľadaná informácia sa získa určitým výpočtom,
   - číselný - efektívne vyjadrenie kvantity (množstva),
   - obrazový (názorný, 2D, 3D, 4D) - hľadaná informácia sa získa sledovaním modelu, efektívne vyjadrenie tvarov a farieb, jasné znázornenie akcie alebo scény, ale môže ukázať aj to, čo nie je vidieť (napr. mapy), môže eliminovať nepodstatné javy, grafické schémy sú štruktúrované a teda priamo znázorňujú štruktúru popisovaného problému.

**Požiadavky na model:**

- úplnosť,
- presnosť,
- minimálna redundancia,
- jednoduchosť, čitateľnosť (7 ± 2),
- konzistencia,
- hierarchia úrovní,
- pomer grafika (väčšina) - text (menšina).

**Systémová metodológia a typy vytváraných modelov**:

***technika*** - je popis činností (operácií) pri riešení problému a je skôr konkrétna, pretože popisuje čo robiť a ako to urobiť - je teda pomôcka, nástroj,

***metóda*** - je zaznamenaná skúsenosť, oddelená od pôvodného nositeľa

Je k dispozícii
pre opakovné použitie

K realizácii metódy
nepotrebujeme jej autora

***možnosti***, ako úspešne realizovať činnosť:

- pozveme si odborníka, ktorý to vie urobiť,
- sami to robíme tak, ako to robí tento odborník (= metóda),

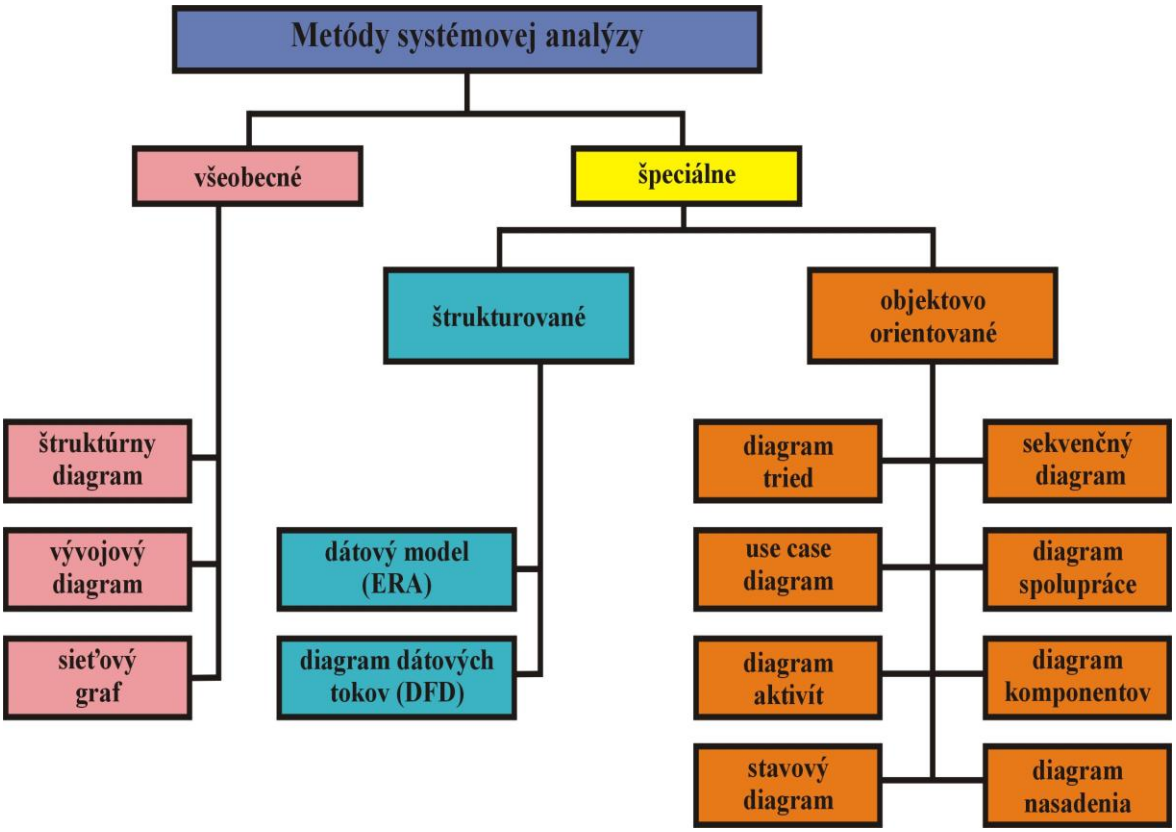
***metodika*** - je postup, ako zvoliť operácie vhodné na riešenie problému a je skôr abstraktná, predstavuje návod prečo to robiť práve tak a ako to robiť.

Tabuľka 3 Metódy systémovej analýzy

| **1. Všeobecné metódy** | **2. Špeciálne metódy** |
|---|---|
| • zaujímajú nás **objekty** | • zaujímajú nás **dáta** |
| • možné použiť pre znázornenie **ľubovoľnej reality** | • používame výhradne pre navrhovanie **komunikačných systémov** |
| • zobrazujeme všetky spozorované objekty a procesy | • sústredíme sa len na to, čo bude spracované v počítači – objekty a procesy mimo dosah informačných a komunikačných technológií nepopisujeme |

*Zdroj: Vlastný návrh*

Rozdelenie metód systémovej analýzy môžeme graficky znázorniť podľa obrázka 5.




Obrázok 5 Grafické znázornenie metód systémovej analýzy (vlastný návrh podľa George - Batra - Valacich - Hoffer, 2004)

**Teoretické zázemie projektovania komunikačných systémov:**

Projektovanie komunikačných systémov nie je veda, ale oblasť praktickej ľudskej činnosti, ktorá čerpá potrebné teoretické základy z početných teoretických a aplikovaných vedných disciplín ako matematika, logika, systémová veda, kybernetika, teória riadenia, počítačová veda atď. Cieľom projektovania potom nie je ďalší rozvoj príslušných teoretických odborov, ale využitie ich poznatkov pri riešení reálnych problémov.

**Teória systémov, tiež systémová teória (systems theory), všeobecná teória systémov (general systems theory), systémová veda (systems science)** (George - Batra - Valacich - Hoffer, 2004) je vedná disciplína v logicko-matematickej oblasti, ktorá sa zaoberá formuláciou a dedukciou všeobecných systémových zákonov, aplikovateľných na všetky systémy určitého typu bez ohľadu na povahu ich prvkov a vzťahov. Je to všeobecný teoretický prístup k vystihnutiu najvšeobecnejších rysov a zákonitostí rôznych typov systémov na základe ich izomorfie (homomorfie). Metodológia teórie systémov

predstavuje spojovací mostík medzi filozofickou metodikou a špecifickými metódami jednotlivých vied. Predmetom skúmania je systém, metódou skúmania je systémový prístup, systémová analýza a aj syntéza a aplikačnými oblasťami sú všetky oblasti, v ktorých prebieha príprava a zhodnocovanie rozhodnutia a riešenie zložitých problémov ako kybernetika, operačný výskum, a rôzne inžinierske disciplíny.

**Inžinierstvo (engineering)** - je systematická aplikácia vedeckých poznatkov pri návrhu a tvorbe cenovo efektívnych riešení praktických problémov a:

- vychádza z vopred určeného cieľa, existujúcich obmedzení a požadovanej spoľahlivosti,
- uplatňuje sa pri tvorbe rozsiahlych, zložitých a technologicky náročných artefaktov, t. j. objektov vytvorených človekom,
- riešenie rešpektuje existujúce štandardy a je zdokumentované.

**Inžinierske „pravidlá hry“ - od konkrétnych ku všeobecným:**

- podnikové predpisy, best practices - príklady najlepšej praxe,
- štandardy,
- princípy a metódy danej inžinierskej disciplíny,
- všeobecné inžinierske princípy a metódy.

Inžinierske disciplíny uplatňujúce sa pri projektovaní a prevádzke komunikačných systémov a oblasti ich pôsobnosti sú uvedené v tabuľke 4.

Tabuľka 4 Inžinierske disciplíny

| | Analýza, Návrh, Realizácia a Prevádzka |
|---|---|
| **systémové inžinierstvo (systémová analýza)** | systémov |
| **priemyslové inžinierstvo** | výroby |
| **operačný výskum (manažérska veda)** | rozhodnutí |
| **projektový management** | projektov |
| **informačné inžinierstvo** | informačných systémov |
| **softvérové inžinierstvo** | softvéru |
| **organizačné inžinierstvo** | organizačných a riadiacich štruktúr |
| **webové inžinierstvo** | webových aplikácií |
| **znalostné inžinierstvo** | expertných systémov |

*Zdroj: Vlastný návrh*

**Softvérové alebo aj informačné inžinierstvo** - hlavné okruhy činností:

A. analýza
   - problémov,
   - podnikových procesov,
   - informačných potrieb a užívateľských požiadaviek,

B. návrh softvéru, alebo IS:
   - programovanie,
   - tvorba užívateľského rozhrania,
   - tvorba dokumentácie,
   - testovanie,

C. implementácia a integrácia softvéru, alebo IS,

D. prevádzka, zabezpečenie kvality, údržba softvéru, alebo IS.

**Systém** - slovo pochádza z gréčtiny (http://sk.wikipedia.org) - je niečo zložené, zoskupené v celok; spojenia, skupina, oddelenie, sústava, skladba, ale aj štátne zriadenie. Je to tiež usporiadaná rozmanitosť nejakých objektov buď materiálnych alebo ideálnych, ďalej to môže byť množina prvkov a väzieb medzi nimi s dynamickým, účelovým správaním, ktoré môže byť považované za jeden celok. Akonáhle na objekte dokážeme definovať jeho prvky a ich vzťahy, ktoré tvoria celistvosť objektu - teda štruktúru a fungovanie objektu,

potom sme do tohto objektu zaviedli systém.

***Typológia systémov:***

Systémy môžeme rozdeliť podľa vlastností prvkov, z ktorých sú zložené, ďalej podľa toho, akého typu sú vzťahy medzi prvkami a podľa správania:

A. *prirodzené* - *umelé*: rozlišujúcim kritériom je spôsob vzniku objektu a to s prispením resp. bez prispenia človeka,
B. *statické* - *dynamické*: rozlišujúce kritérium je stabilita stavu systému v čase,
C. *abstraktné* - *konkrétne*: rozlišujúcimi znakmi sú prvky, alebo vlastnosti, a ich vzťah k realite, napríklad pojmy resp. fyzikálne objekty,
D. *uzavreté* - *otvorené*: rozlišujúcim kritériom sú vstupy - výstupy, napr. vzťah k okoliu,
E. *jednoduché* - *zložité*: rozlišujúcim kritériom sú prvky a väzby a ich počet,
F. *deterministické* - *indeterministické (pravdepodobnostné, stochastické)*: rozlišujúcim kritériom je správanie, resp. možnosť jeho predvídania,
G. *vertikálne* - *horizontálne*: rozlišujúcim znakom sú väzby a teda existencia alebo neexistencia podriadenosti,
H. *homogénne* - *heterogénne*: rozlišujúce kritérium je postavené na podobnosti atribútov (vlastností) jednotlivých prvkov,
I. *„tvrdé“* - *„mäkké“*: rozlišujúcim kritérium je miera určitosti pri definovaní systému.

**Statické poňatie systému (štruktúru)** môžeme rozdeliť na:

- prvky,
- vlastnosti (atribúty) prvkov,
- usporiadanie prvkov (väzby, vzťahy),
- subsystém,
- okolie,
- hranice.

Statický pohľad na systém ho chápe ako celok zložený z dvoch tried: z triedy **prvkov** a z triedy **vzťahov** (relácií, väzieb) medzi prvkami. Týmto reláciám hovoríme **štruktúry**. V systéme sú prvky spojené štruktúrami do **jednotného celku**.

***Štruktúra:***

- je množina vzájomných vzťahov, ktorými sú späté prvky vo vnútri systému, usporiadané spravidla podľa jednotiaceho princípu a ktoré umožňujú predvídať správanie systému,
- je to taktiež skladba alebo usporiadanie prvkov a väzieb.

***Prvok systému:***

- je ďalej nedeliteľná časť celku (na danej rozlišovacej úrovni tvorí ďalej nedeliteľný celok, ktorého štruktúru nechceme alebo nemôžeme rozlíšiť),
- časť systému, v ktorej prebieha proces transformácie.

***Atribút (vlastnosť, charakteristika) prvku*** - určuje prvok po kvalitatívnej alebo kvantitatívnej stránke.

**Typy prvkov podľa umiestnenia v systéme:**

- *vnútorné* - má väzby len s prvkami toho istého systému,
- *hraničné* - *vstupné, výstupné* - má, alebo nemá aspoň jednu väzbu s prvkom, ktorý nie je prvkom systému,
- *externé* - *interné* - má aspoň jednu väzbu s prvkom, ktorý je prvkom systému) - vstupy a výstupy,
- *tranzitívny prvok* - prechádza systémom, určitú dobu je aj jeho súčasťou (externý - vnútorný - externý).

***Subsystém*** - je systém, ktorý je časťou (prvkom) iného systému.

***Vzťah (relationship)*** - je spojenie medzi prvkami alebo ich množinami.

***Okolie systému (prostredie - environment)*** - je to súbor prvkov, ktoré nie sú časťami systému, ale ktorých zmena môže spôsobiť zmenu stavu systému, a tých prvkov, ktorých vlastnosti sa môžu meniť správaním systému.

***Hranice - rozhrania systému (boundary - interface)*** - sú to množiny hraničných prvkov systému, sú to rozhrania systému, kadiaľ vstupujú prvky z okolia a vystupujú výstupy zo systému a kde platí vzťah rozhranie = väzba medzi systémami.

**Dynamické poňatie systému (funkcie, správanie)** - sem patria:

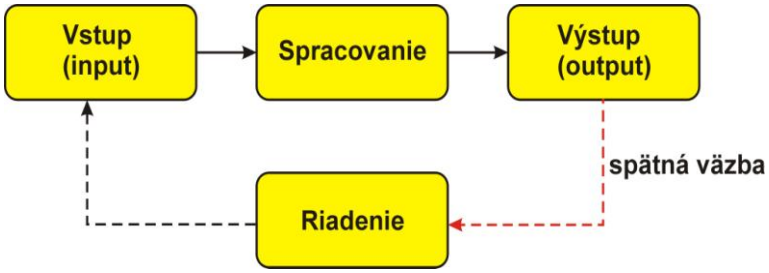
- vstupy,

- výstupy,
- proces (spracovanie - transformácia, riadenie),
- správanie,
- cieľ.

Dynamický pohľad na systém ho chápe ako spracovateľskú jednotku so vstupmi, výstupmi, riadením a spätnou väzbou, vyznačujúcou sa cieľovým správaním.

***Správanie systému*** - je spôsob realizácie cieľov a všeobecná charakteristika reakcie systému na podnety z okolia.

***Cieľ*** - je budúci stav, usporiadanie alebo správanie, ku ktorému systém smeruje, alebo ktoré sú systému vytýčené. Formuláciu cieľa ovplyvňujú neuspokojené potreby všetkého druhu.




Obrázok 6 Základné procesy, operácie a funkcie prebiehajúce v rámci systému (vlastný návrh)

***Proces (process)*** - je základná dynamická jednotka systému, vymedzená svojím začiatkom a koncom a transformuje vstup na výstup, alebo upravuje správanie systému.

***Vstup (input)*** - je všetko to, čo vstupuje do systému (externé prvky vstupujúce do systému).

***Výstup (output)*** - je zase všetko to, čo vystupuje zo systému (externé prvky vystupujúce zo systému) a je to výsledok procesu, alebo konečný stav systému.

***Väzba (relationship), tok (flow)*** - určuje postupnosť procesov, t. j. určuje, že výstup niektorého procesu je súčasne vstupom iného určitého procesu.

***Typy procesov:***

A. *základný proces* - označovaný aj ako spracovateľský proces - transformuje vstup na

výstup. Transformácia je pravidlo (predpis), podľa ktorého sa menia vstupy systému na jeho výstupy a je to zmena stavu nejakého objektu do iného, jasne diferencovaného stavu, napríklad pretvorením, pridaním, alebo odstránením určitých prvkov.

B. *riadiaci proces* - upravuje správanie systému. *Riadenie (management)* je vymedzovanie funkcie, cieľov systému a pôsobenie na systém s cieľom dosiahnuť jeho žiaducich funkcií, jeho približovanie k definovanému cieľu. Zahŕňa monitorovanie, vyhodnocovanie spätnej väzby, ovládanie a reguláciu štruktúry a aj správanie systému.

***Typy riadiacich procesov:***

- *sledovanie (monitorovanie)* je spojité alebo nepretržité získavanie relevantných informácií o stave a správaní systému, rozhodnutie, či systém smeruje k dosiahnutiu svojho cieľa. Regulačný proces potom vykoná prípadné úpravy vstupov a spracovateľských procesov.
- *ovládanie, regulácia (control)* je pôsobenie na systém, ktorého zmyslom je udržiavať systém alebo jeho výstupy vo vopred stanovených medziach,
- *obmedzenie (constraints)* je proces diktovaný nejakými vonkajšími faktormi, ktoré nemôžeme ovplyvniť (napr. legislatíva, veľkosť objektu, finančné zdroje),
- *cieľové správanie* - je množina transformácií, jednorazových alebo postupných stavov vedúcich z ľubovoľného daného prípustného východiskového stavu systému do stavu definovaného ako cieľový stav,
- *cieľavedomé správanie* - je špeciálna forma cieľového správania, typická pre človeka a spoločenské skupiny, keď je cieľ vedome určovaný a systém sa ho snaží dosiahnuť a následne aj udržať.

***Spätná väzba*** - sú to informácie o výkone, výstupe a správaní systému, je to väzba medzi výstupom a vstupom prvku, subsystému alebo systému, ktorá spôsobuje to, že vstup je závislý na výstupe a zaisťuje použitie časti výstupu zo systému ako vstupu do systému.

Na zabezpečenie riadneho fungovania systému je nevyhnutná nejaká forma riadenia resp. regulácie. Podmienkou riadenia je fungovanie spätnej väzby, ktorá poskytuje odpoveď aj na otázku: Je stav alebo správanie systému v súlade s jeho stanoveným cieľom? Pritom rozoznávame:

a) *pozitívnu spätnú väzbu* - dáta zo spätnej väzby uľahčujú a urýchľujú transformáciu

v rovnakom smere ako predchádzajúce výstupy, efekt je kumulatívny a výsledkom je exponenciálny rast až explózia, či exponenciálny pokles, alebo až zánik,

b) *negatívna spätná väzba* - dáta zo spätnej väzby prinášajú výsledok v opačnom smere k predchádzajúcim výsledkom, efekt stabilizuje systém a dochádza k zaisteniu rovnováhy alebo rovnovážneho pohybu, príp. dynamiky v súlade s cieľmi systému.

**Systémová analýza a syntéza**

- ***analýza - redukcionismus*** - je izolovanie a zjednodušenie systému, takže môžu byť analyzované a zdokonalené jeho rôzne prvky a z danej štruktúry určujeme správanie (funkciu),
- ***syntéza - popis vzťahov*** - je popis vzájomného pôsobenia jednotlivých prvkov systému a z požadovanej funkcie (správania) odvodzujeme štruktúru,
- ***analýza a syntéza*** - je myšlienkové alebo faktické rozkladanie celku na súčasti a opätovné spájanie častí v jeden celok

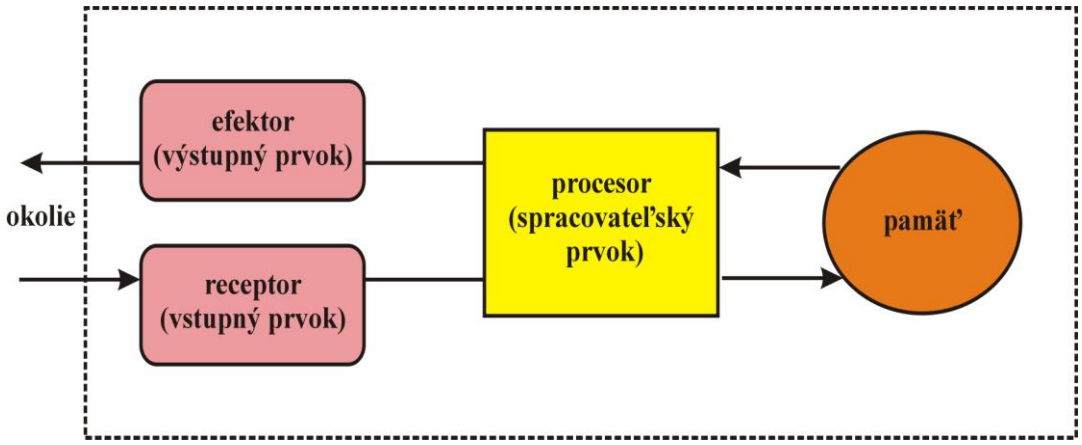
**Zásady správnej analýzy**:

- zachytenie celého rozsahu analyzovaného celku,
- jasné ohraničenie celku i jeho častí,
- vzájomne sa vylučujúce časti,
- neprekrývajúce sa časti,
- rovnaká úroveň veľkosti jednotlivých častí.

**Typické problémy riešené komunikačnými systémami:**

- potreba informácií pre poznanie, pre rozhodovanie, pre realizáciu určitej činnosti,
- zložitosť,
- znovupoužitelnosť,
- automatizácia,
- komunikácia,
- bezpečnosť,
- spoľahlivosť,
- minimalizácia rizika.

Všeobecný model KS by sme mohli znázorniť aj podľa obrázka 7.




Obrázok 7 Všeobecný model KS (vlastný návrh podľa Encyclopaedia Britannica)

**Jednotlivé komponenty modelu umožňujú realizovať základné ciele KS:**

- receptor: získavanie informácií,
- pamäť: ukladanie informácií (ich fixácia v priestore a čase),
- procesor: transformácia (spracovanie) informácií,
- efektor: prenos informácií, sprístupnenie informácií.

**Automatizovaný KS -** je KS fungujúci s podporou IKT. Vďaka tejto podpore môžu byť procesy získavania, spracovania, ukladania a komunikácie informácií realizované čiastočne alebo úplne bez priamej účasti človeka; zvyčajne sú sprevádzané aj digitalizáciou dátovej základne. Vzhľadom k rýchlemu prenikaniu IKT do informačnej praxe a s tým súvisiacim klesajúcim podielom „tradičných“, manuálnych informačných systémov toto označenie a vyčleňovanie automatizovaných informačných systémov do osobitnej kategórie v súčasnosti stráca na význame.

**Cieľ tvorby automatizovaného komunikačného systému:** definovanie štruktúry digitálne uložených dát a vytvorenie programu, ktorý bude realizovať požadované činnosti.

**Prvky KS:**

*Subsystém 1 - ľudia:*

- tvorcovia (autori) informácií,
- používatelia informácií (klienti),
- spracovatelia, správcovia, sprostredkovatelia informácií.

*Subsystém 2 - informácie:*

a) <u>informácie ako ekonomický zdroj:</u>
   - 2 základné typy zdrojov každého ekonomického subjektu:
     1) fyzické - pracovníci, materiál, stroje, zariadenia, peniaze
     2) nehmotné (ideálne, abstraktné pojmy) - dáta, informácie

   Na rozdiel od iných ekonomických zdrojov má informácia určité špecifiká - je expanzívna (rozpínavá), obmedzená iba časom a kapacitou ľudského poznania. Jej povahou je šírenie a skôr sa reprodukuje ako spotrebováva používaním, môže byť len zdieľaná, nie však zameňovaná v transakciách a je komprimovateľná syntakticky aj sémanticky.

b) <u>informácie ako ekonomická komodita („tovar“)</u>

*Subsystém 3 - prostriedky zaznamenávania, uchovávania, zabezpečenia, triedenia, vyhľadávania a šírenia informácií:*

- jazyky,
- IKT (hardvér - PC a periférie, sieťové prvky, softvér),
- pracovné postupy a metódy,
- materiálne zabezpečenie (budovy, serverovne, ...).

*Vstupy (zdroje):*

- dáta, informácie,
- požiadavky, dotazy.

*Výstupy:*

- informačné služby (informácie, odpovede na otázky),
- informačné produkty.

**Funkcie KS** - sú konkrétne procesy podporujúce základné ciele KS.

**Procesy (činnosti, spracovanie)**:

- získavanie informácií,
- spracovanie informácií (evidencia, organizácia, kategorizácia, konverzia, zmena médií, triedenie, vyhľadávanie, agregácia, odvodzovanie nových informácií),
- uloženie informácií (zaznamenávanie, zhromažďovanie na nosiči),
- prenos informácií,
- sprístupnenie informácií (tlač, zobrazenie, šírenie atď.).

**Graf (diagram) -** je nástroj pre zobrazenie výsledkov systémovej analýzy:

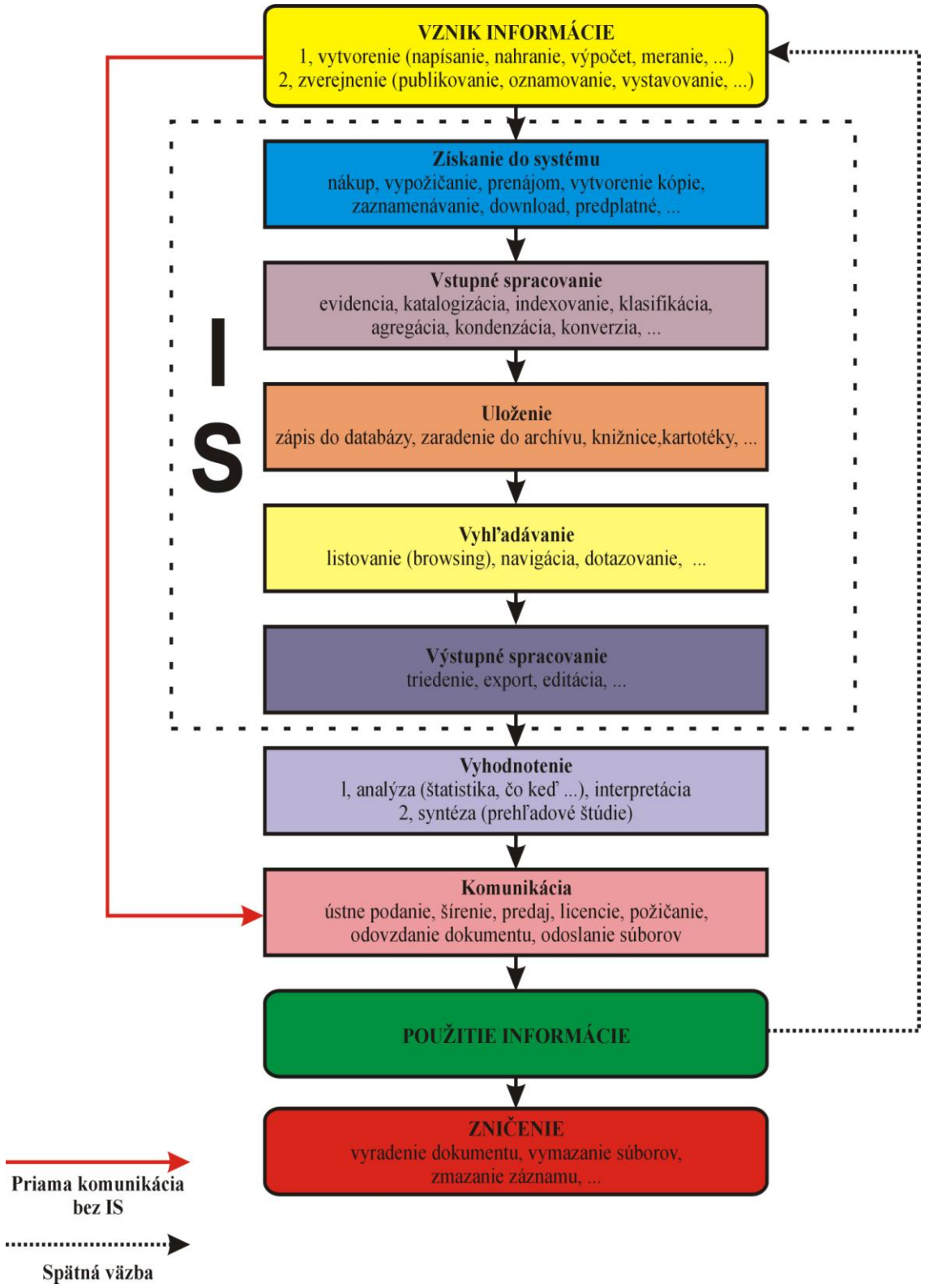
- určitý útvar (rovinný, priestorový), znázorňujúci vzťahy (väzby, relácie) medzi prvkami systému prostredníctvom množiny uzlov a hrán,
- umožňuje aj dynamický, funkčný pohľad na dáta (orientovaný graf).

**Uzol -** pri znázorňovaní grafu v rovine sa uzly zobrazujú spravidla ako body tejto roviny.

**Hrana -** je spojnica dvoch uzlov v grafe, príp. čiara začínajúca a končiaca v tom istom uzle (tzv. slučka); hrany značíme koncovými uzlami.

**Cesta -** je postupnosť nadväzujúcich hrán.

Životný cyklus informácie v informačno-komunikačnom systéme môžeme znázorniť podľa obrázka 8.




Obrázok 8 Životný cyklus informácie v KS (vlastný návrh podľa Page-Jones, 2001)

**Typy grafov:**

a) **súvislý - nesúvislý**

*súvislý graf:* medzi každými dvoma uzlami existuje aspoň jedno spojenie (medzi ľubovoľnou dvojicou uzlov existuje nejaká cesta)

b) **orientovaný - neorientovaný**

hranám je, alebo nie je priradená orientácia (smer)

orientovaný graf (usmernený graf, postupový diagram)

c) **ohodnotený - neohodnotený**

hranám alebo uzlom sú, alebo nie sú priradené hodnoty (vzdialenosť, čas, náklady atď.)

ohodnotená hrana alebo uzol má priradené jedno alebo viac čísel, zvyčajne nezáporných (napr. trvanie, nároky na zdroje)

d) **hranovo definovaný - uzlovo definovaný**

činnosť zobrazujú buď hrany alebo uzly

*hranovo definovaný graf*:

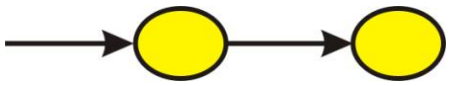
- činnosť znázorňuje hrana
- uzol znázorňuje okamih začatia alebo ukončenia jednej, popr. viacerých činností

*uzlový definovaný graf:*

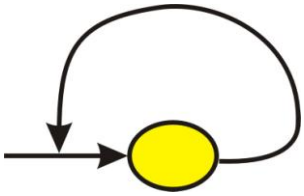
- činnosť znázorňuje uzol
- hrana znázorňuje vzťah medzi jednotlivými činnosťami

**Logické operácie znázorňované v grafoch:**

- **následnosť** - taktiež postupnosť, sekvenčné smerovanie, sekvencia, prechod

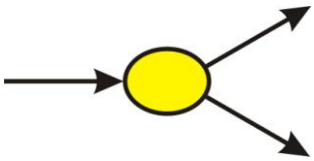


- **smyčka** - taktiež opakovanie procesov, cyklus, rekurzia, iterácia

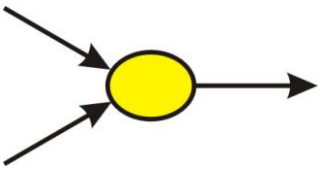


- **vetvenie - 1 vstup, viacej výstupov**
  - vetvenie, rozhodovanie - alternatívny sled činností, výber alternatív, selekcia,

    OR - split, xor - exclusive OR, „buď/alebo“
  - paralelné smerovanie - súčasne prebiehajúce činnosti, AND - split, „a zároveň“



- **spojovanie - viac vstupov, 1 výstup**
  - **spojovanie** - zbiehanie alternatívnych vetiev, OR - join
  - **synchronizácia** - vyústenie paralelných činností do jednej, AND - join

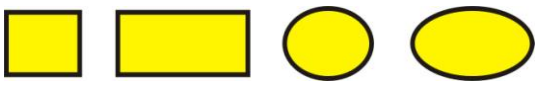


- **uzly** - v hranovo definovanom grafe označujú udalosti - začiatok, alebo koniec činnosti; v uzlovo definovanom grafe označujú činnosti

- **hrany** - v hranovo definovanom grafe označujú vykonanie činnosti (šípka ukazuje smer toku); v uzlovo definovanom grafe označujú vzťah - postupnosť činností

- **dummy(falošná, fiktívna hrana)** - následnosť udalostí za sebou v čase, pričom sa nevykonáva žiadna činnosť a nespotrebovávajú sa žiadne zdroje; znázorňuje, že nasledujúca činnosť (za uzlom 2) môže byť začatá až po dokončení predchádzajúcej činnosti

- **prvok systému (element)** - identifikácia: názov, prípadne vlastnosti (atribúty); znázornenie: štvorec, obdĺžnik alebo kruh, ovál

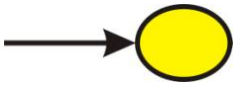


- **subsystém** - znázornenie: štvorec, obdĺžnik alebo kruh, ovál (rovnako ako prvky systému)
- **okolie - prostredie systému (environment)** - znázornenie: slovný popis, štvorec, obdĺžnik alebo kruh, ovál (rovnako ako prvky systému)
- **hranice - rozhranie systému (boundary - interface)** - znázornenie: prerušovaná čiara

- **vzťah (relationship)** - znázornenie: čiara spojujúca prvky

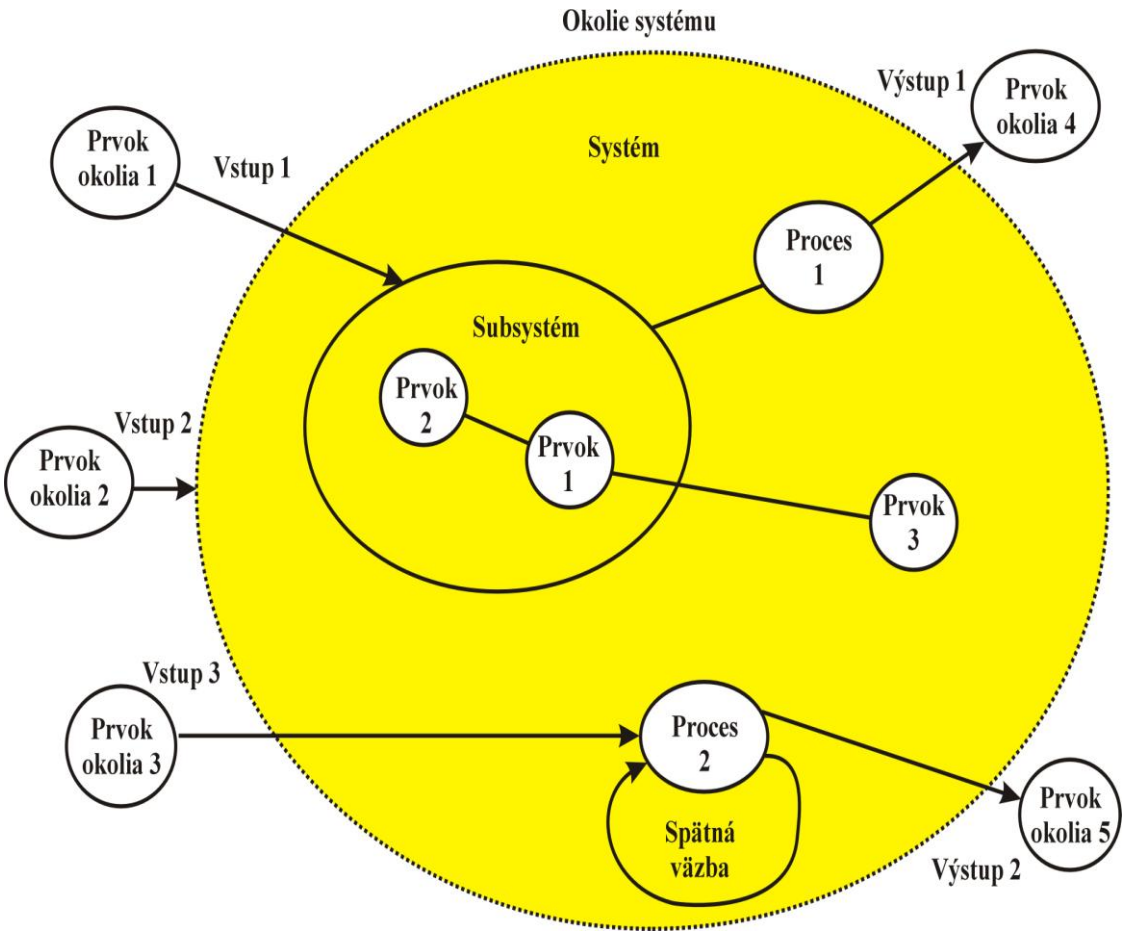


- **vstup (input)** - znázornenie: orientovaná čiara (šípka)



- **výstup (output)** - znázornenie: orientovaná čiara (šípka)

- **proces (process)** - znázornenie: štvorec, obdĺžnik alebo kruh, ovál (rovnako ako prvky systému)




Obrázok 9 Príklad vzťahov medzi prvkami v systéme (vlastné spracovanie)

**Funkčný popis systému - využitie funkčnej (procesnej) analýzy v súčasných KS:**

- *Návrh KS v organizácii*
- *Modelovanie*
- *Dokumentácia procesov v organizácií*
- *Operačná analýza/výskum (manažérska veda)* - hľadá optimálne riešenie zložitých problémov rozkladom zložitého procesu na čiastkové činnosti
- *Sieťová analýza* - grafická metóda operačnej analýzy (znázornenie závislostí čiastkových činností pomocou sieťového grafu)
- *Príprava podkladov pre počítačový program, ktorý bude procesy podporovať alebo samostatne vykonávať (automatizovať)* - správa podnikových procesov (BPM, workflow management)
- *Reengineering procesov podniku* (BPR - business process reengineering)
- *Tvorba „znalostnej bázy“ podniku* - implicitné (tacitné) vedomosti o procesoch

(best practices) sa prevádzajú na explicitné - vyjadrené znalosti v modeloch a v dokumentoch.

**Vývojový diagram (flowchart) je:**

- orientovaný, súvislý, uzlovo definovaný zvyčajne neohodnotený sieťový graf,grafické znázornenie definície, analýzy alebo metódy riešenia problému, v ktorom sú symboly používané pre znázornenie operácií, dát, toku, zariadení atď.,nástroj pre modelovanie procesov v systéme - vyjadruje logickú štruktúru procesu alebo operácie, t. j. súvislosti (vzťahy) jednotlivých činností.**Využitie vývojových diagramov v projektovaní komunikačných systémov:**

definovanie tzv. algoritmov

**algoritmus -** je jednoznačný a vyčerpávajúci popis, ktoré operácie a v akom poradí sa majú vykonať, aby bolo možné úlohu vyriešiť

1. ***vývojový diagram programu*** - je návrh štruktúry počítačového programu (bloková schéma),
2. ***dynamický model systému*** - je zobrazovanie algoritmov transformačných procesov pri modelovaní systémov; ilustruje časti (prvky) a toky KS (napr. diagram aktivít, sekvenčný diagram a stavový diagram v UML),
3. ***procesná analýza, workflow management***

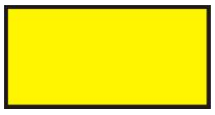
   základná štruktúra:

vstup → spracovanie → výstup

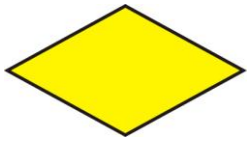
**Základné symboly vývojových diagramov:**

**A. symboly spracovania:**

- **spracovanie (akcia, proces)** - akákoľvek operácia, ktorej výsledkom je transformácia dát



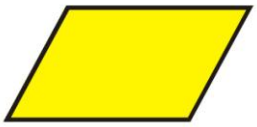
- **rozhodovanie (vetvenie)**



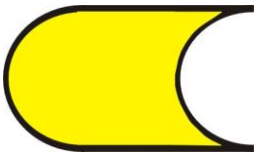
- **paralelný režim**

**B. symboly dát:**

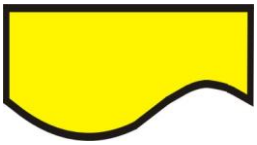
- **vstup / výstup dát - bez určenia média**



- **pamäť, uloženie dát - bez určenia média**



- **tlačený vstup / výstup dát (dokument)**



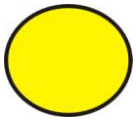
**C. symboly spojenia:**

- **tok dát alebo riadenie**

**D. zvláštne symboly:**

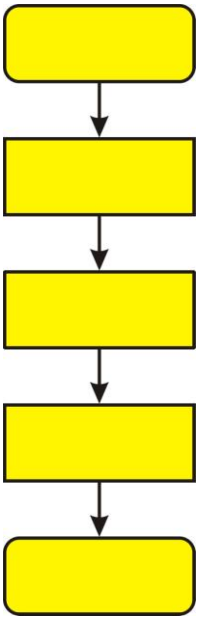
- **medzná značka**

- **spojka**

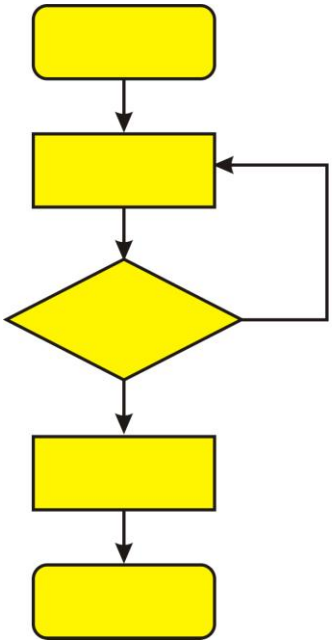


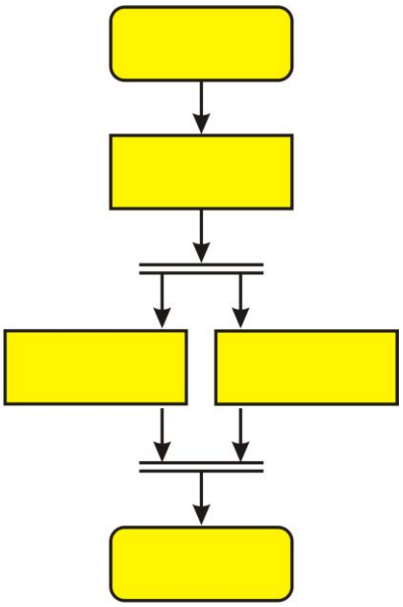
**Zobrazované logické operácie vo vývojových diagramoch:**

- **následnosť (postupnosť)**



- **smyčka**
- **vetvenie (OR, AND)**
- **spojovanie (OR, AND)**





**Modelovanie, Model podnikových procesov (tiež mapa procesov, procesná mapa) je:**

- konceptuálny (implementačne nezávislý) model všetkého, čo sa v organizácii deje,
- obdoba organizačnej schémy (organizačná schéma je statická, procesné mapa je dynamická),
- zobrazenie procesov a ich interakcií v systéme od počiatočného bodu po jeden, alebo viac koncových bodov. Dôraz sa kladie na zobrazenie vzťahov procesov a objektov s nimi súvisiacich.

**Využitie modelu:**

- podklad pre návrh KS v organizácii,
- podklad pre reengineering procesov v organizácii,
- podklad pre procesný manažment (procesné riadenie - business process management, workflow management),
- dokumentácia procesov v podniku (napr. pre certifikát kvality podľa ISO 9000).

**Organizačný proces:**

- skupina aktivít, ktorých vykonaním sa pridá hodnota pre zákazníka,
- vzájomne prepojené čiastkové činnosti, ktoré vo svojej postupnosti transformujú vstupy na požadované výstupy,
- tok práce, postupujúce od jedného pracovníka (oddelenia) k druhému.

**Typy procesov v organizácii:**

- kľúčové - pomocné,
- manuálne - automatizované (IT - intensive),
- kolaboratívne (assembly line) - individuálne (once and done).

*Kľúčový, primárny, realizačný („core“) proces* - v praxi to je produkcia výrobkov, alebo služieb, kvôli ktorým organizácia existuje - hodnota sa porovnáva potrebami zákazníkov.
Výstup: hodnota pre zákazníka.
*Pomocný, podporný (supporting) proces* - zabezpečuje činnosti a zdroje pre kľúčové procesy (riadenie, predaj a marketing, finančné riadenie, personalistika).
Výstup: hodnota pre kľúčový proces.

**Atribúty podnikových procesov**:

- *vstupy:* predvolené zdroje (suroviny, materiál, kapacity strojov, ľudia, kapitál, technológie, dáta, informácie, znalosti), dodávatelia, výstupy z iných procesov,
- *výstupy:* konečné výsledky slúžiace zákazníkovi (výrobok alebo služba),
- stakeholder: zainteresovaný subjekt
  - vlastník: osoba zodpovedná za efektívnosť daného procesu,
  - zákazník (interný, externý): preberá výstup procesu,
  - ďalší: investori, konkurencia, regulátori, management, zamestnanci, dodávatelia, partneri, predajcovia atď.

