

# **Informační systém pro operativní sdílení servisních dat**

Information system for operational sharing of service data

Bc. Michal Kouřil

---

Diplomová práce  
2009



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2008/2009

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal KOUŘIL**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Informační systém pro operativní sdílení servisních dat**

Zásady pro vypracování:

1. Prostudujte teorii modelování při tvorbě informačních a databázových systémů.
2. Provedte analýzu současných toků dat a informační vazby v oblasti údržby, oprav a instalace technologických zařízení v elektronické výrobě, které představují zázemí při práci servisního týmu, způsob jejich sběru a prezentace.
3. Navrhněte strukturu modifikovatelného informačního systému pro operativní sdílení servisních dat s přístupem přes www stránky včetně zabezpečení a správy vytvořeného systému.
4. Realizujte ověřovací instalaci navrženého informačního systému formou zabezpečené otevřené webové aplikace s možností uživatelského rozšiřování a modifikace. Provedte praktické ověření jeho funkčnosti.
5. Zpracujte příslušnou dokumentaci a prezentaci navrženého systému. Diskutujte případné doporučení pro modifikace a úpravy.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Gilmore,W.J.: **Velká kniha PHP & MySQL**, Zoner press, ISBN 80-86815-20-X.
2. Kosek,J.: **PHP tvorba iterativních internetových aplikací**, Grada Publishing, ISBN 80-7169-373-1.
3. Holzschlag,M.E.: **HTML a CSS jdi do toho**, Grada Publishing, ISBN 80-247-1454-X.
4. Schneider,R.D.: **MySQL oficiální průvodce tvorbou, správou a laděním databází**, Grada Publishing, ISBN 80-247-1516-3.
5. Gutmans,A., Bakken,S.S., Rethans,D.: **Mistrovství v PHP 5**, Computer Press, ISBN: 978-80-251-1519-0.
6. Kofler,M.: **Mistrovství v MySQL 5**, Computer Press, ISBN: 978-80-251-1502-2.
7. Farana,R.: **Databáze:speciální postupy**, Český svaz vědeckotechnických společností, ISBN: 80-02-01876-1.

Vedoucí diplomové práce:

**Ing. Petr Neumann, Ph.D.**

Ústav elektrotechniky a měření

Datum zadání diplomové práce:

**20. února 2009**

Termín odevzdání diplomové práce:

**27. května 2009**

Ve Zlíně dne 13. února 2009



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Tato diplomová práce se zabývá tvorbou informačního systému pro potřeby servisního týmu v oblasti údržby, oprav a instalací technologického zařízení v elektronické výrobě. Samotný informační systém má tvořit zázemí pro servisní tým, z kterého mohou čerpat informace pro servisní práci. Systém bude přístupný přes www stránky, čímž uživatel dosáhne přístupu k informacím např. z místa servisního zákroku. Pro vytvoření aplikace byl zvolen programovací jazyk PHP, databázový systém MySQL a uživatelské rozhraní je vytvořeno pomocí jazyka HTML a CSS.

Klíčová slova: HTML, PHP, CSS, MySQL, Analýza, Návrh, Implementace

## **ABSTRACT**

This diploma thesis deals with creation of information system for the need of a service team in the field of installation, maintainance and repairs of a technology equipment in electronic production. The informational system itself shall be the background for the service team that can pull out of it the information they need for a service work. System will be accessible via a website, allowing the user to access the information for instance at the service site. The program language chosen for creation of the application is PHP, the database system is MySQL and the user interface is created by HTML and CCS.

Keywords: HTML, PHP, CSS, MySQL, Analysis, Design, Implementation

Poděkování:

Děkuji panu Ing. Petru Neumannovi PhD. za poskytnuté rady, cenné připomínky i trpělivost při vedení diplomové práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>10</b>
<b>1 TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 TERMINOLOGIE INFORMAČNÍHO SYSTÉMU</b> .....	<b>12</b>
1.1 STRUKTURA INFORMAČNÍHO SYSTÉMU .....	13
<b>2 OBECNÁ KLASIFIKACE SYSTÉMU</b> .....	<b>14</b>
2.1 DALŠÍ KLASIFIKAČNÍ KRITERIA .....	14
2.2 OBSAH INFORMAČNÍHO SYSTÉMU .....	15
2.3 TVORBA IS .....	15
<b>3 METODIKA VÝVOJE INFORMAČNÍHO SYSTÉMU</b> .....	<b>16</b>
3.1 ANALÝZA .....	17
3.1.1 Strukturovaná metodika .....	18
3.1.2 Objektově orientované metody .....	19
3.1.2.1 Základní koncepty objektového přístupu.....	20
3.1.2.2 OMT (Object Modeling Technique).....	22
3.2 NÁVRH SYSTÉMU – DESIGN .....	26
3.2.1 Systémový návrh systému – design.....	26
3.2.2 Objektový návrh systému – design.....	27
3.3 IMPLEMENTACE .....	28
3.3.1 Hosting .....	29
3.3.2 Open Source .....	30
<b>4 DATOVÉ MODELY</b> .....	<b>31</b>
4.1 KONCEPTUÁLNÍ MODEL .....	32
4.1.1 E-R model .....	32
4.2 LOGICKÝ MODEL .....	33
4.2.1 Hierarchický model .....	33
4.2.2 Síťový model.....	34
4.2.3 Relační model.....	35
4.2.4 Objektově orientovaný model .....	36
4.2.5 Objektově relační model .....	37
4.2.6 XML model .....	37
4.3 FYZICKÝ MODEL.....	37
<b>5 ŽIVOTNÍ CYKLUS INFORMAČNÍCH SYSTÉMŮ</b> .....	<b>40</b>
5.1 VODOPÁDOVÝ MODEL ŽIVOTNÍHO CYKLU .....	40
5.2 PROTOTYPOVÝ MODEL ŽIVOTNÍHO CYKLU.....	42
5.3 SPIRÁLOVÝ MODEL ŽIVOTNÍHO CYKLU .....	43
<b>6 METODIKA TVORBY INFORMAČNÍCH SYSTÉMŮ</b> .....	<b>45</b>