**Základné konštrukty používané v softvérových nástrojoch na tvorbu vývojových diagramov - napríklad notácie podľa modulu v BPMN, Business Process Model PowerDesigneru a ďalších:**

**začiatok**

**koniec**

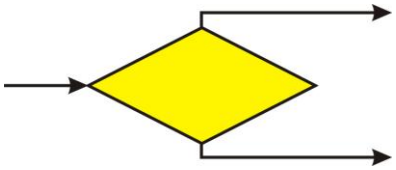
**proces**

**zdroj**

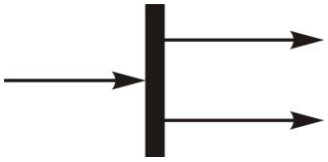


**tok**

**rozhodovanie**

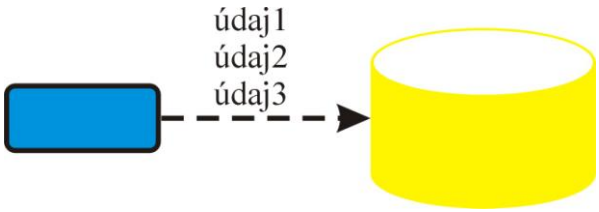


**synchronizácia**

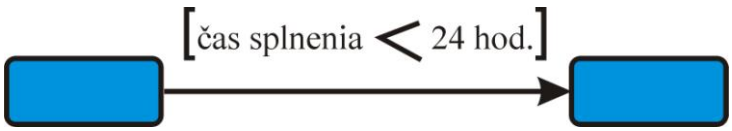


**formát správy**

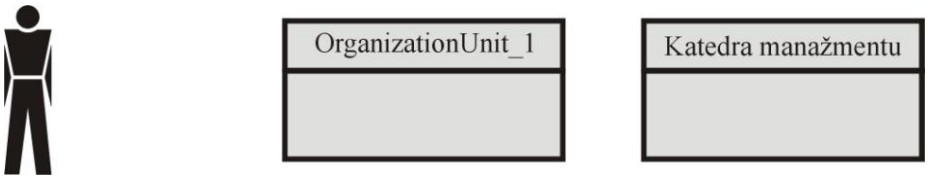
**dátový tok**




**podmienka**




**organizačná jednotka**



**Workflow manažment** - je počítačová podpora procesov v organizácii, je to automatizácia podnikového procesu vcelku alebo jeho časti, v ktorej priebehu sú dokumenty, informácie alebo úlohy odovzdávané od jedného účastníka k druhému v súlade s procesnými pravidlami tak, aby sa dosiahlo, alebo prispelo k plneniu podnikových cieľov (http://web.sks.cz).

**Workflow management system** - je systém, ktorý podrobne definuje, spravuje a realizuje „toky práce“ prostredníctvom programu, ktorého operácie sú riadené počítačovou reprezentáciou - modelom - logiky procesu poskytovaných služieb pri automatizácii procesov v organizácii a zahŕňa:

- administráciu,
- smerovanie informácií,
- definovanie úloh a pravidiel,
- monitorovanie - sledovanie priebehu jednotlivých krokov procesov,
- kontrolu - napr. dodržiavanie stanovených termínov,
- pri prechode do nového kroku sú automaticky odovzdávané aktivity ďalšiemu užívateľovi a v jeho neprítomnosti jeho zástupcovi,
- generovanie a distribúciu elektronických upovedomení,
- spúšťanie IT nástrojov a aplikácií (napr. textový, tabuľkový editor; e-mail a pod.)

**Model pre workflow management** - je to špeciálna aplikácia vývojových diagramov - spojenie vývojového diagramu a organizačnej schémy. V tomto modeli definujeme:

- procesy (štruktúrované alebo čiastočne štruktúrované procesy) a tie sa ďalej členia na kroky procesov (worksteps) a aktivity (činnosti, ktoré sa vykonávajú v kroku procesu),
- následnosť alebo rovnobežnosť krokov procesu,
- podmienky, za ktorých dôjde ku správnemu kroku,
- lehoty na realizáciu aktivít,
- priradenie užívateľov (pracovníkov) jednotlivým krokom,
- programy, ktoré sa spúšťajú počas jednotlivých krokov,
- dáta, potrebné na realizáciu jednotlivých krokov,
- dáta produkované jednotlivými krokmi.

**Najčastejšie uplatnenie:** modelovanie obehu dokumentov - t. j. informácií o realizovaných alebo plánovaných činnostiach organizácie.

**Spoločné črty všetkých metodík projektovania komunikačných systémov:**

1. *viac oddelených pohľadov na KS - separované oblasti*:
   - dáta - procesy
   - celok - časť (detail)
   - niekoľko úrovní hierarchie
   - konkrétne (realita) - abstraktné
   - všeobecné / model
   - „architektúry“ (napr. princíp troch architektúr):

     A. model reality,

     B. model KS,

     C. implementovaný KS
2. *modelovanie (reality aj informačného systému)*
3. *bstrakcia reality*

**Základné prvky modelov KS**:

- ***Entity, triedy***

  *entita* - je akýkoľvek reálny objekt, prvok, ktorý nás zaujíma

  *trieda* - združuje entity sa spoločnými vlastnosťami; je to popis množiny objektov, ktoré zdieľajú rovnaké atribúty, operácie, vzťahy a sémantiku

  *inštancia* - je konkrétny výskyt určitej entity (triedy, typu) naplnený hodnotami, ktoré vyjadrujú jeho stav alebo zmenu stavu, je to objekt v IS
- ***Atribúty***

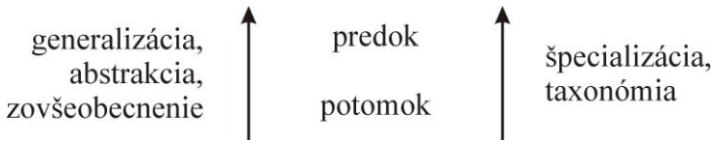
  *atribút* - je charakteristika (vlastnosť) entity alebo triedy; definuje hodnoty, ktorých môže nadobúdať jej inštancia
- ***Operácie, funkcia, procesy***

  *operácia* je akákoľvek udalosť, ktorá nás zaujíma

  *funkcia* združuje viac operácií

- ***Vzťahy*** - sú sémantické spojenia medzi dvoma alebo viacerými prvkami modelu a poznáme:
  1. *hierarchické vzťahy (*generické, *rododruhové)*

     **dedičnosť, tiež generalizácia (abstrakcia, zovšeobecnenie) - špecializácia** - je to vzťah, ktorý definuje jednu entitu prostredníctvom inej entity), pričom entita - potomok dedí vlastnosti svojho predka




  2. *nehierarchické vzťahy*

     kauzálne (príčina - dôsledok), partitívne (celok, kontajner - časť), finálne (účel - prostriedok)

     **Asociácia** - je všeobecný sémantický vzťah medzi prvkami modelu, ktorý špecifikuje spojenie medzi ich inštanciami.

     **Agregácia (kolektivizácia), kompozícia** - je forma asociácie, ktorá vyjadruje vzťah celok - časť; celok iba združuje prvky (jednotlivé komponenty), nedefinuje však ich spoločné charakteristiky.

     **Agregácia** - element môže „prežiť“ svoj kontajner, prípadne sa môže stať súčasťou iného kontajnera. Príklad: Zamestnanec (časť) - Oddelenie (celok).

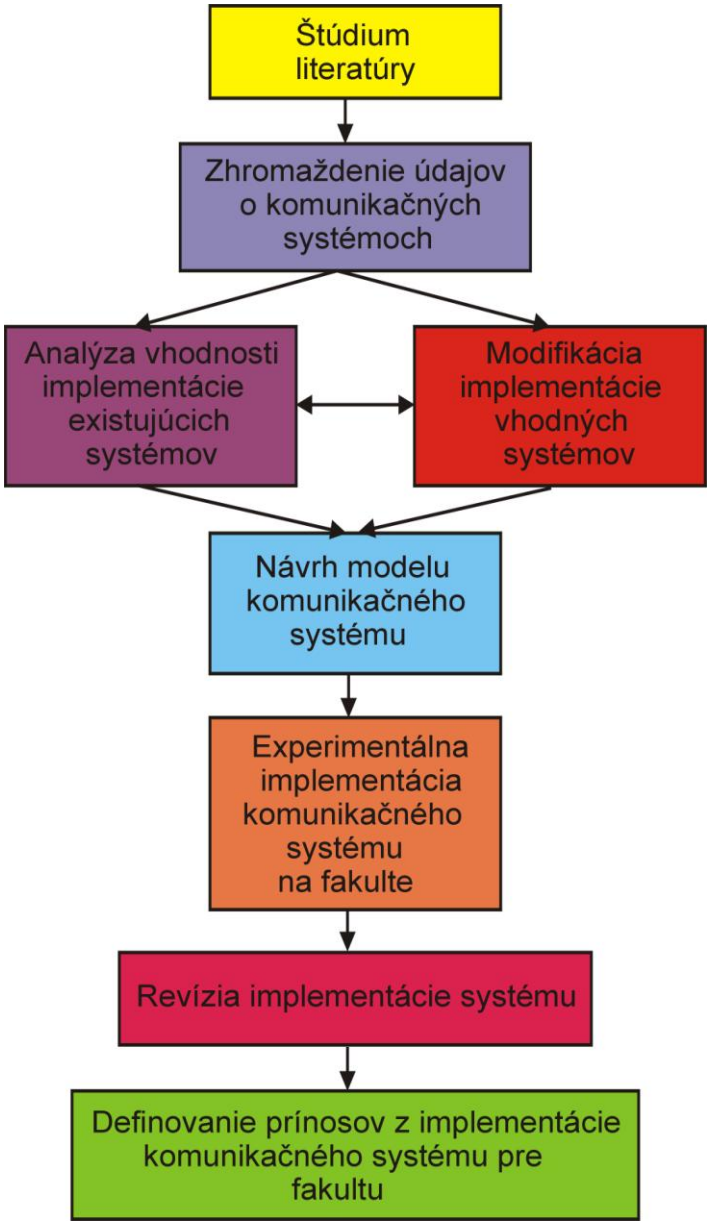
     **Kompozícia** je silnejšia väzba - zrušením kontajnera automaticky zrušíme aj obsiahnutý element; daný element môže byť súčasťou práve jedného kontajnera. Príklad: Okno (časť) - Miestnosť (celok).

     **Závislosť** je vzťah medzi dvoma elementmi modelu, v ktorom zmena jedného (nezávislého) elementu ovplyvní druhý (závislý) element; vzťah, v ktorom jeden element využíva služby alebo zariadenia iného elementu a zahŕňa aj postupnosť, teda následnosť (napr. v čase).

     **Násobnosť (multiplicity)** - ukazuje počet hodnôt (kardinalít), ktorých môže nadobúdať príslušná úloha vo vzťahu (rola označuje úlohu, ktorú má objekt videný objektom z druhej strany vzťahu).

## 3.3 Postup riešenia dizertačnej práce

Postupnosť riešenia dizertačnej práce zobrazuje obrázok 10.




Obrázok 10 Kroky riešenia dizertačnej práce (vlastný návrh)

**Pred tvorbou samotného modelu komunikačného systému, sme si museli popísať jednotlivé typy prístupov v návrhu, (projektovania) KS:**

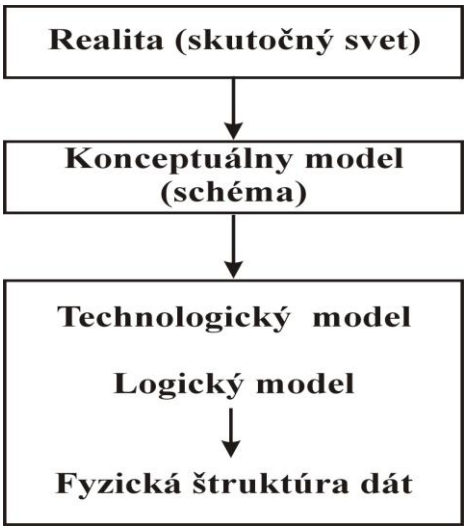
- **zdrojový - funkčný**

  *zdrojový prístup:* stanovenie dátového modelu reality a nového systému (zaujímajú nás používateľské potreby), tvorí sa informačná (dátová) analýza

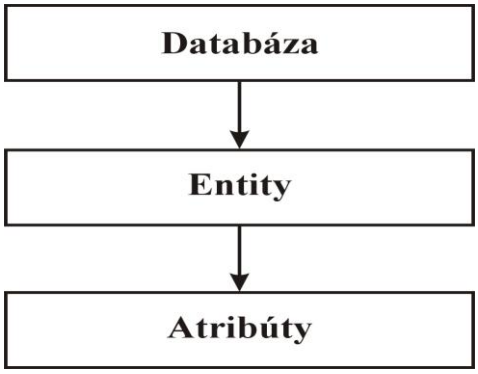
  *funkčný prístup:* stanovenie cieľov, ktoré by mal nový systém spĺňať (zaujíma nás fungovania systému), vykonáva sa funkčná analýza

- **top-down - bottom-up**

  *top-down (zhora nadol)* je to postup od abstraktného ku konkrétnemu




  *bottom-up (zdola nahor)* je postup od konkrétneho k všeobecnému




- **štruktúrovaný - objektovo orientovaný**

  *štruktúrovaný prístup*: aplikácia má podobu hierarchie funkcií, ktorá je realizovaná štruktúrovanými programami

Štýl práce:

**AKCIA ⟶ OBJEKT**

*objektovo orientovaný prístup*: funkcie sú realizované pomocou „programových fragmentov“ nad stabilnými štruktúrami databáz

Štýl práce:

**OBJEKT ⟶ AKCIA**

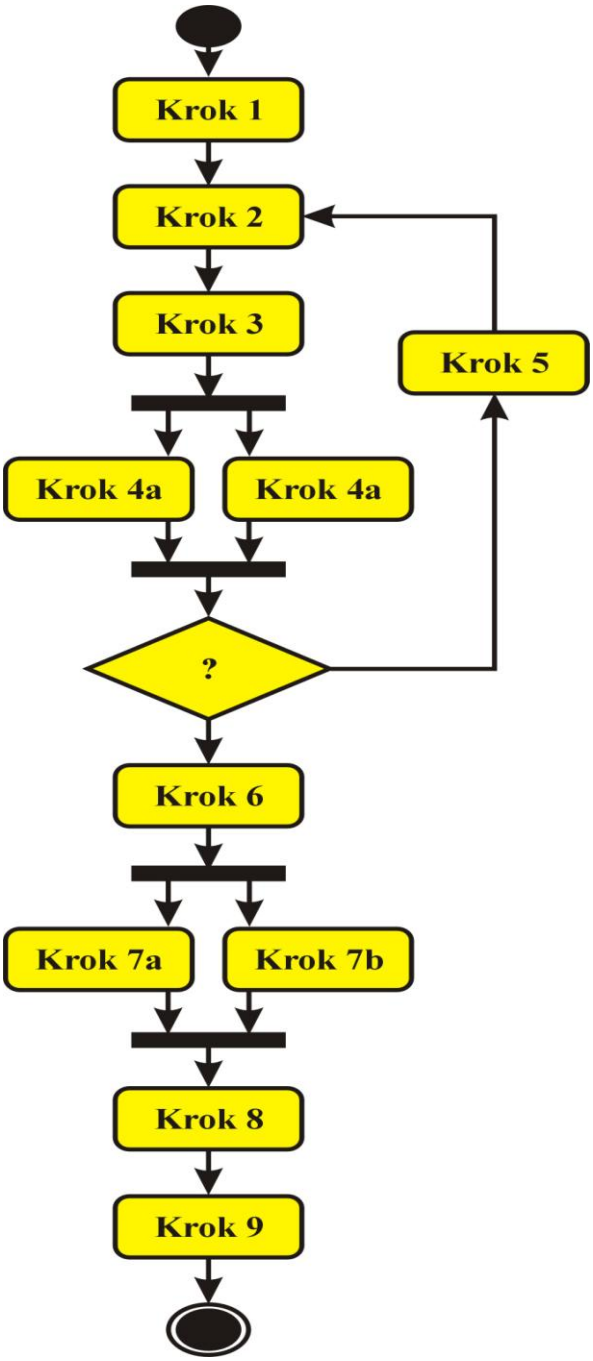
- **rigorózne (striktné) - agilné (light)**
- **plánovacie - konštrukčné - implementačné**

**Entitno - relačný model (ERA)**

alternatívne názvy: ERA, ERD, E-R, ER model (schéma, diagram) - je to množina pojmov slúžiacich k statickému popisu aplikácie na konceptuálny úrovni, je to model popisujúci objekty, ktoré nás zaujímajú, ich vlastnosti a ich vzájomné vzťahy.

**ERA diagram** je grafický prostriedok pre analýzu a zobrazenie dátového modelu systému.

Metodika tvorby ERA diagramu je zobrazená na obrázku 11.




Obrázok 11 Metodika tvorby ERA diagram (vlastný návrh podľa Analecto, 2004)

**Slovne by sme mohli návod na postupnosť krokov popísať takto:**

1. Zvoľte jednu primárnu entitu zo špecifikácie požiadaviek.
2. Určite atribúty, ktorých hodnoty sa majú pre túto entitu zaznamenávať. Označte prípadné kľúče, teda identifikátory, a vytvorte vzorové údaje.
3. Popíšte slovne navrhnutú entitu, jej atribúty a kľúče.

4a. Preverte funkčné vzťahy (závislosti) atribútov a v prípade potreby entitu normalizujte.

4b. Preverte atribúty navrhnutej entity (pokiaľ možno v spolupráci s používateľom) a zistite, či je potrebné zaznamenávať informácie o jednom alebo viacerých atribútoch v novej samostatnej entite.

5. Ak je vhodné vytvoriť ďalšiu entitu, zakreslite ju do diagramu a vráťte sa na krok 2.

6. Spojte entity vzťahmi, ak tieto existujú. Popíšte slovné vzťahy medzi entitami z oboch strán.

7a. Preverte zoznam atribútov a určite, či niektoré z nich potrebujú byť identifikované prostredníctvom dvoch poprípade viacerých entít. Ak áno, umiestnite atribút na príslušný vzťah, ktorý spája dané entity.

7b. Preverte, či v diagrame nemáte „slučky“, ktoré môžu indikovať nadbytočné vzťahy. Ak je vzťah skutočne redundantný, odstráňte ho.

8. Vytvorte vzorové údaje.

9. Predveďte navrhnutý model, teda diagram, aj slovný popis užívateľovi. Ak je to potrebné, upresnite diagram.

**Možné prístupy k tvorbe ERA diagramu:**

- *zdola nahor (bottom-up)* - najprv zostavíme zoznam atribútov, potom ich zoskupíme do entít
- *zhora nadol (top-down)* - najprv definujeme entity, potom ich naplníme atribútmi

**Pravidlá návrhu „správnych“ ERA diagramov (Analecto, 2004):**

- Zobrazujeme iba dáta a ich vzťahy, žiadne procesy.
- Každý atribút zobrazujeme iba raz - cieľom je štruktúrovať zoznam atribútov, nie napríklad znázorňovať prepojenie v relačnej databáze.
- Zobrazujeme zoskupenie dát pre účely databázy, nie pre účely výstupov (kombináciu atribútov z rôznych entít a prípadné duplicity realizujú až pohľady - formuláre alebo zostavy).
- Zobrazujeme iba perzistentné (trvalé) dátové objekty - dáta, ktoré chceme vygenerovať výpočtami a agregáciami, nemodelujeme.

- Zobrazujeme iba nevyhnutne nutné vzťahy, t. j tie, ktoré k niečomu využijeme a nezobrazujeme: odvodené vzťahy, kruhové závislosti (slučky), redundantné vzťahy.

**Diagram dátových tokov (DFD) data flow diagram -** je grafický prostriedok návrhu a zobrazenia funkčného modelu systému.

**Funkčný model:**

- pohľad na realitu ako na súhrn neustále vznikajúcich rôznych udalostí,
- popis procesov a ich nadväzností,
- opis procesov transformácie informácie a ich vzájomných vzťahov.

**Udalosť - stimul - reakcia**

***Udalosť*** - je to, čo nastane a systém na to musí reagovať, je to stimul, ktorý spúšťa spracovanie vo vnútri systému.

***Typy udalostí:***

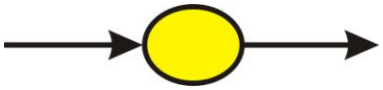
- príchod dát do systému z okolia,
- udalosť spojená s časom,
- riadiaca udalosť - vyžiadanie reakcie riadiacim prvkom vo vnútri systému.

***Stimul*** - je dátový tok, ktorý oznamuje systému, že udalosť nastala.

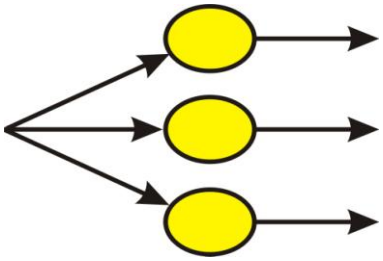
***Reakcia*** - je výstupný dátový tok do okolia a zároveň uloženie dát v systéme.

***Typy reakcií na udalosť:***

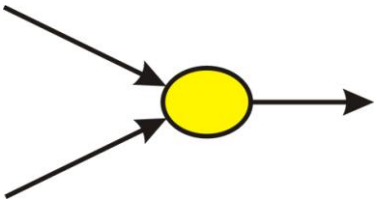
- jedna udalosť - jedna reakcia (proces)



- jedna udalosť - rôzne reakcie (procesy)

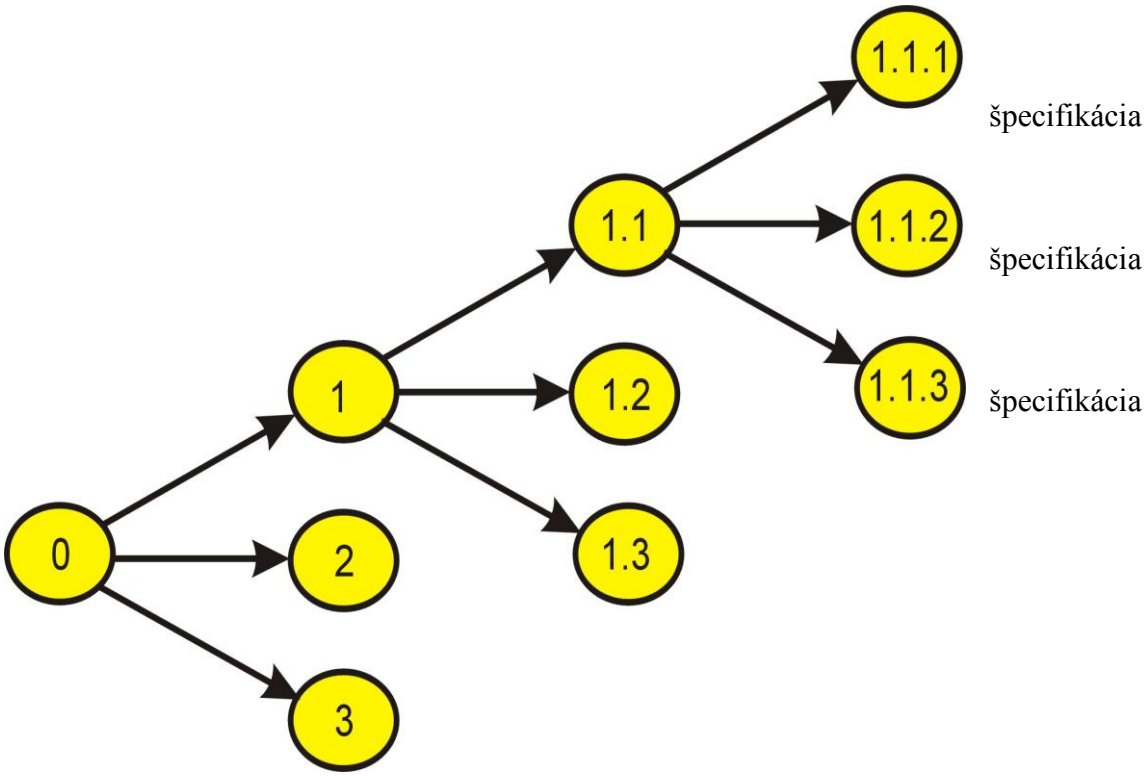


- viac udalostí - rovnaká reakcia (proces)



**Hierarchický princíp tvorby DFD (top-down):**

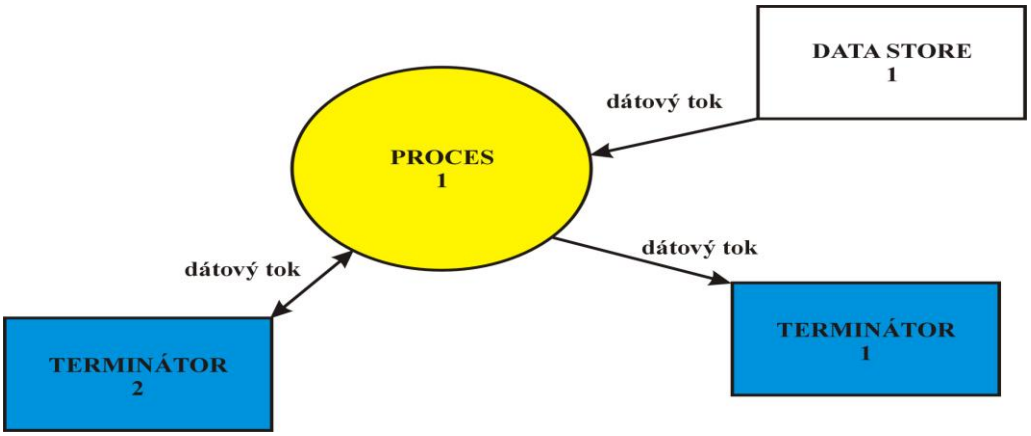
| | | | |
|---|---|---|---|
| **kontextový diagram (úroveň -1)** | **úroveň 0** | **úroveň 1** | **úroveň 2** |



**Kontextový diagram tvoria:**

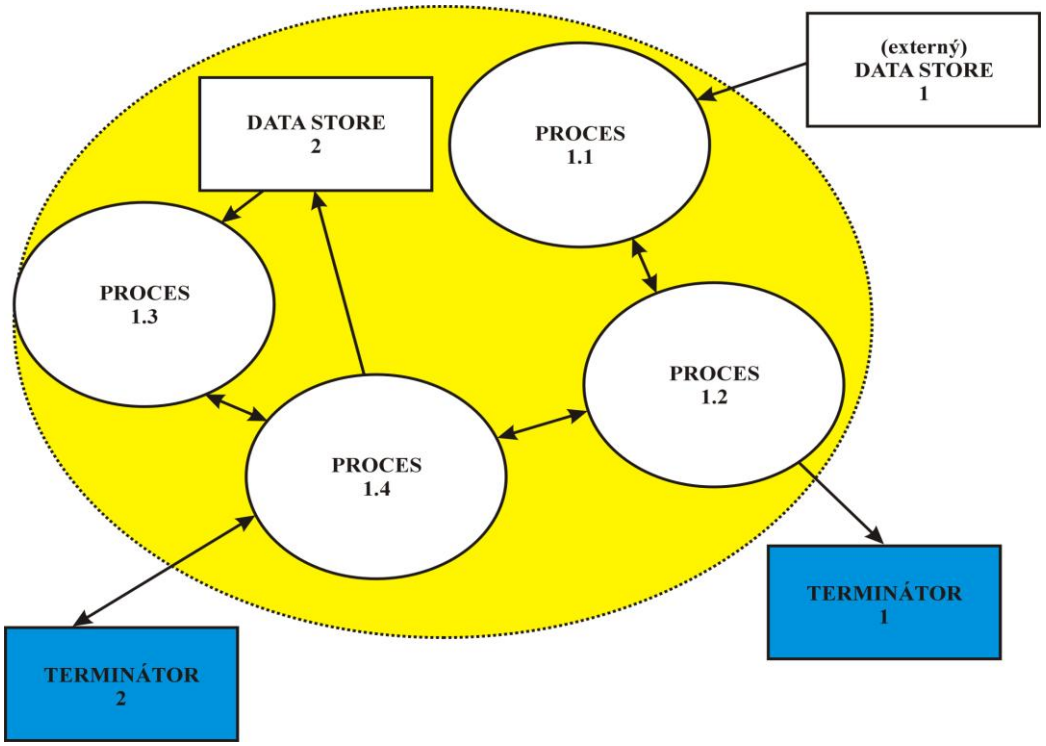
- ľudia, organizačné jednotky, systémy, ktoré s modelovaným systémom komunikujú,
- dáta, ktoré do systému prichádzajú z okolia a ktoré musí spracovať,
- dáta, ktoré systém produkuje,
- datastory zdieľané systémom a terminátory - zdroje alebo miesto určenia dát mimo definovaný systém,
- zoznam udalostí, na ktoré musí systém reagovať.

Kontextový diagram môžeme znázorniť aj podľa obrázka 12.




Obrázok 12 Kontextový diagram (vlastný návrh)

Diagram dátových tokov môžeme znázorniť podľa obrázka 13.




Obrázok 13 Diagram dátových tokov (vlastný návrh)

**Odporúčaný postup tvorby DFD:**

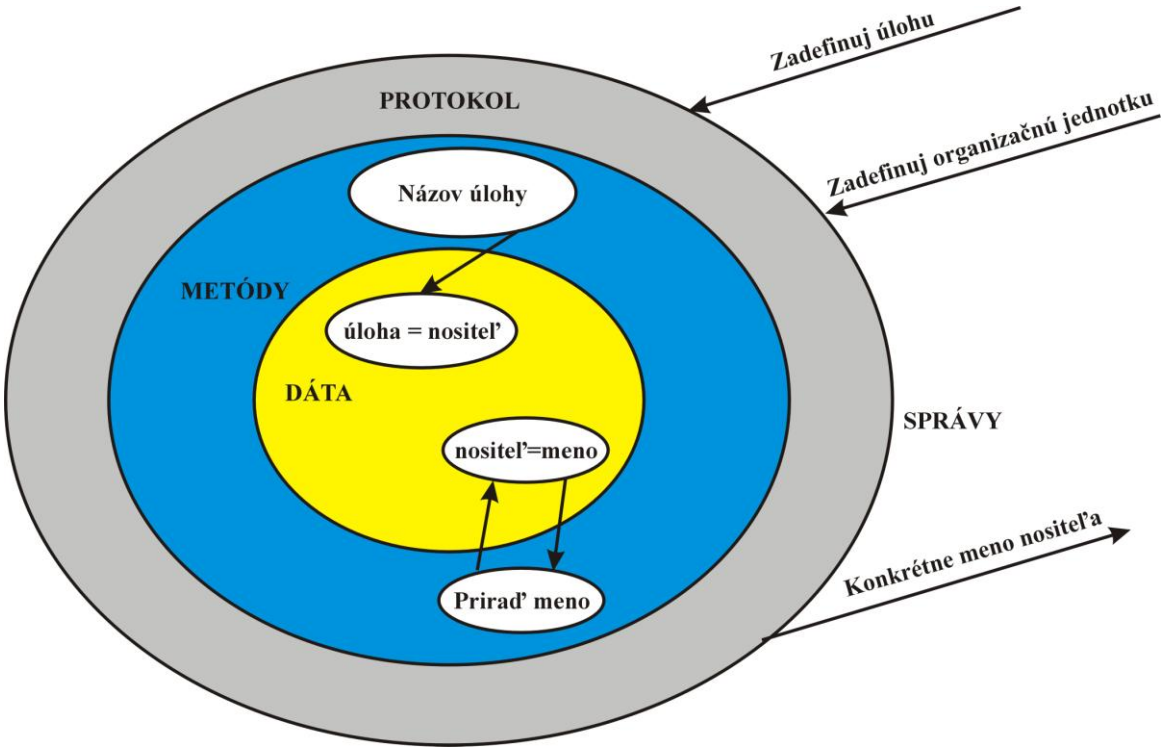
1. vytvoriť kontextový diagram,
2. vytvoriť zoznam udalostí,
3. pre každú udalosť vytvoriť proces (proces = reakcia na udalosť),
4. každý proces pomenovať podľa reakcie na udalosť,
5. ku každému procesu doplniť vstupy, výstupy, prípadne DataStore, pritom je potrebné dopredu vedieť aké dáta proces potrebuje a čo je jeho výstupom,
6. kontrola konzistencie - to znamená, že všetky vstupy a výstupy z kontextového diagramu sa musia objaviť v DFD.

**Objektová metodológia**

***Objekt*** - je čokoľvek reálne či abstraktné (vec, entita) s jasne definovanou rolou v danom kontexte, o čom uchovávame údaje a metódy, ktoré s týmito údajmi manipulujú (Buchalcevová, 2009). Je to samostatná nedeliteľná entita, vie o sebe, kto je, objekt pozná sám seba (má atribúty naplnené konkrétnymi hodnotami), robí konkrétne „niečo“ a má na to metódy, pozná možnosti svojho použitia a všetky prípustné operácie - má však aj

zodpovednosť, vie spracovávať správy a následne ich aj distribuovať. Každý objekt má 3 charakteristiky: stav, identitu a správanie - teda množinu povolených prechodov medzi stavmi.

Príklad objektovej metodológie by sme mohli znázorniť podľa obrázka 14.




Obrázok 14 Príklad objektovej metodológie (vlastný návrh)

**Princípy objektovo orientovaného prístupu:**

- sebaidentifikácia objektov - objekt sám o sebe vie, kto je, takže pamäť obsahujúca objekty obsahuje aj informácie (v objektoch) o štruktúre ich obsahu (t. j. o atribútoch a operáciách);
- zapuzdrovanie objektov (encapsulation) - zoskupenia atribútov a operácií do štruktúry objektov, na ktoré sa dá ďalej odkazovať jediným názvom a v ktorých operácie predstavujú jediný nástroj pre prístup k hodnotám atribútov a ich zmenám; dáta sú chránené (zapuzdrené - zabalené) v schránke (shell), ktorá obsahuje metódy daného objektu. To má za následok obmedzenie možností zmien takto chránených dát;
- klasifikácia (classification) - je združovanie objektov s podobnými vlastnosťami do tried;

- dedičnosť (inheritance) - je tiež označované ako hierarchia, zovšeobecnenie, generalizácia a teda špecializácia, je to tiež vzťah predok (superclass, Bázová trieda, nadtyp) - potomok (subclass, odvodená trieda, podtyp), ale je to aj mechanizmus umožňujúci tvorbu nových objektov (resp. tried) na základe objektov už existujúcich. Novo vytváraný objekt dedia všetky vlastnosti objektu už existujúceho a nový objekt môže navyše pridať vlastnosti nové, alebo môže podľa potreby niektoré z existujúcich vlastností modifikovať (upraviť implementáciu zdedených metód);
- polymorfizmus (polymorphism) - mnohotvárnosť - je rôzna odozva rôznych objektov na prijatie rovnakej správy (1 správa, 2 rôzne objekty, 2 rôzne metódy).

**Postup pri objektovej analýze:**

1. identifikácia objektov,
2. definícia atribútov a metód objektov,
3. identifikácia tried, zoskupenie objektov do tried,
4. opis vzťahov medzi triedami.

**Základné prínosy objektovo orientovaného prístupu:**

- podpora komplexných objektov (complex, composite objects) - skladanie objektov - zložitý objekt sa skladá z jednotlivých komponentov, čo sú opäť objekty,
- možnosť opakovaného použitia (reuse, reusability) objektov - je to vlastnosť, ktorá umožňuje raz vytvorený objekt použiť vo viacerých súvislostiach - v podsystémoch, aplikáciách a podobne.

**Unified Modeling Language (UML)** - je štandardizovaný jazyk pre tvorbu diagramov v objektovo orientovaných modeloch analýz a návrhov KS. Nie je viazaný na konkrétnu metodiku. Základným účelom UML je umožniť a uľahčiť komunikáciu ako v tíme projektantov, tak i medzi projektantom a obstarávateľom.

Umožňuje modelovať:

- objekty (entity),
- triedy,

- atribúty,
- operácie (funkcie),
- vzťahy.

***Základné prvky jazyka UML:***

- reťazec,
- ikona - grafický symbol,
- 2D symbol - ikona s textom,
- spojka (path).

***Doplnky UML notácie:***

- package (balíček, balík, zväzok, modul, subsystém) - logická skupina elementov modulu,
- stereotyp doplnenie ďalšieho významu štandardnému elementu prostredníctvom kľúčových slov (keywords).

***Príklady:***

| | |
|---|---|
| závislosť (dependency) | <<Include>> <<extend>> |
| balík (package) | <<subsystem>> |
| trieda (class) | <<type>> |

**Základné súčasti UML:**

- UML Notation Guide - notácia UML (grafický syntax)
- UML Semantics - metamodel UML (sémantika)
- OCL - Object Constraint Language - textový jazyk pre popis obmedzení modelov
- Špecifikácia prevodu do výmenných formátov (CORBA IDL, XML DTD)

***Notácia UML - typy diagramov:***

Jeden model možno zobraziť viacerými diagramami, čo zaisťuje viac pohľadov na rovnaký systém:

*diagram tried* - model statickej štruktúry systému

*Use Case diagram* - model funkcionality systému

*sekvenčný diagram* - model časovej dynamiky vnútri užívateľského scenára - Use Case

*diagram aktivít* - model dynamiky jednotlivých Use Case a operácií v triedach

*diagram spolupráce* - model interakčnej dynamiky vo vnútri Use Case

*stavový diagram* - model životného cyklu objektu

*diagram komponentov* - model komponentov a ich spolupráca

*diagram nasadenia* - model rozloženia komponentov príbehu systému

**Uplatnenie princípov objektovo orientovaného prístupu v UML diagramoch (Horný, 2000):**

- **skladanie objektov:** agregácia alebo kompozícia (diagram tried); extend (Use Case diagram); composite activity (diagram aktivít); package (všetky modely)
- **opakované použitie objektov:** include (Use Case diagram); dedičnosť (diagram tried); interface - rozhranie (diagram tried).

**Riadenie a plánovanie projektu KS**

***Projekt je:***

- plán - návrh rozsiahlej jednorazovej operácie
- samotná rozsiahla jednorazová operácia - jej priebeh a riadenie

***Charakteristiky projektu:***

- unikátnosť, prípadná neopakovateľnosť;
- presne stanovený cieľ;
- stanovené obmedzenia - čas, zdroje, náklady;
- spravidla obsahuje aj riziko neúspechu.

***Projekt (projektovanie) KS*** - je to riadený proces vyvolaný za účelom obstarania alebo adaptácie (zmeny) KS. Zvyčajne sa člení na etapy (napr. analýza požiadaviek, návrh systému, implementácia, dokumentácia, zaistenie kvality), zvyčajne končí odovzdaním do rutinnej prevádzky.

***Trojimperativ riadenia projektu (triple constraint)*** - kľúčové projektové ciele, ktoré musia byť dosiahnuté súčasne:

1. ČO - špecifikácia prevedenia - čo sa má urobiť
2. KEDY - časový plán - do kedy sa to má urobiť
3. ZA KOĽKO - finančný rozpočet - za koľko sa to má urobiť

Úspešný projekt je ten, ktorý dosiahol stanoveného cieľa:

- v plánovanom čase;
- s plánovanými nákladmi;
- s pridelenými zdrojmi;
- bez negatívnych dopadov.

***Projektovanie = etapy + kontrolné body + dokumentácia***

***Etapy*** (fázy, aktivity, úseky) - skladajú sa z jednotlivých činností (krokov). Pre každú etapu definujeme:

- obsah a cieľ (PREČO);
- produkty (ČO);
- vstupy a výstupy (ZÁVISLOSTI);
- techniky (AKO);
- metriky (KOĽKO, KEDY, KVALITA);
- roly, zodpovednosť (KTO).

***Kontrolné body sú:***

- koncové body nejakých konkrétnych aktivít (činností)
- nasledujú po každej etape - rozhoduje sa, či pokračovať alebo nie v projekte

forma: *checkpoints* - kontrolné body

*milestones* - míľniky (čiastkové ciele)

*deliverable* - „k odovzdaniu“: zreteľný (hmatateľný) výstup nejakej aktivity (činnosti)

***Dokumentácia*** - je prostriedok vyjadrenia, prezentácia projektu - každá etapa sa začína a končí dokumentom.

Typy dokumentácie:

- *užívateľská* - manuály, návody, help pre koncového užívateľa (k používaniu IS) a pre administrátora (na prevádzkovanie KS);
- *projekčné* - technická dokumentácia - modely, špecifikácia dát, architektúra systému);
- *manažérska* - zmluvy, plány projektu, záznamy z porád, záznamy z kontrol a oponentúry.

**Typy projektov KS** - možnosť rôznej kategorizácie za použitia rôznych uhlov pohľadu na projekt - aký systém, čo, ako, pre koho sa projektuje.

**AKÝ systém sa projektuje?**

1. *data-intensive* - skôr statické systémy s veľkým objemom údajov
2. *software-intensive* - skôr dynamické systémy s veľkým množstvom procesov (operácií), často prebiehajúcich v reálnom čase (real-time applications)

**ČO sa projektuje?**

1. celý KS,
2. časť KS - napríklad HW - sieťové prvky, SW - nový modul aplikačného softvéru,
3. služba spojená s KS, resp. jej využitia napríklad outsourcing, ASP (Application Service Providing - poskytovanie aplikačných služieb).

**AKO sa projektuje?**

1. návrh úplne nového KS:
   - vývoj nového softvéru,
   - inštalácia hotového softvérového produktu („krabicový“ softvér),
   - inštalácia a užívateľské prispôsobenie - customizácia hotového softvérového produktu (napr. ERP, automatizovaný úlohovač),

- zostavenie novej aplikácie z hotových komponentov - systémová integrácia,

2. zmeny v už existujúcom KS:
   - rozšírenie systému,
   - reengineering, redesign, refaktoring.

**BPR - business process reengineering - radikálna zmena podnikových procesov -** predpokladom úspechu projektu IS je definovanie a analýza podnikových procesov, ktoré majú byť informačným systémom podporované. Výsledkom tejto analýzy je často zistenie, že niektoré procesy sú vykonávané neefektívne, redundantne či dokonca nesprávne. Projekt IS môže byť preto spojený is projektom úprav „reálnych" procesov v organizácii.

**Legacy system -** je funkčný systém „zdedený" po predchodcoch, spravidla založený na už prekonanej technológii a už ďalej nevyvíjaný.

Riešené problémy: integrácia do existujúceho KS

nahradenie novým systémom

**PRE KOHO sa projektuje?** - Kto je *Zákazník*, Kto je *Koncový užívateľ*?

1. **interný projekt** - výsledok práce je určený pre vlastnú organizáciu
2. **projekt na zákazku** - vzťah dodávateľ - obstarávateľ
3. **projekt pre voľný trh** - obstarávateľom je vlastná firma
4. **projekt ako subdodávka komplexného riešenia** - obstarávateľom je systémový integrátor

## 3.4 Tvorba výkonnostných parametrov vhodných na posúdenie prínosov z implementácie modelu fakultného komunikačného systému

Problémom mnohých projektov inštalácií komunikačných systémov je zlá komunikácia pracovníkov zodpovedných za modelovanie podnikových procesov

a softvérových návrhárov. Metodika Rational Unified Process (RUP - podrobnejšie sme sa o nej už zmienili v kapitole 1.6.) prepája obe činnosti na úrovni riadenia projektu, pre lepšiu komunikáciu sa využíva spoločná terminológia - významové slovníky. Pre tvorbu modelu v organizáciách sa využívajú prostriedky UML - jednotlivé scenáre sú znázornené pomocou organizačného modelu prípadov použitia, ich realizáciu opisuje objektový model organizácie. U systémov, ktoré nie sú zviazané s konkrétnym podnikovým procesom, sa táto činnosť spravidla vypúšťa.

**Pri tvorbe výkonnostných parametrov vhodných na posudzovanie prínosov z implementácie komunikačného systému, musíme mať popísané konkrétne postupy pri implementovaní:**

***Postupy v rámci tvorby modelu komunikačného systému:***

- *Základná charakteristika organizácie* postupov spočíva v tom, že analytik organizačných procesov na základe konzultácií so zadávateľom spíše základné charakteristiky organizácie a formuluje jej víziu:
  - opis organizácie;
  - formulácia vízie organizácie;
  - tvorba slovníka organizačných pojmov.
- *Definovanie organizačných procesov* - cieľom tohto kroku je identifikovať organizačné procesy (pomocou modelu prípadov použitia) a určiť, na ktoré sa ďalší vývoj systému zameria. Pre zjednotenie terminológie sa priebežne dopĺňa slovník organizačných pojmov:
  - tvorba slovníka pojmov organizácie;
  - nájdenie aktérov organizácie a prípadov použitia.
- *Podrobný popis procesov v organizácii* - procesy identifikované v predchádzajúcom kroku sú pridelené tímom analytikov, ktorí ich podrobne popíšu pomocou štruktúrovaných modelov možných prípadov použitia. Návrhár procesov v organizácii vykoná detailný popis jednotlivých prípadov použitia.

  Správnosť výsledného modelu potom overí návrhár procesov organizácie v spolupráci

so zadávateľom - teda tím v úlohe akéhosi revízora modelu organizácie:

  - štruktúrovanie modelu organizácie a prípadov použitia;
  - podrobný popis prípadov použitia v organizácii;
  - revízia modelov organizácie - prípadov použitia.

- *Popis realizácie procesov v organizácii* - po tom, čo boli jednotlivé procesy vymedzené a podrobne popísané pomocou modelu prípadov použitia, je možné pristúpiť k popisu ich realizácie (nájsť ich účastníkov, vstupy, výstupy a určiť väzby medzi nimi). Výsledkom tohto kroku je objektový model organizácie:
  - nájdenie účastníkov a entít procesov v organizácii;
  - tvorba objektového modelu organizácie.
- *Systematizácia objektového modelu organizácie* - v objektovom modeli vytvorenom v predchádzajúcom kroku sa pripoja špecifikácie jednotlivých účastníkov - ich zodpovednosti a entity. Potom sa v spolupráci so zadávateľom vykoná revízia objektového modelu organizácie:
  - podrobný popis entít procesov organizácie;
  - podrobný popis účastníkov procesov v organizácii;
  - revízia objektového modelu organizácie.
- *Určenie možností automatizácie procesov* - na základe popisu realizácie procesov v organizácii sa určia možnosti ich realizácie a z nich vyplývajúce požiadavky na komunikačný systém:
  - určenie možností automatizácie procesov v organizácii.

**Správa požiadaviek** - požiadavky sú kritériami, ktorých splnenie podmieňuje úspech projektu. Požiadavky je preto potrebné systematicky evidovať a zodpovedajúcim spôsobom dokumentovať a zapracovávať ich prípadné zmeny. Správu požiadaviek by sme teda mohli definovať ako:

- systematický prístup k identifikácii a dokumentácii požiadaviek na systém,
- proces zabezpečujúci zhodu medzi obstarávateľom a dodávateľom v prípade zmeny požiadaviek na systém.

***Postupy v rámci správy požiadaviek:***

- *Analýza problémovej oblasti* - kľúčovou činnosťou v rámci analýzy problémovej oblasti je tvorba vízie. Túto činnosť vykonáva systémový analytik na základe konzultácií so zainteresovanou osobou - väčšinou z radov obstarávateľa. Jej cieľom je príslušný problém presne identifikovať a popísať, určiť iné osoby, ktorých sa dotýka a stanoviť, čo všetko by úspešné riešenie malo spĺňať. V priebehu správy požiadaviek sa vytvára spoločný slovník, ktorý má za úlohu zjednotiť používanú terminológiu a pomôcť odstrániť možné duplicity spôsobené rozdielnou formuláciou rovnakého problému.

  V rámci tohto kroku sa tiež identifikujú základné závislosti medzi jednotlivými požiadavkami a kľúčoví aktéri systému, teda osoby, alebo iné systémy, ktoré systém využívajú:

  - zistenie potrieb zainteresovaných osôb;
  - tvorba vízie;
  - tvorba spoločného slovníka;
  - identifikácie závislostí;
  - nájdenie aktérov a prípadov použitia.

- *Identifikácia požiadaviek zadávateľa* - ak preukáže úvodná verzia vízie, že sa oplatí v projekte pokračovať, vykoná sa „zber“ požiadaviek zainteresovaných osôb napr. formou elektronických príspevkových skupín. Následne potom systémový analytik usporiada informácie, odstráni z nich možné duplicity a na ich základe týchto informácií doplní víziu a určí základné prípady použitia systému. Výsledkom by mal byť úvodný model prípadov použitia obsahujúci kľúčové prvky komunikačného systému:

  - zistenie potrieb zainteresovaných osôb;
  - tvorba vízie;
  - tvorba spoločného slovníka;
  - identifikácie závislostí;
  - nájdenie aktérov a prípadov použitia.

- *Definícia systému* - tento krok predstavuje ďalšie spresnenie modelu prípadov použitia,

jeho výsledkom by mali byť kompletné špecifikácie požiadaviek na úrovni systému:

- tvorba spoločného slovníka;
- identifikácia závislostí;
- nájdenie aktérov a prípadov použitia.

- *Identifikácia kľúčových prípadov* použitia, stanovenie priorít - následne po tom, čo boli určení takmer všetci aktéri aj prípady použitia systému, možno pristúpiť k stanoveniu ich priorít i možných rizík. Softvérový architekt v tomto kroku nájde kľúčové prípady použitia pre architektúru komunikačného systému a vytvorí jej prvý náčrt - obsahujúci iba pohľad prípadov použitia. Priority prípadov použitia sa potom vložia do vízie:
  - tvorba vízie;
  - stanovenie priorít prípadov použitia;
  - identifikácia závislostí.
- *Spresnenie definície systému* - tento krok predpokladá, že na úrovni systému boli identifikovaní všetci aktéri a aj prípady použitia a ku všetkým existuje aspoň stručný popis. Analytik prípadov použitia detailne popíše jednotlivé prípady použitia, a to vstupné aj výstupné podmienky a postupy v rámci prípadu použitia.

  Pri systémoch, ktorých dôležitou časťou je užívateľské rozhranie, sa na základe presnej špecifikácie prípadov použitia vykoná jeho návrh:
  - detailné špecifikácie prípadov použitia;
  - modelovanie užívateľského rozhrania;
  - prototypovanie používateľského rozhrania.
- *Riadenie zmien požiadaviek* - v priebehu projektu podávajú jednotliví účastníci požiadavky na zmeny. Systémový analytik ich najprv klasifikuje a určí, či ide o požiadavky na odstránenie chýb softvéru, či ide zmenu súčasnej požiadavky, alebo či sa jedná o pridanie novej funkcionality. Požiadavky sa potom popíšu pomocou vyššie uvedených postupov, odošlú sa na zhodnotenie „revízorovi požiadaviek“ (zamestnanec, alebo kolektív zodpovedajúci za časti systému, ktorých sa konkrétna zmena dotkne). V prípade, že požiadavky sú relevantné a z hľadiska časového plánu a rozpočtu aj plne realizovateľné, zabezpečí ich zapracovanie do projektu:
  - identifikácia závislostí;

- úprava štruktúry požiadaviek;
- revízia požiadaviek.

Analýza a návrh prepája požiadavky na systém s jeho implementáciou. Zatiaľ čo modely prípadov použitia vytvárané v rámci správy požiadaviek odpovedajú na otázku Čo sa bude vytvárať?, návrh by mal riešiť Ako sa to bude vytvárať? - to znamená nadefinovať architektúru komunikačného systému vrátane jeho členenia na jednotlivé komponenty.

***Postupy v rámci analýzy a návrhu:***

- *Úvodný návrh architektúry* - cieľom tohto kroku je na základe analýzy prípadov použitia a ďalších požiadaviek nájsť kľúčové prvky a vytvoriť analytický model systému. V tomto štádiu by sa ale skôr než o komplexný návrh systému malo jednať o náčrt riešenia, na základe ktorého bude možné posúdiť jeho uskutočniteľnosť:
  - analýza architektúry;
  - analýza prípadov použitia;
  - zhodnotenie správnosti navrhovanej architektúry.
- *Spresnenie návrhu architektúry* - tento krok predstavuje prechod od analýzy k návrhu. Analytický model je doplnený o aspekty týkajúce sa implementácie a nasadenia komunikačného systému - softvérový architekt teraz na základe analýzy prípadov použitia identifikuje jednotlivé prvky návrhu - budúce komponenty. V tomto štádiu sa tiež rieši súbežnosť procesov a praktická distribúcia systému:
  - identifikácia prvkov návrhu;
  - popis súbežnosti;
  - popis distribúcie.
- *Návrh jednotlivých subsystémov* - po tom, čo je k dispozícii definícia základných prvkov návrhu, vytvorí k nim návrhár presnejší popis ich funkcionality, aby ich bolo možné implementovať ako jednotlivé komponenty. Teraz sa tiež navrhuje hierarchická štruktúra modelu implementácie:
  - návrh prvkov;

- návrh subsystémov;
- revízia návrhu.

- *Návrh databázy* - návrhár databázy určí perzistentné triedy a navrhne k nim príslušné dátové štruktúry:
  - návrh databázy

***Postupy v rámci implementácie:***

- *Určenie praktickej distribúcie systému* - softvérový architekt na základe modelu návrhu definuje spôsob implementácie jednotlivých komponentov komunikačného systému. Tento pracovník by mal mať určité skúsenosti s používaným implementačným prostredím a integráciou, pretože nesprávne zvolená štruktúra systému často podstatne sťaží súbežný vývoj komponentov systému:
  - štruktúrovanie modelu implementácie.
- *Plánovanie integrácie systému* - cieľom tohto postupu je naplánovať, ktoré subsystémy sa budú v rámci aktuálnej iterácie implementovať a v akom poradí sa vykoná ich integrácia:
  - plánovanie integrácie systému.
- *Implementácia komponentov* - implementátori vytvoria zdrojový kód komponentov a vytvoria kompiláciu. Následne ho otestujú a odstránia prípadné chyby. Revízor kódu potom skontroluje, či zdrojový kód spĺňa formálne požiadavky dané pravidlami programovania:
  - implementácia komponentov;
  - testovanie komponentov;
  - opravy chýb;
  - revízia kódu.
- *Integrácia subsystémov* - v prípade, že subsystém je vyvíjaný tímovo, zabezpečí poverený člen tímu integráciu jednotlivých komponentov. Po tom, čo je celý subsystém otestovaný, je zaradený do repozitára pre integráciu:
  - integrácia subsystému.

- *Integrácia systému* - softvérový integrátor postupne pridáva jednotlivé subsystémy podľa plánu integrácie, v každom kroku je podsystém otestovaný:
  - integrácia systému.

***Postupy v rámci nasadzovania komunikačného systému:***

- *Plánovanie nasadzovania* - pred vlastným nasadením komunikačného systému je potrebné určiť jeho spôsob nasadzovania a vytvoriť jeho časový plán. Vzhľadom k tomu, že okrem samotného softvéru sa spravidla dodávajú aj ďalšie položky (manuály, hardvérové komponenty a iné), je potrebné tieto tiež dokumentovať a zohľadniť v pláne:
  - vytvorenie zoznamu potrebného materiálu;
  - vytvorenie plánu nasadzovania.
- *Tvorba dokumentácie* - aby pracovníci obstarávateľa mohli produkt samostatne používať, je potrebné im poskytnúť dokumentáciu v podobe užívateľských príručiek a manuálov. U niektorých produktov sa tiež vytvárajú materiály pre školenie užívateľov:
  - tvorba „HELPu“ - pomocníka užívateľských manuálov;
  - tvorba podkladov pre školenia.
- *Vytvorenie inštalácie* - cieľom tohto postupu je vytvorenie inštalácie systému, ktorá je určená koncovým užívateľom. V závislosti na štádiá vývoja sa môže jednať o testovaciu (beta) alebo konečnú verziu komunikačného systému:
  - vytvorenie inštalačných programov.
- *Beta testovanie* - akákoľvek už funkčná verzia systému je odovzdaná určitej skupine koncových užívateľov, aby vykonali jej testovanie pri bežnej prevádzke. Zistené závady sú zaevidované aj zapracované a v „ostrej“ verzii produktu už aj odstránené:
  - riadenie beta testov.
- *Preberacie testy* - pred samotným odovzdaním produktu sa zvyčajne vykonajú testy, v ktorých sa obstarávateľ presvedčí o jeho správnej funkčnosti. Testovanie sa spravidla uskutočňuje už v nasadzovanom prostredí, teda v systéme zadávateľa:

- riadenie preberacích testov.

- *Výroba softvérového balíka* - tento krok sa uplatňuje v prípade, že produkt je distribuovaný ako krabicový softvér. Ak sú preberacie testy úspešné, odovzdá vedúci nasadzovania produkt do „výroby“ a vykoná jeho overenie:
  - „výroba“ - jedná sa o napaľovanie produktu na CD, DVD, kopírovanie manuálov;
  - kontrola vyrobeného produktu - po napaľovaní sa zistí funkčnosť dátového nosiča.
- *Poskytnutie prístupových práv k stiahnutiu SW* - táto činnosť sa uplatňuje v prípade, že produkt je distribuovaný po akejkoľvek sieti.

Najvýznamnejším testovacím nástrojom je práve preto už spomínaná metodika **RUP**. Táto testovacia disciplína je rovnako ako ďalší jej obsah postavená na najlepších praktikách, resp. kľúčových princípoch v najnovších testovacích verziách. V prípade testovania sa jedná najmä o „priebežné overovanie kvality“ a „iteratívny vývoj“, resp. „trvalé zameranie na kvalitu“ a „predkladanie výsledkov v iteráciách“.

Priebežné overovanie kvality v metodike vychádza z už uvedeného rozdelenia na kvalitu procesnú a kvalitu produktu. Každý artefakt, aktivita alebo vedľajší produkt procesu má metodikou jasne stanovené kontrolné body a miera splnenia týchto bodov predstavuje meradlo kvality (Buchalcevová - Kučera, 2008). Zrozumiteľne definované ciele sú súčasťou opisu všetkých aktivít, vďaka tomu majú členovia tímu prehľad o svojich povinnostiach a vytvárajú artefakty, ktoré sú skutočne potrebné. Všetci členovia tímu sa tak podieľajú na kvalite procesu aj kvalitu produktu rovným dielom, čo vyvracia často sa vyskytujúce názory, že za kvalitu nesie zodpovednosť len testovací tím, alebo tím starajúci sa o oblasť zabezpečenia kvality.

Iteratívny prístup k vývoju umožňuje realizovať testovanie počas celého projektu. Intenzita testovania sa potom odvíja od počtu iterácií použitých pri implementácii komunikačného systému. Hlavnými cieľmi testovej disciplíny je preverenie interakcií medzi objektmi, preverenie správnej integrácie všetkých komponentov do systému, preverenie, či boli všetky požiadavky korektne zaimplementované, a ubezpečenie, že všetky identifikované chyby sú pred nasadením systému vyriešené.

Testovú disciplínu metodika opisuje ako disciplínu, ktorá sa v konkrétnych princípoch odlišuje od zvyšných základných disciplín, ako je implementácia, analýza,

návrh a podobne. Ako najpodstatnejší rozdiel je metodikou uvádzaný prístup disciplíny k vyvíjanému systému. Zatiaľ čo ostatné disciplíny sa snažia produkt uviesť do kompletného a bezchybného stavu, testovanie je naopak zamerané na odhalenie chýb alebo chýbajúcich častí. Predstavuje protipól zvyšným hlavným disciplínam a nezávisle na nich overuje, či výstupy týchto disciplín zodpovedajú požiadavkám.

Na prvý pohľad sa teda môže zdať, že testovanie je jediná z hlavných disciplín metodiky RUP, ktorá do produktu nevnára žiadne nové vlastnosti a ani sa nepodieľa na tvorbe viditeľnej hodnoty produktu. Prostredníctvom testovej disciplíny je však významne ovplyvnená kvalita systému, ktorú je možné považovať už za celoorganizačnú hodnotu.

**Úloha v testovej disciplíne**

S testovou disciplínou sú v metodike spojené štyri základné úlohy - *tester, analytik testov (test analyst), návrhár testov (test designer)* a *manažér testov (test manager)*. V praxi môže byť podľa potreby každá rola obsadená jednou alebo viacerými osobami, alebo pri menších projektov implementovaného komunikačného systému je niekedy vhodné priradiť viac rolí jednej osobe. Metodika dodáva, že s testovaním čiastočne súvisí aj úloha z ďalších disciplín. Všetky aktivity v metodike sú bez výnimiek zakončené preventívnym krokom, spočívajúcom v zhodnotení prínosu a hlavne v overení výsledkov aktivity.

**Tester**

Úloha testera spočíva v prvom rade vo vykonávaní testov, čo je kľúčová činnosť testovej disciplíny, a ďalej má tester na starosti evidenciu výsledkov týchto testov. V metodike sa nachádza nasledujúci výpočet aktivít spojených s rolou testera: implementácia testu, implementácia testovej suity, prevedenie testovej suity a prípadná analýza zlyhania testu.

***Implementácia testu***

Implementáciou testu sa rozumie tvorba jedného alebo viacerých artefaktov, označovaných ako testový skript, prípadne pojmom špecifikácia testovej procedúry. Ide o dokument, špecifikujúci sekvenciu akcií pre vykonanie samotného testu. V prvom rade je

potrebné zvoliť spôsob implementácie testovacieho skriptu, ktorý závisí od typu testu, pri ktorého neskoršom prevedení bude skript využitý. Pokiaľ pôjde o manuálne testovanie, skript by mal byť vytváraný najlepšie v podobe textových inštrukcií, ktorými sa tester bude pri vykonávaní testu riadiť. V prípade použitia v automatizovanom teste by sa malo jednať o programové inštrukcie (skripty) pre testovací nástroj. Inštrukcie môžu vzniknúť jednak naprogramovaním, a tiež záznamom skriptu, ak to nástroj umožňuje.

***Implementácia testovej suity***

Aktivita implementácie testovej suity slúži podľa metodiky k vytvoreniu kombinácií testov, ktoré budú poskytovať hodnotnejšie informácie, než by tomu bolo u samostatne spúšťaných testov. Na začiatku tejto aktivity je potrebné preskúmať vhodné oblasti, kde by vytvorenie testovej suity malo význam, a určiť testy a ďalšie súčasti, ktoré by suita mala obsahovať. Pretože suita pozostáva z viacerých testov, je nutné zistiť, či a prípadne aké závislosti medzi testami bude treba vyriešiť. Ide najmä o zabezpečenie správnej nadväznosti v poradí susedných dvojíc testov a zjednotenie ich vstupných a výstupných podmienok. V metodike sa zdôrazňuje, že s použitím opakovaného vývoja veľakrát pri prvotnej implementácii nevznikne konečná verzia suity a dochádza často k jej úpravám. Preto by sa mal klásť dôraz na umožnenie neskorších úprav. Suitu je ďalej potrebné implementovať tak, aby bola schopná sa reštartovať alebo v lepšom prípade pokračovať v behu aj po havárii niektorého z testov.

***Prevedenie testovej suity***

Metodika pre roly testera neponúka žiadnu aktivitu, ktorá by zabezpečovala vykonanie testu. Namiesto toho existuje iba aktivita prevedenia testovej suity, čo možno vysvetliť tak, že podľa definícií je test možné pokladať za testovú suitu obsahujúcu iba jeden prvok.

***Analýza zlyhania testu***

Aktivita analýzy zlyhania testu spočíva v preskúmaní testovacieho logu a identifikáciu všetkých chýb nájdených v priebehu testovej suity. Je potrebné vykonať rozlíšenie chýb, ktorých príčinou je samotná testová suita, z ktorej log vznikol,

a zabezpečiť ich opravu. Zvlášť je vykonaná evidencia všetkých ostatných chýb, ktoré pochádzajú z testovaného systému. Je potrebné vykonať podrobný rozbor nájdených chýb v systéme, čo najdetailnejšie popísať problém a pokúsiť sa na základe predchádzajúcich skúseností odhadnúť príčinu chyby. Ďalej podľa potreby chyby, za kontrolovaných podmienok, znovu vyvolať a prijať ďalšie opatrenia, ktoré pomôžu pri odstraňovaní týchto chýb. Aby nájdené chyby boli riešené, je nevyhnutná ich prezentácia členom tímu, ktorí zodpovedajú za príslušné opravy, a je nevyhnutná dokumentácia chyby - to znamená jej vloženie do systému pre evidenciu chýb či zabezpečenie novej zmenovej požiadavky.

**Analytik testov**

Analytik testov má podľa metodiky na starosti identifikáciu a určenie potrebných testov, sledovanie podrobného postupu testov a výsledkov každého testovacieho cyklu a vykonáva hodnotenie celkovej kvality. V kompetencii analytika testov sú tieto aktivity: identifikácia testových nápadov, určenie výsledkov testov, preverenie zmien v builde, definícia potrieb zhodnotení a monitorovanie závislostí, identifikácia cieľov testu, definícia detailov testu.

V súčasnej dobe sa v oblasti tvorby výkonnostných parametrov na posudzovanie prínosov z implementácií akýchkoľvek komunikačných systémov môžeme stretnúť so súčasným využívaním objektových i štruktúrovaných metodík, hoci pritom v oblasti implementácií systémov sa využívajú dnes už výhradne objektové programovacie jazyky. Prečo tomu tak je? Sú totiž oblasti návrhu a vývoja, kde je prirodzenejšie využiť práve štruktúrované metodiky. Ďalším dôležitým dôvodom je určite aj fakt, že veľa výkonnostných parametrov je zameraných na štruktúrované metódy (objektové metodiky pre tvorbu výkonnostných parametrov tu nie sú zase tak dlho). Pritom objektová metodika má svoje nesporné výhody. Tými hlavnými sú triedy a ich zodpovednosť a názornosť vizualizácie. Znázornenie systému pomocou tried, ktoré vznikajú zovšeobecnením objektov reality, umožní akémukoľvek testerovi realitu lepšie pochopiť. Na druhej strane je to jasný a prirodzený prostriedok pre zachytenie štruktúry reality a je zrozumiteľný aj nevývojárom komunikačných systémov, čiže taktiež jednoduchým užívateľom. To, že trieda má nejakú zodpovednosť, umožňuje vytvárať jej metódy (operácie triedy), ktoré zabezpečujú určitú činnosť a hlavne sú s triedou úzko zviazané, teda sú jej súčasťou.

Takáto trieda - ak je dobre navrhnutá - potom ako zapuzdrený celok môže byť využívaná opakovane, všade tam, kde sa vyskytujú jej akékoľvek objekty.

Aby bol systém tvorby výkonnostných parametrov schopný posúdiť prínosy z implementácie modelu fakultného komunikačného systému, v súvislosti s priebehom jeho implementácie, musí ich návrhár systému najprv pochopiť a zachytiť. Na zachytenie výkonnostných parametrov je možné využiť viac možností. Začneme u tej vyložene objektovej, pri ktorej možno použiť notáciu UML. Pre určenie vecnej oblasti sa uplatňuje use case diagram. Pre znázornenie sekvenčného opisu činností prebiehajúcich v procese sa uplatňuje hybridný diagram činností. Hybridný preto, pretože je vhodné, aby obsahoval aj objekty zodpovedné za vykonanie daných činností a objekty tvoriace vstupy a výstupy procesov. Zovšeobecnením objektov z hybridného diagramu činností potom získame diagram tried vyskytujúcich sa v procesoch. Triedy zodpovedných objektov (veľká časť bude zodpovedať prvkom organizačnej štruktúry fakulty) i triedy vstupov a výstupov sa označia príslušnými stereotypmi (ako napríklad <<vstup/výstup>>). Stereotypmi je tiež vhodné označiť triedy zodpovedajúce prvkom organizačnej štruktúry. Následne sa dopracuje use case diagram, to znamená, že sa dopracujú väzby <<uses>> a <<extends>>, aby use case diagram vyjadroval požadovanú funkcionalitu implementovaného komunikačného systému.

Z vyššie uvedeného postupu je na jednej strane zrejmá flexibilita UML, na druhej strane fakt, že UML nie je pre modelovanie procesov navrhnuté a musí teda byť takto užívateľsky prispôsobené. Museli by sa napríklad využiť balíky s označenými stereotypmi. Uvedený postup je možné použiť aj v našom prípade, problémom totiž je, že niektoré modelovacie nástroje diagram aktivít neponúkajú. Preto sa v praxi používajú aj iné metódy pre tvorbu výkonnostných parametrov. Tieto iné metódy bývajú podporované špecializovanými modelovacími nástrojmi, ktoré okrem hierarchického rozpadu modelovaných procesov KS ponúkajú aj možnosti simulácie priebehu procesov, hľadanie „úzkych hrdiel“, atď., teda podklady vhodné pre možný reengineering procesov - nápravu priebehov procesu, či už čiastkové, postupné, alebo aj radikálne.

UML ako vizuálne modelovanie je silnou vlastnosťou moderných metodík. Umožňuje graficky znázorniť procesy, závislosti a iné aspekty softvéru a záležitostí okolo neho. Vizuálne modelovanie napomáha objasneniu predstáv medzi zadávateľom a dodávateľom. Najmä u rozsiahlejších projektov umožňuje vytvorenie modelu zobrazujúceho rôzne aspekty implementovaného komunikačného systému jednoducho

zachytiť skutočnosti, ktorých pripomienky by v inej podobe boli neprehľadné a menej zrozumiteľné. Model ako výsledok vizuálneho modelovania predstavuje zjednodušenie reality, vedúce k vyššej abstrakcii, čo umožňuje sústrediť úsilie najprv na samotný problém vo všeobecnej rovine a až neskôr na jeho konkrétnu realizáciu. Metodika RUP aj odporúča pri vizuálnom modelovaní použiť práve jazyk UML, pokrývajúci rôzne typy modelov, označovaných v terminológii jazyka ako diagramy, napríklad diagram prípadov použitia, diagram komponentov, diagram aktivít a ďalšie.

**Identifikácia mechanizmov testovateľnosti**

Popis aktivity identifikácie mechanizmov testovateľnosti vyžaduje objasnenie dvoch použitých termínov: *mechanizmus* a *testovateľnosť*. Mechanizmus je podľa metodiky RUP definovaný ako špecifické riešenie (opakujúceho sa problému) v konkrétnom prípade. Možno ho tiež popísať ako vlastnosť alebo charakteristiku. Pre väčšiu zrozumiteľnosť metodika uvádza príklady mechanizmov: *súbežnosť*, *distribúcia*, *komunikácia*, *bezpečnosť* a ďalšie.

Pojem testovateľnosti metodika vysvetľuje ako schopnosť vykonať u cieľových prvkov vhodné testovanie. Úspešné dokončenie tejto aktivity úzko súvisí so správnym pochopením softvérovej architektúry produktu a podoby cieľových prostredí, čo je obsahom prvého kroku. So získanými znalosťami je potom potrebné zostaviť zoznam potenciálnych mechanizmov. Do tohto zoznamu sú ďalej zaradené už existujúce mechanizmy automatizovaného aj manuálneho testovania, napríklad schopnosť tohto nástroja spúšťať súbežne testy viacerých užívateľov, z viacerých pracovných staníc naraz. Zo zoznamu sú na záver vybrané mechanizmy, ktoré budú použité a tento výber je prezentovaný členom testovacieho tímu a podľa potreby aj implementátorom.

Pred stanovením prvkov testovateľnosti metodika odporúča objasnenie poznatkov o určitých oblastiach, a teda aj o fakultnom komunikačnom systéme. Aktivita tak začína zoznámením s cieľovými prvkami k otestovaniu a mechanizmy, ktoré by pri testoch cieľových prvkov bolo možné použiť. Ďalej je vyžadované skúmanie budúcich dynamických aspektov komunikačného systému, čo sú udalosti, procesy, prvky a ďalšie. Je tiež potrebné získať informácie o rozhraniach, ktorá bude systém používať pre komunikáciu s okolitým fakultným prostredím.

Aby sa zabránilo chaotickému vykonávaniu testových činností, je metodikou

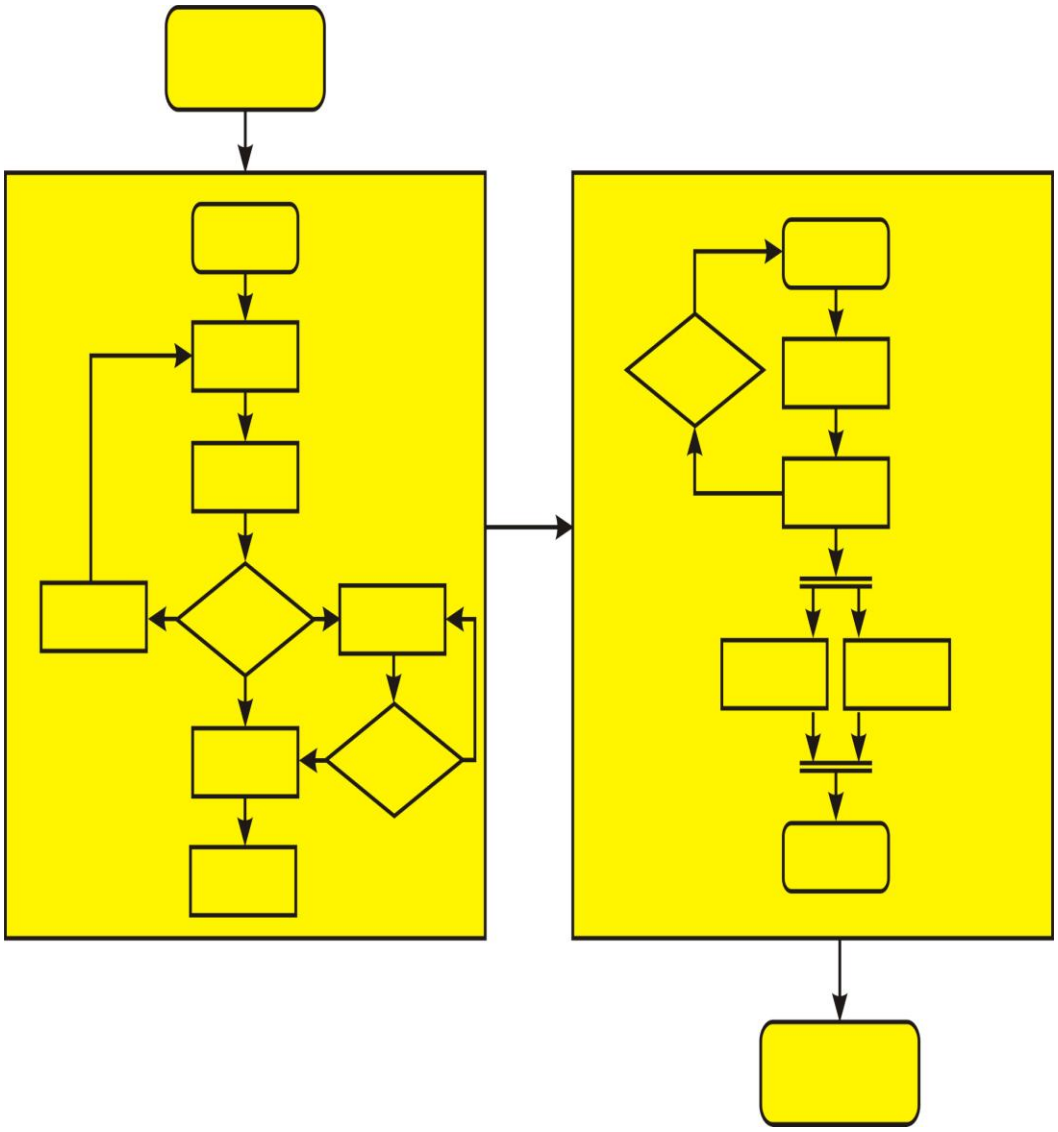
odporúčané použitie testovej infraštruktúry. Je potrebné vykonať rozhodnutie o podobe súčasťou infraštruktúry, ako sú šablóny testových skriptov, spoločne používané procedúry a premenné, spôsob oddelenia testových dát od skriptov a ďalšie. Aktivita je zakončená určením požiadaviek na testovú infraštruktúru, ktorá bude prvky testovateľnosti využívať.

**Pre tvorbu vývojových diagramov platia všeobecné postupy (http://web.sks.cz):**

- Tvorte diagram tak, ako by ste pripravovali program pre počítač - nespoliehajte sa na intuitívne rozhodovanie alebo na zaužívané postupy.
- Pokiaľ sa jedná o zložitejší problém, vytvorte niekoľko jednoduchších diagramov, postupne riešených na rôznych úrovniach podrobnosti (najprv všeobecnú blokovú schému, potom detailnejšie členenie jednotlivých blokov) a rozdeľte rozsiahlu detailnú schému do viacerých dielov – stránok.
- Každý diagram musí mať aspoň jeden začiatok a aspoň jeden koniec.
- Každý proces zobrazujte len raz.
- Každý proces musí mať aspoň jeden vstup a aspoň jeden výstup.
- Každý rozhodovací proces musí mať len jeden vstup a viac než jeden výstup.
- Každý proces musí smerovať k jasne definovanému ukončeniu.
- Akonáhle na proces nič ďalšie nenadväzuje, je to KONIEC.
- Snažte sa o prehľadnosť, zrozumiteľnosť, dodržujte úpravu: toky dát zobrazujte zhora nadol a zľava doprava; neprekrižujte čiary tokov dát; nespojujte dve vstupné čiary do jednej výstupnej v rovnakom bode.

**Validácia - potvrdenie správnosti vývojového diagramu:** priechod všetkými vetvami - každá cesta by mala viesť od začiatku po koniec; musí byť jasný, názorný priechod testovacích dát.

Hierarchické členenie vývojového diagramu by sme mohli znázorniť aj ako v obrázku 15.



Obrázok 15 Hierarchické členenie vývojového diagram (vlastný návrh)

# 4 Výsledky práce

## 4.1 Prípadová štúdia 1

**Microsoft Office Outlook ako nástroj fakultného komunikačného systému na PHF EU**

Outlook je súčasťou balíka Office už od prvej verzie. Vylepšuje možnosti štandardného programu Outlook Express, ktorý sa nachádza vo Windows XP, alebo Windows Mail, ktorý nájdeme vo Windows Vista. Jeho hlavnou výhodou je pokročilá správa aj viacerých účtov, práca s kontaktmi, úlohami a kalendármi. „Rozumie“ si tiež so serverom Exchange, ktorý je používaný vo väčších firmách a ktorý uľahčuje prácu s e-mailami, skupinami používateľov, a navyše ponúka webový prístup k pošte. Vďaka nemu si môžeme poštu prečítať kdekoľvek, kde je dostupný internetový prehliadač. Tento webový prístup ponúka rad e-mailových freeserverov (servery, ktoré ponúkajú mail poštu zadarmo), na Slovensku je to napríklad najpoužívanejší Gmail.com a potom ďalšie ako Zoznam.sk, Centrum.sk, alebo Post.sk. Webový prístup je ale v mnohých ohľadoch obmedzujúci, neustále sa načítajú pomerne rozsiahle webové stránky, so správami sa nedá pracovať lokálne, a čo je najdôležitejšie - zle sa prikladajú rôzne prílohy. Preto je lepšie sa pripájať práve pomocou klienta elektronickej pošty, ktorý je nielen že pohodlnejší, ale tiež ponúka viac možností: správy možno filtrovať, vyhľadávať v nich, zaraďovať ich, triediť do rôznych zložiek, archivovať a podobne. Naviac sú stiahnuté lokálne na pevnom disku, takže s nimi možno pracovať aj vtedy, ak nie sme práve pripojení k internetu.

**Poštové protokoly**

Najznámejším e-mailovým protokolom, ktorý slúži na komunikáciu medzi klientskym počítačom a serverom (počítač v internete, na ktorý vaša pošta prichádza a cez ktorého je odosielaná iným používateľom), je nepochybne POP3 (Post Office Protocol version 3). Tento protokol „ovláda“ všetky programy pre prácu s poštou a súčasne ho ako základný typ ponúkajú aj všetky e-mailové servery. POP3 je výhodný pre užívateľov, ktorí

majú časovo obmedzené pripojenie k internetu. Správy si zo svojho poštového servera stiahneme, odpojíme sa a môžeme si čítať doručené e-maily ľubovoľne dlho a byť pri tom offline (odpojení od internetu). Po stiahnutí správ zo servera do klienta (tzn. počítača, do ktorého budeme sťahovať správy) sa správy na serveri môžu zmazať a tým sa uvoľní miesto pre ďalšie správy. Ak máme malú e-mailovú schránku, nemusíme sa báť, že bude preplnená a nová pošta nebude môcť prísť.

Na druhej strane je to aj nevýhodné. Keď sa do pošty budeme chcieť pozrieť aj odinakiaľ ako zo svojho počítača - napríklad cez webové rozhranie daného poštového servera - nebude to možné. E-maily budú uložené v našom počítači a schránka na internete bude obsahovať len neprečítané správy, prípadne bude prázdna. Preto umožňujú e-mailoví klienti zanechať na serveri kópiu správy. E-mail tak nebude presunutý, ale iba skopírovaný. Toto riešenie je vhodné vtedy, ak pracujeme s poštou z viacerých miest.

Protokol POP3 má ale aj jedno veľké obmedzenie - nemôžeme si vybrať, aké správy sa do počítača stiahnu. Dopredu ani nevieme, koľko ich na server dorazilo a teda ako dlho bude sťahovanie trvať - to spoznáme, až so sťahovaním začneme. Ak nám teda do schránky chodia spamy alebo zavírená pošta, nemôžeme to ovplyvniť; jedine ak by spamy filtroval priamo poštový server. Hoci v rámci Microsoft Office Outlook nám samostatná časť, a to Outlook spam, filtruje do špeciálnej zložky rôzne spamy a spravidla sa mu darí odchytiť všetku nevyžiadanú poštu, ale občas do spamu zaradí aj poštu vyžiadanú, takže je potrebné túto zložku čas od času skontrolovať.

Lepším protokolom pre prácu s poštou je IMAP (Internet Message Access Protocol), avšak ten nepodporujú všetky poštové servery. V súčasnej dobe sa používa štvrtá verzia IMAP. Na rozdiel od protokolu POP3 je optimalizovaný pre prácu v dlhodobo pripojenom režime, keď správy zostávajú uložené na serveri a priebežne sa sťahujú vtedy, keď sú potrebné. Rozdiely zahŕňajú podporu pre prácu viacerých pripojených klientov zároveň, uchovávanie stavov správ na serveri, podporu viacerých zložiek a prehľadávanie správ na strane servera. Môžeme tak používať jedného poštového klienta v práci a jedného doma, a pritom v oboch uvidíme rovnakú poštu prijatú aj odoslanú, rovnaké kontakty a podobne.

Tretím dôležitým e-mailovým protokolom je SMTP (Simple Mail Transfer Protocol). Ide o protokol určený pre prenos správ elektronickej pošty (e-mailov) medzi stanicami. SMTP zaisťuje doručenie pošty, ktorú odošleme inému užívateľovi, do jeho poštovej schránky. Z nej si ju užívateľ môže kedykoľvek vybrať buď cez webové rozhranie, alebo pomocou poštového klienta. Používa k tomu vyššie uvedené protokoly

POP3 alebo SMTP.

## Základná práca s e-mailami

### *Založenie nového účtu*

Aby sme mohli sprevádzkovať poštového klienta, musíme poznať od správcu svojej schránky prihlasovacie meno a heslo - to, ktoré používame aj pre prípadný webový prístup. V závislosti od toho, či používame pre príjem pošty protokol POP3 alebo IMAP, potom ešte potrebujeme adresu poštového servera a súčasne aj adresu SMTP servera, ktorý slúži na odosielanie pošty. Niekedy býva server odchádzajúcich správ rovnaký ako server prichádzajúcich správ, niekedy sú odlišné - pre odosielanie správ sa spravidla používa SMTP server poskytovateľa internetu. U freemailov majú adresy serverov spravidla tento formát: pop3.mail.sk a smtp.mail.sk. Kým server prichádzajúcich správ je vždy zaheslovaný, aby sa k našim správam nedostal nikto nepovolaný, u servera odchádzajúcich správ to neplatí vždy, spravidla žiadne heslo zadávať nemusíme. Ak server predsa len vyžaduje overenie, musíme požadované údaje zadať už pri konfigurácii e-mailového konta. Overenie nebýva potrebné v prípade, ak používame SMTP server svojho poskytovateľa internetu. V takom prípade ale môžeme poštu odosielať len vtedy, keď sa k internetu pripájame práve cez tohto konkrétneho poskytovateľa.

### *Nastavenie účtu*

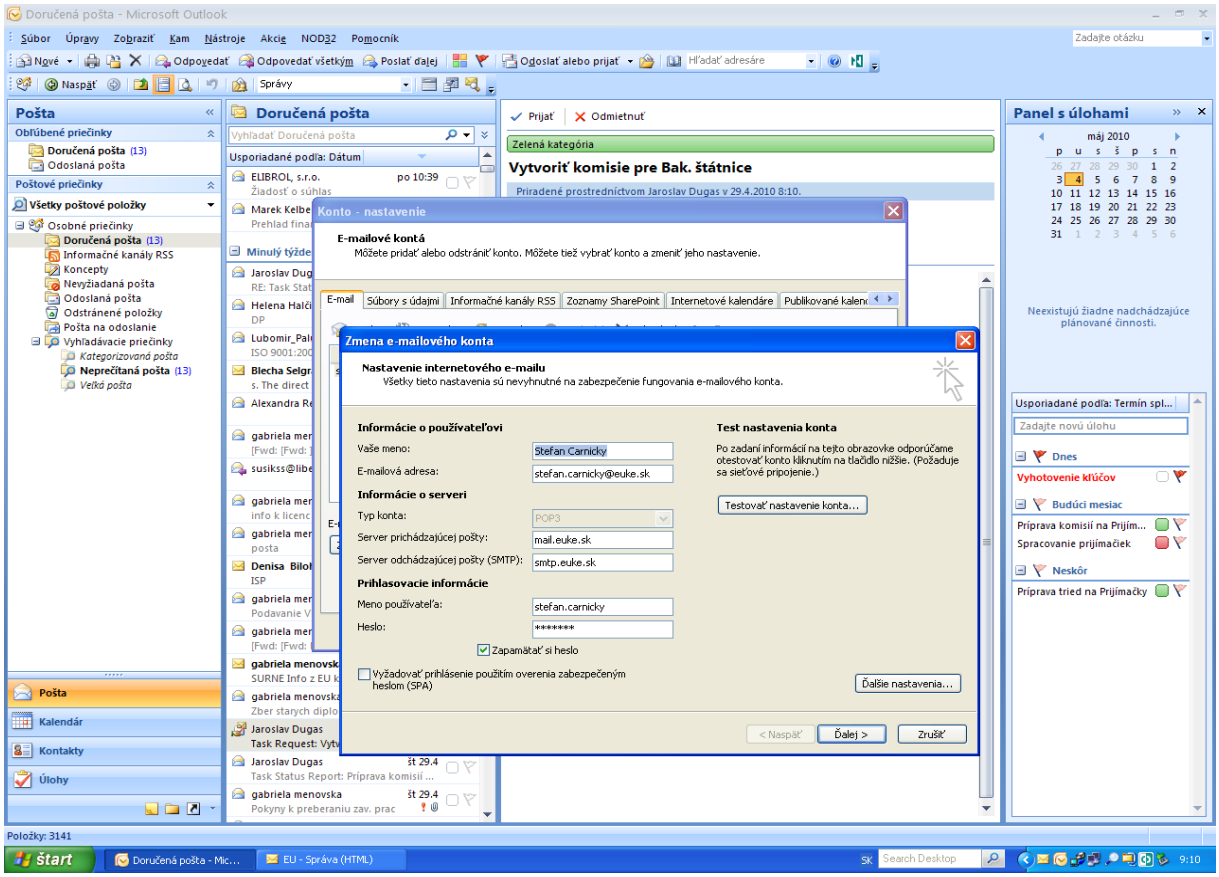
Na nastavenie účtu sa dostaneme z Outlooku z ponuky *Nástroje, Nastavenie kont...* a príkazu *E-mailové kontá*. Dostaneme sa do hlavného konfigurátora pošty, kde vidíme zoznam už nakonfigurovaných účtov, kde v prípade užívateľov PHF bude tento zoznam pravdepodobne prázdny. Kliknime na ikonu *Nové* vľavo hore (záložka *E-mail*). Následne vyberieme hneď prvú položku *Microsoft Exchange Server, POP3, IMAP* alebo *HTTP*, a v nasledujúcom dialógu postupne vyplníme základné prístupové údaje k svojej e-mailovej schránke: meno, e-mail a heslo. Ak chceme nechať Outlook, aby nakonfiguroval prístup ku schránke automaticky, ponecháme voľbu *Konfigurovať ručne* nastavenia servera alebo ďalšie typy servera. Manuálne nastavenie bude vo väčšine prípadov potrebné, nie je to však nič zložité.