6.1	CASE (COMPUTER AIDED SOFTWARE ENGINEERING) .....	45
<b>7</b>	<b>POUŽITÉ WEBOVÉ TECHNOLOGIE.....</b>	<b>47</b>
7.1	HTML.....	47
7.2	CSS.....	47
7.3	JAVASCRIPT .....	48
7.4	PHP.....	49
7.4.1	PHP 5 .....	49
7.5	MYSQL.....	50
7.5.1	phpMyAdmin .....	50
<b>8</b>	<b>EFEKTIVNOST ZAVÁDĚNÍ IS .....</b>	<b>52</b>
<b>9</b>	<b>ZABEZPEČENÍ PODNIKOVÉHO IS.....</b>	<b>53</b>
9.1	BEZPEČNOSTNÍ HROZBY A RIZIKA .....	53
9.2	ZAJIŠTĚNÁ BEZPEČNOSTI DAT .....	54
9.3	ZAJIŠTĚNÍ BEZPEČNOSTI SLUŽEB.....	54
9.4	ZAJIŠTĚNÍ OCHRANY TECHNICKÝCH PROGRAMOVÝCH PROSTŘEDKŮ .....	54
9.5	ZAJIŠTĚNÍ OCHRANY PŘE ZNEUŽITÍM ČI ODCIZENÍM DAT.....	55
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>56</b>
<b>10</b>	<b>ANALÝZA .....</b>	<b>57</b>
10.1	ZADÁNÍ .....	57
10.2	BLIŽŠÍ SPECIFIKACE .....	57
10.2.1	Obchodní oddělení .....	57
10.2.2	Servisní oddělení.....	58
10.2.3	Seznam zákazníků a zařízení .....	58
10.3	DŮLEŽITÉ FAKTORY.....	58
10.3.1	Forma požadovaného systému .....	58
10.3.2	Velikost požadovaného systému .....	59
10.3.3	Správa dat.....	59
10.4	ANALÝZA PROCESŮ VE FIRMĚ POMOCÍ OBJEKTIVĚ ORIENTOVANÉ METODY .....	59
10.4.1	Vytvoření funkčního modelu.....	59
10.4.1.1	Funkční model prodeje stroje.....	60
10.4.1.2	Funkční model servisního zásahu .....	61
10.4.2	Vytvoření objektového modelu .....	62
10.4.2.1	Objektový model prodeje stroje .....	62
10.4.2.2	Objektový model servisního zásahu stroje.....	63
10.4.3	Vytvoření dynamického modelu .....	64
10.4.3.1	Dynamický model prodeje stroje.....	64
10.4.3.2	Dynamický model servisního zásahu .....	65
10.4.3.3	Zodpovědnosti pracovníků.....	66
10.5	MODELOVÁNÍ ENTIT .....	70
10.5.1	Modelování entit a jejich stavů při prodeji stroje.....	70



10.5.2	Modelování entit a jejich stavů při servisu stroje.....	71
10.6	DIAGRAM TŘÍD .....	72
10.6.1	Diagram tříd prodeje stroje .....	72
10.6.2	Diagram tříd servisu stroje .....	73
10.7	VÝSLEDEK ANALÝZY .....	73
<b>11</b>	<b>NÁVRH SYSTÉMU .....</b>	<b>75</b>
11.1	SYSTÉMOVÝ NÁVRH SYSTÉMU.....	75
11.1.1	Struktura databáze .....	76
11.2	OBJEKTOVÝ NÁVRH SYSTÉMU .....	76
11.2.1	Diagram tříd v systému .....	78
<b>12</b>	<b>IMPLEMENTACE .....</b>	<b>80</b>
12.1	OVĚŘOVÁNÍ A SPRÁVA UŽIVATELŮ.....	80
12.2	MATERIÁLY A KATEGORIE.....	82
12.3	BEZPEČNOST SYSTÉMU .....	83
12.3.1	Přihlášení do systému.....	83
12.3.2	SQL injection .....	84
12.3.3	Uložení přihlašovacích hesel v databázi .....	85
12.4	INSTALOVANÝ SOFTWARE NUTNÝ PRO BĚH APLIKACE .....	85
12.4.1	Instalace.....	85
12.5	POŽADAVKY NA HARDWARE.....	86
	<b>ZÁVĚR .....</b>	<b>87</b>
	<b>SUMMARY:.....</b>	<b>88</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>89</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>91</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>92</b>

## ÚVOD

V současné společnosti můžeme sledovat určité základní rysy ve vztahu k informačním systémům. Je pravda, že bez funkčního a flexibilního systému se dnes neobejde snad již žádná firma. Můžeme sledovat, že jednotlivé podniky vydávají nebývalé investiční prostředky na realizaci informačních systémů. Vše je dáno faktem, že roste potřeba rychlé, včasné a správné informace, čímž dochází ke zlepšení konkurenční schopnosti a úspoře finančních prostředků. Samotná důvěra v informační systémy rychle roste a většina pověřených zaměstnanců firem dokáže po zaškolení ovládat svou uživatelskou část, což také přispívá k rozmachu v oboru této lidské činnosti i navzdory tomu, že aplikace jsou rozsáhlejší, komplexnější a také náročnější.

Uživatel má určité požadavky, které musí kvalitní informační systém splňovat. Jedná se především o plnou podporu strategické orientace podniku a odpovídající funkční spektrum aplikace. Samozřejmostí je integrace, provozuschopnost, flexibilita a udržovatelnost samotného informačního systému jako celku. V neposlední řadě by měl informační systém být efektivně provozuschopný, otevřený, rozšiřovatelný a v rámci možností jednoduchý.

Podnikový informační systém patří mezi strategické faktory prosperity a konkurenceschopnosti firem. Úspěšnost zavedení informačního systému v řízení firmy záleží na strategickém řízení samotného systému. Cílem strategického řízení je dát informačnímu systému a informačním technologiím podnikatelskou hodnotu. Tato podnikatelská hodnota informačního systému musí být nejdříve formulována vizí dávající smysl a cíl dalším aktivitám.