Ak sa automatické nastavenie nepodarí, čo je dosť pravdepodobné, potom prejdeme na manuálnu konfiguráciu. V tomto prípade sa zobrazí ďalší dialóg, kde vyberieme *E-mail v internete* a v ďalšom kroku zadáme konkrétne správne údaje - dôležitý je predovšetkým

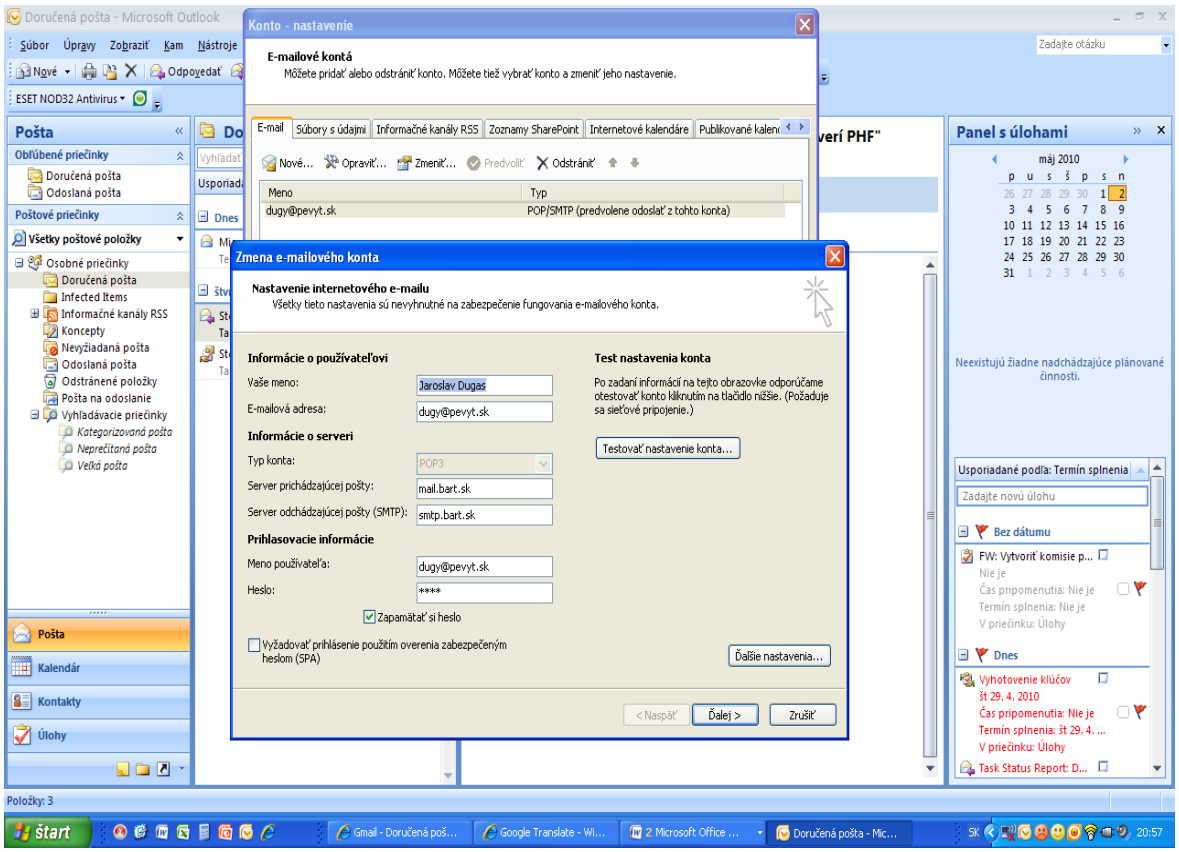
POP3 a SMTP server.

Užitočná je voľba *Zapamätať heslo* - tá zabezpečí, aby sme nemuseli heslo zakaždým zadávať pri sťahovaní e-mailov. Je to síce bezpečnostné riziko, hlavne v prípade, keď sa k nášmu počítaču dostane nepovolaná osoba, ale na druhú stranu je to určite pohodlnejšie.

Obrázok 16 zobrazuje názorné nastavenie e-mailového konta priamo na školský poštový server a obrázok 17 znázorňuje príklad nastavenia e-mailového konta na prihlasovací mimoškolský poštový server:

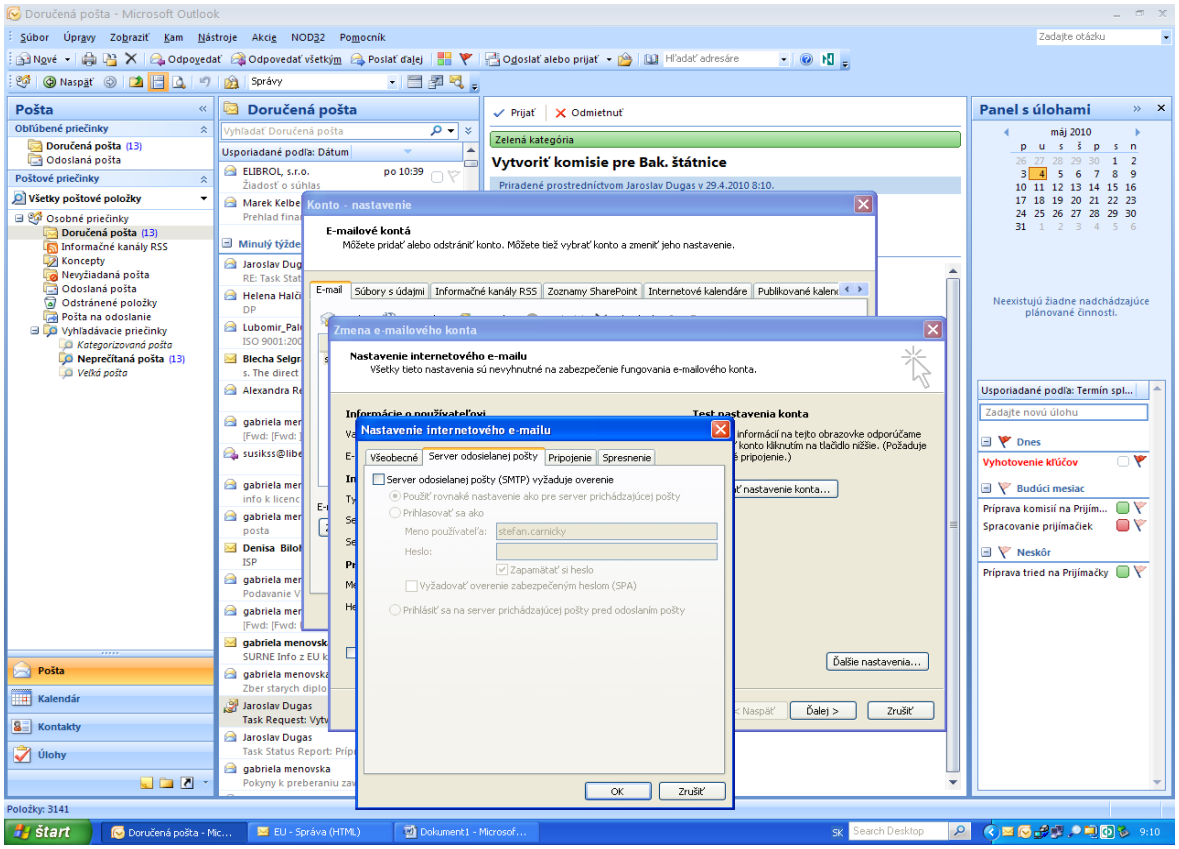



Obrázok 16 Názorné nastavenie e-mailového konta na školský poštový server

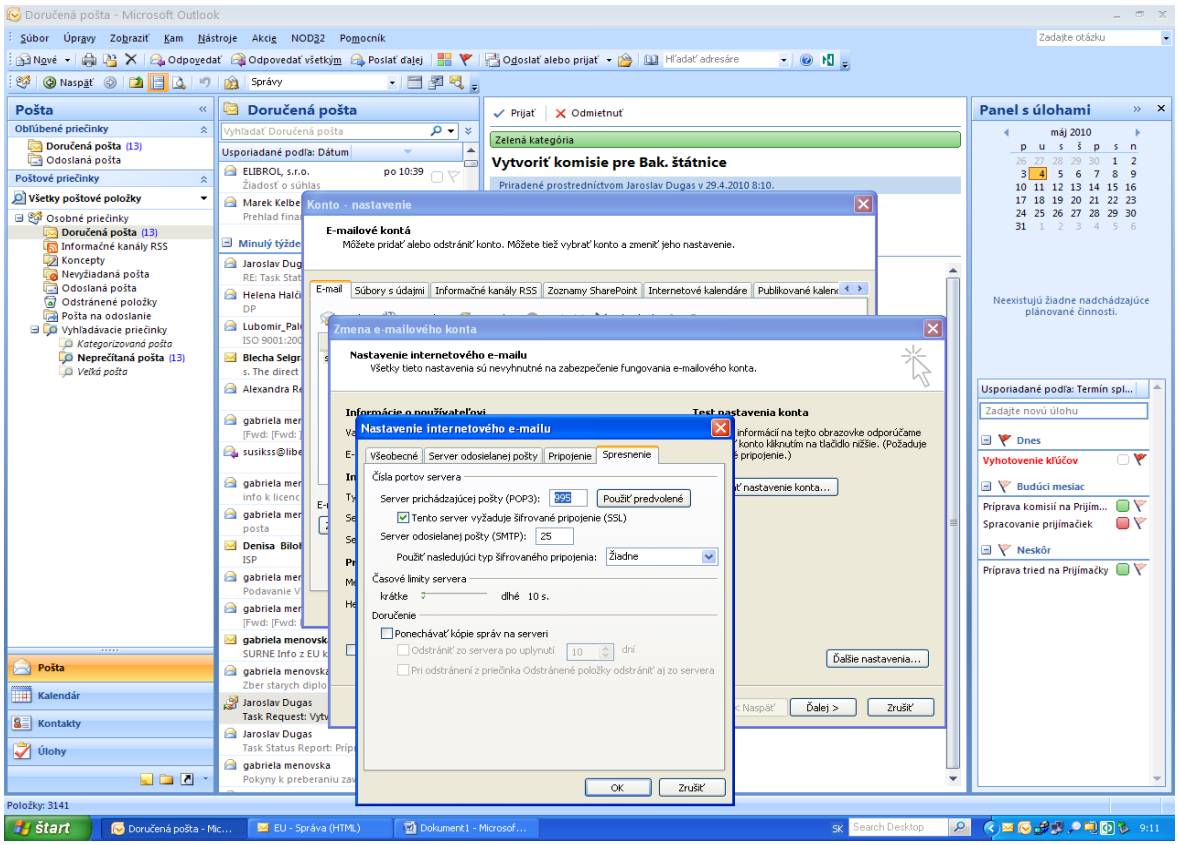



Obrázok 17 Názorné nastavenie e-mailového konta na prihlasovací mimoškolský poštový server

Pritom je veľmi dôležité mať vyplnené aj ďalšie náležitosti, viď obrázok 18 a obrázok 19.

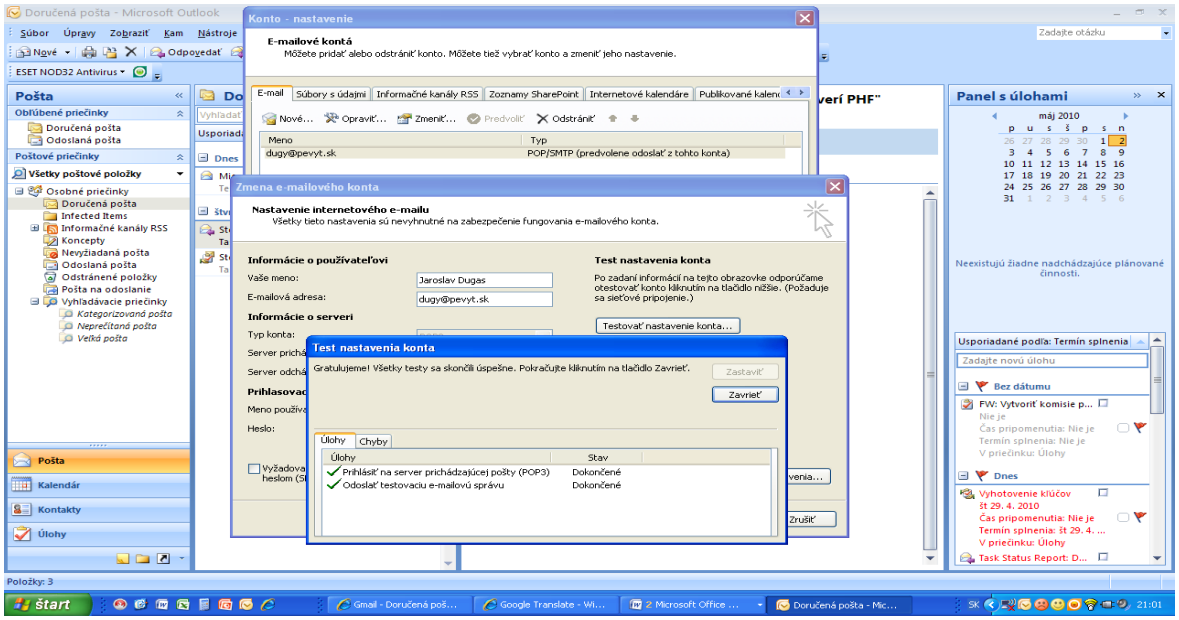



Obrázok 18 Nastavenie internetového e-mailu - server odosielanej pošty




Obrázok 19 Nastavenie internetového e-mailu - server prichádzajúcej pošty

Akonáhle máme všetko správne nastavené, môžeme otestovať, či sa Outlook k e-mailovej schránke správne pripojí - klikneme na ikonu *Testovať nastavenia konta* (obrázok 20).




Obrázok 20 Test nastavenia internetového konta

Pokiaľ všetko prebehne v poriadku, užívateľ prejde do rozšíreného nastavenia *Ďalšie nastavenia* a tu môže na záložke *Spresnenie* ovplyvniť, či sa budú e-maily na serveri po stiahnutí mazať, alebo či tam zostanú. V prípade, že sa chceme k schránke prihlasovať aj odinakiaľ, a to cez webový prehliadač, doporučujeme kópiu ponechávať.

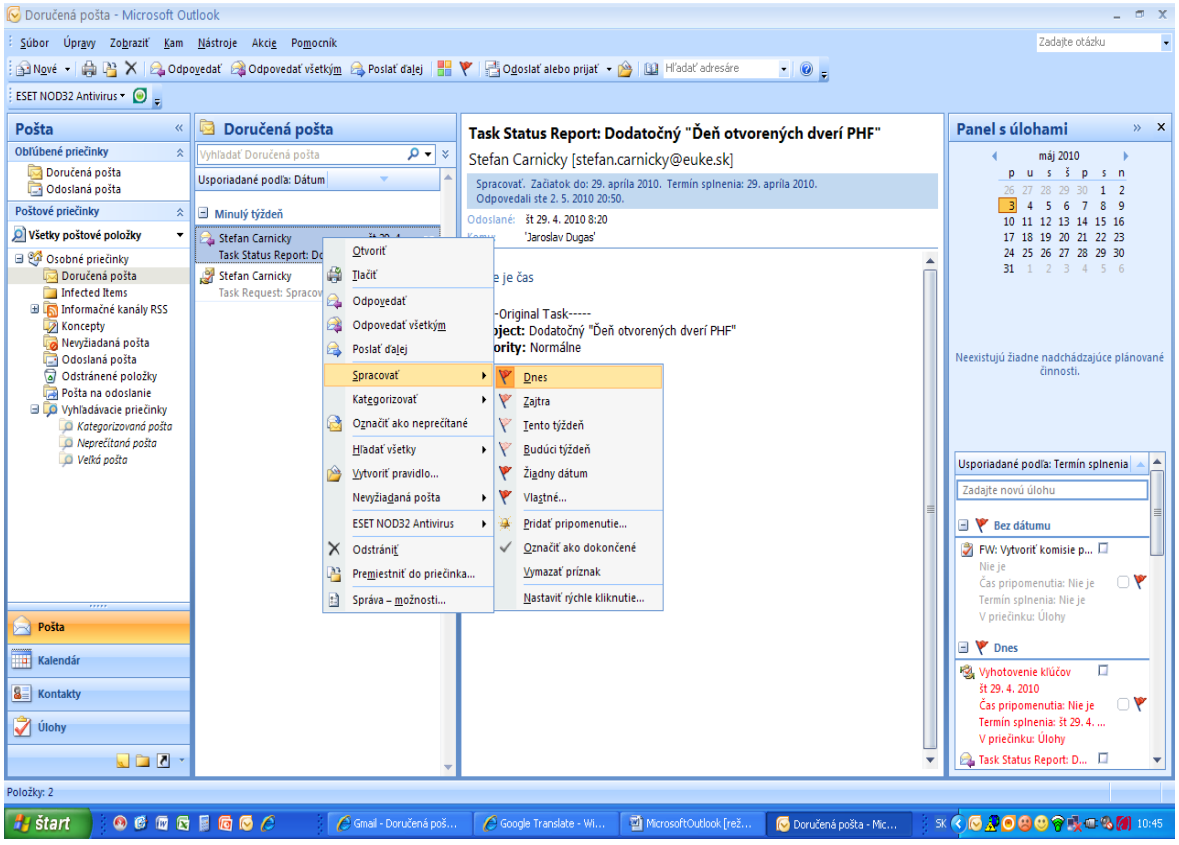
V prípade, že SMTP server, ktorý používame, požaduje overenie pred zaslaním e-mailu, zadáme meno a heslo na záložke *Server odosielanej pošty*. Ak sú údaje rovnaké ako pre príjem správ, zaškrtneme položku *Použiť rovnaké nastavenia*. V opačnom prípadne kolónky pre meno a heslo musíme vyplniť.

Po dokončení konfigurácie jednoducho klikneme na ikonu *Ďalej* a potom na *Dokončiť*. V zozname účtov sa objaví nový, práve pridaný účet, pritom ho môžete kedykoľvek upraviť inkonou *Opraviť*, *Zmeniť* alebo zmazať ikonou *Odstrániť*. Potom sa po návrate do hlavného okna Outlooku okamžite začnú sťahovať e-maily zo servera do priečinka *Doručená pošta*.

## Pokročilá práca s e-mailami

Ak budeme používať Microsoft Office Outlook e-mail ako pracovný nástroj, je možné aby nám pomocou neho ***zadával „nadriadený“ nejaké úlohy***. Tie si môžeme zadať do zoznamu úloh, ktorý je súčasťou Outlooku, čo popisujeme nižšie, alebo môžeme ***označiť priamo daný e-mail na ďalšie spracovanie - priradíte mu „zástavku“***. Od tohto okamihu sa správa chová v podstate ako úloha, ktorú musíme splniť.

*„Zástavka“*, teda príznak na spracovanie, môžeme pridať dvoma spôsobmi. Buď vo výpise správ klikneme na daný e-mail pravým tlačidlom myši a z kontextovej ponuky vyberieme príkaz *Spracovať*, alebo správu otvoríme a z pásu ponúknutých kariet vyberieme ikonu *Spracovať*. Potom vyberieme termín, do ktorého sa má úloha zadaná v našom e-maili splniť - zajtra, pozajtra a pod., prípadne zadáme priamo dátum pomocou voľby *Vlastné*. Ak nemá úloha termín, jednoducho zadáme *Žiadny dátum* (obrázok 21).

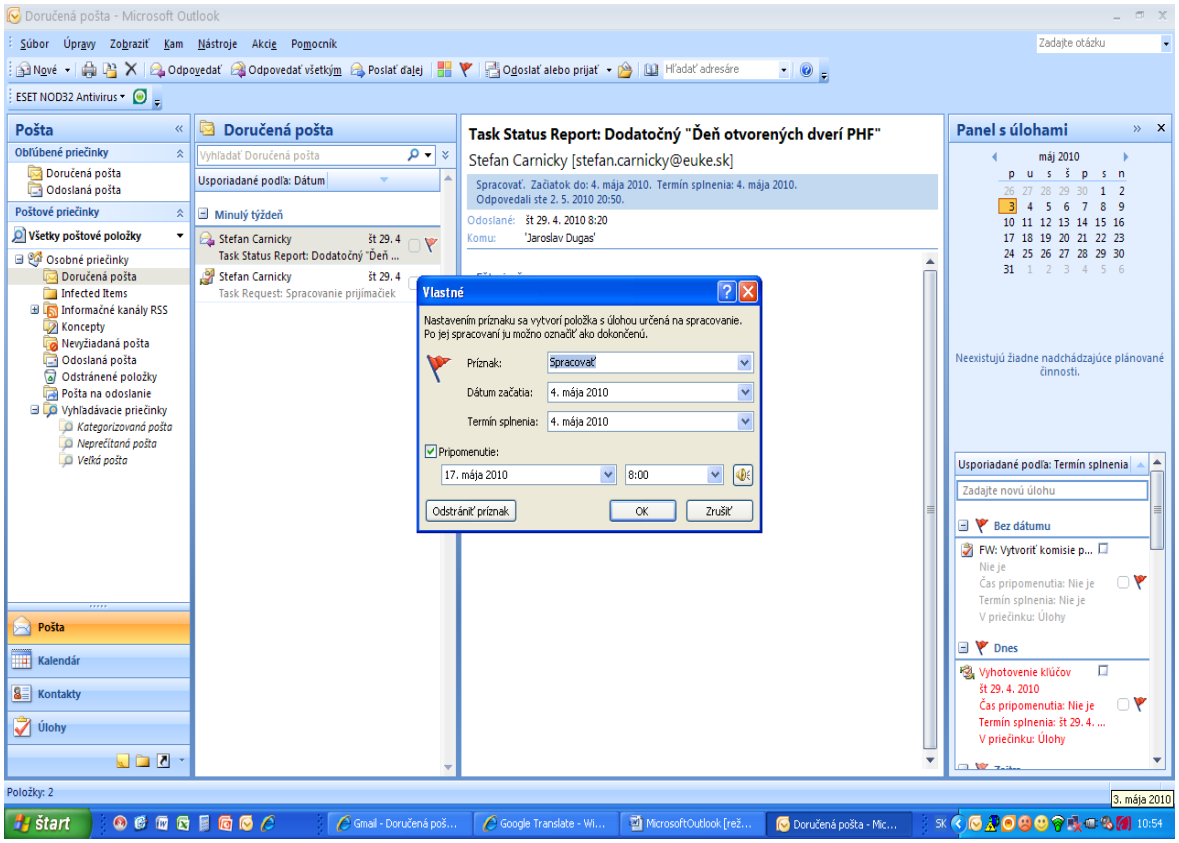



Obrázok 21 Zadanie termínu splnenia úlohy

Od tejto chvíle bude v zozname správ zobrazený u daného e-mailu príznak - „*zástavka*“. V prípade, že nastal deň termínu splnenia, zatiaľ ľahko priehľadná zástavka

zmení farbu na sýto červenú - čiže je jasné, že úlohu je potrebné splniť, prípadne na danú správu musíme odpovedať. Akonáhle je úloha splnená, môžeme príznak odstrániť, prípadne označiť úlohu ako splnenú.

Ak si chceme nastaviť pripomenutie na iný deň, než v ktorý má byť úloha splnená, vyberieme z miestnej ponuky príkazu *Spracovať* voľbu *Pridať pripomenutie*. Tu tiež môžeme zmeniť typ príznaku, miesto *Spracovať* zadáme napríklad *Potrebné pre vedenie*, *Dôležitá informácia*, alebo *Poslať ďalej*. Je to vhodné aj v prípade, ak chceme poslať niekomu e-mail s určitým príznakom, napríklad svojmu podriadenému, a tak môžeme v správe nastaviť príznak *Spracovať* a dať konkrétny termín, kedy má byť zadaná úloha hotová (obrázok 22).



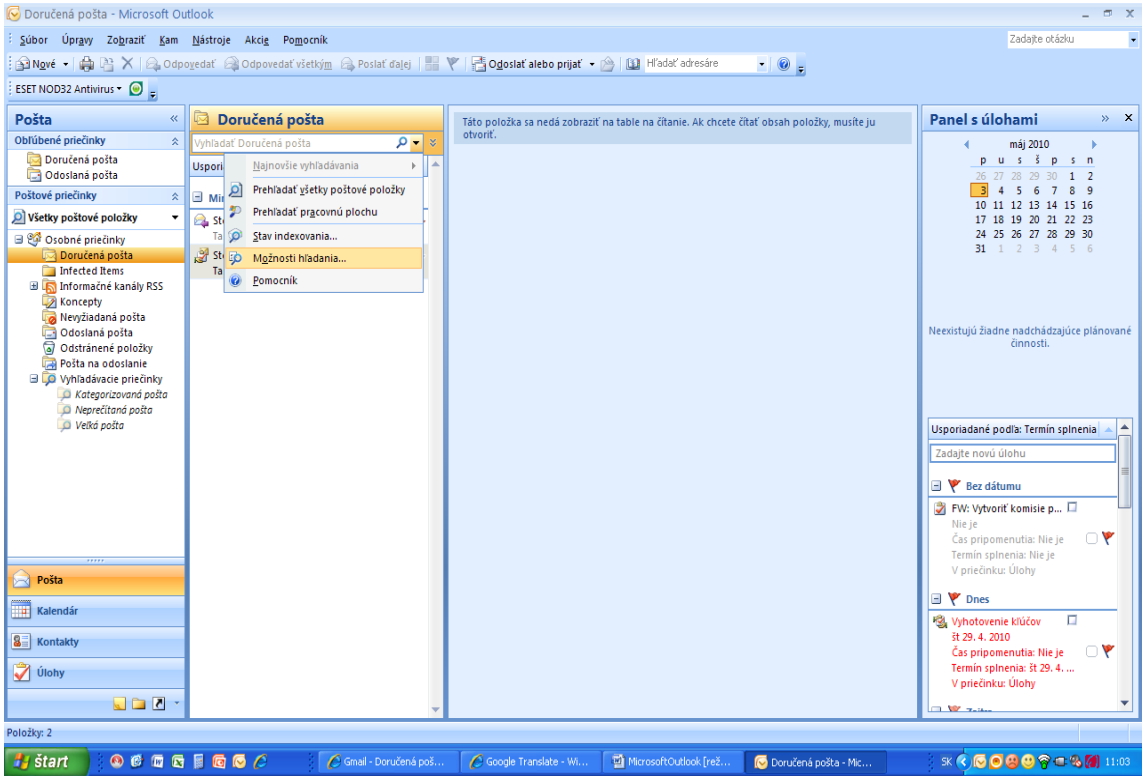

Obrázok 22 Možnosti spracovania úlohy

## Vyhľadávanie a triedenie

Ak svoj e-mail používame skutočne aktívne, nahromadí sa nám veľmi rýchlo veľké množstvo správ, ktoré je nutné nejako triediť, katalogizovať. Nie všetky správy možno ihneď vymazať, niektoré uložíme na vybavenie, niektoré vyhodíme do koša, ale

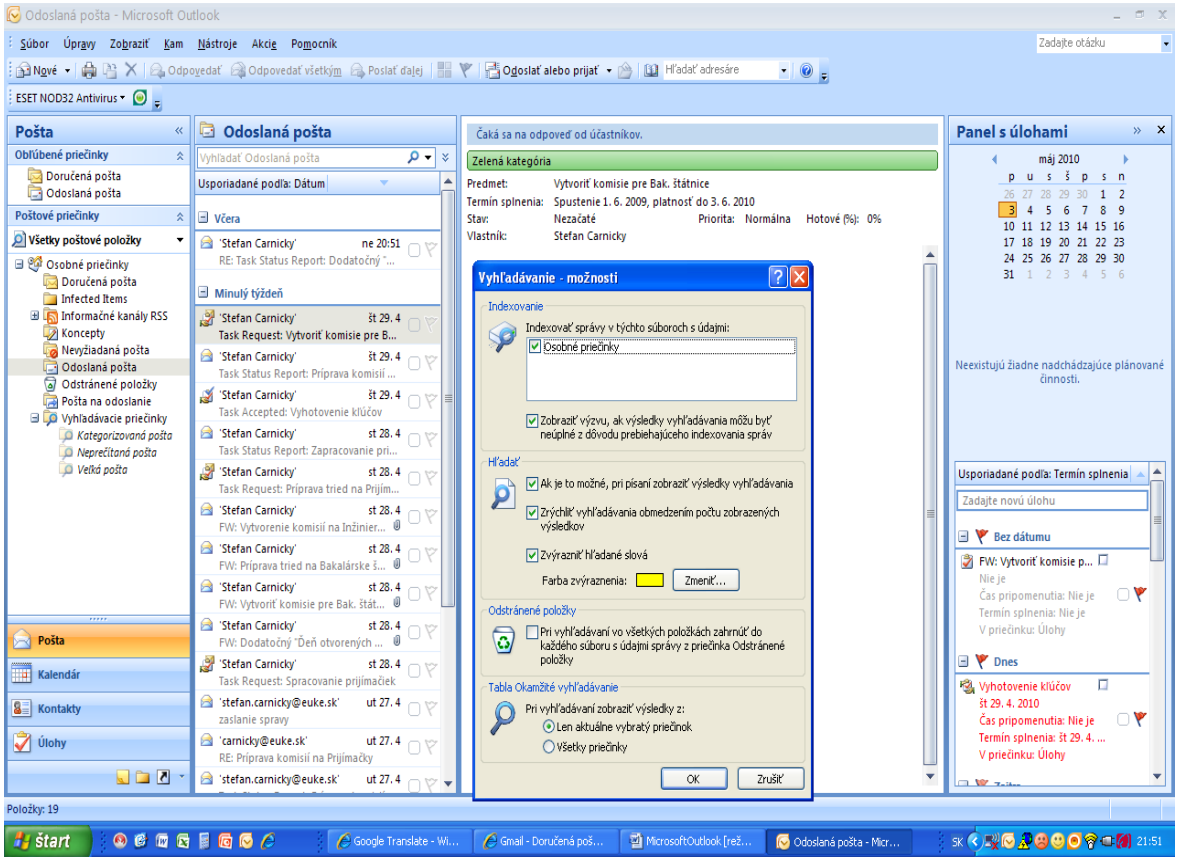
nevymažeme ich úplne - čo ak by sme sa k nim v budúcnosti ešte museli vrátiť? Akonáhle ale počet správ presiahne hoci len pár stoviek, prestaneme sa v nich celkom iste orientovať. Čo teda s tým? Napríklad, ak potrebujete nájsť e-mail, u ktorého síce poznáte obsah, ale nevieme, kto nám ho poslal. V takom prípade je ideálne použiť vyhľadávanie. Do políčka označeného lupou (nad výpisom správ) zadáme kľúčové slovo, prípadne slová, o ktorých sme si istí, že sa v e-maili vyskytovali. Outlook následne vyhľadá všetky správy, ktoré toto slovo obsahujú v tele alebo predmete. Ak je v predmete, bude žlto zvýraznené. Prehľadávať možno jednotlivé zložky osobitne (*Doručená pošta*, *Odstránené položky* atď.). Ak nám nestačí základné hľadanie, klikneme na malú ikonou so šípkou nadol. Zobrazí sa nám dialóg, v ktorom môžeme zadať upresnenie, v ktorých častiach e-mailu slovo hľadáme (obrázok 23).

Aby sme mohli používať pokročilé hľadanie, je potrebné mať nainštalovaný balík Windows Desktop Search - ak nemáme tento doplnok nainštalovaný, zobrazí sa pod oknom pokročilého hľadania odkaz na inštaláciu. Ak chceme povoliť *Okamžité hľadanie*, klikneme naň. Po kliknutí sa zobrazí web Microsoftu, odkiaľ doplnok (4,5 MB) stiahneme, a ak máme administrátorské právomoci aj nainštalujeme. Vďaka pokročilému hľadaniu môžeme podstatne lepšie špecifikovať svoju prípadnú otázku.



Obrázok 23 Vyhľadávanie e-mailov pomocou kľúčových slov

Nastavenie hľadania si môžeme zobraziť výberom položky *Možnosti hľadania*. Je možné zmeniť - napríklad akou farbou sa má zvýrazniť výsledok hľadania - *Farba zvýraznenia*, ďalej či sa má hľadať okamžite pri písaní, alebo či sa má hľadať aj v koši, alebo jednoducho si zvolíme hľadanie vo všetkých priečinkoch (obrázok 24).



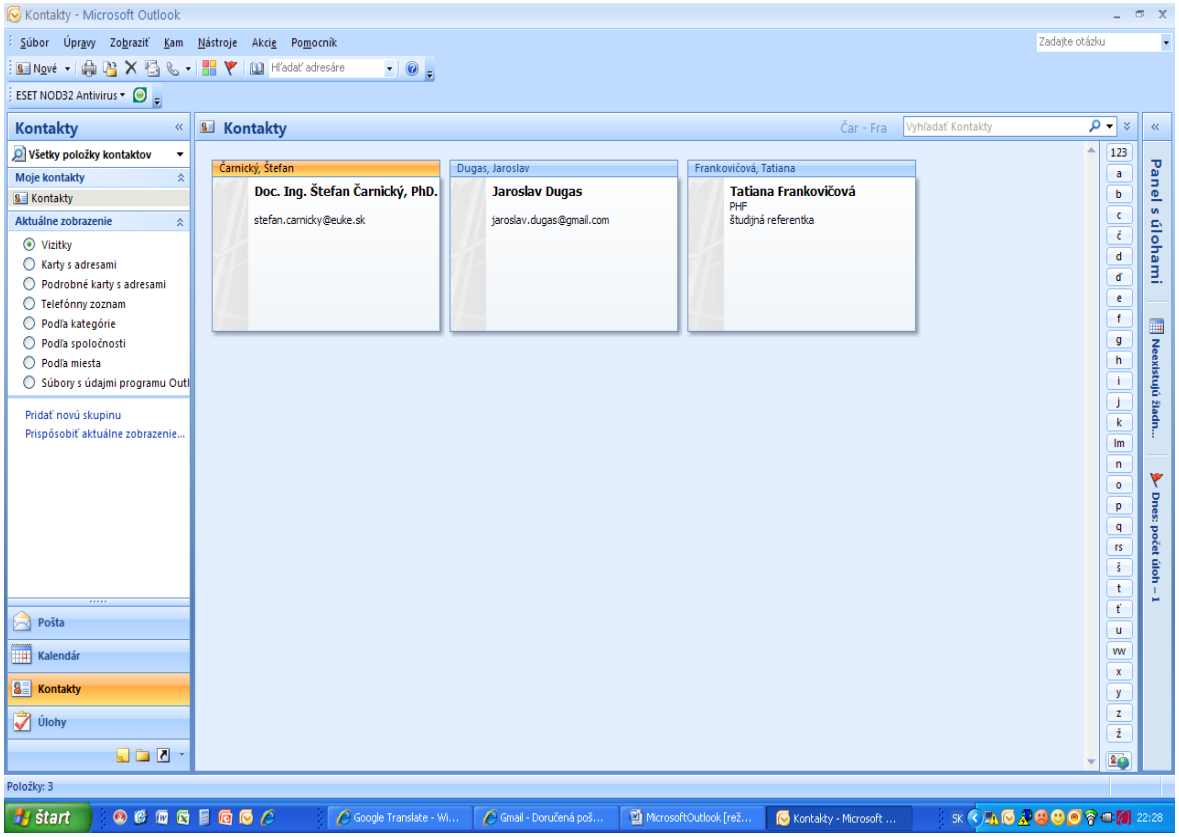

Obrázok 24 Nastavenie možností vyhľadávania

Outlook neslúži len na odosielanie a prijímanie pošty, je to aj veľmi dobrý organizačný nástroj. Dôležitá je predovšetkým správa kontaktov, organizácie času a úlohy. Kontakty možno využiť nielen pre ukladanie e-mailových adries ľudí, s ktorými si často píšeme, ale môžeme si do nich uložiť aj telefónne čísla, adresy a poznámky - napríklad funkciu daného človeka, alebo aktuálny stav komunikácie.

## Radenie kontaktov

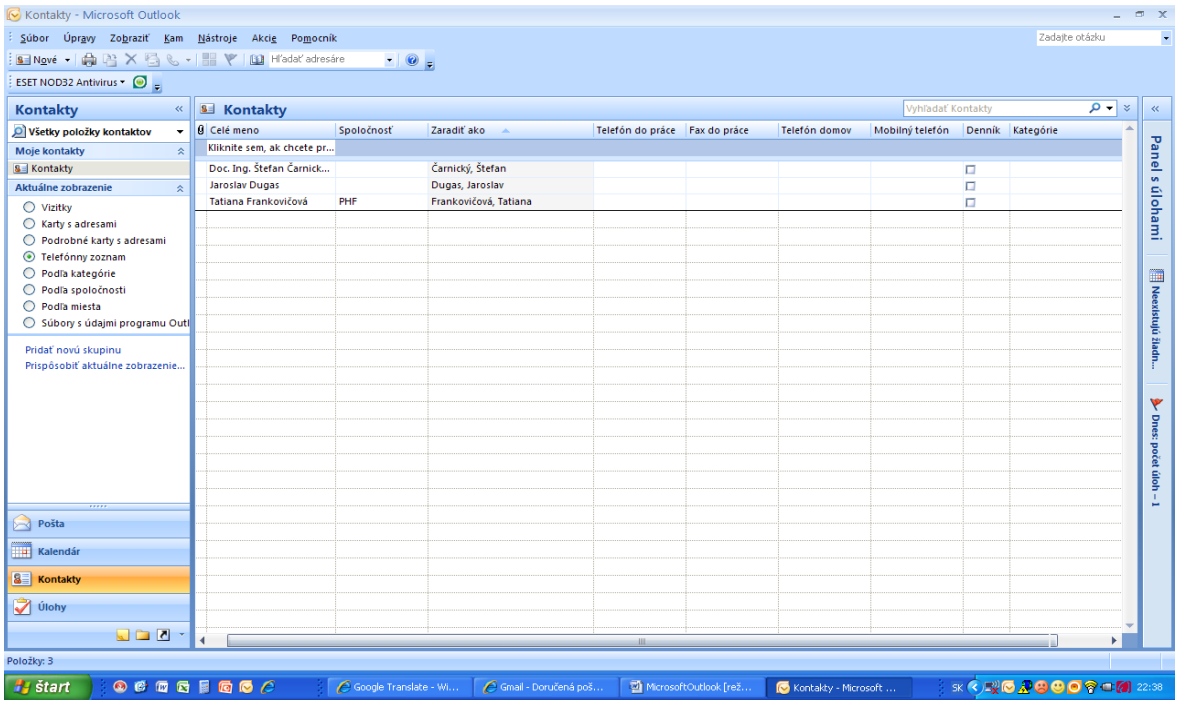
Kontakty zobrazíme kliknutím na položku *Kontakty* v ľavom paneli. Ak nie je panel zobrazený celý, musíme kliknúť na *Aktuálne zobrazenie* a rozbalí sa nám ďalšia ponuka. V hlavnom okne sa teraz zobrazí prehľad kontaktov zobrazených ako vizitky -

každý kontakt je vyobrazený v obdĺžniku, kde sú uvedené rôzne základné informácie ako meno, adresa, e-mail, telefón a pod. Ak máme kontaktov viac, poteší nás možnosť rýchleho vyhľadávania podľa začiatočných písmen mena v stĺpci napravo od vizitiek (obrázok 25).



Obrázok 25 Vyhľadávanie kontaktov

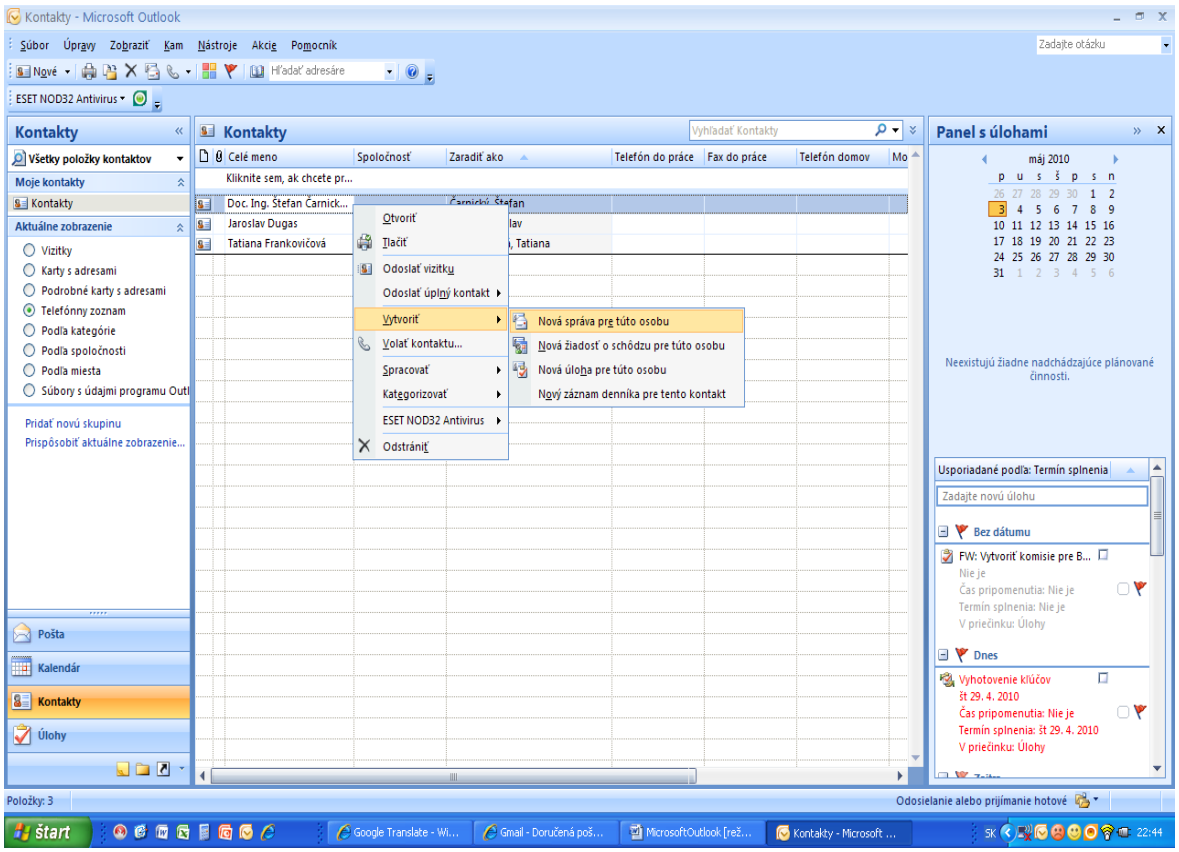
Tento pohľad ale nie je príliš prehľadný, ďaleko lepšie je sa na kontakty pozerať ako na telefónny zoznam - teda na tabuľku zoradených mien pod sebou, hoci každý kontakt má pre seba vyhradený len jeden riadok. Toto zobrazenie je ďaleko úspornejšie, pretože sa na stránku zmestí oveľa viac mien. Jednotlivé pohľady možno prepínať v ľavom stĺpci (obrázok 26).




Obrázok 26 Zoznam kontaktov

**Možnosti poslania správy priamo konkrétnemu „kontaktu“**

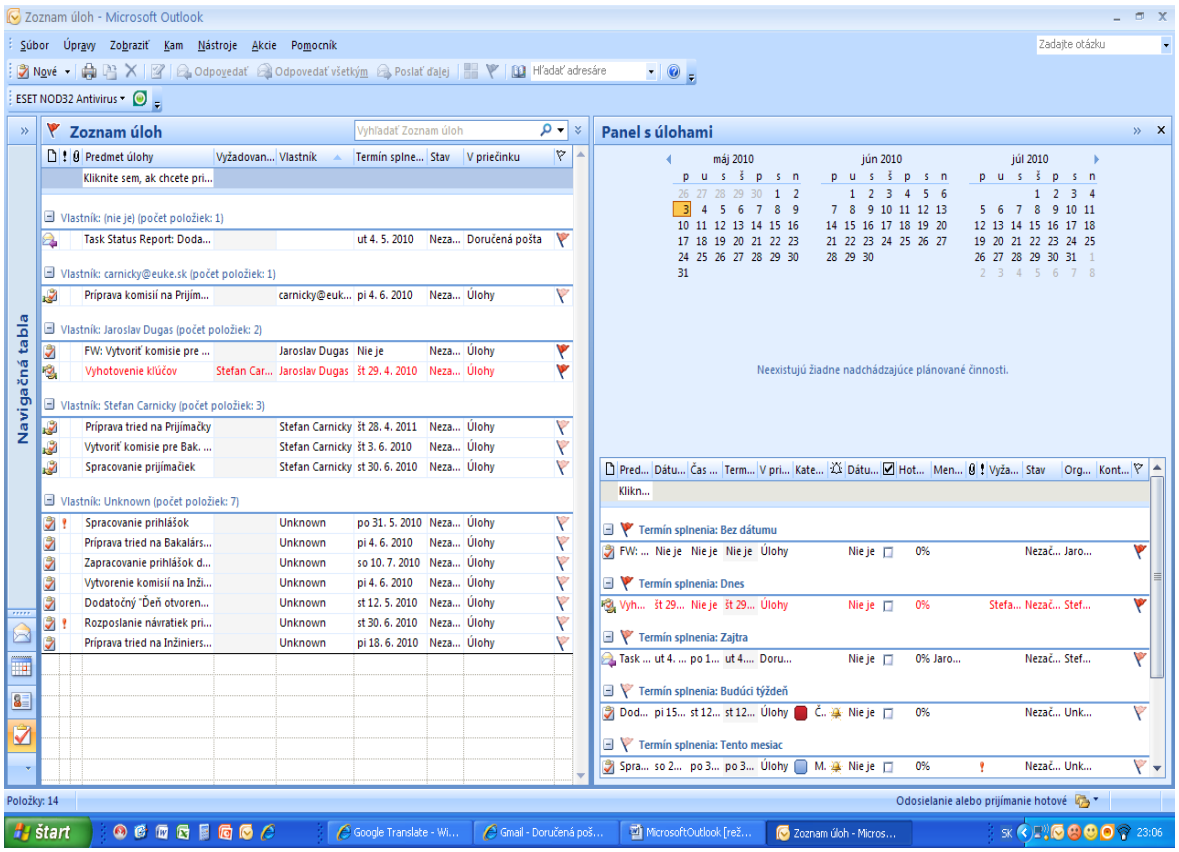
Aby sme nemuseli zakaždým, keď niekomu posielame správu, vypisovať znova e-mailovú adresu, klikneme v dialógu s novou správou na ikonu *Komu*, prípadne *Kópia*. Zobrazí sa nám zoznam všetkých kontaktov, ktoré boli do nášho Outlooku zadané. Napísať správu kontaktu ale možno aj inak - v zozname kontaktov si vyberieme osobu, ktorej chceme napísať e-mail, a z kontextovej ponuky vyberieme príkaz *Vytvoriť* a potom *Nová správa pre túto osobu*. Týmto spôsobom možno osobe zadať aj *Úlohu* a prípadne ju *Pozvať na schôdzku* (obrázok 27).




Obrázok 27 Možnosti poslania správy alebo úlohy konkrétnej osobe

**Organizácia času - úlohy osobne zadávané sami sebe, aby sme na nič nezabudli**

Ak nám nestačí kúsok papiera, na ktorý si zapisujeme „akútne úlohy“, ktorých splnenie aj čakáme, môžeme sa spoľahnúť na Outlook. Jeho úlohy sú prepracované, možno im nastavovať rôzne termíny, ktoré potom samotný Outlook stráži, môžeme im meniť prioritu a zaraďovať ich do rôznych kategórií. Toto dokáže časť Outlooku 2007 *Panel s úlohami*. Ten možno zobraziť napravo od výpisu správ a môže zobrazovať kalendár s aktuálnym dátumom a tučne vyznačenými dňami, keď sa niečo deje, ďalej zoznam udalostí v kalendári a zoznam aktuálnych úloh, prípadne e-mailov, ktoré sme si označili na spracovanie zástavkou (obrázok 28).




Obrázok 28 Zoznam aktuálnych úloh

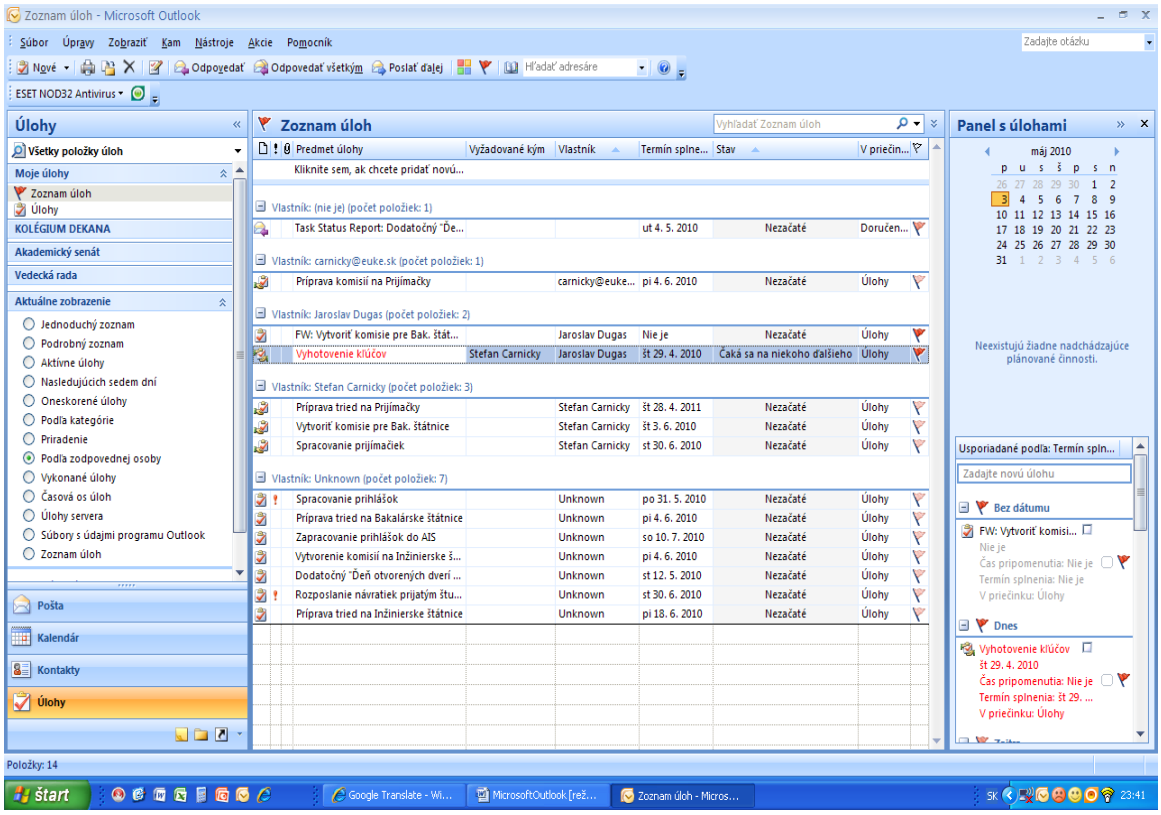
Ak nám *Panel úloh* zaberá príliš veľa miesta, môžeme ho minimalizovať tak, aby bol stlačený len do malého úzkeho prúžku. Potom na ňom ale nič neuvidíme, resp. uvidíme len nadchádzajúcu udalosť a počet nesplnených úloh. Minimalizáciu vykonáme kliknutím na šípku >> vpravo hore, obnovenie vykonáme pomocou šípky <<. Krížik × potom panel úplne zruší a je potrebné ho obnoviť z ponuky *Zobraziť* | *Panel s úlohami* | *Normálne*.

## Zadanie novej úlohy

Do prehľadu úloh sa dostaneme kliknutím na ikonu *Úlohy*, ktoré je umiestnené v ľavej dolnej časti Outlooku. Ak je ľavý panel minimalizovaný, treba sa nastaviť na ikonu v notese s „červenou potvrdzujúcou značkou“. Úlohy sa vypíšu v hlavnom okne rovnako ako e-maily - štandardne sú zobrazené hlavné stĺpce ako predmet, dátum začatia a termín splnenia, prípadne kategórie a termín pripomenutia.

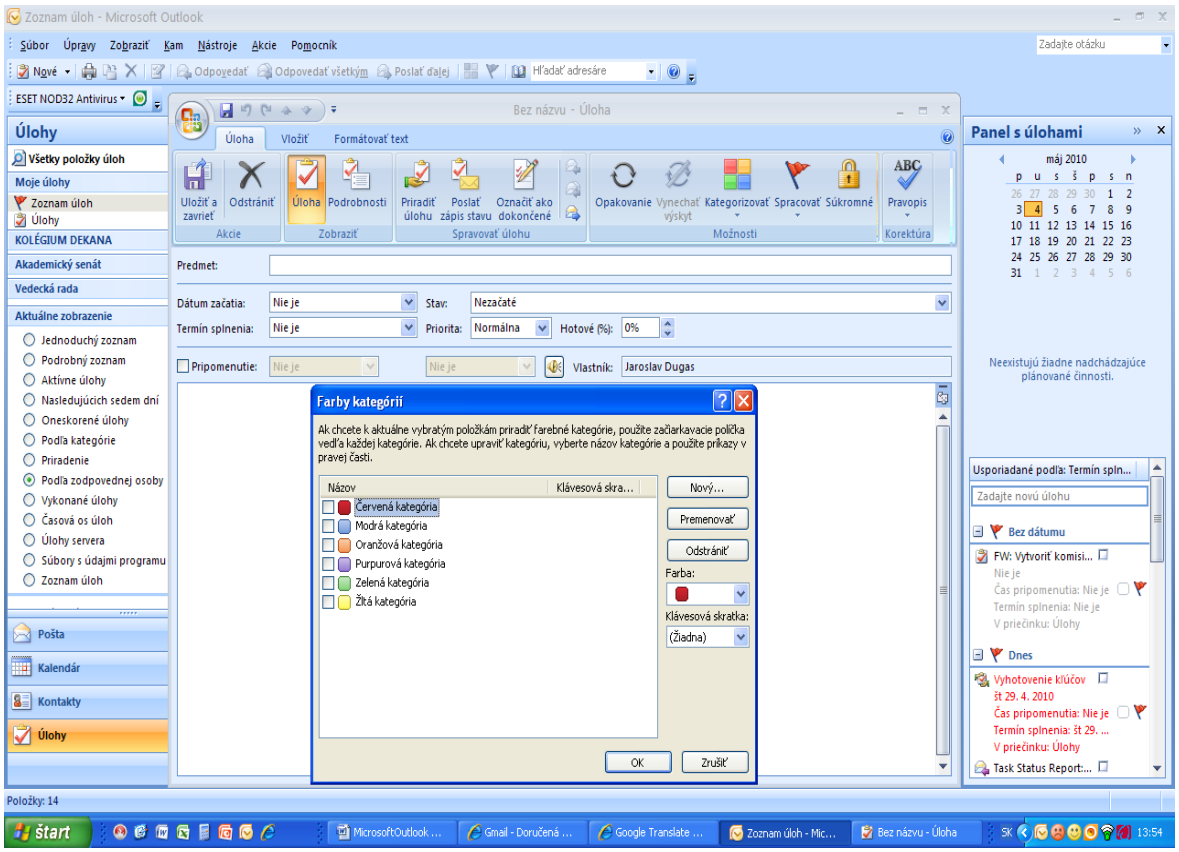
Novú úlohu zadáme do databázy pomocou klávesovej skratky *Ctrl + N*, prípadne z panela nástrojov voľbou *Nové* | *Úloha*. Každá úloha má niekoľko hlavných položiek (obrázok 29):

- ***predmet***: meno úlohy, napríklad „pripraviť miestnosť na štátnice“,
- ***dátum začatia***: dátum, kedy sa na úlohe začalo pracovať,
- ***termín splnenia***: deň, do kedy musí byť úloha dokončená,
- ***stav***: fáza, v ktorej sa práve úloha nachádza - môžeme si tak pomerne dobre označiť, ako sa nám plniť úlohy darí; možnosti sú: *Nezačaté*, *Prebieha*, *Dokončené*, *Čaká na niekoho ďalšieho* a *Odložené*,
- ***hotovo (%)***: **percentuálne** vyjadrenie stavu splnenia - možno využiť v kombinácii so stavom; akonáhle na úlohe začneme pracovať, nastavíme stav *Prebieha* a položku *Hotovo* pri dokončení zadanej úlohy - toto môžeme meniť podľa toho, ako sa nám darí úlohu plniť,
- ***kategória***: každú úlohu môžeme zaradiť do určitej kategórie, takže sa v nich potom lepšie vyznáme - môžeme si ich zoradiť podľa toho, či ide o úlohy pracovné, súkromné alebo iné,
- ***pripomenutie***: aby sme nezabudli, že sa blíži dátum ukončenia, môžeme nastaviť pripomenutie na deň aj presnú hodinu a ešte si vybrať zvuk, ktorý následne Outlook prehrá; hneď v onen deň pustíme Outlook a ak nastane hodina H, vyskočí dialóg s pripomenutím a prehrá sa nastavený zvuk.




Obrázok 29 Zadanie novej úlohy a jej riešiteľa

Pre založenie novej úlohy stačí len vyplniť predmet, všetko ostatné možno vynechať. Ak budeme chcieť pridať aj kategóriu, klikneme na ikonu *Kategorizovať* na paneli *Možnosti*. Štandardne sú tu predvolené kategórie líšiace sa len farbou. Pri prvom zvolení danej kategórie však dostaneme možnosť ju premenovať a súčasne môžeme zmeniť aj farbu, rovnako ako zvoliť nejakú klávesovú skratku - napríklad úlohy prichádzajúce z Kolégia dekana červená farba, Akademický senát modrá farba, atď. Ak zo zoznamu kategórií zvolíme *Všetky kategórie*, zobrazí sa dialóg *Farebné kategórie*, kde môžeme ľubovoľne meniť názvy kategórií a ich farbu, prípadne založiť novú kategóriu, ak nám tie existujúce už nebudú stačiť (obrázok 30).



Obrázok 30 Triedenie úloh podľa kategórií

Každej úlohe možno doplniť aj poznámku, resp. vysvetľujúci text, ktorý úlohu bližšie opisuje a dopĺňa. V tomto okne môžeme text rovnako ako v e-maile ľubovoľne upravovať pomocou štandardných funkcií na páse kariet *Formátovať text*. Na záložke *Vložiť* sú zase funkcie pre vloženie prílohy, obrázka, grafu a ďalších prvkov Office - rovnice, textové polia a podobne.

**Úloha pre niekoho iného**

Toto všetko sa týkalo úloh, ktoré chceme zadať sami sebe. Na PHF EU ale potrebujeme zaviesť určitý systém zadávania a preberania úloh, a to znamená, že Microsoft Office Outlook budeme aktívne využívať na zadávanie úloh nielen svojmu kolegovi alebo podriadenému, ale podľa vopred dohodnutých pravidiel hlavne medzi sebou.

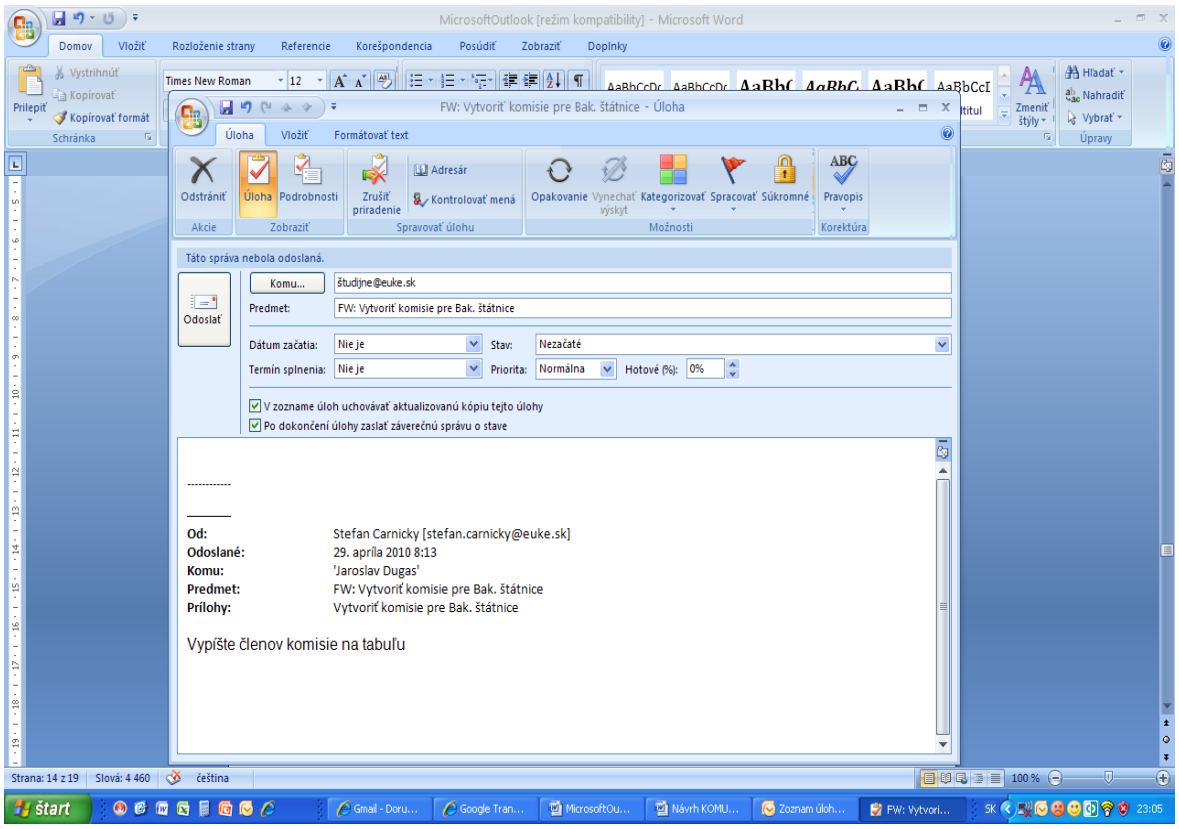
V rámci **Návrhu vytvorenia KOMUNIKAČNÉHO SYSTÉMU na PHF EU z vlastných zdrojov a pre zavedenie „Rozposielania úloh" na PHF EU** potrebujeme:

1. určiť pracovníkov z radov PHF EU zodpovedných za tvorbu úloh,
2. určiť pracovníkov z radov PHF EU zodpovedných za zadávanie a priraďovanie úloh konkrétnym osobám,
3. určiť okruh pracovníkov z radov PHF EU – adresátov úloh.

Po splnení všetkých týchto podmienok a hlavne po odborne a administrátorsky nainštalovanom Microsoft Office Outlook 2007 na PC daných zodpovedných pracovníkov zapojených do komunikačného systému potrebujeme ešte mať nastavený Microsoft Office Outlook 2007 ako predvolený „poštový nástroj", ošetrený bezpečnostnými certifikátmi, a potom sa môžeme naplno pustiť do zadávania úloh pre niekoho iného: klikneme na ikonu *Priradiť úlohu* v páse kariet. To pridá novú položku - *Komu*: sem zadáme e-mail osoby, ktorej chceme a zároveň aj máme kompetencie danú úlohu zadať.

Navyše pribudli dve voľby (obrázok 31):

- ***V zozname úloh udržiavať aktualizovanú kópiu tejto úlohy*** - úlohu bude mať v zozname ten, komu ju zadávame, aj jej zadávateľ. Budeme tak vidieť, ako je úloha postupne plnená.
- ***Po dokončení úlohy zaslať záverečnú správu o stave*** - teda ak druhá strana úlohu splní, dostaneme o tom správu, a na druhej strane akonáhle sme my vyplnili všetky požadované položky, môžeme úlohu uložiť do zoznamu vlastných, prípadne ju podľa kompetencie ***odoslať na splnenie tretej osobe***.




Obrázok 31 Zadanie konkrétnej úlohy

### Radenie úloh

Aby sme sa v úlohách vyznali, najmä ak ich máme veľa, môžeme použiť rad rôznych ponúkaných výpisov. Všetky sú zobrazené v ľavom paneli v sekcii *Prispôsobiť aktuálne zobrazenie*. Asi najprehľadnejší je posledný výpis *Zoznam úloh*, kde sú úlohy zoradené pekne podľa kategórie a následne podľa termínu splnenia, pričom sú viditeľné len tie úlohy, ktoré neboli splnené. Ak si ale necháme zobraziť filter *Podľa kategórie*, uvidíme to isté, ale aj s hotovými úlohami - tie sú v zozname ***prečiarknuté***.

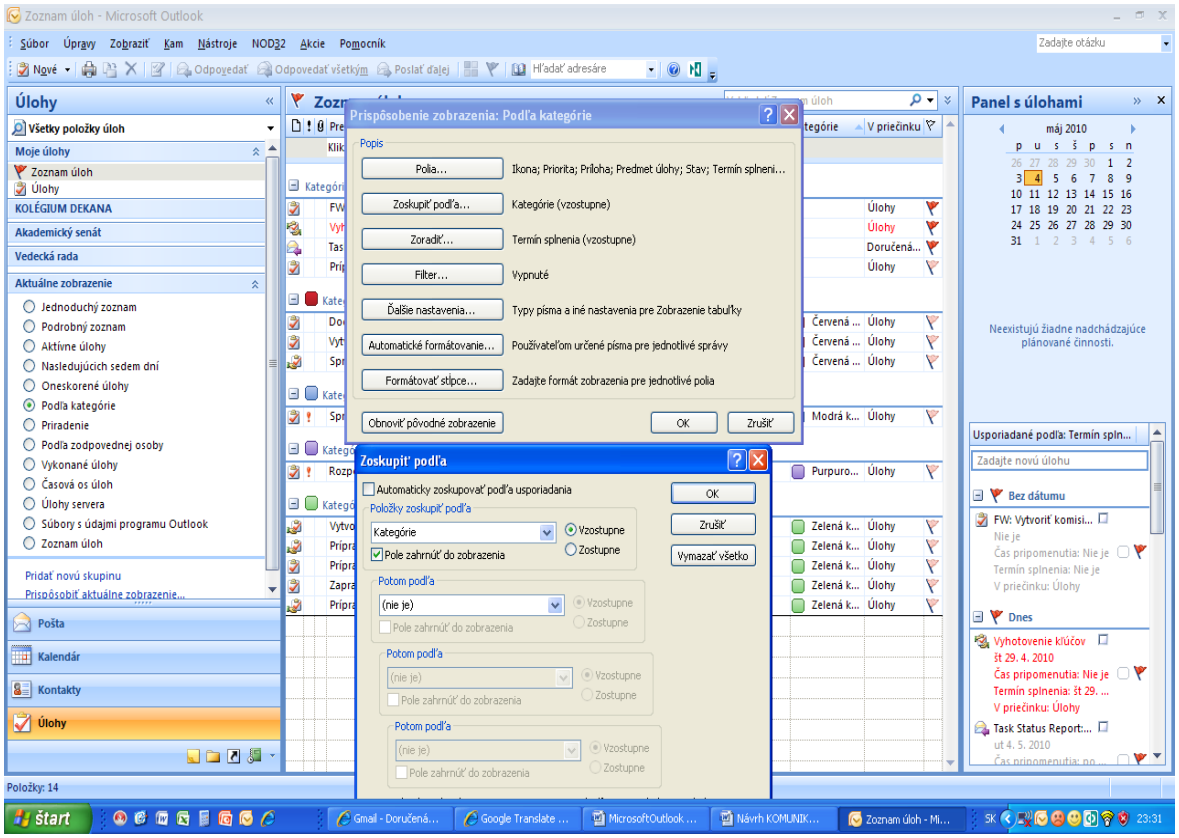
Jednotlivé filtre zobrazenia nie je potrebné podrobnejšie opisovať, sám užívateľ si vyskúša, čo mu vyhovuje najviac.

### Dokončenie úlohy

Akonáhle úlohu dokončíme, označíme ju aj v zozname, nech nám už neprekáža. ***Nie je vhodné ju mazať nastálo, pretože históriu splnených úloh môžeme ešte potrebovať.*** Stačí, keď na úlohu v zozname klikneme a z ponuky vyberieme príkaz *Označiť ako dokončené*, ak máme zapnutý výpis iba aktuálnych úloh, ten dokončený

zmizne. Viditeľný bude iba vtedy, keď si zobrazíme akýkoľvek filter, ktorý ukazuje aj splnené úlohy; napr. *Ukončené úlohy* (obrázok 32).

Úlohu možno splniť aj vtedy, ak jej stav nastavíme na 100%.




Obrázok 32 Triedenie úloh podľa zadefinovaných kategórií

V prípade mobilnej komunikácie z pracovných notebookov s prístupom mimo PHF EU bude nutné zabezpečiť možnosť spúšťania Microsoft Office Outlook 2007 ako „hlavného poštového nástroja“. Veľmi dôležité tiež bude napríklad ošetrenie „Príkazom dekana“ povinnosť práce s daným komunikačným nástrojom, prípadne časom zapracovanie „Rozposielania úloh“ na PHF EU do internej smernice ako zadávacieho a kontrolného nástroja riadenia fakulty.

**Sumárne základ Komunikačného systému na PHF EU bude tvoriť Microsoft Office Outlook 2007 ako súčasť univerzitného inštalačného balíka Microsoft Office:**

- Po samotnej inštalácii balíka Microsoft Office a po úvodnom spustení aplikácie Microsoft Office Outlook 2007 je možné okamžité jednoduché nastavenie „konta“

každého užívateľa na jeho pracovnom PC (stolový počítač, notebook).

- Microsoft Office Outlook 2007 môže jednoducho od začiatku inštalácie slúžiť ako e-mailový nástroj - ale až po správnom nastavení.
- Ďalej je možné aj to, že Microsoft Office Outlook 2007 vôbec nemusí byť používaný ako e-mailový nástroj - hneď pri inštalácií sa jednoducho zvolí možnosť „nekonfigurovania“ a jednoducho „zaškrtnutím“ pokračovania bez e-mailovej podpory a stlačením ikony *Dokončiť* sa ukončí úspešná inštalácia Otlooku 2007.
- V prípade potreby práce na webe nemusíme opúšťať aplikáciu Microsoft Office Outlook a ani nemusíme prechádzať z jednej aplikácie do druhej, aby sme si mohli prezerať hocijakú webovú stránku. Táto aplikácia nám to bezproblémovo umožní. Ak nebudeme teda využívať v aplikácii Microsoft Office Outlook 2007 *POŠTU*, kolónky zostanú prázdne.
- Aj kategóriu *Kalendár* budeme mať po čerstvej inštalácii prázdnu.
- Dôležité bude tiež naplniť kategóriu *Kontakty*. Pritom Microsoft Office Outlook ponúka veľa možností pri vyplňovaní zadávacích dát konkrétnej osoby do *Kontaktov*.
- Najdôležitejšiu kategóriu tvoria *Úlohy* s možnosťou jednoduchého prehľadu akejkoľvek úlohy - hlavne tej nesplnenej!
- Neoceniteľnou výhodou komunikačného systému bude zadanie opakovateľnosti, aby sme budúci rok úlohe priradili len napríklad nový termín, komisiu, zodpovednú osobu a podobne.
- Pre komunikačný systém na PHF EU - akýsi „AUTOMAT“ - by nemal byť problém úlohy priradiť aj niekoľko dní po sebe, alebo vytvoriť rôzne užívateľské časové generovanie rôznych zadaných úloh.
- Aby sme mohli vytvoriť na PHF EU model komunikačného systému a nastaviť optimálny „AUTOMAT“, musíme mať na to nástroj, teda softvér, nainštalovaný na jednom serveri (obrázok 33).

  Je to softvér Windows Small Business Server, pôvodne navrhnutý pre malé organizácie, ktorý umožňuje pracovať so súčasnými technológiami postavenými na najlepších skúsenostiach spoločnosti Microsoft a poskytuje serverové komplexné manažérske riešenie za prijateľnú cenu.

Obrázok 33 Logo softvéru Windows Small Business Server (http://www.microsoft.sk)

**Výhody Microsoft Small Business Servera (http://www.microsoft.sk):**

- chránený a spoľahlivý operačný systém,
- kvalitné riešenia aj pre malé organizácie (nízka obstarávacia cena),
- je možné vzdialene sa pripojiť (cez internet) na ktorýkoľvek počítač alebo server v sieti (cez vzdialené webové pracovisko),
- centrálna správa e-mailov (cez Exchange) - e-maily sú v Outlooku viditeľné v rovnakom stave ako cez webové rozhranie (ak sa napríklad z internetovej kaviarne pripojíte do práce na e-mail a odpoviete na nejakú správu, následne pri zapnutí počítača v kancelárii uvidíte, že ste e-mail čítali, odpovedali, a vidíte aj konkrétnu odpoveď),
- zdieľanie súborov, tlačiarní a rôznych aplikácií,
- interný firemný web (intranet) pre zdieľanie dokumentov a informácií (Sharepoint Services),
- zdieľané zadefinované kalendáre v Outlooku (členovia katedry si môžu navzájom vidieť do časového harmonogramu - veľmi efektívny nástroj napríklad pre sekretárky vedúcich katedier - možnosť dohovárania času schôdzok a podobne),
- poštové schránky s neobmedzenou veľkosťou,
- automatizovaná správa aktualizácií Microsoftu pre celú fakultnú sieť,
- denné správy o stave počítačov (funkčnosť, zapojenie v sieti PHF EU) a serverov so softvérom od spoločnosti Microsoft a možnosť centrálnej správy počítačov v sieti pomocou „Active Directory“ a „Správy zásad skupiny“ (GPO),
- zdieľaný fax - odosielanie z ktoréhokoľvek počítača, príjem na e-mail alebo na ktorúkoľvek zdieľanú tlačiareň,
- možnosť obnovy zmazaných e-mailov,
- možnosť obnovy zmazaných dokumentov (Shadow Copy).

## 4.2 Prípadová štúdia 2

Prípadovú štúdiu 2 tvorí návrh implementácie **Lotus Notes**. Aj tu sa nám podarilo odskúšať si vytváranie, rozposielanie, plnenie úloh a ich následné kontrolovanie aj vyhodnocovanie, a to priamo v ostrej prevádzke vo francúzskej spoločnosti VALEO Slovakia, s. r. o. Košice (ďalej len VALEO). Táto medzinárodná spoločnosť využíva produkt Lotus Notes (ďalej len Lotus) vo všetkých svojich výrobných závodoch a pobočkách po celom svete.

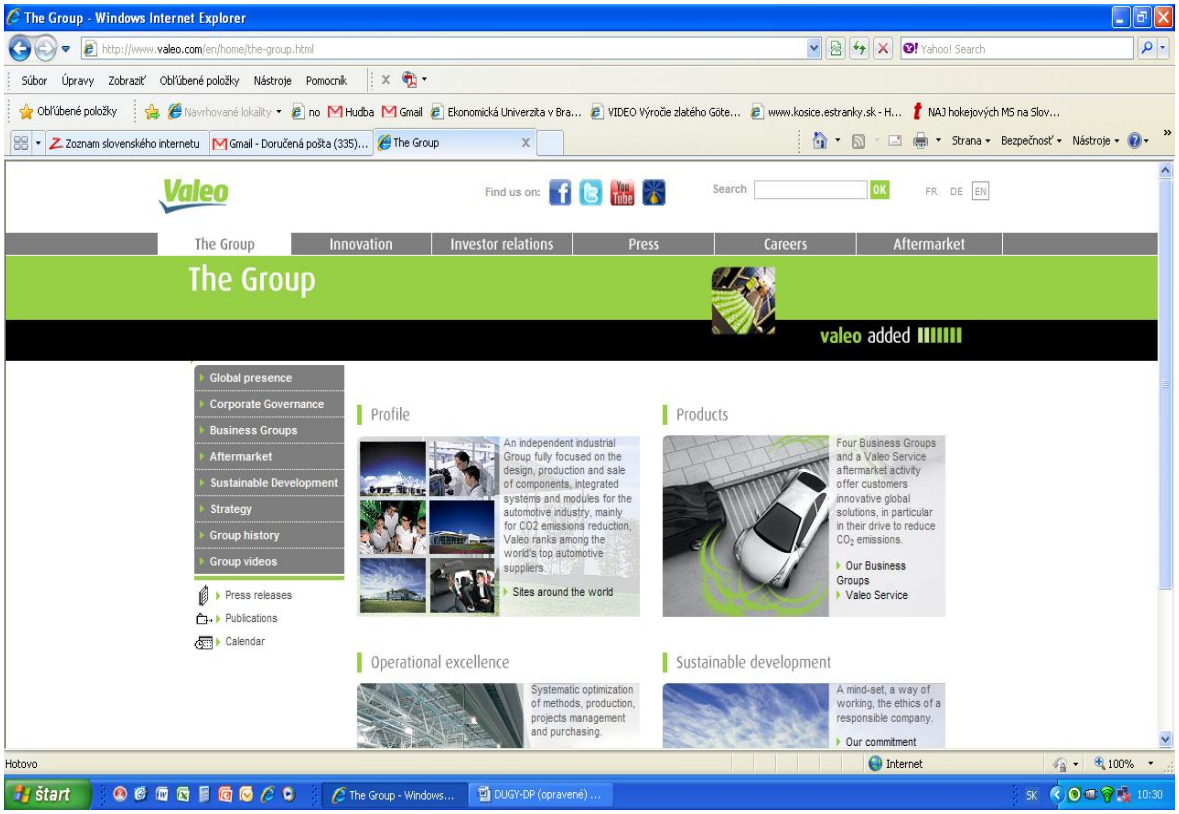
V každej spoločnosti sú v systéme zadefinované konkrétne skupiny pracovníkov z určitých oddelení, ktorí sú zodpovední za plnenie zadefinovaných úloh. Ich plnenie je potom vyhodnocované hneď v prvý pracovný deň nasledujúceho mesiaca. Toto plnenie úloh, hlavne keď je úspešné, je ďalšou čiastkou príplatku ku mzde - sú to odmeny podľa percentuálneho počtu vyriešených úloh. Úloha je úspešne vyriešená, keď je úspešne dokončená a v systéme zaznamenaná tak, že bola splnená do 24 hodín. Podotýkam - nestačí úlohu splniť hneď, ale musí byť aj v systéme označená, že bola úspešne dokončená, a ak sa v systéme nestihne označiť, tak sa nezapočítava do percentuálneho mesačného hodnotenia.

Ak je úloha úspešne splnená a aj v systéme je záznam o jej úspešnom splnení, oznam o dokončení úlohy sa posiela aj zadávateľovi tejto konkrétnej úlohy a ten môže a nemusí túto úlohu „odobriť“. Ak ju zadávateľ úlohy neodobrí, automaticky systém vyhodnotí, že je úloha splnená na 100 % a že neexistuje dôvod, prečo túto úlohu nemá započítavať do mesačného hodnotenia. Ak ovšem zadávateľ úlohy zadanú úlohu nepovažuje za splnenú a to aj v systéme označí, úloha sa nezapočítava do zoznamu úspešne ukončených úloh, ale stále figuruje v systéme ako nesplnená, čo aj systém na záver mesiaca vyhodnotí.

Ako aj ďalej ukážeme, úlohu je možné delegovať inej osobe, alebo aj zadanú úlohu úplne zrušiť, ak je buď už zadaná v systéme niekým iným, alebo jej splnenie nie je možné vykonať pretože je nejasne zadefinovaná alebo je jednoducho nesplniteľná. O všetkých týchto delegovaniach a zrušeniach exitujú záznamy a sú evidované v mesačnom hodnotení.

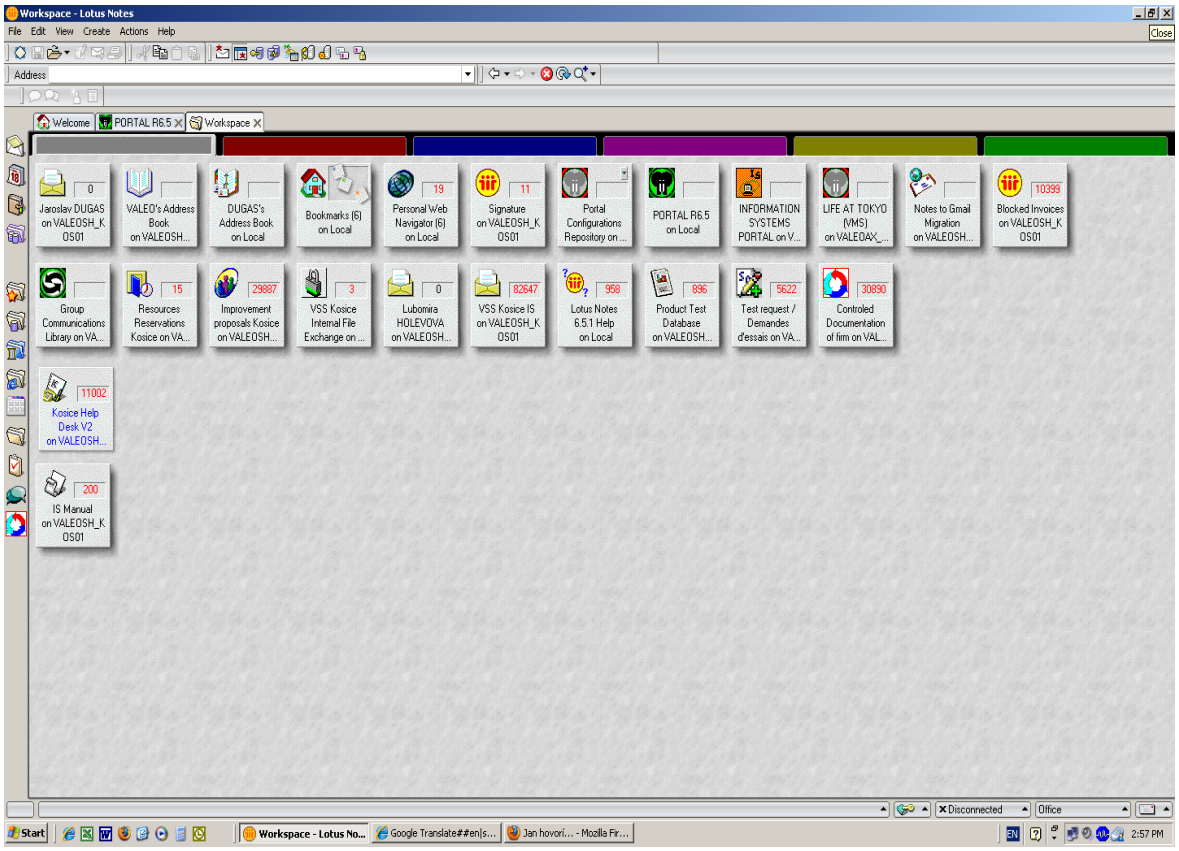
Ešte minulého roku bolo v košickom závode na príplatok ku mzde potrebných 80 % úspešne vyriešených a ukončených úloh do 24 hodín. Tohto roku sa pre košický závod

sprísnili kritéria a stúplo percento úspešne uzatvorených úloh na 90 %. Pritom napríklad v poľských štyroch závodoch zostalo 80 %, a dokonca vo francúzskych závodoch je dlhodobé kritérium splnených úloh nastavené len na 70 %. Všetky nasledujúce obrázky sú robené pomocou Print Screenu priamo na pridelenom osobnom počítači v spoločnosti VALEO.




Obrázok 34 Print screen medzinárodnej webovej stránky spoločnosti VALEO (http://www.valeo.com)

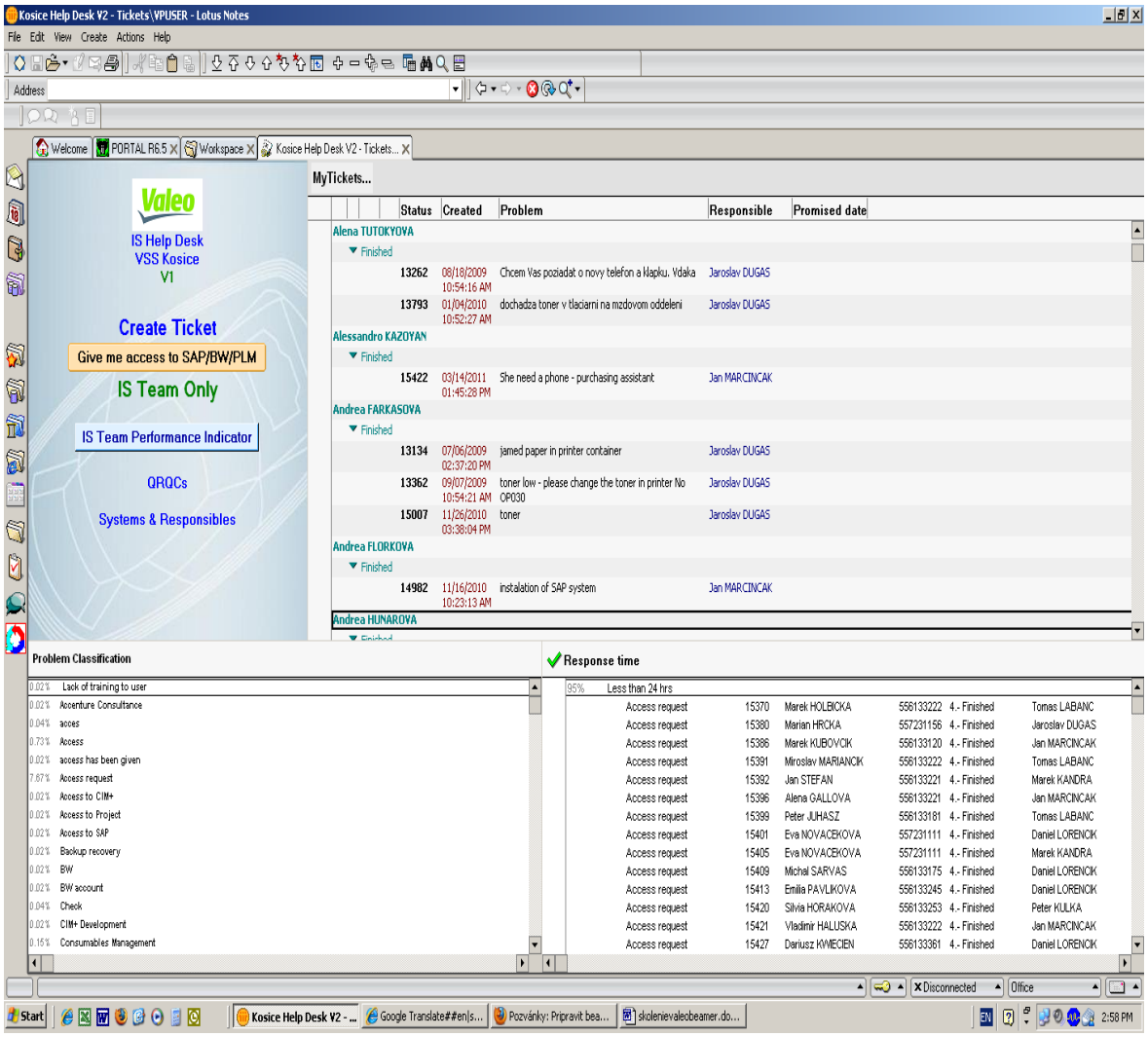
Po zadaní správneho prístupového hesla sa každému užívateľovi otvorí úvodné užívateľské okno Lotusu (obrázok 35).




Obrázok 35 Úvodné užívateľské okno Lotusu

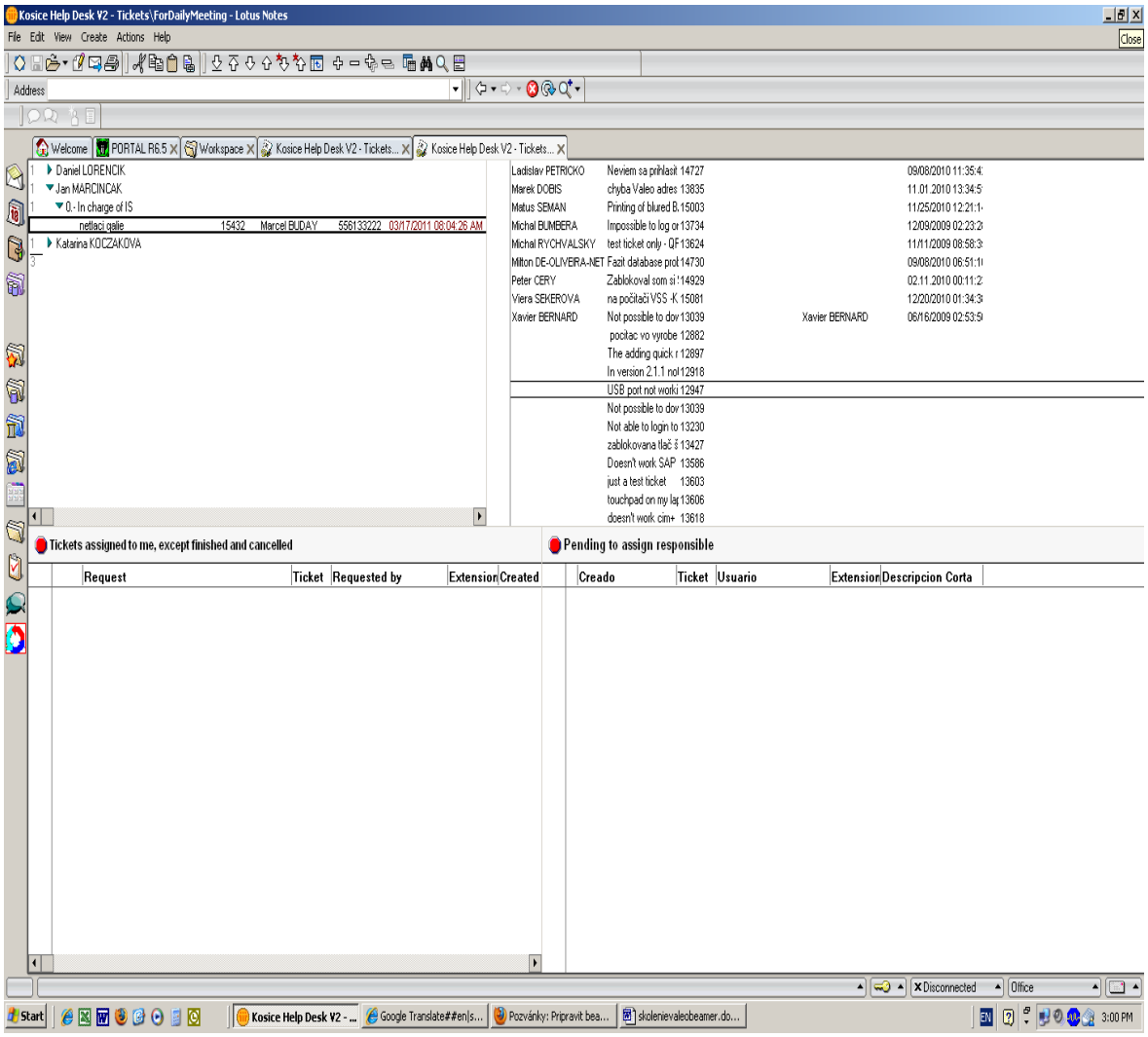
Po spustení programu sa každému užívateľovi otvorí jeho zadefinované pracovné lotusovské prostredie obsahujúce poštový priečinok - e-mailový adresár všetkých zamestnancov, adresár zamestnancov na konkrétnej pobočke, osobný adresár, ktorý si zadefinuje sám užívateľ, menný zoznam všetkých zamestnancov vo všetkých svetových pobočkách s konkrétnym pracovným zaradením, komunikačný portál, portál s databázou, ktorá je prístupná len z počítačov zadefinovaných v Active directory Valeo, pomocník v prostredí Lotusu, Improvement proposals - nástroj na zadávanie zlepšovacích návrhov, databáza pracovných manuálov, nástroj na rezervovanie školiacich miestností a výučbových nástrojov, školiaca - výuková databáza a model Help Desk - prostriedok na zadávanie konkrétnych úloh konkrétnemu užívateľovi.