## I. TEORETICKÁ ČÁST

## 1 TERMINOLOGIE INFORMAČNÍHO SYSTÉMU

Informační a komunikační technologie poslední dobou získávají své nezastupitelné místo nejen v systémech řízení firem a podnikatelských subjektů, ale i ve sférách sociálních, záchranných atd. Dá se říci, že tyto technologie pronikly snad do všech oborů lidské činnosti.

Úspěch informačních a komunikačních technologií vytváří poptávku na stále zdokonalování jak rychlosti poskytování, tak i zpracování stále větších objemů dat. Základem dnešní společnosti by měla být schopnost pracovat se stále většími objemy dat. Schopností práce s daty rozumíme orientaci ve velkém objemu dat a umění je zpracovávat. Výstupem jsou potřebná data, na základě kterých pracovníci vyvozují patřičné závěry.

Slovo **informace** je součástí pojmového aparátu každého člověka, jehož význam je hodně široký. Obecně lze informaci chápat jako ve smyslu sdělování nějaké zprávy, poznatku, události či jevu. Jedná se tedy o název toho, co si vyměňujeme se svým okolím. Kvalitní informace snižuje neznalost nebo nejistotu v konkrétní situaci. Podmínkou kvalitní informace je její cílenost, přesnost, rychlost, podání přiměřeného množství a prezentace vhodnou formou.[4]

Slovo **system** je blízký pojmu celistvosti a struktury. Je chápán jako účelově definovaná množina prvků a vazeb mezi nimi. Dále se užívá jako označení určité části reálného světa s charakteristickými vlastnostmi.

**Informační systém** je účelnou formou využití informačních technologií, kdy člověk může ovlivňovat jeho kvalitu.

### Základním úkolem IS je

- 1) Sběr informací.
- 2) Uchování a zpracování informací do požadovaného tvaru příp. rozsahu.
- 3) Předání informací řídicím úrovním včas [11]

## 1.1 Struktura Informačního Systému

Informační systém se skládá z následujících komponent:

- 1) technické prostředky (hardware) – počítačové systémy různého druhu a velikosti, doplněné o potřebné periferní jednotky, které jsou v případě potřeby propojeny prostřednictvím počítačové sítě a napojeny na paměťový subsystém pro práci s velkými objemy dat;
- 2) programové prostředky (software) – tvořené systémovými programy, řídicími chod počítače, efektivní práci s daty a komunikaci počítačového systému s reálným světem, a programy aplikačními řešící určité třídy úloh tříd uživatelů;
- 3) organizační prostředky (orgware) – tvořené souborem nařízení a pravidel, definující provozování a využívání informačního systému a informačních technologií;
- 4) lidská složka (peopleware) – řešení otázky adaptace a účinného fungování člověka v počítačovém prostředí, do kterého je vřazen;
- 5) reálný svět (informační zdroje, legislativa, normy) – kontext informačního systému.

Má-li být informační systém firmy či instituce efektivní, nesmí být při jeho vývoji zanedbána žádná z jeho složek. [4]



Obrázek 1: Prvky informačního systému

## 2 OBECNÁ KLASIFIKACE SYSTÉMU

Systém charakterizujeme jako specifický model objektu, který dostatečně charakterizuje všechny jeho vlastnosti.

### Požadavky na systém jako model nad objektem

- 1) Zobrazení systémových znalostí
- 2) Zjednodušení složitostí
- 3) Homogenizace popisu
- 4) Měřitelnost vlastností na objektu
- 5) Reprezentace vlastností během časového vývoje , reprezentace genetických vlastností, které reprezentují dědičnost systému [10]

### Okolí systému

Několik vrstev projevu systému vůči okolí

#### VRSTVY:

- 1) Existenční (vnitřní chování systému)
- 2) Akční okolí - prostor, do kterého zasahuje systém svými akčními členy:
  - Energetickými
  - Informačními
  - Hmotnými
- 3) Kontaktní (ještě se zaznamenává přítomnost systému)
- 4) Informační okolí (IS, Telekomunikační s., dál.přenos dat, reklama) [10]

### 2.1 Další klasifikační kritéria

#### Podle struktury:

- 1) Determizované (přesně podle plánu a projektu)
- 2) Stavově determizované (plán závislý na výchozím stavu)
- 3) Stochastické (pravděpodobnostní chování)

**Podle režimu:**

- 1) Statické
- 2) Dynamické

**Podle vztahu k okolí:**

- 1) Otevřené (výměna látek, energie, informací)
- 2) Uzavřené (izolované, výměna pouze jedné veličiny) [10]

## 2.2 Obsah Informačního Systému

Obsah IS můžeme vyměřit těmito hlavními charakteristikami:

- 1) Funkcemi, funkcionalitou IS (hierarchicky uspořádaný souhrn všech operací s daty).
- 2) Procesy, které podporuje. IS by měl v co největší míře podporovat optimalizaci podnikových procesů (procesní řízení firmy).
- 3) Daty, datovými zdroji, které jsou předmětem zpracování v rámci funkcí a procesů IS. [11]

## 2.3 Tvorba IS

- 1) Tvorba IS je vývoj programového vybavení včetně hardwarového pozadí, bezpečnostních mechanismů či pracovních postupů a doporučení.
- 2) Tvorbě nového IS nebo inovaci starého IS předchází pečlivý popis a hodnocení současného stavu a definování cílového stavu.