V košickom závode sa používa Help Desk VSS Kosice, konkrétne pre odbor Informačné technológie (IT) je to, ako vidíme na obrázku 36, IS HELP Desk VSS Kosice V1 a V2.



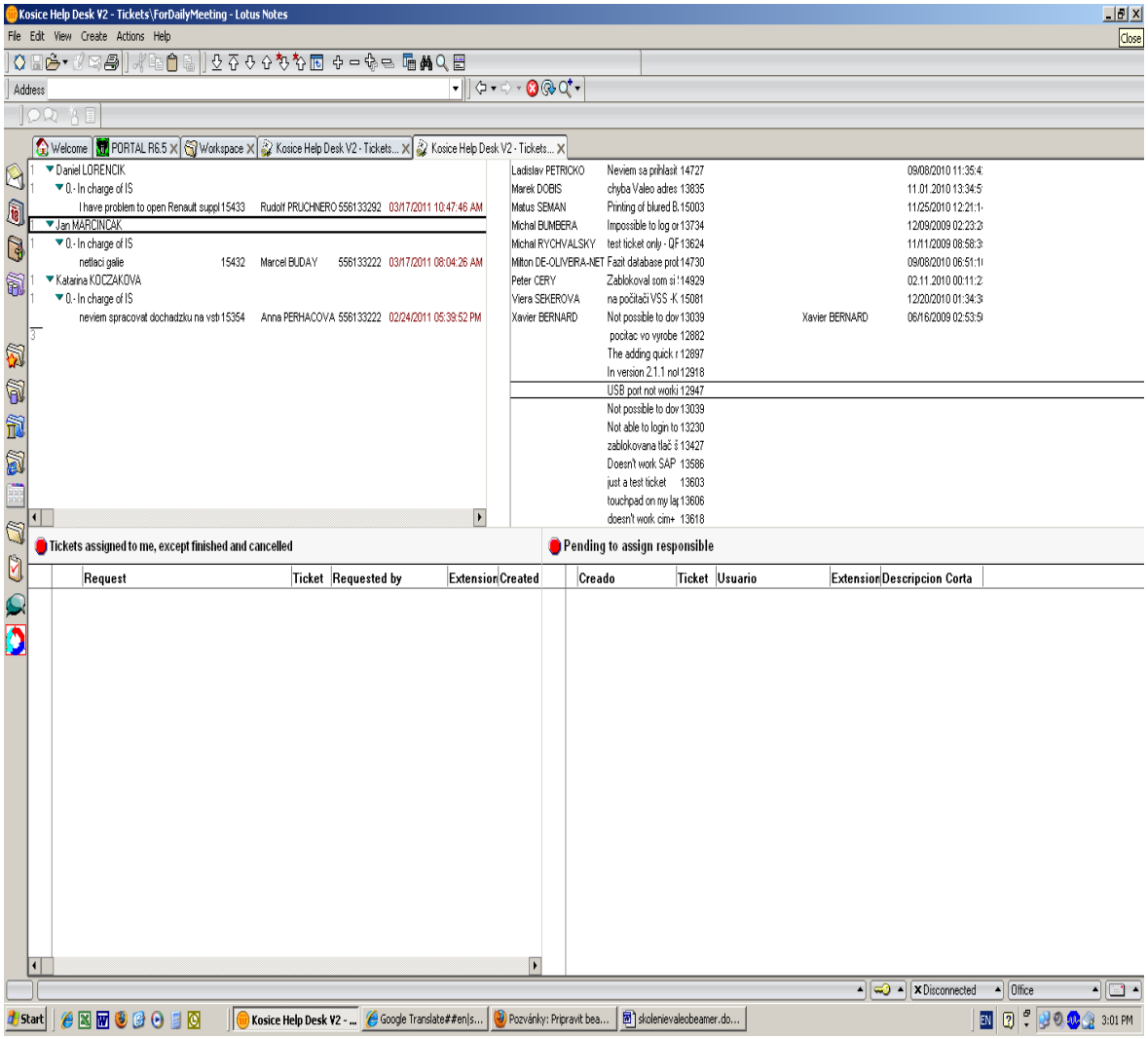

Obrázok 36 Nástroj na zadávanie úloh Help Desk VSS

Po odkliknutí *IS Team Only* sa nám rozbalí zoznam tých pracovníkov IT, ktorým boli zaslané konkrétne úlohy, kde pred každým menom je číslo, ktoré udáva, koľko úloh bolo zaslaných tomuto konkrétnemu zamestnancovi a teda koľko úloh by mal vyriešiť (obrázok 37).




Obrázok 37 Zoznam riešiteľov úloh

Ak si odklikneme každého jedného zamestnanca, rozbalí sa nám skrátený význam jednotlivých zadaných úloh. Pritom ak zadávateľ úlohy potrebuje nejakú konkrétnu pomoc, môže jednoducho kliknúť na preddefinovanú možnú oblasť problému a jednoducho označí prázdne „koliesko“, ktoré je umiestnené pred danou možnou problémovou oblasťou. Ako vidíme na obrázku 38, môže ísť napríklad o problém s Cisco telefónom, s laserovou tlačiarňou, s monitorom, s PC a jeho komponentmi, s RF Gunom - so skenerom čiarových kódov, s čítačkou bezpečnostných kariet, alebo so Zebra tlačiarňami, ktoré sú umiestnené priamo vo výrobnej hale a sú určené na tlač čiarových kódov a informácií o obsahu balíčka, krabice, bedničky alebo celej palety.




Obrázok 38 Skrátený zoznam úloh konkrétnych riešiteľov

Zadávateľ úlohy napíše skráteným záznamom do poľa *Problem Description* a v poli *Detailed Information* buď popíše svoj problém, alebo priamo zadá konkrétnu úlohu na vyriešenie. Niekedy zadávateľ úlohy ani nevie, komu sa úloha posunie na vyriešenie, a dozvie to až vtedy, keď dostane od riešiteľa odpoveď či bola úloha vyriešená, či zariadenie funguje, alebo ako má pokračovať vo svojej ďalšej práci.

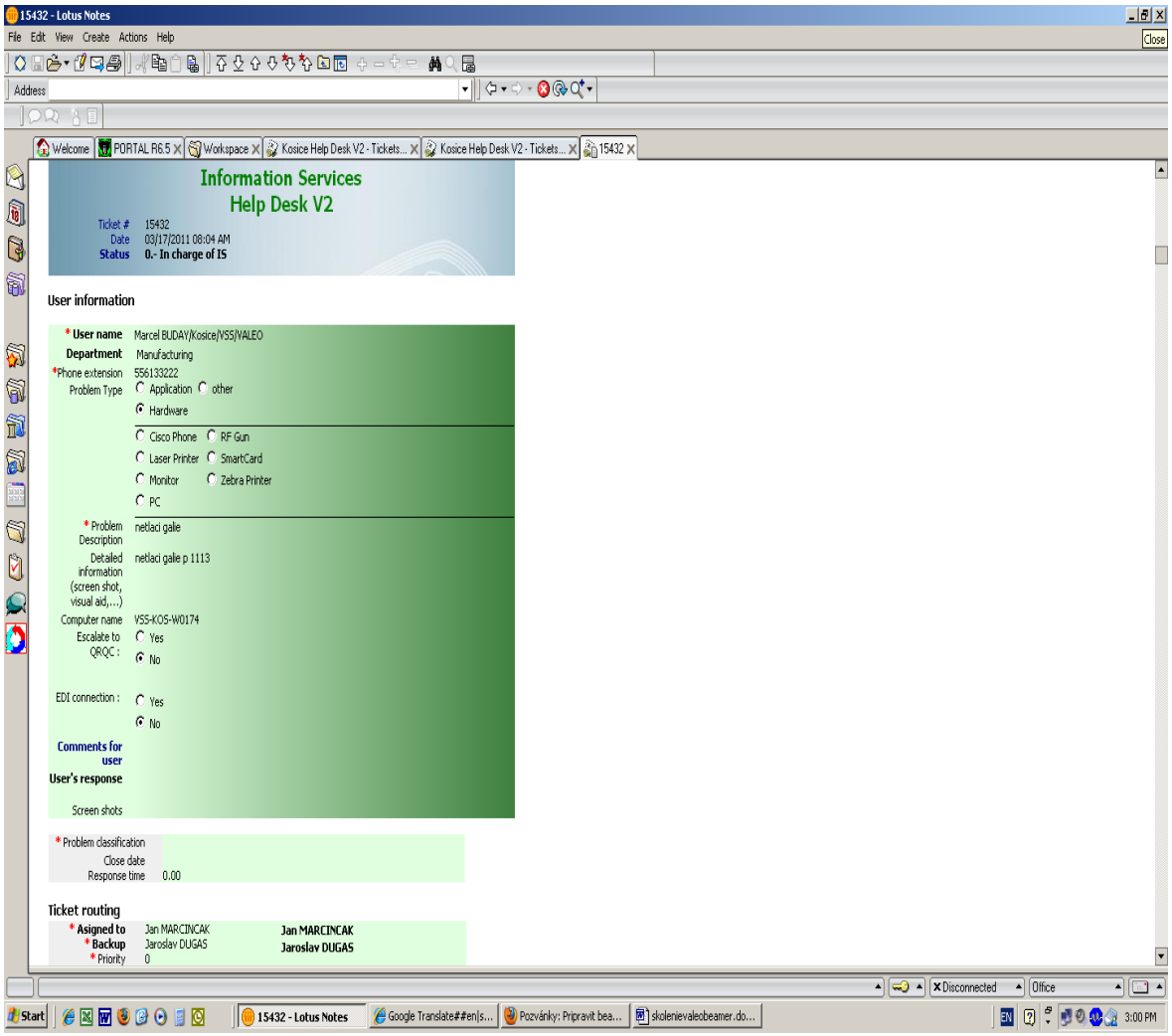
Systém umožňuje aj to, aby úloha bola určite splnená a to takým spôsobom, že je posielaná naraz dvom potencionálnym riešiteľom: *Asignet to: Jan MARCINCAK* a *Backup: Jaroslav DUGAS*. Ak sú naraz títo užívatelia na spoločnom pracovisku, dohodnú sa, kto z nich bude danú úlohu riešiť. Ak by nebol prítomný jeden z týchto zamestnancov v práci, alebo by bol na nejakom inom oddelení, a ak by jeho kolega zadanú úlohu už vyriešil v jeho neprítomnosti a následne by niekam odišiel, pri jeho zápise do systému, že je úloha

splnená, sa okamžite premietne neprítomnému kolegovi, že úloha je už splnená a tak jej už nemusí venovať pozornosť. To ale neznamená, že si nemôže pozrieť aké úlohy boli v daný deň v systéme a ako si poradil jeho kolega pri ich riešení.

Ako vidíme na obrázku 38, každá úloha má svoje poradové číslo, má nielen presný dátum zadania do systému, ale aj presný čas vloženia do systému, potom je uverejnený zadávateľ úlohy a oddelenie, na ktorom daný zadávateľ úlohy pracuje, skrátený a podrobnejší popis problému alebo úlohy, ktorú treba vyriešiť, a taktiež je tam uvedené číslo počítača, z ktorého zadávateľ úlohu zadával. Čísla počítačov, z ktorých je úloha zadávaná, sa generuje automaticky, aby v prípade problému sa mohol pracovník IT na diaľku napojiť na uvedené PC a aby tak vedel priamo užívateľovi pomôcť pri jeho probléme.

Na obrázku 39 je v kolónke *Problem Type* zadaná možnosť *Hardware*, ale je možné zadať ešte možnosť *Application*, a tam si zadávateľ môže zadeklarovať konkrétnu úlohu v oblasti ako konkrétna aplikácia, alebo konkrétne databázové prostredie zamerané na kancelársku prácu, alebo na výrobnú aplikáciu. Sú to aplikácie: Lotus Notes; SAP, CIM+, Microsoft Office, PSC, PDM, Catia, WinTerm a ďalšie.

Ak zadávateľ úlohy nevie konkrétne zaradenie pre túto úlohu, alebo úloha je „niečo nové a nezatriedené“, jednoducho označí *Other*, úloha príde vedúcemu oddelenia a ten rozhodne, komu ju pridelí.




Obrázok 39 Presné zadefinovanie úlohy

## 4.3 Úpravy navrhovaného modelu fakultného komunikačného systému

V Prípadovej štúdii 1 sme navrhli ako prvý krok implementácie komunikačného systému na univerzite ekonomického zamerania Microsoft Office Outlook a ako druhý krok sme ešte v tej istej štúdii navrhli „Automat“ na automatické rozposielanie úloh. V Prípadovej štúdii 2 sme navrhli implementáciu komunikačného systému Lotus Notes. Možné úpravy navrhovaného modelu fakultného komunikačného systému máme znázornené vo vývojových diagramoch, ktoré sú súčasťou práce a sú umiestnené v kapitole Prílohy.

Pri hľadaní akéhosi „vzoru“ alebo príkladu implementácie KS v univerzitnom

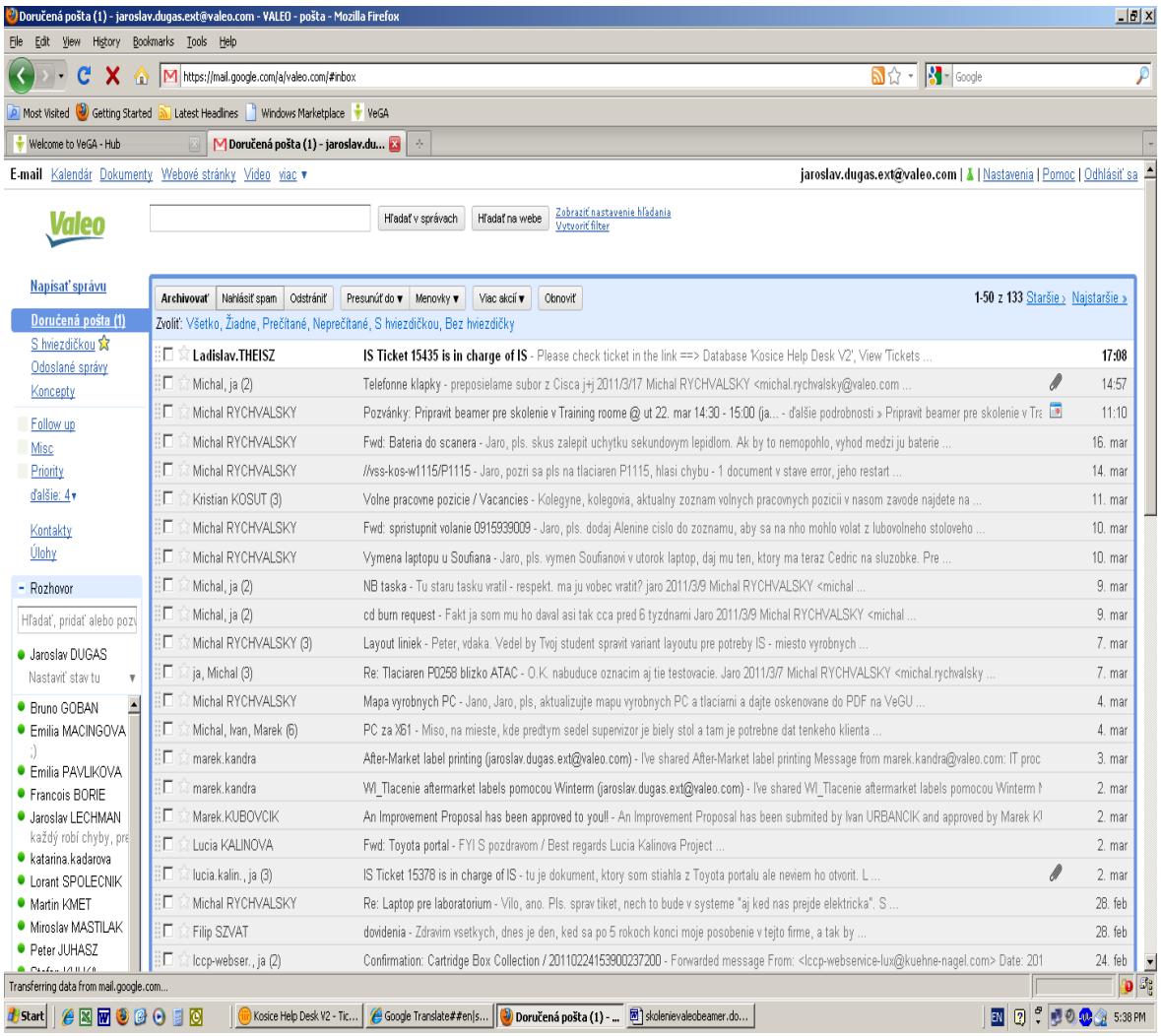
prostredí sme sa stretli s implementáciou Lotus Notes na Vysokej škole ekonomickej v Prahe. Táto univerzita má naimplementovanú verziu 6.5. Veľmi nás zaujala možnosť, že budeme mať konečne nejaký vzor implementácie KS, vhodného aj pre našu fakultu, prípadne pre našu univerzitu. Sklamaním pre nás však bolo zistenie, že Vysoká škola ekonomická v Prahe využíva Lotus len ako celouniverzitný e-mailový nástroj a zatiaľ sa o jeho ďalšom využití ani neuvažuje. O implementáciu, správny chod KS a o poštových klientov sa na tejto škole stará Výpočetní centrum VSE, ktoré má na svojej internetovej stránke síce veľmi pekne popísané možnosti Lotusu, ale implementovaná je „len“ e-ailová časť tohto, dá sa povedať už robustnejšieho, KS.

V Prípadovej štúdii 2 sme si priamo v praxi - vo firme Valeo - odskúšali v rámci Lotusu posielanie, prijímanie a vyhodnocovanie zadaných úloh. Je pravda, že v rámci celosvetovej agendy Lotusu sa v materskej firme vo Francúzsku stará o tento KS sedem zamestnancov a v košickej pobočke Valea dvaja ľudia, ale je to tak preto, pretože na akékoľvek servre spoločnosti Valea sa vedia napojiť len hlavní vrcholoví manažéri zo svojich notebookov pomocou softvéru Juniper, ku ktorým sa pripája špeciálna čítačka kariet špeciálne zadefinovaných prístupových kariet konkrétnych vrcholových manažérov, ale napríklad správcovia serverov a databáz to nevedia a prípadné poruchy musia vyriešiť priamo na mieste. Všetci zamestnanci Valea si vedia spustiť Lotus len na svojich pridelených počítačoch, resp. notebookoch, a je to zabezpečené jedným špecifickým prístupovým súborom, ktorý je jedinečný pre každého užívateľa a je nakopírovaný priamo na jeho počítač.

V prípade možnej implementácie Prípadovej štúdie 2 by sme ďalej navrhovali prepojenie Lotusu s univerzálnym, globálnym e-mailovým prostredím, ktoré umožní nielen prístup k e-mailom z akéhokoľvek počítača na svete, ale zároveň by v e-maili konkrétnemu užívateľovi došla aj konkrétna úloha, ktorú má splniť a ktorá by bola potom aj automaticky sledovaná a vyhodnocovaná. Takýmto univerzálnym prostredím by mohol byť aj napríklad google.com, pomocou ktorého si dnes väčšie podniky prenajímajú e-mailovú on-line prístupnú celoorganizačnú komunikáciu.

Ako vidíme na obrázku 40, konkrétnemu užívateľovi prišiel e-mail do jeho e-mailovej schránky od užívateľa menom Ladislav THEISZ. Teda ak akýkoľvek užívateľ zadá nejakú úlohu buď konkrétnemu riešiteľovi danej úlohy, alebo konkrétnemu oddeleniu, ktoré je zodpovedné za vyriešenie tejto úlohy, a túto úlohu zadáva priamo v Lotuse, spolupráca Lotusu s globálnym e-mailovým prostredím umožní to, že nositeľovi

úlohy príde e-mail do jeho poštovej schránky nech sa v danom okamihu nachádza kdekoľvek na svete, a nemusí byť ani prítomný vo Valeu. Takýto e-mail zaslaný v rámci spolupráce medzi Lotusom a googlom je sprevádzaný *Predmetom mailu*, kde je stále pripojený konkrétny oznam: *IS Ticket (konkrétne číslo úlohy) is in charge of IS*.

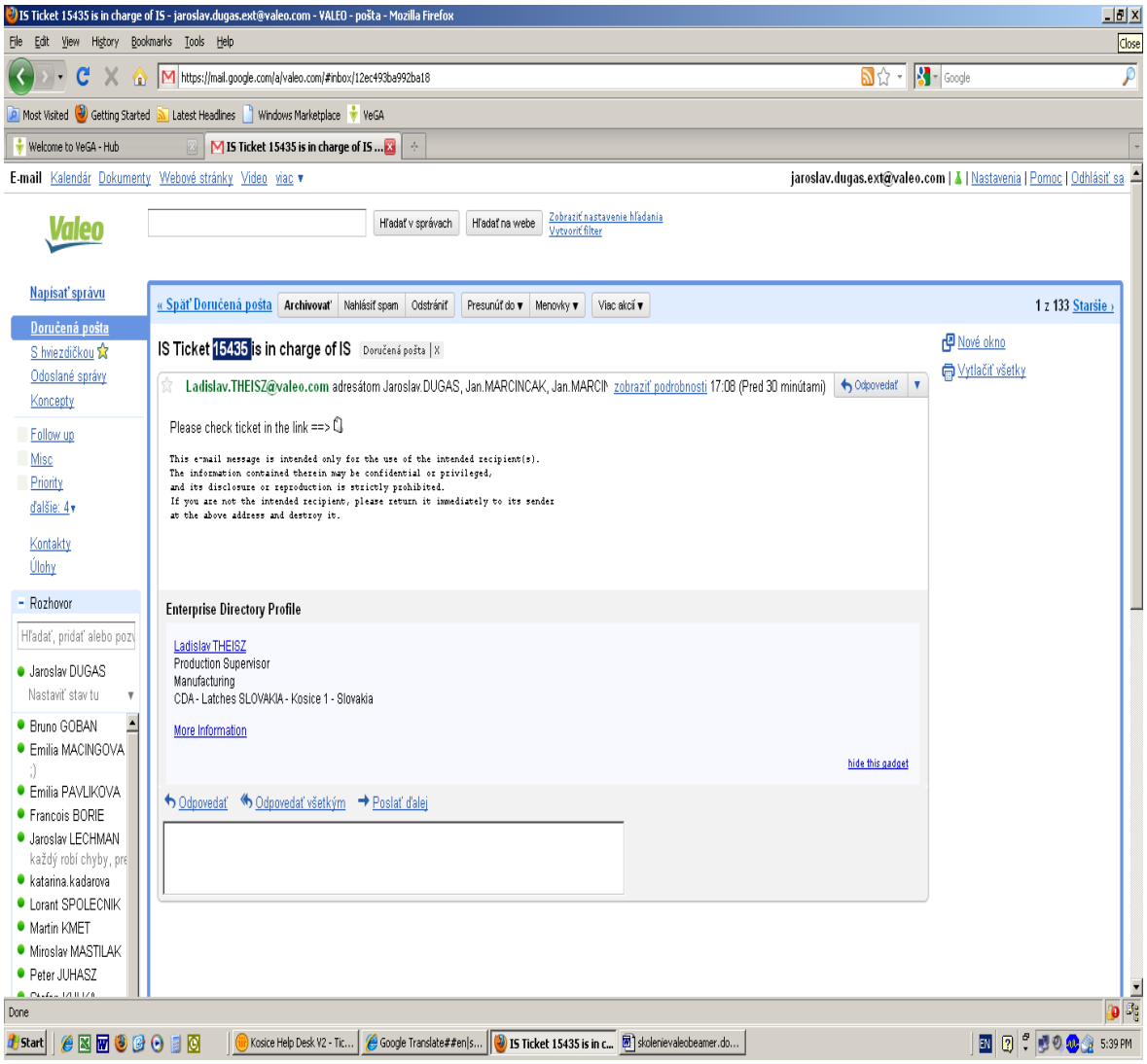



Obrázok 40 Možnosť zaslania úlohy na e-mail konkrétho užívateľa

Po rozbalení takéhoto e-mailu sa dozvieme, komu ešte túto úlohu na vyriešenie poslal konkrétny zadávateľ, v našom prípade Ladislav THEISZ. Pritom ak je napríklad neprítomný potencionálny riešiteľ tejto úlohy, úlohu môže úspešne vyriešiť jeho prítomný kolega, o čom sa po splnení úlohy vyhotoví automatický záznam v systéme, a tak zamestnanec neprítomný v práci, užívajúci si dovolenku alebo len jednoducho neprítomný zo služobných dôvodov, sleduje, ako sa úloha do systému zadala a kto ju úspešne vyriešil.

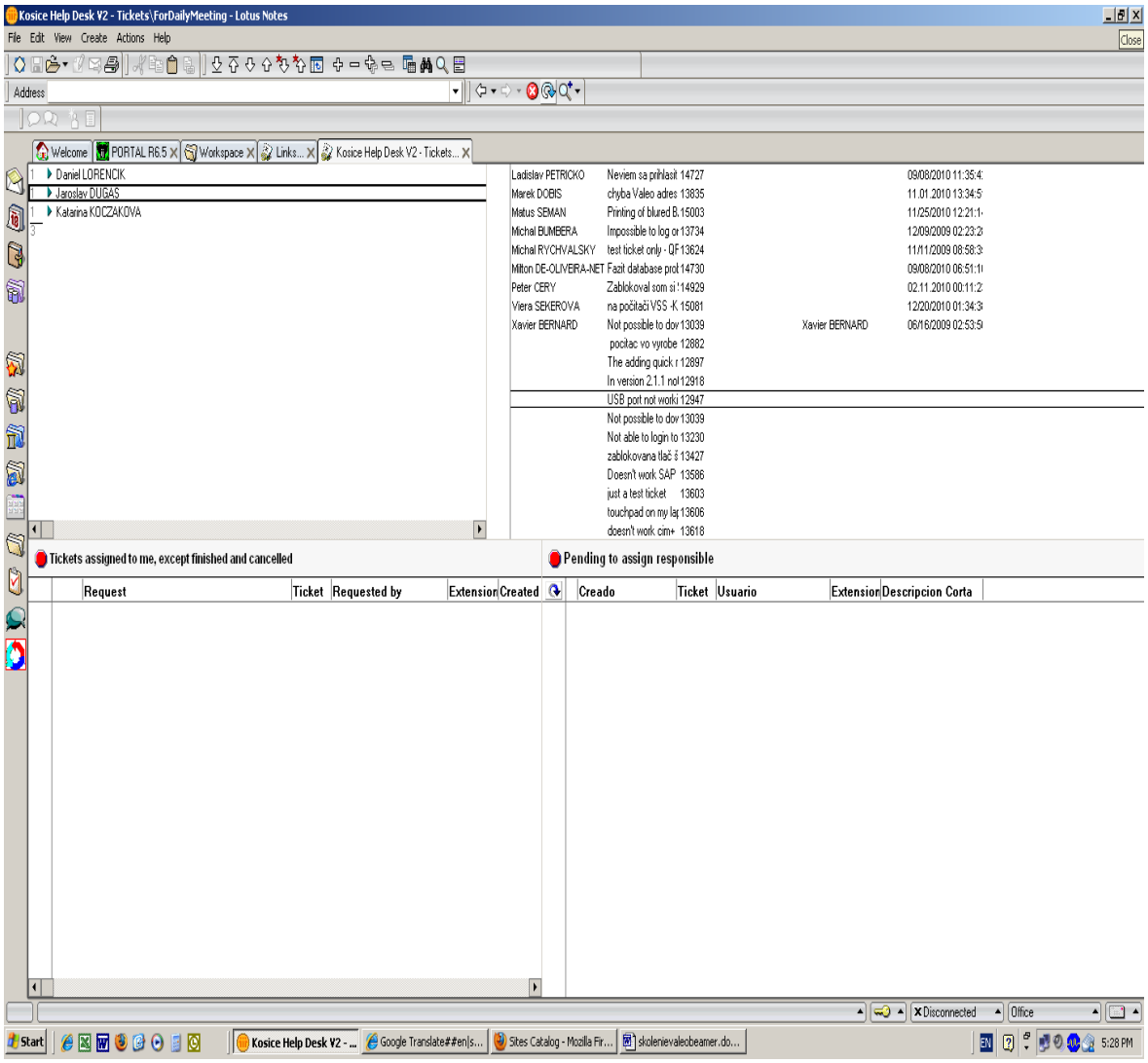
V každom takomto e-maili v rámci komunikačnej spolupráci Lotusu a googlu príde v e-maili odkaz na riešenie úlohy - napríklad v podobe:

*Please check ticket in the link==>* 🗐




Obrázok 41 Odkaz na riešenie úlohy pri spolupráci Lotusu a googlu

Po kliknutí na odkaz v e-maili sa nám spustí Lotus a vidíme, že v aktuálnom čase v rámci nášho oddelenia úlohu majú splniť traja zamestnanci, z ktorých každý má za úlohu splniť po jednej zadanej úlohe (obrázok 42).

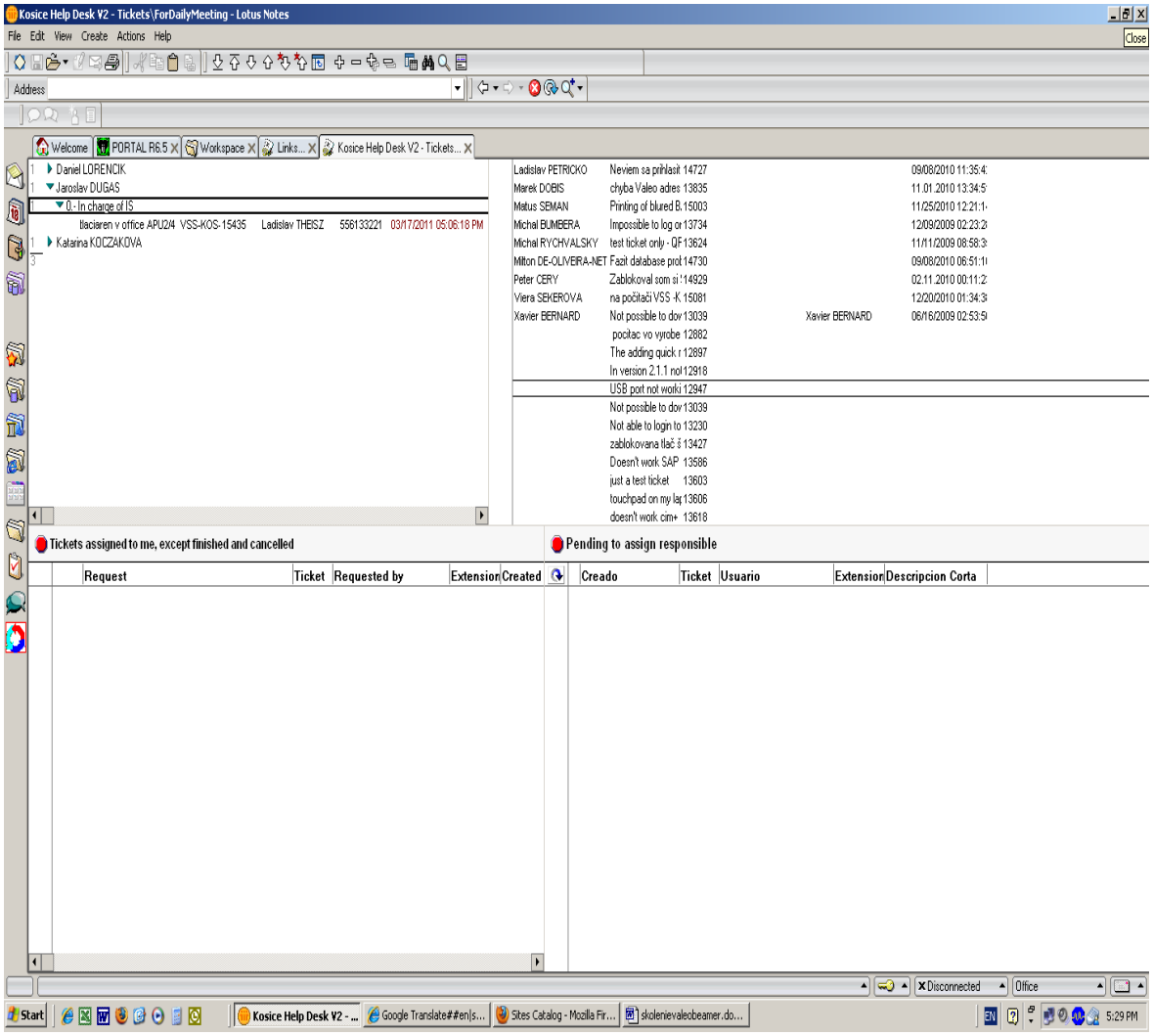



Obrázok 42 Znázornenie riešiteľov úloh pri spolupráci Lotusu a googlu

Vidíme, že každý užívateľ má splniť jednu úlohu, ale časovo dopredu nikdy nie je jasné, ako dlho bude trvať vyriešenie zadanej úlohy. V spoločnosti Valeo, ako už bolo spomenuté, sa má úloha splniť do 24 hodín, aby bola v zozname hodnotených úloh. Tento parameter sa môže ľubovoľne nastaviť, hlavne z dôvodu, keď je dopredu poslaná úloha na vyriešenie, ktorá je napríklad rozsiahlejšia a aj chvíľu potrvá, kým bude úspešne splnená na 100 %. Môže to byť napríklad úloha zaslaná na študijné oddelenie v znení: *Vytvorte 5 komisií na inžinierske štátnice*. Takýchto podobných úloh môže byť aj viac, kde riešenie nie je možné splniť do 24 hodín z konkrétnych dôvodov: nie je napríklad jeden deň prítomný člen katedry zo služobných alebo súkromných dôvodov, nie je dostupný ani telefónom ani e-mailom a tak študijná referentka, zodpovedná za zostavenie komisií, nemôže bez jeho potvrdenia účasti zoznam členov uzavrieť. Preto je vhodné, aby sa časový

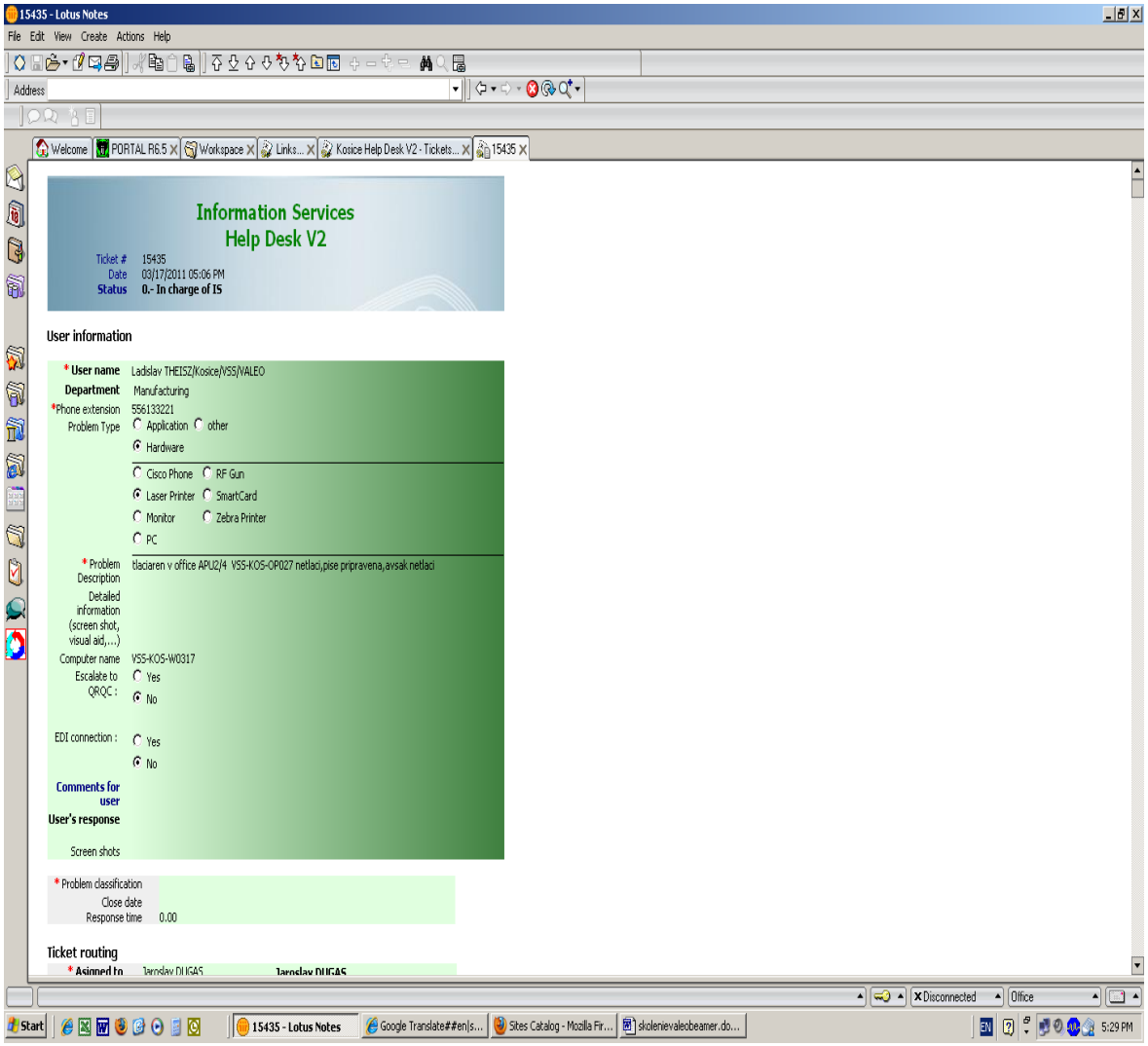
parameter hodnotenia splnených úloh dal posúvať práve z takýchto konkrétnych dôvodov, respektíve aby bol prípadne vytvorený „časový sortér“ spĺňania jednotlivých úloh.

Vráťme sa ešte k zadanej úlohe pre riešiteľa Jaroslav DUGAS. Ak si odklikneme meno, rozbalí sa nám v skratke znenie úlohy. Je to potrebné v tom prípade, keď máme vyriešiť viac úloh, aby sme ich nemuseli naraz odklikávať všetky aby sme vedeli, ktorá úloha čo obsahuje a ktorej sa musíme venovať skôr. Čiže vidíme, že znenie úlohy je: *tlaciaren v office APU2/4 VSS-KOS-*. Podľa tohto krátkeho popisu ešte síce nevieme, aké je presné znenie zadanej úlohy, ktorú máme vyriešiť, ale aj znenie tohto skráteného popisu nám vie napomôcť rozlíšiť túto úlohu od ďalších zadaných úloh (obrázok 43).




Obrázok 43 Skrátený popis zadanej úlohy

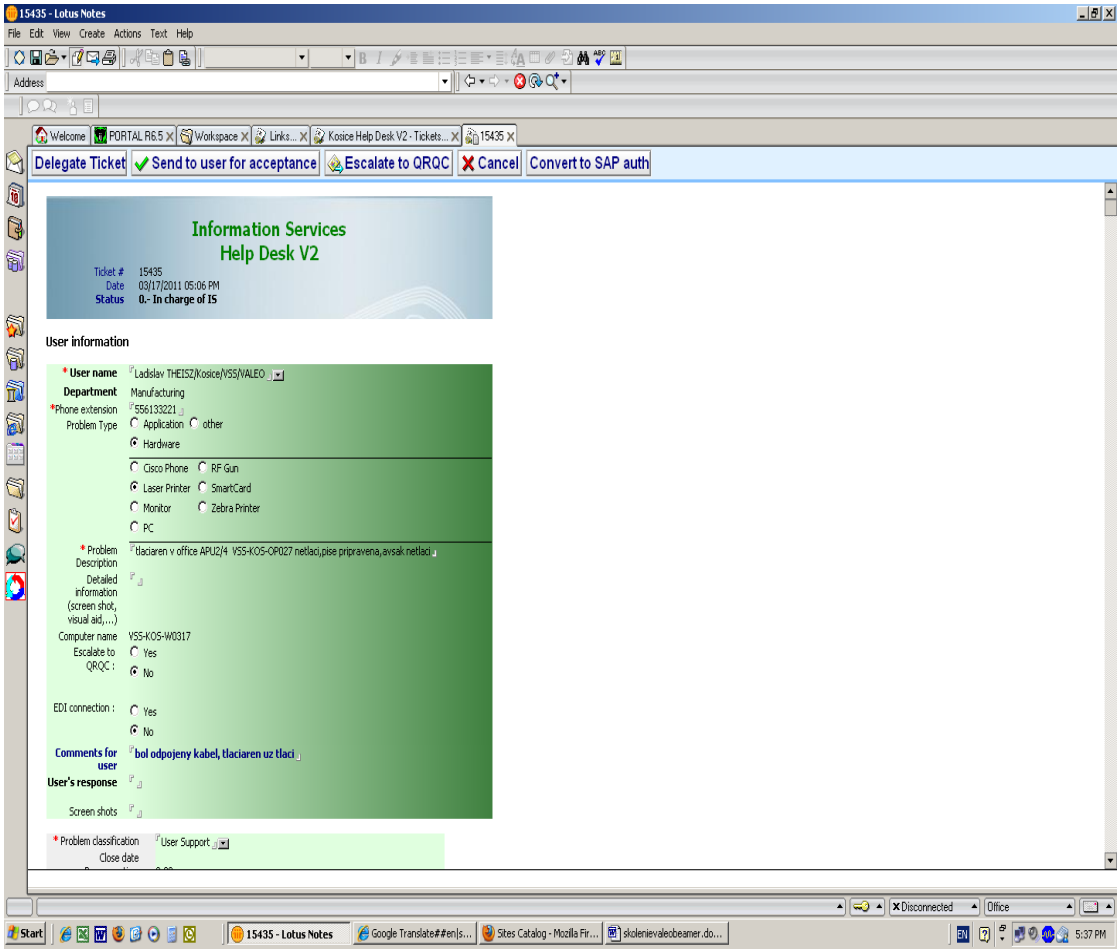
Ak si ale odklikneme úlohu, dozvieme sa všetko o zadanej úlohe, ktorú máme vyriešiť: Číslo úlohy je 15435, bola vytvorená 17. 3. 2011 presne o 17.06. Bola zaradená do skupiny *Hardware* a konkrétne do podskupiny *Laser Printer* a v plnom znení: *tlaciaren v office APU2/4 VSS-KOS-OP027 netlaci, pise pripravena, avsak netlaci*. Komunikácia medzi manažmentom a zamestnancami vo Valeu prebieha v angličtine, a tak často absentuje nainštalované lokálne slovenské prostredie na niektorých PC a preto aj v tomto prípade absentuje slovenská diakritika pri písaní tejto úlohy so žiadosťou na úspešné vyriešenie. Avšak niektorí THP pracovníci menej zdatnejší v angličtine napríklad v technickom popise problému pre oddelenie IT radšej problém popíšu v slovenčine, aby bolo jasnejšie, o aký problém ide (obrázok 44).




Obrázok 44 Znázornenie popisu konkrétnej úlohy

Po splnení úlohy pracovník zodpovedný za splnenie tejto úlohy v komentári pre zadávateľa úlohy (Comments for user) popíše spôsob vyriešenia úlohy a odošle to do systému tým, že odklikne možnosť: *Send to user for acceptance*. Oznam o splnení úlohy dôjde zadávateľovi úlohy a ten, ak je spokojný s vyriešením úlohy, tak to aj v systéme odobrí. Ak odobrenie nestihne do 24 hodín, teda do času, kedy úloha má byť splnená, tak to systém vyhodnotí ako splnenie úlohy.

Pre budúcnosť by bolo vhodné nechať väčší časový priestor zadávateľovi úlohy na odobrenie, pretože sa môže stať, že zadávateľ úlohy až po dlhšom čase zistí, že úloha predsa len nie je splnená na 100 %. Nositeľ úlohy, ako vidíme na obrázku 45, by mal mať možnosť úlohu delegovať na niekoho iného, ak úlohu nevie vyriešiť alebo nie je v jeho kompetencii - ale to tiež by malo byť vopred preddefinované v systéme, aby nedochádzalo k zbytočnému prieťahu pri plnení zadaných úloh, keď si nositelia úlohy danú úlohu presúvajú medzi sebou a tá stále zostáva nevyriešená. Úloha by sa mala dať vo výnimočných prípadoch aj zrušiť.




Obrázok 45 Komentár riešiteľa úlohy zadávateľovi

# 5 Diskusia

## 5.1 Ďalšie smery vedeckého výskumu implementácií komunikačných systémov

Každý úspešný a teda aj skúmateľný projekt návrhu implementácie KS by mal obsahovať tri základné fázy, a to *usporiadanie (Align), architektúru (Architect)* a *zostavenie (Assembly)*. V každej z týchto fáz by malo byť vytvorených niekoľko častí návrhov, ktoré by následne produkovali nejaké viditeľné výsledky. Fáza Align všeobecne zabezpečuje dodržanie požiadaviek na systém, fáza Architect vytvára komponentovo orientovanú architektúru a fáza Assembly dodáva riešenia zložením vývojových častí a jednotlivých komponentov.