Bere se v úvahu

- 1) Analýza stavu IS konkurence nebo klíčových partnerů
- 2) Hodnocení stávajícího aplikačního software
- 3) Trendy IS
- 4) Výsledky SWOT analýzy
- 5) Podnikové cíle, priority podniku
- 6) Požadavky uživatelů. [11]

### 3 METODIKA VÝVOJE INFORMAČNÍHO SYSTÉMU

K vytvoření informačního systému se pouhý hierarchický rozklad ukázal jako nedostatečný, přestože by byl specializován. Proto je nutné návrh rozčlenit do několika etap, ve kterých se budou řešit odlišné části realizace. Metodika vývoje informačního systému se stala základním standardem a návodem vztahujícím se na:

- 1) pracovníky vývoje
- 2) organizaci práce
- 3) práci s informacemi o vyvíjeném informačním systému
- 4) SW a HW prvky
- 5) ekonomické otázky vývoje IS
- 6) projektovou dokumentaci k IS
- 7) způsob řízení v jednotlivých fázích vývoje IS

Metodika umožní vytvořit:

- 1) definici kritérií kvality pro každou etapu vývoje IS
- 2) komunikační standard tvůrců i uživatelů vyvíjeného IS
- 3) lepší kooperaci vývojových týmů
- 4) lepší specializaci projektových týmů na jednotlivé etapy
- 5) tvorbu kvalitní a aktuální dokumentace
- 6) větší pružnost k potřebám uživatelů
- 7) kvalitnější řízení prací na projektech

Rozčlenění návrhu systému do etap:

- 1) analýza
- 2) návrh systému - design
- 3) implementace



V praxi to znamená, že analýza rozhodne co se bude dělat. Design určí jak se to bude dělat a implementace je samotné provedení.

### 3.1 Analýza

Analýza systému usiluje o celistvý a komplexní pohled na studovaný systém ve všech jeho vnitřních i vnějších souvislostech. Obtížnost analýzy složitých, rozsáhlých a málo průhledných systémů však vede k tomu, že je účelné získávat celistvý a komplexní pohled postupně. Proto analýzu dělíme na dvě části:

- 1) analýza struktury systému
- 2) analýza chování systému

Je nezbytné mít přehled o cílech a o cestách jak je dosáhnout. Dále je nutné kromě znalostí cest, které povedou k řešení, znát i cesty, které nepovedou k řešení a proč k řešení nepovedou. Tím eliminujeme navrhování zbytečných řešení, jenž by stejně nebyly využity k vyřešení problematiky.

Největší problém nastává při popisu požadavků za strany zákazníka, kdy mnohdy v oboru informačních systémů laická veřejnost vysvětluje odborné skupině tvůrců informačních systémů své požadavky na funkcionalitu a nabízené služby požadovaného systému. To vše klade vysoké nároky na porozumění samotným požadavkům.

Analýzou vzniká logické řešení problému, bez ohledu na prostředí, v němž poběží a bez ohledu na způsob, jakým bude realizován

Bez dobře udělané analýzy se vytvoří pouhá změť nespolupracujících řešení nebo řešení, které pracují proti sobě, kdy práce jedné části zruší práci jiné části navrhovaného systému.

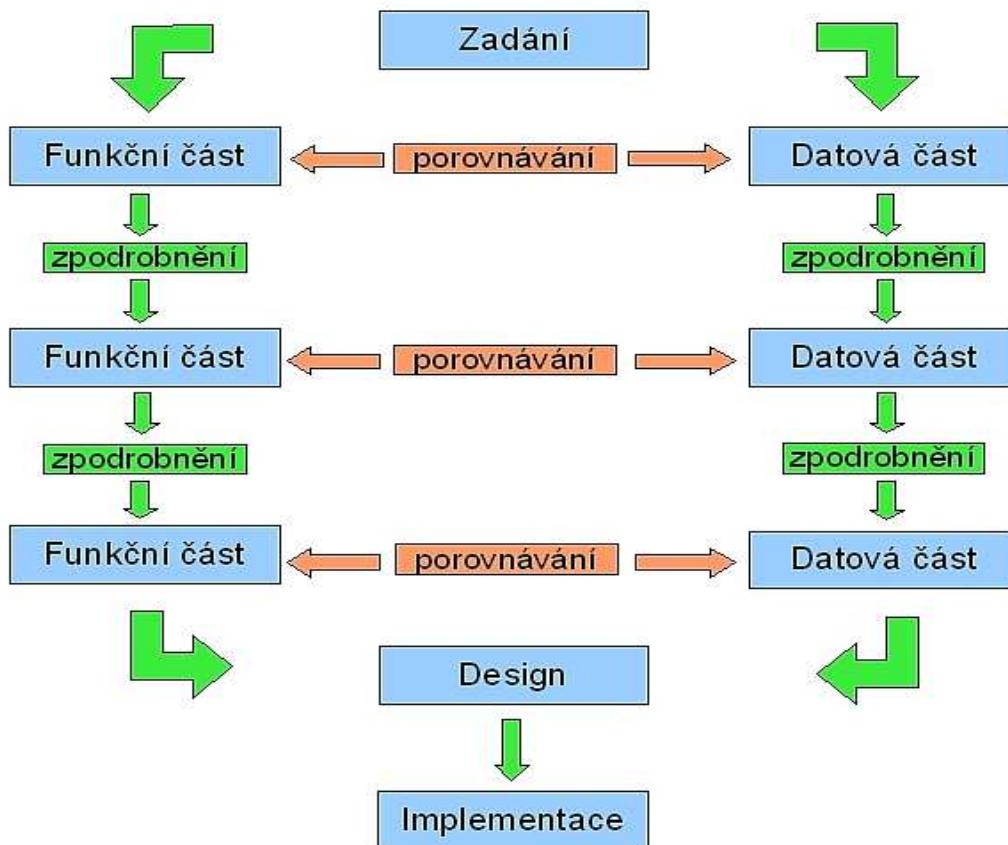
Naopak bezvadně udělaná analýza se všemi požadavky a náležitostmi ještě není vytvořený systém.

Problém analýzy řeší strukturované metodiky.

### 3.1.1 Strukturovaná metodika

Smyslem analýzy struktury systému je vyšetření vztahů mezi prvky a vazbami systému, a to v převážně statické podobě. Hlavní metodou při analýze struktury je logický rozbor vycházející ze znalosti problematiky a vhodně utříděných dat.

Princip strukturované metodiky spočívá v rozkladu analýzy na funkční část a datovou část. Protože bylo nutné při tvorbě funkční a datovou část sjednotit, dochází k paralelní tvorbě se vzájemným porovnáváním. Jedná se o myšlenku, která je úspěšně používána například ve strojírenství, architektuře atd. V realitě tahle myšlenka aplikovaná na příkladu strojírenství nebo architektuře funguje tak, že je jako plán budoucího trojrozměrného výrobku použito několik dvourozměrných výkresů.



Obrázek 2: Strukturovaná metodika

Při analýze složitých struktur si obvykle rozdělíme požadovaný systém na řadu dílčích kroků, při níž přecházíme od globálního pohledu k detailům. Z tohoto důvodu jsou některé metody analýzy struktury uváděny ve dvou verzích:

a, globální

určujeme pouze prvky a vazby struktur, které vykazují určitou vlastnost bez podrobnějších informací o vzájemných vztazích mezi prvky a vazbami.

b, detailní

kdy získáváme informace o vzájemných vztazích mezi prvky a vazbami.

### 3.1.2 Objektově orientované metody

Objektově orientované metody analýzy vznikly jako nástupce strukturované metodiky analýzy. Jejich rozšíření nenabývalo takové dynamiky jak se předpokládalo. Tento fakt byl dán především zvykem a zaběhlou praxí analytiků v užívání starší strukturované metodiky.

Používání strukturované metodiky se může zdát jako nejvhodnější způsob dosažení cíle, což je ale omyl, protože výsledný systém může být velmi nestabilní. Příčinou jsou případné změny v zadání, kdy je nutné provést masivní restrukturalizaci. Objektově orientované metody se zaměřují nejprve na identifikaci objektů (dat) a data jsou podstatně stabilnější než algoritmus. To je jednoznačná výhoda objektově orientovaných metod.

Některé známé objektově orientované metody:

- 1) OMT (Object Modeling Technique)
- 2) OOAD (Object Oriented Analysis and Design)
- 3) OOA (Object Oriented Analysis)
- 4) SDM (System Development Metodology)
- 5) SSADM (Structured Systems Analysis and Design Method)
- 6) MERISE

Samotné metody jsou principiálně shodné a liší se jen v detailech.

Výhody této metody:

- 1) zvyšuje se kvalita vyvíjeného produktu v důsledku používání systémového přístupu, který zajišťuje přesně definované kroky pro zajištění kvality
- 2) zjednodušuje se plánování a řízení, protože jsou dopředu známy jednotlivé kroky vývoje systému
- 3) usnadňuje se komunikace mezi osobami které se účastní vývoje systému (zákazník, analytik, řešitel), protože jsou používány jednotné prostředky pro komunikaci (tabulky, grafy, terminologie)
- 4) zvyšuje se efektivita práce analytického týmu, protože jsou používány racionální techniky a přístupy
- 5) nezávislost na jednotlivých osobách analytického týmu (migrace zaměstnanců mezi firmami)

### ***3.1.2.1 Základní koncepty objektového přístupu***

Všechny následující prvky jsou používány v UML. Notace UML je v současné době jedinou používanou objektovou notací v analýze, přičemž využívá všech prvků a dřívějších notací objektového přístupu.

#### ***3.1.2.1.1 Abstrakce***

Jedná se stanovení vlastností „atributů“ a operací „metod“ objektu, které budou v modelu použity.

#### ***3.1.2.1.2 Třída***

Představuje zobecnění reálných modelovaných objektů v analyzovaném systému. Každý objekt třídy má své popisné charakteristiky „atributy“, kdy atribut může mít nějaké omezení. Objekty, které jsou prvky definované třídy, se vyznačují určitým způsobem chování, které popisujeme prostřednictvím „metod“. Popisné charakteristiky metod se nazývají „příznaky“.

Pokud jedna třída přímo využívá jinou třídu mluvíme o „závislosti“.

Rozhraní (interface) představuje skupiny operací, které stanovují chování třídy a její vztah k jiným třídám. Vztah mezi třídou a rozhraním nazýváme „realizace“.

Další vlastností třídy je „viditelnost“ jejích atributů. Existují tři úrovně viditelnosti:

- 1) public
- 2) private
- 3) protecte

#### 3.1.2.1.3 Dědičnost

Popisuje stav, kdy objekt dědí atributy a metody třídy do které patří.

Rozlišujeme „nadtřídu“ a „podtřídu“.

#### 3.1.2.1.4 Polymorfismus

Jedná se o dostatečnou znalost třídy určující jak provést určitou operaci - může být společná i pro více tříd.

#### 3.1.2.1.5 Zapouzdření

Umožňuje skrytí informací samotných objektů tříd před okolím. Informace je možné zpřístupnit přes definovaný okruh metod. Slouží k rozlišování dat a operací soukromých a veřejných v závislosti na místě pohledu.

#### 3.1.2.1.6 Synergie a objektová struktura

Popisuje situaci, kdy objekt je spojením dat a chování. Objektová struktura určuje co objekt je a ne jak je užít.

#### 3.1.2.1.7 Asociace

Určuje vztah mezi objekty. Známe „jednosměrné“ a „obousměrné“. Každá třída má v asociaci danou určitou „roli“ – tu je nutné zapsat na konec asociace ke třídám, s nimiž souvisejí. Vztahu mezi objekty může být přiřazeno určité „pravidlo“. Počet vazeb objektů jedné třídy k počtu vazeb objektů druhé třídy vyjadřuje „násobnost“.

#### 3.1.2.1.7.1 Vazba

Jedná se o samotnou vazbu mezi objekty „link“.