Cieľom Align je teda prepojenie business processes a programového riešenia. Táto fáza by teda mala obsahovať tieto hlavné aktivity:

- musí byť vykonaná analýza „Business processes“, vďaka ktorej môžu byť identifikované dôležité systémové funkcie (use cases),
- ďalej sa musí vykonať základné dôležité rozhodnutie o architektúre systému - napríklad o komponentoch technickej infraštruktúry - HW, a je tu tiež vhodné rozdeliť pole aplikácií na funkcionálne segmenty - komponenty, ktoré potom môžu byt vyvíjané aj paralelne.

Vo fáze Architect je hlbšie rozpracovaná architektúra business processes a technická architektúra.

Fáza Assembly sa detailnejšie zaoberá dizajnom, vývojom a nasadením KS. Pre kompletizáciu a správu komponentov sa využíva model „Vytvoriť - Spravovať - Užívať“, ktorý je podporovaný produktom Select Component Manager.

Ďalšie smery vedeckého výskumu implementácií komunikačných systémov budú v budúcnosti veľmi závisieť hlavne od úspešného zavedenia Metodiky RUP tesne

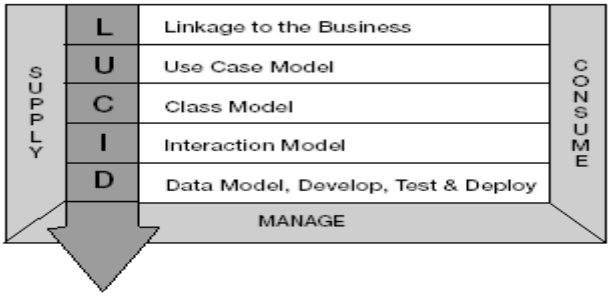
prepojenú s Metodikov SELECT Perspective. Táto metodika bola vytvorená práve za účelom napomáhania vedeckého výskumu úspešných implementácií informačných a komunikačných systémov. Je objektovo orientovaná a bola vytvorená Stuartom Frostom a Paulom Allenom z firmy SELECT Software Tools. Vychádza z kombinácie objektovo orientovaných techník modelovania (OMT) Jamesa Rumbaugha a z Objektovo orientovaného softvérového inžinierstva (OOSE) Ivara Jacobsona (Sterman, 2010). Súčasná verzia metodiky ponúka podporu jednotlivých fáz komponentovo orientovaného vývoja (Component Based Development - CBD), t. j. tvorby komponentov, riadenie popísaných komponentov a tvorby aplikácií z týchto komponentov.

Koncept CBD spočíva v tom, že inštitúcia namiesto toho, aby vždy vytvárala novú a novú aplikáciu od začiatku, použije už vytvorené moduly - komponenty, ktoré zapojí do kompletného systému. S rastúcim množstvom komponentov potom klesá samotná doba tvorby komplexného systému a jeho náročnosť. Komponenty sa môžu s každým novým projektom vylepšovať.

Tak ako metodika RUP, tak aj Select Perspective zahŕňa okrem UML aj modelovanie organizačných procesov a dátové modelovanie. Môžeme teda konštatovať, že metóda Select Perspective stavia na siedmich základných princípoch:

- začína znázornením procesov, a z tých potom vychádzajú ďalšie analýzy,
- komponentovo orientovaný vývoj,
- obsahuje integrované modelovacie techniky, a to modelovanie business processes, dátové modelovanie a časť UML,
- iteratívny, inkrementálny vývoj,
- paralelný rozvoj, v súvislosti s CBD môže paralelne pracovať niekoľko tímov vyvíjajúcich jednotlivé komponenty,
- architektúra založená na službách (Service-Based Architektúra), časťou Select Perspective sú Perspective Patterns, čo sú pripravené rámce pre technickú a business architektúru,
- programové nástroje pre návrh systému, správu komponentov, automatické generovanie kódu.

Dôležitou súčasťou metodiky Select Perspective je balík modelovacích techník, ktorý sa označuje LUCID. Ide o skupinu praktík pre návrhy, vývoj komunikačných systémov a výskum implementácií týchto systémov (Sterman, 2010).




Obrázok 46 LUCID - súčasti balíka modelovacích techník (Sterman, 2010)

Select Perspective zahŕňa pre modelovanie business processes dva druhy diagramov, a to procesný hierarchický diagram (Process Hierarchy Diagram) a diagram procesných reťazcov (Process Threads Diagram). Procesný hierarchický diagram sa používa pre grafické znázornenie vzťahov medzi úrovňami možných nespojitých procesov. Na vrchole sa nachádza akákoľvek celá firma, spoločnosť, inštitúcia, tá je potom pomocou stromu rozkladaná na kľúčové procesy a vlákna až na elementárne procesy (elementary business process - EBP). Diagramy procesných reťazcov sa používajú pre znázornenie závislostí vo vnútri procesných reťazcov.

Use cases znázorňujú kľúčové služby, ktoré musí komunikačný systém ponúkať a vykonávať. Jeden use case obvykle korešponduje s jedným alebo viacerými elementárnymi business processes. Pri modelovaní tried (Class Modeling) sa vytvára pohľad na celkovú štruktúru komunikačného systému. Ďalším modelom je model interakcií (Interaction Model). V rámci tohto modelu sú ponúkané dva UML diagramy, a to diagram spolupráce (Collaboration Diagram) a diagram sekvencií (Object Sequence Diagram). Diagram sekvencií obsahuje naviac oproti notácii UML textový stĺpec. Časť Data Modeling, Develop, Test and Deploy - dátové modelovanie, vývoj, testovanie a nasadenie - obsahuje dátové modelovanie.

Tvorcovia metodiky Select Perspective ho zaradili z dôvodu potrieb znázorňovať diagramy relačných databáz, ktoré sú najrozšírenejším typom používaných databáz v praxi. Využívajú sa k tomu diagramy ERD (Entity Relationship Diagram). Select Enterprise používa technológiu pre mapovanie medzi triedami UML a relačnými dátovými modelmi. Pre vývoj ponúka nástroje Select Perspective generovanie programových šablón pre rozšírené programovacie jazyky Java, C + + a Visual Basic. Kódy a modely následne synchronizovane spravuje Select Enterprise.

Dátový model môže mať dve základné formy, a to logický a fyzický model. Logický model znázorňuje entity, podobné triedam v objektovo orientovanom prístupe, konkrétne „objekt“ sa označuje ako výskyt entity, ich atribúty a vzťahy medzi entitami. Fyzický model je odvodený z logického modelu a zachytáva štruktúru tabuliek relačnej databázy, pri ktorých vychádza z entít, a pri tvorbe stĺpcov tabuliek vychádza z atribútov a väzieb medzi tabuľkami.

## 5.2 Prínosy pre fakultu

Implementácia komunikačného systému prebieha za účelom následnej profesionálnej a hlavne manažérskej komunikácie. Fakultná komunikácia je špecifickým typom komunikácie, ktorá, na rozdiel od všeobecnej komunikácie, vzniká a používa sa v určitom, presne definovanom prostredí, je obsahovo vyprofilovaná, čo sa odráža vo výrazových prostriedkoch a komunikačných nástrojoch a postupoch, ktoré komunikátori v komunikačnom procese musia rešpektovať. Podľa prostredia, v ktorom táto fakultná komunikácia prebieha, môže mať množinový, interpersonálny alebo kombinovaný charakter.

- ***Množinový charakter*** fakultnej komunikácie, ktorá prebieha v konkrétnom prostredí, je daný výskytom množinových momentov v komunikačnom procese, kde príjemcom – publikom je viac či menej organizovaná množina ľudí s rozličnými zaradeniami, očakávaniami, predstavami, záujmami.
- ***Interpersonálny charakter*** fakultnej komunikácie je daný komunikačnými schopnosťami a zručnosťami jednotlivých účastníkov komunikačného procesu, ktoré môžu byť limitované komunikačnými vzorcami, komunikačnou situáciou a použitými komunikačnými nástrojmi na fakulte.
- ***Kombinovaná profesionálna*** fakultná komunikácia má jasný komunikačný cieľ, a preto musí byť postavená na premyslenom výbere nositeľa informácií a komunikačných kanálov – formálnych aj neformálnych a na písomne vopred spracovaných štandardných, odvodených a zadefinovaných komunikačných formulárov. Špecifickým typom profesionálnej fakultnej komunikácie by mala byť manažérska komunikácia, ktorá okrem základného komunikačného cieľa –

sprostredkovanie a výmena informácií - by mala aj plniť cieľ nástroja riadenia a vedenia ľudí vo fakultnom pracovnom procese.

V univerzitnom prostredí prebieha, resp. mala by prebiehať len profesionálna komunikácia. Jej nositeľom a protagonistom je „manažér“ - dekan, prodekani, vedúci katedier; účastníkom člen kolégia dekana, akademického senátu, člen katedry, zamestnanec, študent.

Prínosná profesionálna fakultná manažérska komunikácia v rámci fakultného komunikačného systému je charakterizovaná nasledovným spôsobom:

a) Profesionálna manažérska komunikácia by mala byť vždy cielená. Jej vnútorná štruktúra by mala byť tvorená tými istými zložkami ako napríklad interpersonálna komunikácia, ale každá zložka má kvalitatívne iný obsah. Komunikátor, ak chce sprostredkovať určité cielené informácie v pracovnom prostredí, musí používať nielen iný vyjadrovací nástroj, iné výrazové prostriedky, ale aj im zodpovedajúce prostriedky, komunikačné toky, primerané komunikačné nástroje a techniky, ktoré urýchľujú prenos úlohy smerom k prijímateľovi s cieľom nasmerovať ho na plnenie týchto pracovných úloh a zadaných pracovných povinností. V tomto procese sa tak od komunikátora, zadávateľa úlohy, ako aj od prijímateľa, nositeľa úlohy, vyžadujú určité komunikačné zručnosti, ktoré možno nie sú súčasťou osobnostnej výbavy človeka, ale tvoria určité rozvinutie a zdokonalenie osobnostných dispozícií, určitú nadstavbu, ktorú „manažér“ nadobúda buď v pracovnom procese cez semester, teda „za pochodu“, alebo si ju osvojuje cielene v rámci výchovno-vzdelávacieho procesu.
b) Profesionálna fakultná komunikácia by sa mala vyznačovať predovšetkým používaním obsahovo definovaných kategórií a termínov, dávajúcich zmysel vo všetkých súvislostiach a logických väzbách vychádzajúcich z teoretických a prakticky overených poznatkov, skúseností, a záväznosťou, ktorá je vyjadrená stupňom úspešnosti riešenia komunikovaných zadefinovaných plniteľných úloh.
c) Fakultná manažérska komunikácia by mala teda byť prísne špecifická a hlavne odborná. Profesionálna fakultná komunikácia je možno pre laikov neznalých problémov v tomto vednom odbore nezrozumiteľná, ale je charakteristická predovšetkým používaním odborných termínov, používaním špecifických výrazových prostriedkov, používaním konkrétnych techník a nástrojov, jasným definovaním komunikačného zámeru, stupňom záväznosti a zodpovednosti voči príjemcom úlohy, spôsobom riešenia komunikačných konfliktov, ktoré by v rámci fakultného

komunikačného procesu mohli vzniknúť.

d) Fakultná komunikácia by mala byť vecná. Znamená to, že „manažér“ v rámci komunikačného procesu by mal aktívne eliminovať možné subjektívne komunikačné bariéry, ktoré môžu vzniknúť tak na strane zadávateľa úlohy, ako aj na strane adresáta, príjemcu úlohy, ktoré môžu „zahmliť“ predmet komunikácie. Tieto bariéry väčšinou vyplývajú z osobnostných charakteristík účastníkov celofakultného komunikačného procesu. Medzi najčastejšie subjektívne bariéry, ktoré môžu byť problémovým predmetom komunikácie, patria napríklad: odosielateľ správy a jej prijímateľ majú rozdielne vnímanie, prijímateľ ignoruje doporučené informácie, adresát úlohy túto úlohu chápe „po svojom“, možná zmena psychického stavu komunikantov vplyvom stresu, predsudky, nedôvera, podozrievanie, sympatia resp. antipatia, nízka schopnosť aktívne počúvať, ale aj rozdielne tempo myslenia a riešenia zadaných úloh. Ich vznik môžu vyvolať napríklad nesprávne zvolené riešenie zadanej úlohy, nesprávne určenie času a miesta, neodborné filtrovanie informácií alebo dokonca účelové pozmeňovanie informácií a podobne.

Ak si všetci účastníci komunikačného procesu na PHF uvedomia, že profesionálna komunikácia pomocou navrhovaného komunikačného systému je kvalitatívne vyšším stupňom od sociálnej komunikácie a že aj pre ňu platí základné pravidlo, že aj profesionálnu komunikáciu sa musí každý človek naučiť, ak sa, chce do procesu výmeny informácií zapojiť, tak potom táto komunikácia v určitom presne definovanom prostredí má obrovský potenciál napomáhať kvalitnému fakultnému ale aj mimofakultnému komunikovaniu. Hlavným prínosom implementácie fakultného komunikačného systému by malo byť aj vyriešenie možných následných problémov:

- nejasné alebo veľmi stručné formulovanie hlavných myšlienok, príkazov a odporúčaní na riešenie pracovného problému alebo definovanej zadanej úlohy;
- doteraz absenčná tvorba spätnej väzby (feedbacku) - možné vyriešenie pomocou fakultného komunikačného systému;
- neschopnosť predchádzať problémovým a konfliktným situáciám;
- neprofesionálne zvládnutie komunikačných bariér;
- možná nečestnosť a neotvorenosť v komunikačnom procese;
- dodržiavanie komunikačných zásad a komunikačnej etiky na fakulte.

## 5.3 Prípadné prínosy pre iné univerzitné inštitúcie

Pri určovaní významu komunikácie v organizácii, akou univerzitná inštitúcia je, musíme brať do úvahy, že komunikácia je jej neoddeliteľnou súčasťou, alebo dokonca aj vplýva na niektoré vedné disciplíny. Najužšie je vnútorná komunikácia v inštitúcii spojená s jej univerzitnou podnikovou kultúrou. Podniková kultúra určuje hodnoty, ktoré by sa následne mali uznávať a dodržiavať aj v rámci inštitucionálnej komunikácie. Hodnotenie internej komunikácie v každej univerzitnej inštitúcii je veľmi komplikované a hlavne dosť náročné, pretože sa tu stretávajú faktory subjektívnosti a individuálnosti, v protiklade so snahou o komplexné a objektívne hodnotenie.

## 5.4 Prípadné prínosy pre teóriu manažmentu znalostí, teóriu informačných technológii, teóriu komunikačných systémov

Prínosy implementácie KS možno rozdeliť na aspekty ***technologické*** a ***manažérske***.

Medzi popredné technologické aspekty možno zaradiť:

- integráciu dát z viacerých KS,
- zníženie záťaže primárneho KS,
- rozšírenie funkcionality oproti štandardným zostavám KS,
- reporty dostupné na internom portáli podľa vopred zadefinovaných práv,
- inteligentná reprezentácia dát vzhľadom na históriu,
- reporty prispôsobené potrebám podľa organizačnej štruktúry inštitúcie,
- poskytovanie kvalitných informácií vedenia inštitúcie v agregovanej podobe,
- rýchly prístup k analýze veľkého objemu dát,
- možnosť spracovávania elektronických dokumentov a ich následná distribúcia,
- podpora elektronického podpisu.

Z manažérskeho hľadiska patrí medzi hlavné prínosy implementácie KS:

- vyhodnocovanie trendov,
- navrhovanie a vyhodnocovanie plnenia plánov a inštitucionálnej stratégie,
- denná informovanosť „manažérov“ o plnení zadaných úloh,

- riadený tlak na rast vzdelávacej inštitúcie,
- vzájomné previazanie a odovzdávanie informácií o plnení úloh,
- zrýchlenie a zjednodušenie činností pri kolobehu riadenej dokumentácie,
- jednoduché riadenie a uchovávanie externých dokumentov,
- efektívne riadenie a udržiavanie záznamov o kvalite,
- jednoduché a efektívne vedenie interných previerok - auditov a riadenie prípadných nápravných opatrení,
- zavedenie Workflow manažmentu,
- postupná optimalizácia vykonávaných činností a odstránenie duplicít v jednotlivých procesoch a v zadávaní úloh,
- trvalý rast kvality a úrovne vzdelávacieho procesu,
- rozsiahle možnosti manažérskeho vyhodnocovania v reálnom čase podľa viacrozmerných ukazovateľov.

Prínosy implementovaného KS v čase nemožno stanoviť tak jednoducho, ako napríklad hodnotu nákladov na obstaranie a prevádzku KS. V rámci práce je implementáciou nového KS navrhnutý spôsob kvantifikácie reálneho prínosu reálnym pokrytím potreby inštitúcie pre sledovaný organizačný proces. Kvantifikácia procesu je zdrojom informácií o prínose reálneho pokrytia potreby inštitúcie.

# Záver

Tak ako na PHF EU, aj vo všetkých inštitúciách aplikujúcich určitý postup zavedenia manažmentu znalostí pri implementácii svojho komunikačného informačného systému je tento postup zavádzania zároveň postupom zlepšovania tvorby pridanej hodnoty, akejkoľvek konkurenčnej výhody, je cestou k inováciám, a v neposlednom rade je zdrojom zmeny aj samotnej podnikovej kultúry, kde prekvitá tvorivosť ale aj zadefinovaná disciplína rozhodovania sa na základe znalostí. Vlastník dát často disponuje nejakou apriórnou znalosťou, ktorá môže poslúžiť pri redukcii rozmerov riešeného problému a skúmanie najčastejších typov takýchto znalostí, možností ich využitia a vplyv ich zahrnutia alebo ignorovania na výsledok dolovacieho algoritmu môže priniesť zaujímavé poznatky napríklad aj pri implementácii modelu fakultného komunikačného systému.

Výstupom dizertačnej práce sú výsledky, ktoré budú mať prínos tak pre teóriu manažmentu znalostí, teóriu informačných technológii, teóriu komunikačných systémov, tak aj pre našu fakultu, ako možno aj pre iné fakulty našej univerzity, a možno aj pre iné univerzity na Slovensku. Za hlavný teoretický prínos považujeme zavedenie nových pojmov popisujúcich rôzne kategórie modelov komunikačných systémov ako aj ich následné implementácie, ktoré môžu byť predmetom výskumu, a tak - na rozdiel od existujúcich neuniverzitných modelov - môžu byť predmetom ďalších vedeckých diskusií.

Ďalším teoretickým prínosom je vyplnenie prázdneho miesta v teórii, do ktorej sa metodológia implementácie modelu fakultného komunikačného systému dostáva iba výnimočne - ide o meranie prínosov z implementácie modelu komunikačného systému. V neposlednom rade si od dizertačnej práce sľubujeme ďalšie nevyriešené otázky, ktoré môžu byť predmetom výskumu v budúcnosti. Pre fakultnú prax bude navrhnutá metodológia aplikovateľná bez veľkých nárokov na technické a hlavne organizačné zabezpečenie v rámci implementácie komunikačného systému, a prínosná, keďže pre manažment poskytne komplexný, ucelenejší informačný obraz o každodennej činnosti všetkých zložiek fakulty.

# Zoznam použitej literatúry


1. ADAMEC, S. - HORNÝ, S. - ROSICKÝ, A. 1997. *Projektování informačních systém*ů. Praha : Vysoká škola ekonomická, 1997, 89 s. ISBN 80-7079-848-3.
2. ANALECTO, A. 2004. Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil. In: *Proceedings of the 4th International SPICE Conference on Process Assesment and Improvement*. Lisbon, Portugal : 2004, pp 33 - 37.
3. ARLOW, J. - NEUSTADT, I. 2003. *UML a unifikovaný proces vývoje aplikací: Průvodce analýzou a návrhem objektov*ě *orientovaného softwaru*. Brno : Computer Press, 2003, 387 s. ISBN 80-7226-947-X.
4. BAGUI, S. - EARP, R. 2003. *Database design using entity-relationship diagrams*. Boca Raton : Auerbach Publications, 2003, 264 s. ISBN 0849315484.
5. BALÁŽ, P. - VERČEK, P. 2002. *Globalizácia a nová ekonomika.* Bratislava : SPRINT vrfa, 2002, 175 s. ISBN 80-89085-06-7.
6. BASL, J. - BLAŽÍČEK, R. 2008. *Podnikové informační systémy*. 2. vydanie. Praha : Grada Publishing, 2008, 288 s. ISBN 978-80-247-2279-5.
7. BÉBR, R. - DOUCEK, P. 2005. *Informační systémy pro podporu manažerské práce.* Praha : Professional Publishing, 2005, 216 s. ISBN 80-86419-79-7.
8. Beta distribution. In: *Wikipedia, The Free Encyclopedia*. [online]. c. 2010 [cit. 15. 4. 2010]. Dostupné na: <http://en.wikipedia.org/wiki/Beta_distribution>
9. BUCHALCEVOVÁ, A. - KUČERA, J. 2008. Hodnocení metodik vývoje informačních systémů z pohledu testování. In: *Systémová integrace*, 2008, roč. 15, č. 2, s. 42 - 54. ISSN 1210-9479.
10. BUCHALCEVOVÁ, A. - LEITL, M. 2006. Průzkum používání agilních metodik v ČR. In: *Objekty 2006*. Praha : PEF ČZU, 2006, s. 125 - 136. ISBN 80-213-1568-7.
11. BUCHALCEVOVÁ, A. - PAVLÍČKOVÁ, J. 2002. *Základy softwarového inženýrství objektov*ě *orientovaný přístup*. 1. vyd. Praha : Vysoká škola ekonomická, 2002, 216 s. ISBN 80-245-0268-2.
12. BUCHALCEVOVÁ, A. 2005. *Metodiky vývoje a údržby informačních systémů: Kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha : Grada, 2005, 163 s. ISBN 80-247-1075-7.

13. BUCHALCEVOVÁ, A. 2009. *Metodiky budování informačních systémů*. 1. vyd. Praha : Oeconomica, 2009. ISBN 978-80-245-1540-3.
14. BUREŠ, V. 2007. *Znalostní management a proces jeho zavádění*. Praha : Grada Publishing, 2007, 216 s. ISBN 978-80-247-1978-8.
15. *Capability Maturity Model® Integration (CMMI$^{SM}$) - Version 1.2 - Staged Representation, Technical Report CMU/SEI-2002-TR-012*. [online]. The Software Engineering Institute, 2002. [cit. 7. 10. 2007]. Dostupný na: <http://www.sei.cmu.edu/cmmi/>
16. CARDA, A. - KUNSTOVÁ, R. 2003. *Workflow: Nástroj manažera pro řízení podnikových procesů*. 2. rozšíř. a aktualiz. vyd. Praha : Grada Publishing, 2003, 156 s. ISBN 80-247-0666-0.
17. CEARLEY, D. 2009. *Top 10 Strategic Technologies for 2010*. Cannes : Gartner Symposium ITexpo 2009.
18. COLLISON, CH. - PARCELL, G. 2005. *Knowledge management*. Praha : Computer Press, 2005, 236 s. ISBN 80-251-0760-4.
19. ČARNICKÝ, Š. - MESÁROŠ, P. 2009. *Informačné systémy podnikov*. Bratislava : EKONÓM, 2009, 265 s. ISBN 978-80-225-2676-0.
20. ČARNICKÝ, Š. - VAVRINČÍK, P. 2003. *Informačné systémy a reinžiniering v podnikovom manažmente.* Bratislava : EKONÓM, 2003, 142 s. ISBN 80-225-1725-9.
21. ČEPELOVÁ, A. a kol. 2005. *Základy manažmentu: Praktikum*. Bratislava : EKONÓM, 2005, 228 s. ISBN 80-225-1952-9.
22. ČERNÝ, J. 2001. *Základy teorie systémů*. Praha : Vysoká škola ekonomická, 2001, 125 s. ISBN 80-245-0231-3.
23. *ČSN ISO/IEC 15504-4: Informační technologie - Posuzování procesu, Část 4: Návod na zlepšování procesu a určování způsobilosti procesu*. ČNI, 2006.
24. DLOUHÝ, M. 2007. *Simulace podnikových procesů*. Brno : Computer Press, 2007, 201 s. ISBN 9788025116494.
25. FOTR, J. - DĚDINA, J. 1997. *Manažérské rozhodování.* Praha : Ekopress, 1997, 207 s. ISBN 80-901991-7-8.
26. FRIEBELOVÁ, J. - KLICNAROVÁ, J. 2007. *Rozhodovací modely pro ekonomy*. České Budějovice : Jihočeská universita v Českých Budějovicích, 2007, 135 s. ISBN 9788073940355.

27. GÁLA, L. - POUR, J. - TOMAN, P. 2006. *Podniková informatika.* Praha : Grada Publishing, 2006, 484 s. ISBN 80-247-1278-4.
28. Gamma distribution. In: *Wikipedia, The Free Encyclopedia*. [online]. c. 2010 [cit. 15. 4. 2010]. Dostupné na: <http://en.wikipedia.org/wiki/Gamma_distribution>
29. GAMMA, E. - HELM, R. - JOHNSON, R. - VLISSIDES, J. 2003. *Návrh program*ů *pomocí vzorů: Stavební kameny objektov*ě *orientovaných program*ů. Praha : Grada, 2003, 386 s. ISBN 80-247-0302-5.
30. GAZDA, M. 2010. *Simulačné modely a nástroje*. [online] [cit. 15. 4. 2010]. Dostupné na: <http://www.sjf.tuke.sk/transferinovacii/pages/archiv/transfer/5-2002/pdf/141-143.pdf>
31. GEORGE, J. F. - BATRA, D. - VALACICH, J. - HOFFER, J. A. 2004. *Object-Oriented Systems Analysis and Design*. Upper Saddle River : Prentice Hall, 2004, 525 s. ISBN 0-13-113326-8.
32. HÁLEK, I. - PALATOVÁ, D. - ŠKAPA, R. 2005. *Systémy řízení*. Brno : Masarykova univerzita, 2005, 112 s. ISBN 8021036508.
33. HAMMOND, S. - KEENEY, R. L. - RAIFA, H. 1998. The Hidden Traps in Decision Making. In: *Harvard Business Review*, 1998. [online]. Reprinted in 1/2006, p. 118 - 125. [cit. 13. 8. 2007]. Dostupné na: http://harvardbusinessonline.hbsp.harvard.edu/hbsp/hbr/articles/article.jsp?articleID =R01 08G&ml_action=get-article&pageNumber=1&ml_subscriber=true&referral=2533
34. HAYES, R. M. 2001. *Models for Library Management, Decision-Making and Planning*. London : Academic Press, 2001, 278 s. ISBN 0123341515.
35. HORNÝ, S. - PALOVSKÁ, H. - VÁCLAVÍKOVÁ, M. 2001. *Systémová metodologie: Materiály ke cvičení*. Praha : Vysoká škola ekonomická, 2001, 133 s. ISBN 80-245-0132-5.
36. HORNÝ, S. 2000. *Analýza a návrh systém*ů. Praha : Vysoká škola ekonomická, 2000. ISBN 80-245-0007-8.
37. http://consilium.europa.eu/ueDocs/cms_Data/docs/pressData/en/ec/00100-r1.en0.htm
38. http://europa.eu/legislation_summaries/education_training_youth/lifelong_learning/c11090_en.htm
39. http://hn.hnonline.sk/c1-22134480-po-dni-ky-dos-ta-nu-sof-tver-od-mic-ro-sof-tu-inter-ne-tom

40. http://kiwiki.fmtnuni.sk/mediawiki/index.php/Open_source_n%C3%A1hrada_MS_Small_Business_server_2003_so_sharepoint_services
41. http://objekty.vse.cz/Objekty/Rup4#kap43
42. http://objekty.vse.cz/Objekty/Rup4#kap47
43. http://online.gerrardconsulting.com/iseb/otherdocs/ieee829mtp.pdf
44. http://sk.wikipedia.org/
45. http://swn.softnov.cz/
46. http://technet.microsoft.com/en-us/library/bb497060.aspx
47. http://vedomostnaspolocnost.vlada.gov.sk/4753/vedomostna-spolocnost.php
48. http://vedomostnaspolocnost.vlada.gov.sk/4754/lisabonska-strategia.php
49. http://vedomostnaspolocnost.vlada.gov.sk/4790/programove-vyhlasenie-vlady-sr.php
50. http://web.sks.cz/users/ku/DOKUMENTY/pri_syl.pdf
51. http://web.sks.cz/users/ku/pri/BPM.htm
52. http://www.apress.com/resource/bookfile/3497
53. http://www.cssi.cz
54. http://www.cvtisr.sk
55. http://www.dmreview.com
56. http://www.euroinfo.gov.sk/europedia/posobenie-eu/lisabonska-strategia-eu-7b2.html
57. http://www.flexense.com/
58. http://www.gartner.com/it/page.jsp?id=593207&format=print
59. http://www.chip.cz/
60. http://www.ibm.com
61. http://www.ietf.org/rfc/rfc2821.txt
62. http://www.ietf.org/rfc/rfc2822.txt
63. http://www.ikaros.cz
64. http://www.iminerva.sk/default.aspx?ami=1200&smi=1210
65. http://www.itnews.sk/spravy/software/2010-08-18/c135308-systemy-crm-pre-male-firmy?ref=rss
66. http://www.itnews.sk/tituly/infoware/free-clanky/2008-11-05/c82880-podporuju-sucasne-informacne-systemy-procesy-organizacii-dostatocne
67. http://www.itnews.sk/tituly/infoware/free-clanky/2011-03-24/c138959-iw-cloudove-sluzby-su-skutocne-skoro-pre-kazdeho

68. http://www.itnews.sk/tituly/infoware/free-clanky/2011-03-31/c139136-iw-prehlad-dodavatelov-cloud-computingu-v-sr
69. http://www.itsm.sk
70. http://www.microsoft.sk
71. http://www.olapreport.com/fasmi.htm
72. http://www.pcrevue.sk/
73. http://www.pcworld.sk/
74. http://www.siia.net/etis/2010/ppt/BusinessProfilesPresentations.pdf
75. http://www.valeo.com
76. http://www.vlada.gov.sk/
77. http://www.w3.org/TR/wsdl
78. http://www.wallstreetandtech.com/it-infrastructure/showArticle.jhtml?articleID=212700913
79. http://www-01.ibm.com/software/cz/rational/
80. http://xprogramming.com/blog/technical-debt-2/
81. CHLAPEK, D. - ŘEPA, V. - STANOVSKÁ, I. 1999. *Techniky a nástroje vývoje informačních systémů. 1. díl*. Praha : Vysoká škola ekonomická, 1999, 151 s. ISBN 80-245-0005-1.
82. International Software Testing Qualification Board - Glossary working Party. In: *ed. Veenendaal, E. van: Standard glossary of terms used in Software Testing. Version 1.3*. 31. 5. 2007, 39 s.
83. *ISO/IEC 12207: Systems and software engineering - Software life cycle processes*. 2008.
84. *ISO/IEC 15504-3: Information technology - Process assesment - Part 3: Guidance on performing an assesment*. Geneva : 2004.
85. *ISO/IEC 15504-5: Information technology - Process assesment - Part 5: An exemplar Process Assessment Model*. 2006.
86. *ISO/IEC FDIS 15288: Systems and Software Engineering - System life cycle processes*. 2007.
87. *ISO/IEC FDIS 15289: Systems and Software Engineering - Content of systems and software life cycle process information products (Documentation)*. 2005.
88. *ISO/IEC FDIS 15504-2: Information technology - Process Assesment - Part 2: Performing an Assesment*. 2002.
89. *ISO/IEC TR 15271 Information technology - Guide for ISO/IEC 12207 (Software*

*Lifecycle Processes)*. 1998.

90. IVANIČKA, K. 2000. *Manažérske informačné systémy.* Bratislava : STU, 2000, 103 s. ISBN 80-227-1369-4.
91. JABLONSKÝ, J. 2002. *Operační výzkum: kvantitativní modely pro ekonomické rozhodování*. Praha : Professional Publishing, 2002, 323 s. ISBN 8086419231.
92. JAYANT, O. K. 2009. *Management Information Systems*. Pune : Nirali Prakashan, 2009, 248 s. ISBN 9788185790909.
93. JOHNSON, J. 2006. *My Life is Failure*. The Standish Group International, Inc., 2006. ISBN 1-4243-0841-0.
94. KELEMEN, J. a kol. 2007. *Pozvanie do znalostnej spoločnosti.* Bratislava : Iura Edition, 2007, 268 s. ISBN 978-80-8078-149-1.
95. KENDALL, K. E. - KENDALL, J. E. 2002. *Systems Analysis and Design*. 5 ed. Prentice Hall, 2002, 944 s. ISBN 0-13-041571-5.
96. KEYS, P. 1991. *Operational research and systems: the systemic nature of operational research.* New York : Springer, 1991, 288 s. ISBN 978-0-306-43642-0.
97. KOKLES, M. - ROMANOVÁ, A. 2002. *Informačný vek*. 2. vydanie. SPRINT vrfa, 2002, 305 s. ISBN 80-89085-09-1.
98. KOONTZ, H. - WEIHRICH, H. 2007. *Essentials of Management*. NewDelhi : Tata McGraw-Hill Publishing, 2007, 479 s. ISBN 9780070620308.
99. LAPORTE, C. Y. - APRIL, A. - RENAULT, A. 2006. Applying ISO/IEC software engineering standards in small settings, Historical perspectives and initial achievements. In: *Proceedings of the SPICE conference*. Luxembourg : 2006, pp. 57 - 62.
100. LEITNER B. 2010. *Viackriteriálne (multikriteriálne) rozhodovanie (Rozhodovacia analýza)*. [cit. 8. 4. 2010] Dostupné na: <http://fsi.uniza.sk/ktvi/leitner/2_predmety/OA/Cvicenia/01_VR_1.pdf>
101. MALÝ, J. - VÍTEK, M. 2000. *Teorie systémů II*. 2. vyd. Hradec Králové : Gaudeamus, 2000, 198 s. ISBN 80-7041-759-5.
102. MALÝ, J. 1998a. *Projektování informačních systémů I*. Hradec Králové : Gaudeamus, 1998, 147 s. ISBN 80-7041-518-5.
103. MALÝ, J. 1998b. *Projektování informačních systémů II*. Hradec Králové : Gaudeamus, 1998, 61 s. ISBN 80-7041-530-4.
104. MALÝ, J. 2000a. *Projektování informačních systémů III*. Hradec Králové : Gaudeamus, 2000, 130 s. ISBN 80-7041-771-4.

105. MALÝ, J. 2000b. *Teorie systémů I*. 2. vyd. Hradec Králové : Gaudeamus, 2000, 66 s. ISBN 80-7041-741-1.
106. MATIAŠKO, K. 2001. *Dátové sklady ako informačný zdroj pre podporu rozhodovania*. Bratislava : STU, 2001. ISBN 80-7226-969-0.
107. MÄKINEN, T. - VARKOI, T. 2008. Analyzing a Process Profile for Very Small Software Enterprises. In: *Proceedings of the SPICE 2008 conference*. Nuremberg : 2008.
108. MESÁROŠ, P. - MESÁROŠ, F. - ČARNICKÝ, Š. 2008. *Manažment znalostí a perspektívy jeho uplatnenia v slovenských podnikoch.* Bratislava : EKONÓM, 2008, 204 s. ISBN 978-80-225-2490-2.
109. MIZLA, M. - BAŠISTOVÁ, A. 2002. *Manažérska diagnostika*. Bratislava : EKONÓM, 2002, 80 s. ISBN 80-225-1567-1.
110. MLADENIC, D. - LAVRAC, N. - BOHANEC, M. - MOYLE, S. 2003. *Data mining and decision support: Integration and collaboration*. Springer, 2003, 304 s. ISBN 1402073887.
111. MLÁDKOVÁ, L. 2005. *Moderní přístupy k managementu. Tacitní znalosti a jak je řídit.* Praha : C. H. Beck, 2005, 195 s. ISBN 80-7179-310-8.
112. MOLNÁR, Z. 2000. *Efektívnost informačních systémů*. Praha : Grada Publishing, 2000, 180 s. ISBN 80-7169-410-X.
113. MONAHAN, G. E. 2000. *Management Decision Making: spreadsheet modeling, analysis, and application*. Cambridge : Cambridge University Press, 2000, 717 s. ISBN 0521781183.
114. MORSE, P. M. - KIMBALL, G. E. 2003. *Methods of Operations Research*. Dover Publications, 2003, 176 s. ISBN 0486432343.
115. OCTABA, H. - PIETINI, M. 2008. *Software Process Improvement for Small and Medium Enterprises: Techniques and case studies*. Information Science Reference. New York : Hershey, 2008. ISBN 978-1-59904-906-9.
116. Online Analytical Processing. In: *Wikipedia, The Free Encyclopedia.* [online]. c. 2010 [cit. 12. 4. 2010]. Dostupné na: <http://en.wikipedia.org/wiki/Online_analytical_processing>
117. PAGE-JONES, M. 2001. *Základy objektově orientovaného návrhu v UML*. Praha : Grada, 2001, 367 s. ISBN 80-247-0210-X.
118. PLAMÍNEK, J. 2008. *Řešení problémů a rozhodování: jak přinutit problémy, aby pracovaly ve váš prospěch*. 1. vyd. Praha : Grada, 2008, 144 s. ISBN

9788024724379.

119. POLÁK, J. - MERUNKA, V. - CARDA, A. 2003. *Umění systémového návrhu: Objektově orientovaná tvorba informačních systémů pomocí původní metody BORM*. Praha : Grada, 2003, 195 s. ISBN 80-247-0424-2.

120. POWER, D. J. 2002. *Decision Support Systems: concepts and resources for managers*. Westport : Quorum Books, 2002, 251 s. ISBN 156720497X.

121. *Process Maturity Profile*. [online]. Software Engineering Institute, c. 2008. Dostupný na: <http://www.sei.cmu.edu/appraisalprogram/profile/profile.html>

122. RAVINDRANATH, B. 2003. *Decision Suport Systems and Data Warehouses*. New Delhi : New Age Publications, 2003, 212 s. ISBN 8122414510.

123. *Riadenie financií a informácií: Porozumieť informáciám*. Bratislava : City University, 2004. ISBN 80-89045-66-9.

124. ROWLEY, J. E. 1990. *The basics of systems analysis and design for information managers [Základy systémové analýzy a projektování pro informační manažery]*. 1. ed. London : Clive Bingley, 1990, 158 s. ISBN 0-85157-453-X.

125. ŘEPA, V. - HAUSMANN, O. - HOJDAR, M. 1997. *Řízení projektu informačního systému: Materiály ke kursu.* 1. vyd. Praha : Vysoká škola ekonomická, 1997, 119 s. ISBN 80-7079-248-5.

126. ŘEPA, V. 1999. *Analýza a návrh informačních systém*ů. 1. vyd. Praha : Ekopress, 1999, 403 s. ISBN 80-86119-13-0.

127. ŘEPA, V. 2006. *Podnikové procesy: Procesní řízení a modelování.* 1. vyd. Praha : Grada, 2006, 265 s. ISBN 80-247-1281-4.

128. SZARKOVÁ, M. a kol. 2002. Komunikácia v manažmente. Bratislava: Ekonóm. 198 s. ISBN 80-225-1585-X.

129. SCHMULLER, J. 2001. *Myslíme v jazyku UML*. Knihovna programátora. Praha : Grada, 2001, 359 s. ISBN 80-247-0029-8.

130. Simulation language. In: *Wikipedia, The Free Encyclopedia*. [online]. c. 2010 [cit. 8. 4. 2010]. Dostupné na: <http://en.wikipedia.org/wiki/Simulation_language>

131. SKALSKÁ H. 2006. *Stochastické modelování*. Hradec Králové : GAUDEAMUS, 2006, 162 s. ISBN 80-7041-488-X.

132. Software and Systems Engineering Vocabulary. [online]. IEEE Computer society, c. 2008 [cit. 30. 1. 2008]. Dostupný na: http://pascal.computer.org/sev_display/index.action

133. *Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version*

*1.2: Method Definition Document*. [online]. SEI/CMU, c. 2006 [cit. 8. 10. 2007]. Dostupný na: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06hb002.pdf>

134. STERMAN, J. D. 2010. *Skeptikův průvodce počítačovým modelováním*. [online]. Preložil Marek Šusta. [cit. 10. 4. 2010]. Dostupné na: <http://www.proverbs.cz/media/art/Skeptikuv_pruvodce.pdf >
135. STOFFOVÁ, V. - STOFFA, J. 1999. Základné termíny informačných, multimediálnych a didaktických technológií. In: *Medacta '99 - Učiteľ pre tretie tisícročie*. Nitra : Univerzita Konštantína Filozofa, 1999, s. 64 - 69. ISBN 80-967746-2-X.
136. SVOBODA, S. 2000. *Informační systém podnikatelských subjektů*. Praha : Vysoká škola ekonomická, 2000, 304 s.
137. ŠEŠERA, Ľ. - MIČOVSKÝ, A. - ČERVEŇ, J. 2001. *Datové modelování v příkladech*. Praha : Grada, 2001, 152 s. ISBN 80-247-0049-2.
138. TRUNEČEK, J. 2003. *Znalostní podnik ve znalostní společnosti.* Praha : Professional publishing, 2003, 312 s. ISBN 80-86419-35-5.
139. TRUNEČEK, J. 2004. *Management znalostí*. Praha : C. H. Beck, 2004, 131 s. ISBN 80-7179-884-3.
140. TURBAN, E. - MEREDITH, J. R. 1991. *Fundamentals of management science*. 5. ed. Homewood : Irwin, 1991, 1010 s. ISBN 0256083738.
141. TVRDÍKOVÁ, M. 2000. *Zavádění a inovace informačních systémů ve firmách*. 1. vyd. Praha : Grada, 2000, 110 s. ISBN 80-7169-703-6.
142. VÁGNER, I. 2007. *Systém managementu*. Brno : Masarykova univerzita, 2007, 432 s. ISBN 978-80-210-4264-3.
143. VEBER, J. 2002. *Management. Základy, prosperita, globalizace.* Praha : Management Press, 2002, 704 s. ISBN 80-7261-029-5.
144. VETRÁKOVÁ, M. 2002. Komunikácia v práci manažéra. Zvolen: Bratia Sabovci, s. r. o. 203 s. ISBN 80-8055-630-X.
145. VLASÁK, R. - BULÍČKOVÁ, S. 2003. *Základy projektování informačních systémů*. 1. vyd. Praha : Karolinum, 2003, 144 s. ISBN 80-246-0727-1.
146. VOŘÍŠEK, J. 1997. *Strategické řízení informačního systému a systémová integrace.* Praha : Management Press, 1997, 323 s. ISBN 80-85943-40-9.
147. VOŘÍŠEK, J. a kol. 2008. *Principy a modely řízení podnikové informatiky*. 1. vyd. Praha : Oeconomica, 2008, 446 s. ISBN 978-80-245-1440-6.
148. VRANA, I. - RICHTA, K. 2005. *Zásady a postupy zavádění podnikových*

*informačních systémů: Praktická příručka pro podnikové manažery*. 1. vyd. Praha : Grada, 2005, 187 s. ISBN 80-247-1103-6.

149. VRANA, I. 1997. *Projektování informačních systémů*. Praha : Česká zemědělská univerzita, 1997, 117 s.

150. VYMĚTAL, J. - DIAČIKOVÁ, A. - VÁCHOVÁ, M. 2005. *Informační a znalostní management v praxi*. Praha : LexixNexis CZ, 2005, 399 s. ISBN 80-86920-01-1.