#### 3.1.2.1.8 Agregace

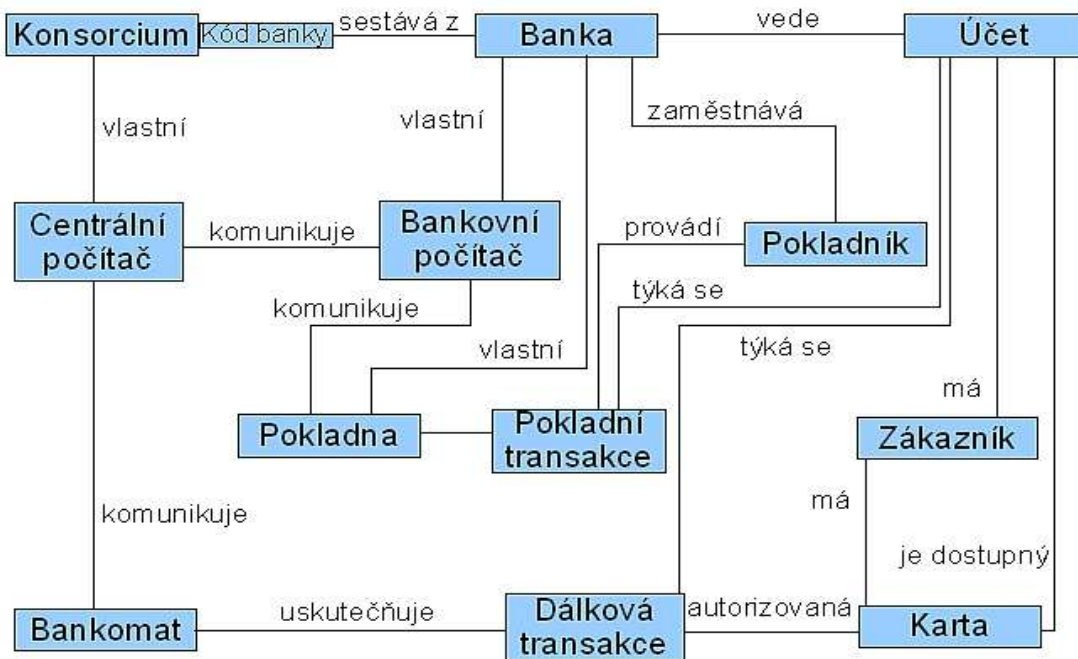
Znázorňuje složení objektu třídy z více objektů jiných tříd. Vyšším řádem agregace je složenina „kompozice“.

Pro znázornění objektově orientovaných metod analýzy popíše podrobněji metodu OMT.

### 3.1.2.2 OMT (*Object Modeling Technique*)

Během analýzy jsou vytvořeny tři vzájemně vztážené modely: objektový model, dynamický model a funkční model. Každý model má svoje nezastupitelné místo a nelze jej vynechat. Celý proces analýzy pak vypadá takto:

- 1) Specifikace problému
- 2) Sestavení objektového modelu:
  - identifikace objektových tříd
  - vytvoření datového slovníku, který popíše třídy atributy a vzájemné propojení
  - doplnění atributů objektů a atributů vazeb
  - rozdělení tříd do modulů, podle vazeb a funkční provázanosti
  - organizování a zjednodušení objektových tříd pomocí dědičnosti.[14]

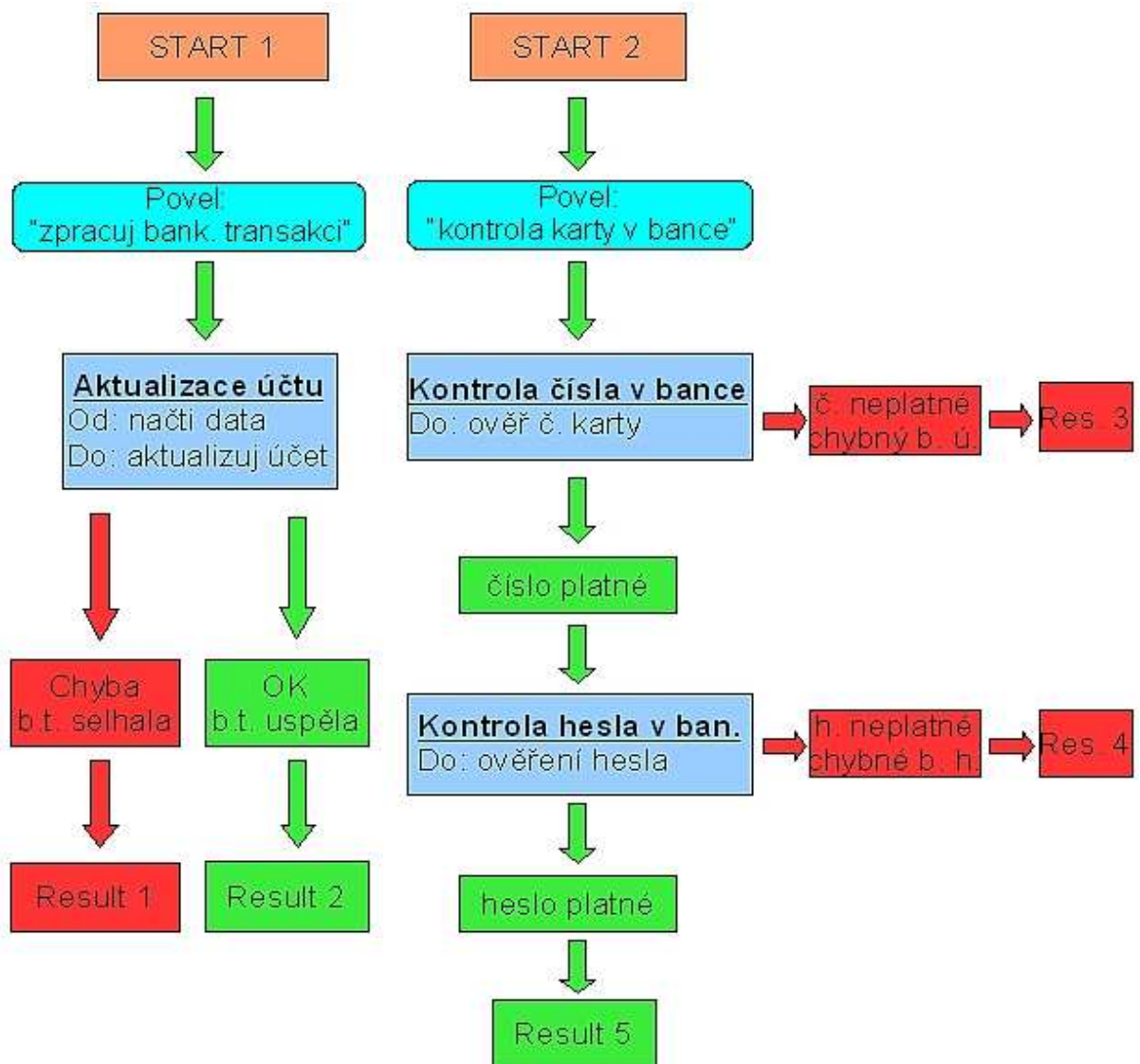


Obrázek 3: Objektový model

## 3) Tvorba dynamického modelu:

- příprava scénářů pro typické průchody aplikací
- identifikace jednotlivých událostí od objektů
- příprava diagramu toku událostí pro systém
- tvorba stavového diagramu pro každou třídu s důležitým dynamickým chováním
- kontrola konzistence a úplnosti událostí, sdílených jednotlivými stavovými diagramy [14]

Výsledkem je dynamický model



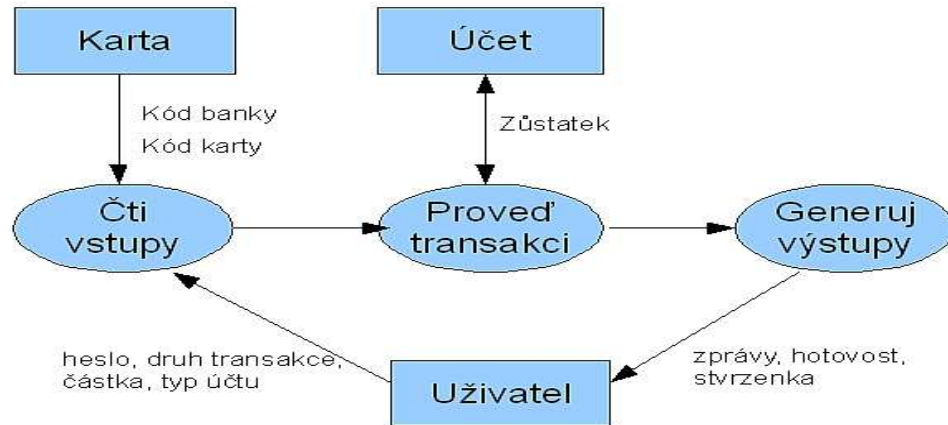
Obrázek 4: Dynamický model

## 4) Konstrukce funkčního modelu:

- identifikace vstupních a výstupních hodnot
- objevení funkčních závislostí
- popis, co dělá každá funkce
- identifikace omezení
- specifikace kritérií pro optimalizaci [14]



Výsledkem je funkční model



Obrázek 5: Funkční model

5) Ověření a vylepšení všech tří modelů:

- přidání klíčových operací (objevených během přípravy funkčního modelu) do objektového modelu
- ověření, že třídy, jejich vzájemné propojení, atributy a operace jsou konzistentní a kompletní (při dané úrovni abstrakce). Porovnání modelu se specifikací problému a důležitými doménovými znalostmi (tj. znalostmi z daného oboru) a testování modelu pomocí různých scénářů
- tvorba detailnějších scénářů, tj. různých variací základních scénářů. Použití scénářů typu "Co, kdyby..." pro další a lepší testování.
- opakování všech kroků dle potřeby [14]

Výsledkem je dokument analýzy skládající se ze:

- 1) specifikace problému
- 2) objektového modelu
- 3) dynamického modelu
- 4) funkčního modelu

## 3.2 Návrh systému – design

Vstupem pro návrh systému jsou dostatečně zpracované podklady analýzy systému. Analýza nám určí, co se má udělat a návrh systému (design) nám určí, jak se to má udělat. Samozřejmě, že nelze striktně oddělit analýzu od návrhu systému. Praxe ukázala, že je nezbytné, aby tvůrci analýzy byli v přímém kontaktu s tvůrci návrhu systému. Při této komunikaci a přírůstkovém životním cyklu projektu dochází k nevyhnutelnému prolnutí těchto dvou, na prvý pohled odlišných, činností. Reálně dochází k prohlubování analýzy za již běžícího procesu návrhu systému.

Samotný proces návrhu systému můžeme rozdělit na dvě kategorie:

- 1) systémový návrh systému (design)
- 2) objektový návrh systému (design)

### 3.2.1 Systémový návrh systému – design

Návrh systému je první stádium, ve kterém je vybrán základní přístup pro řešení problému. Je vybrána architektura systému, tj. celková organizace dílčích komponent systému, tzv. subsystémů. Tato fáze vývoje se skládá z následujících kroků:

- 1) Uspořádání systému do subsystémů
- 2) Identifikování a rozpoznání základní souběžnosti (tj. souběžně prováděných akcí)
- 3) Přidělování subsystémů procesorům a úkolům
- 4) Vybrání základní strategie pro implementaci
- 5) Identifikování globálních zdrojů a stanovení mechanismů pro kontrolu přístupu do těchto zdrojů
- 6) Vybrání přístupu pro implementaci softwarového řízení
- 7) Zvažování hraničních podmínek
- 8) Ustanovení vzájemných priorit. Výsledkem je tzv. dokument systémového návrhu (System Design Dokument) = struktura základní architektury pro systém + rozhodnutí o celkové strategii a postupu řešení. [14]

### 3.2.2 Objektový návrh systému – design

Během objektového návrhu podrobně rozpracováváme model vytvořený během analýzy. Výsledek této bližší specifikace je pak brán jako výchozí při implementaci. Postup je následující:

- 1) Získání operací pro objektový model z ostatních modelů:
  - nalezení operace pro každý proces z funkčního modelu
  - definování operace pro každou událost z dynamického modelu v závislosti na implementaci
- 2) Návrh algoritmů pro implementaci operací:
  - vybrání algoritmů s minimální náročností na implementaci
  - vybrání příslušných datových struktur, vhodných pro jednotlivé algoritmy
  - pokud je to potřeba, definujeme nové interní třídy a operace
  - přiřazení odpovědnosti za operace, které nejsou součástí žádné třídy
- 3) Optimalizace přístupů k datům:
  - opětovné uspořádání výpočtů pro zvětšení efektivity
  - úschova mezivýsledků pro zabránění opětovnému přepočítávání komplikovaných výsledků
- 4) Upravení struktury tříd pro zvýšení možností dědičnosti.
- 5) Návrh implementace propojení tříd.
- 6) Určení přesné reprezentace atributů objektu.
- 7) Rozdělení tříd do modulů. Výsledkem je tzv. Dokument návrhu (Design Document) = Detailní Objektový Model + Detailní Dynamický model + Detailní Funkční model. [14]