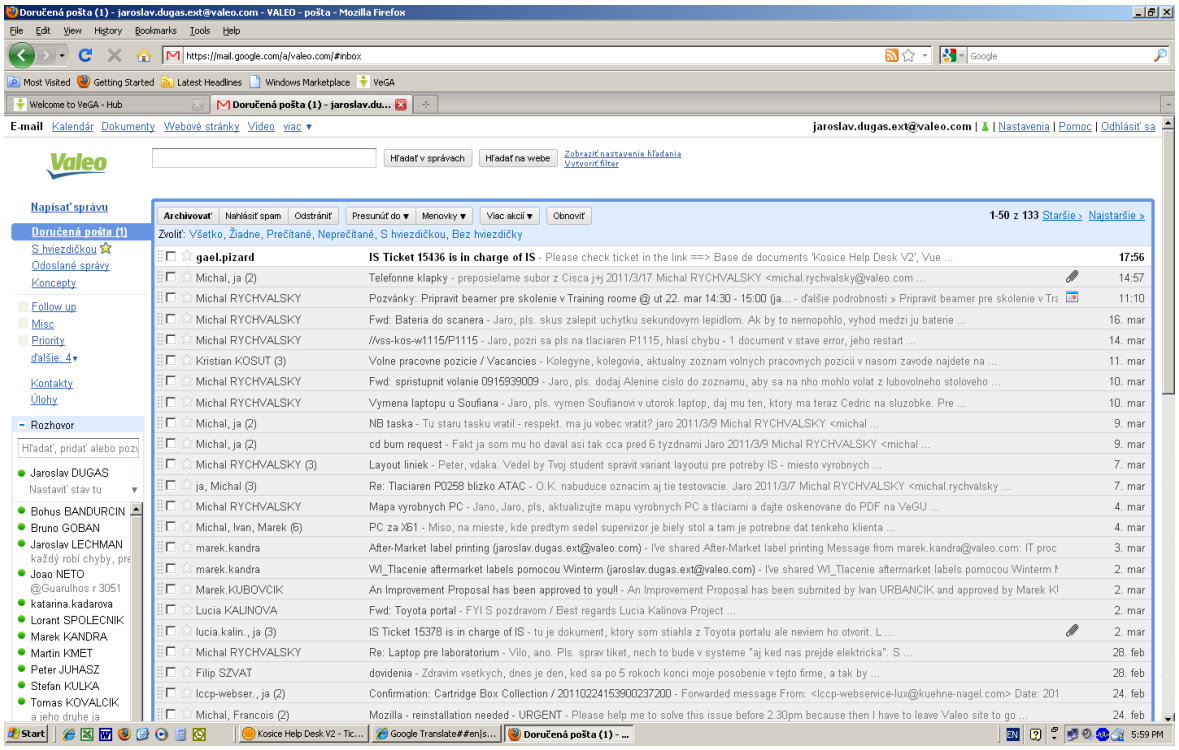
151. *Witness Getting Started Materials*. [online]. [cit. 10. 2. 2010]. Dostupné na: <http://www.dynamic.co.kr/Witness_Training_Center/witness_down/Witness_PwE/WITNESS_Getting_Started_Materials.pdf>

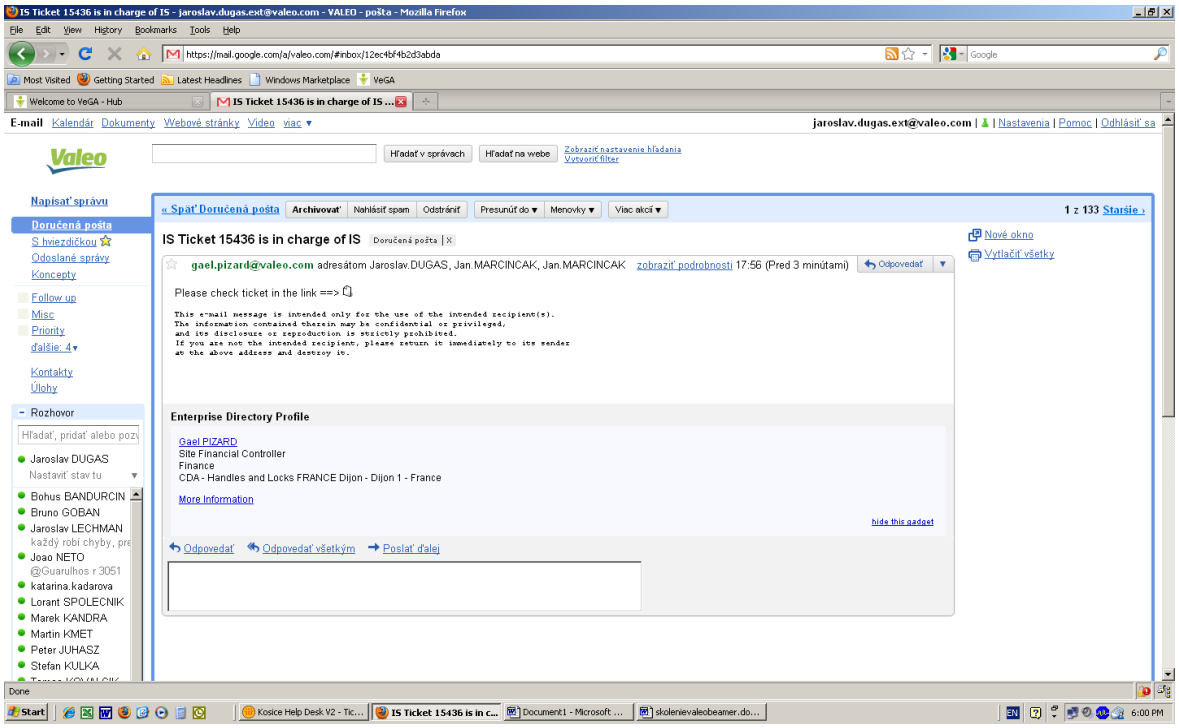
152. WÖHE, G. - KISLINGEROVÁ, E. - MAŇASOVÁ, Z. 2007. *Úvod do podnikového hospodářství.* 2. přeprac. a dopl. vyd. Praha : C. H. Beck, 2007, 928 s. ISBN 9788071798972.

153. ZINS, Ch. 2007. Conceptions of information science. In: *Journal of the American Society for Information Science and Technology*. 2007, Volume 58, Issue 3, s. 335 - 350.

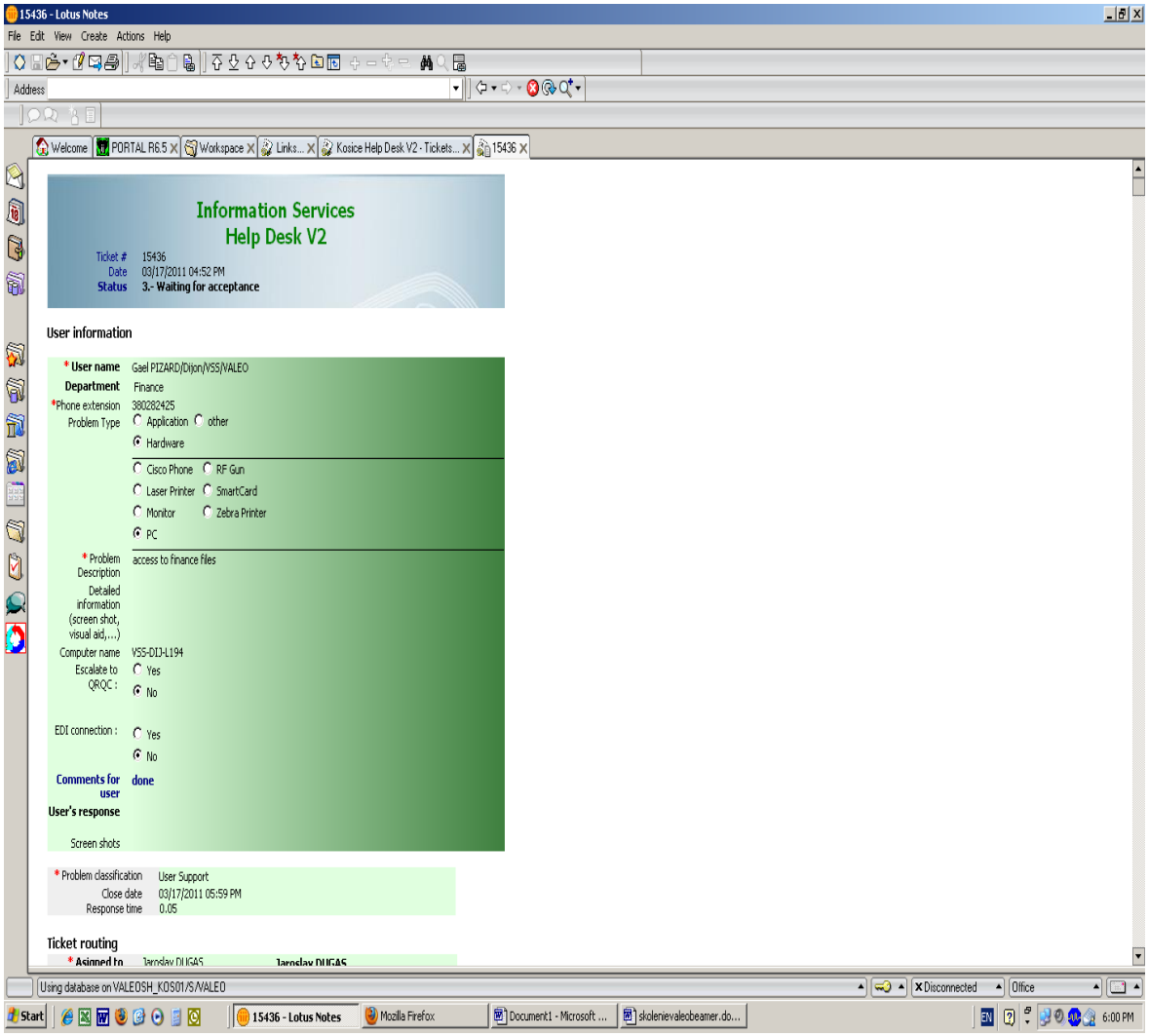
# Prílohy

**Príloha č. 1:** Možnosť zadania úlohy z inej pobočky - napríklad z Dijonu do Košíc - a možné vyriešenie zadanej úlohy:



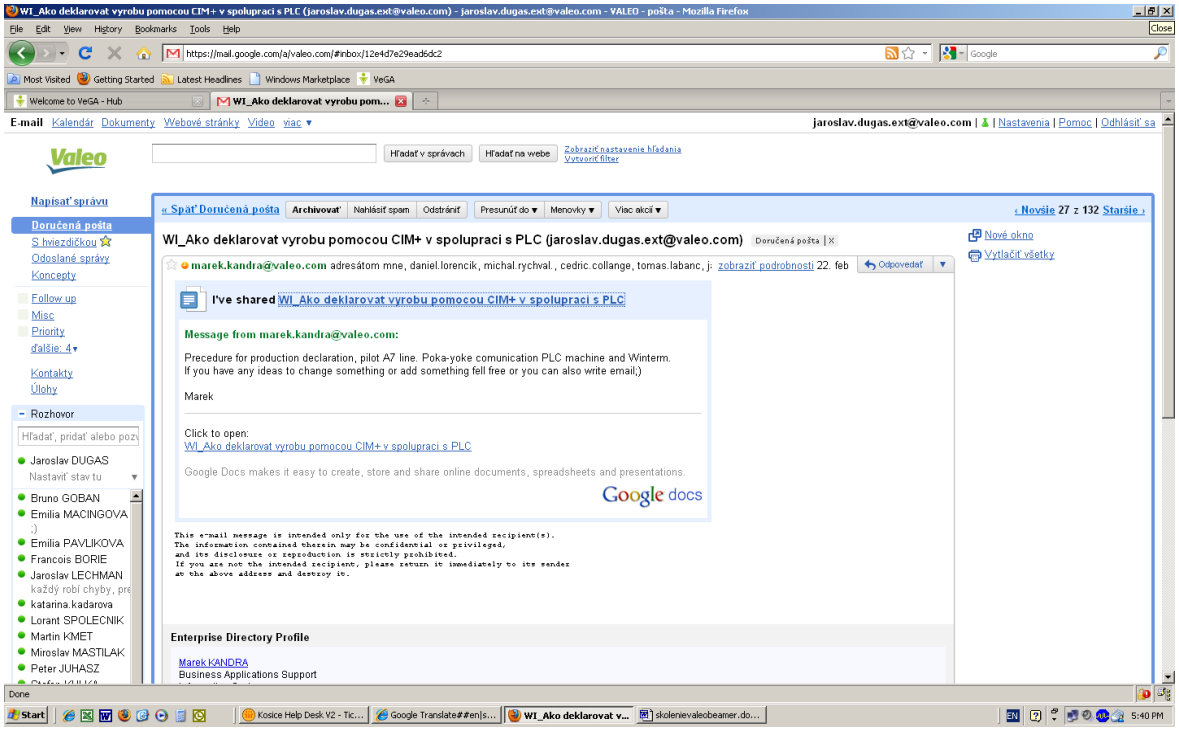



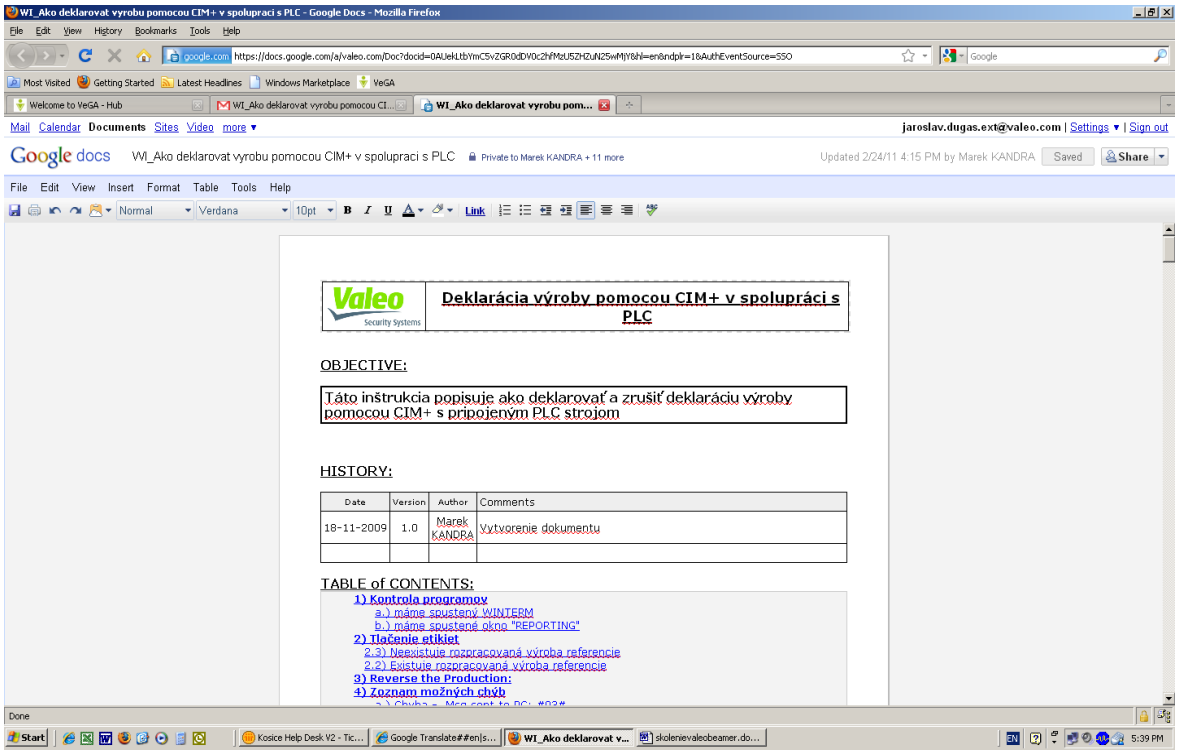
Čiže po rozbalení úlohy vidíme, že finančný pracovník z pobočky Dijon by potreboval prístup k finančným súborom v košickej pobočke Valea:



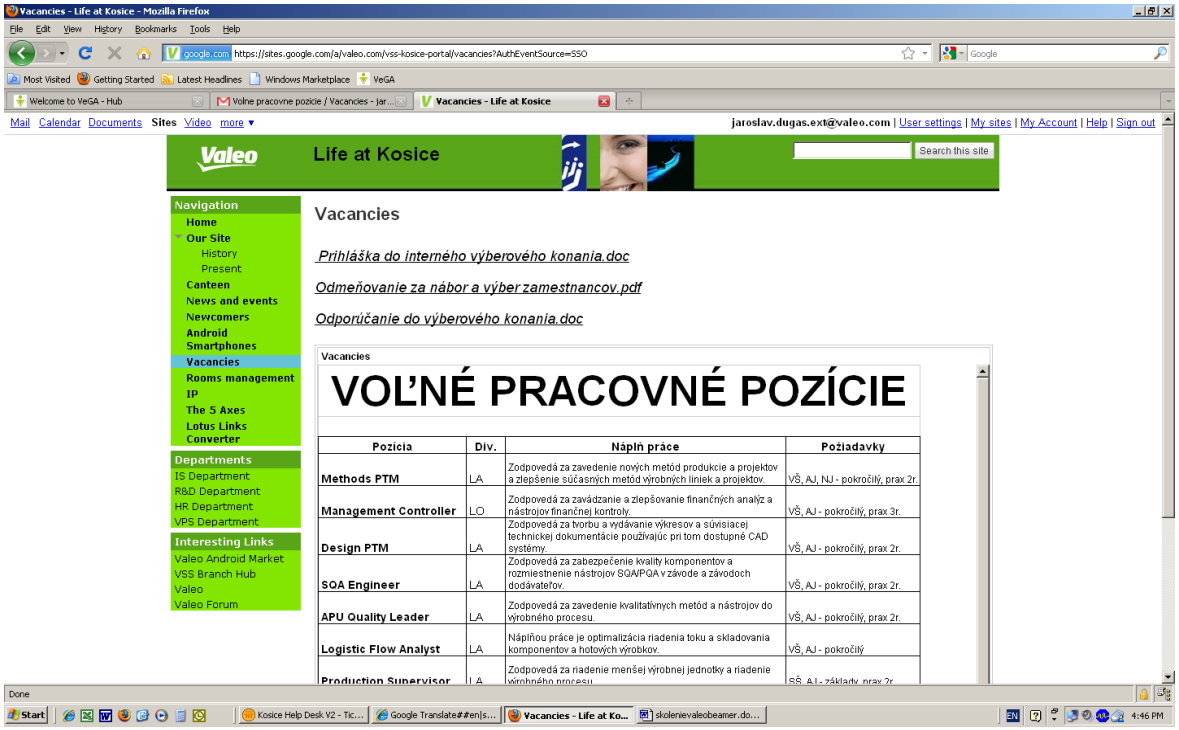

Po splnení úlohy a vyplnení komentára pre zadávateľa úlohy by oznam o splnení úlohy prešiel celú elektronickú cestu z Košíc do Dijonu.

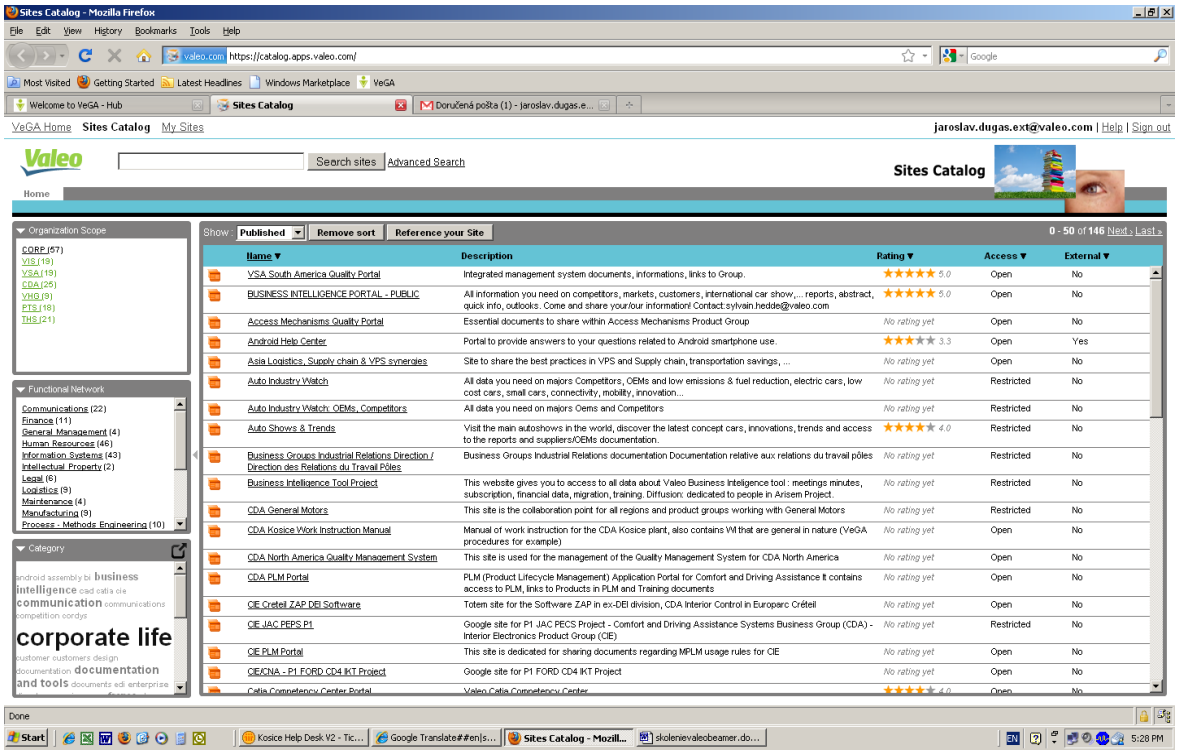
**Príloha č. 2:** Možnosť prezerania pracovného postupu aj mimo svojho počítača v spolupráci Lotusu a Google



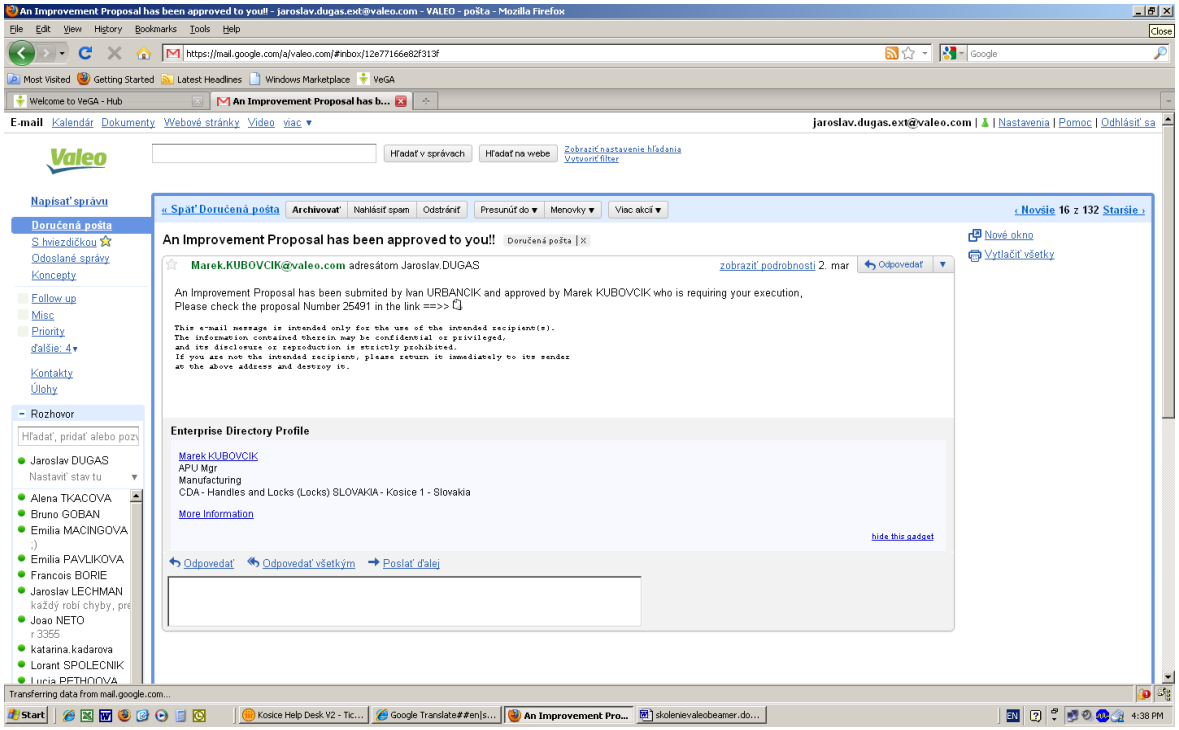



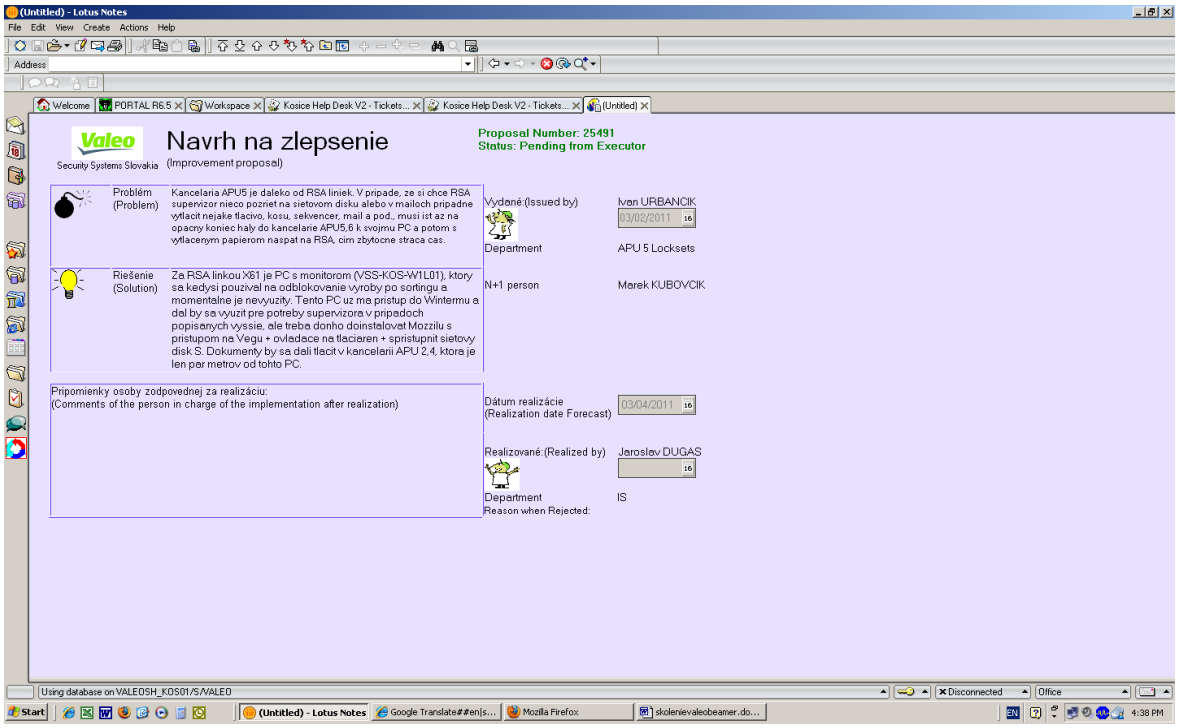
**Príloha č. 3:** Možnosť prezerania aktualít mimo svojho počítača v možnej spolupráci Lotusu a Google



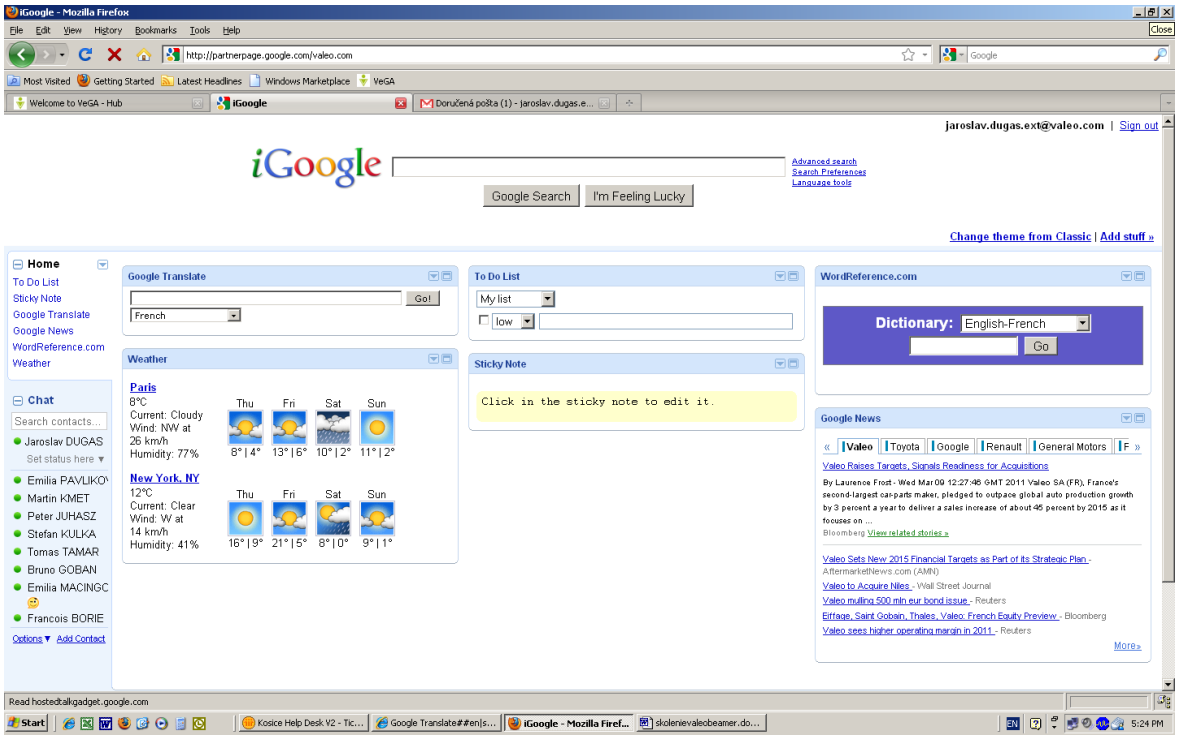



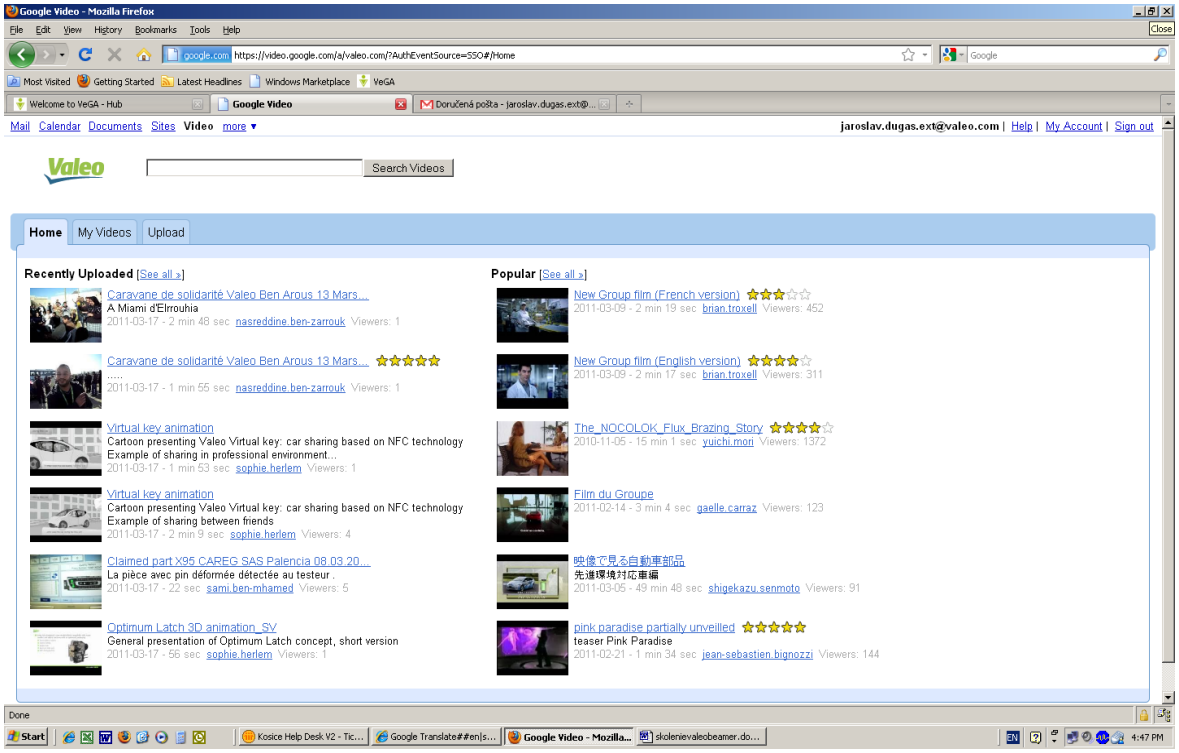
**Príloha č. 4:** Možnosť zadávania a riešenia zlepšovacích návrhov v možnej spolupráci Lotusu a Google



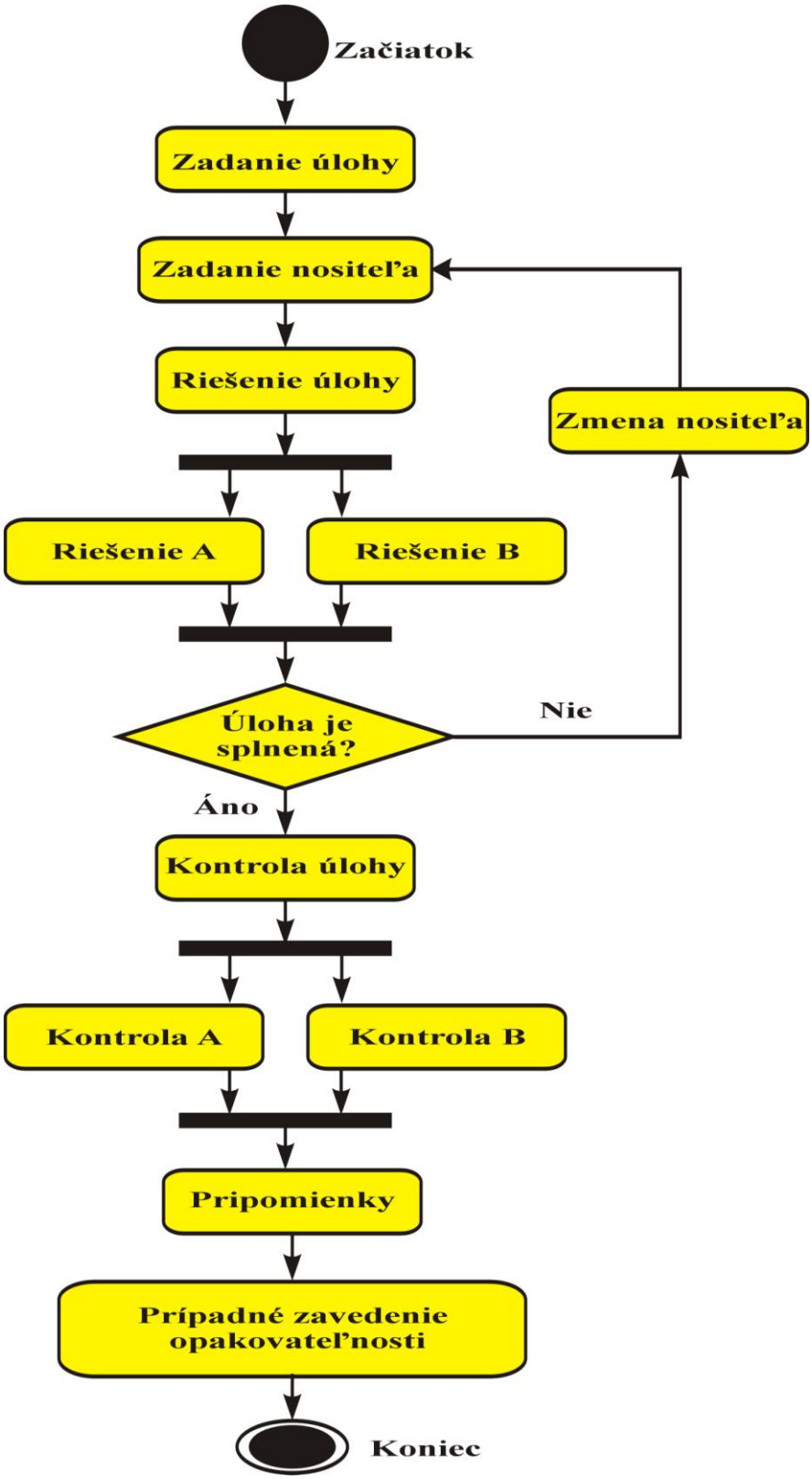



**Príloha č. 5:** Možnosť postupného prechodu Lotusových aplikácií úplne pod Google

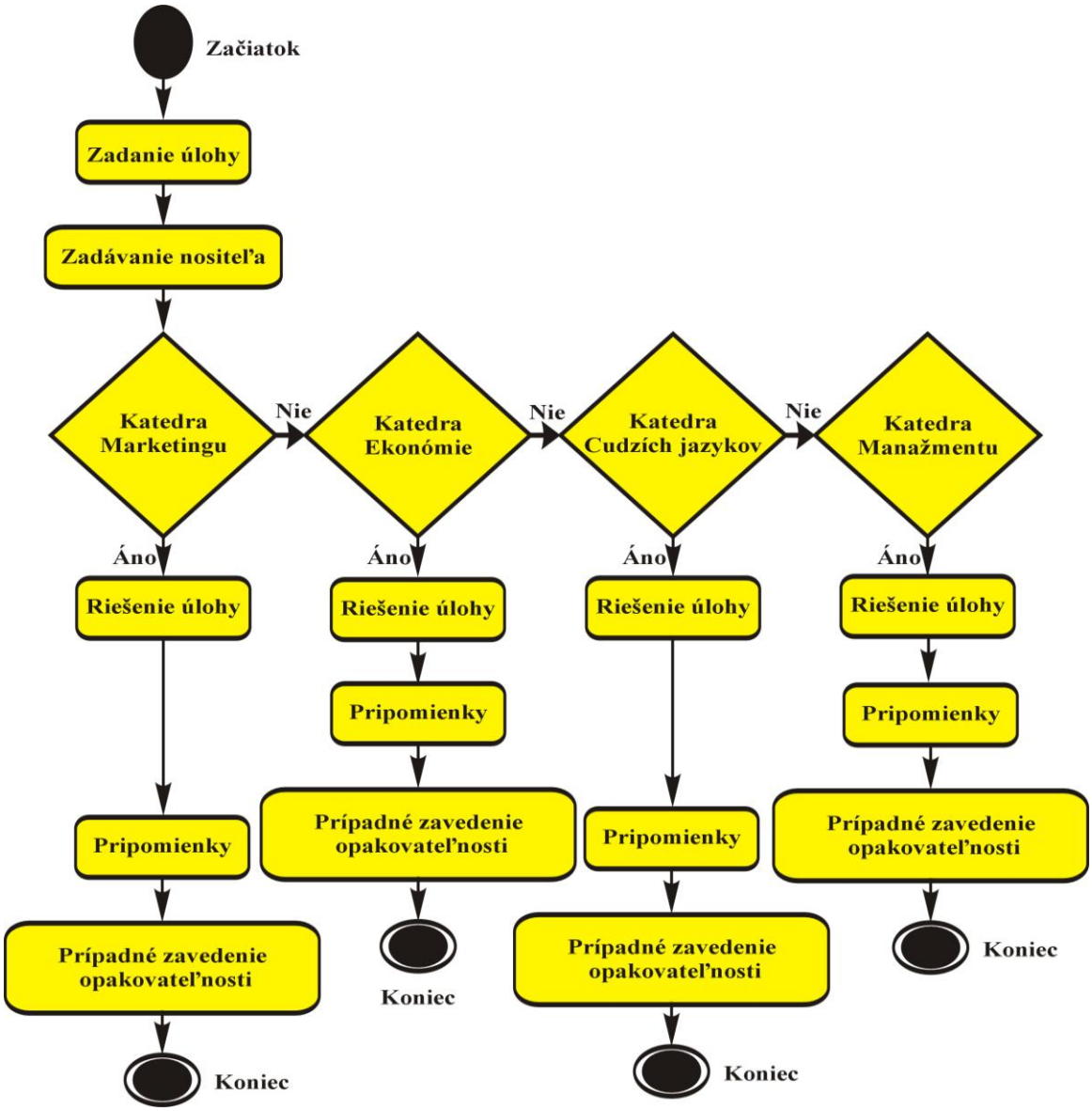


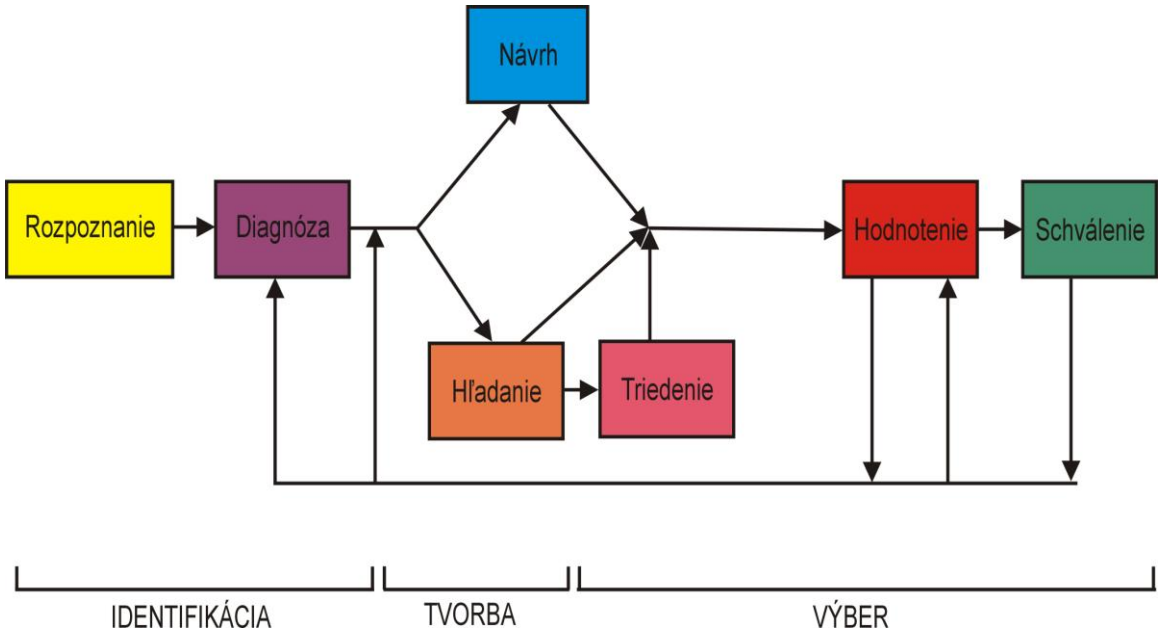
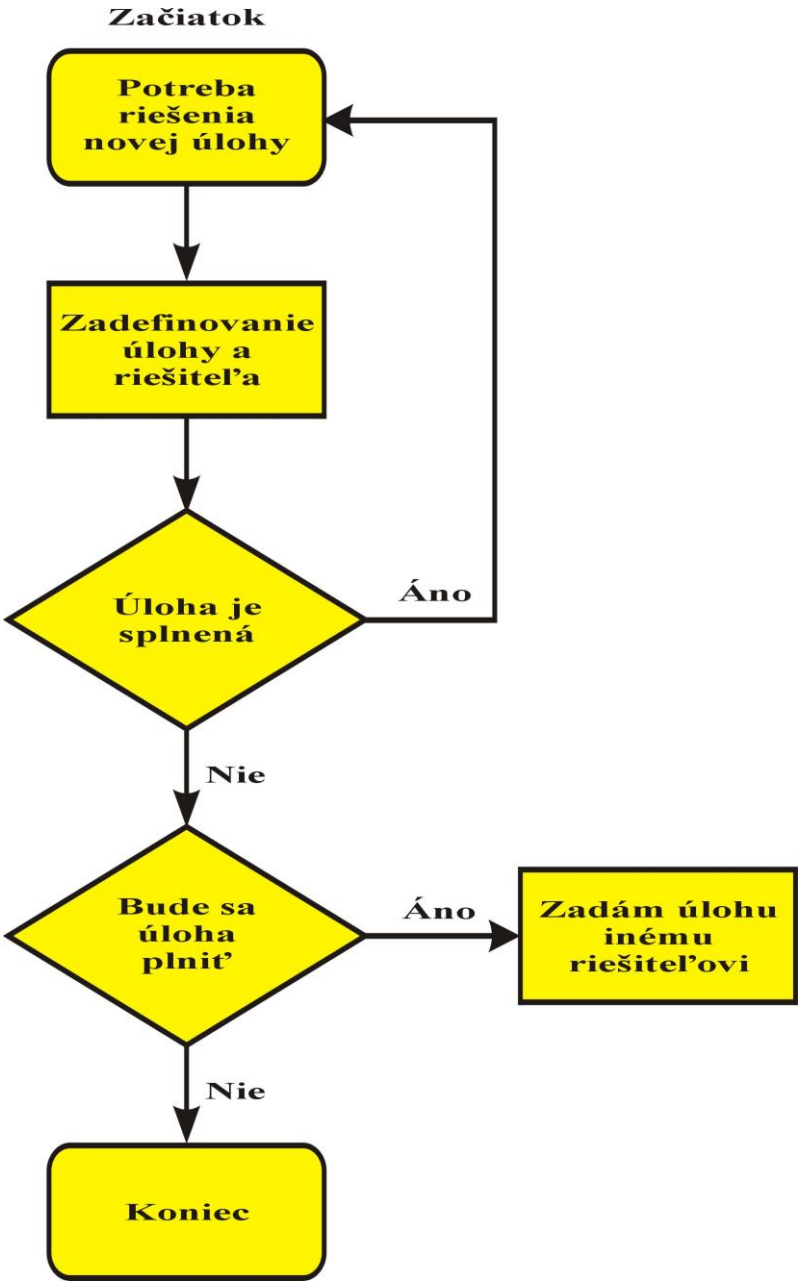
**Príloha č. 6:** Vývojový diagram zadania úlohy



**Príloha č. 7:** Vývojový diagram nositeľa úlohy



**Príloha č. 8:** Diagram postupu rozhodovania pri tvorbe zadávania úloh



**Príloha č. 9:** Diagram postupu rozhodovania pri potrebe zadávania úloh