### 3.3 Implementace

Implementací je proces uskutečňování teoreticky stanovené myšlenky za účelem jejího dalšího použití. Samotné implementaci předchází analýza zadání, návrh systému a očekávaných výsledků. Nesoulad mezi funkčním předpokladem systému a skutečností může být způsoben chybou implementace nebo chybou samotné metody. Správná metoda je funkční nezávisle na způsobu implementace. Správná metoda je funkční nezávisle na způsobu implementace. [15]

Implementace tedy znamená postup zavedení informačního systému – čin je implementovaná myšlenka.

Informační systém většinou není možné ve firmě pouze nainstalovat jako každý jiný software. Je vyžadováno větší úsilí včetně vzájemné spolupráce dodavatele informačního systému, firmy a jejích zaměstnanců. Jedná se o nákladný a velmi důležitý krok. Má za úkol popsat dosavadní procesy a tyto nadefinovat do informačního systému tak, aby provádění těchto procesů bylo jednoduché, rychlé, efektivní a hlavně provázané s procesy, na kterých závisí.[16]

Implementace a kroky s ní související mají následnou strukturu:

- 1) Definice procesu
- 2) Naprogramování, případně úprava stávajícího kódu
- 3) Testování
- 4) Přenos do produkčního prostředí [16]

V praxi bývají nainstalovány tři téměř totožné systémy:

- 1) Vývojový – zpracován od programátorů. V případě že je plně funkční a zákazník je spokojen, je předán do dalšího systému.
- 2) Testovací - obsahuje kopii produktivních dat. Zde se testuje kód vytvořený ve vývojovém systému. Pokud je vše v pořádku, přechází do dalšího systému, pokud ne vrací se do předcházejícího systému.

- 3) Produkční systém – zde se nachází ostrá produkční data. Naprogramované změny se na tato data aplikují pouze po jejich důkladném otestování. Na tomto systému pracují běžní uživatelé.[16]

Pokud je systém správně implementován, přináší uživateli řadu výhod:

- 1) zefektivnění a zrychlení práce s daty
- 2) centralizaci dat a snížení chyb
- 3) zvýšení bezpečnosti dat
- 4) rychlejší výstupy pro vedení firmy [16]

### 3.3.1 Hosting

Standardně bývá informační systém implementován a umístěn přímo do systému firmy, organizace nebo společnosti. Informační systém však lze provozovat pomocí hostingu, kdy uživatelé přistupují do systému pomocí webového prohlížeče.

Webhosting je pouze samotné umístění stránek na serveru poskytovatele. Aby se uživatelé internetu ke stránkám dostali, je potřeba mít zaregistrovanou doménu (např. s názvem společnosti). [17]

Výhody:

- 1) snížení nákladů na hardware – pomůže organizacím které nedisponují dostatečným HD pro provoz IS.
- 2) o správu dat se stará dodavatel IS

Nevýhody:

- 1) firemní data jsou umístěna mimo firmu, což je vnímáno jako prvek snížení bezpečnosti dat
- 2) jeden server poskytovatele sdílí až tisíce zákazníků, čímž může nastat porucha či přetížení webu jednoho zákazníka a následné omezení až vyřazení z provozu ostatních zákazníků

- 3) nutností je připojení k internetu, bez kterého se k informačnímu systému nelze připojit

### 3.3.2 Open Source

Když využíváte klasický ERP software, nemáte přístup ke zdrojovým kódům vaší aplikace. Jste závislí na dodavateli a většinou si nemůžete upravovat systém sami. Naopak v případě open source ERP získáváte přístup ke zdrojovému kódu, který můžete prohlížet, sdílet a měnit. Pokud zdrojové kódy dále sdílíte, musíte ostatním poskytnout stejná práva, jako jste získali vy sami. [18]

Výhody:

- 1) možnost prohlížení zdrojového kódu
- 2) možnost sdílení zdrojového kódu
- 3) možnost modifikace zdrojového kódu

Nevýhody:

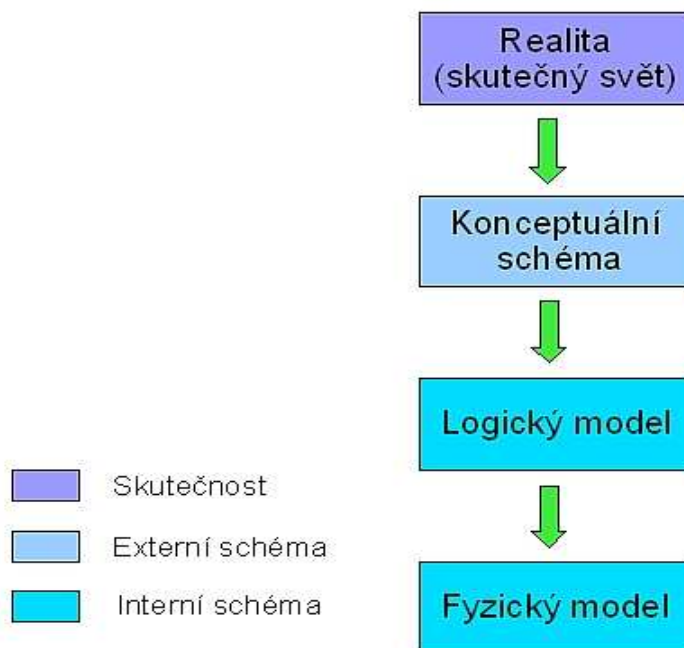
- 1) obavy o bezpečnost informačního systému
- 2) servisní problém

## 4 DATOVÉ MODELY

Datové modely jsou typy schémat při vytváření Informačního systému. Můžeme je rozdělit dle úrovně pohledu na samotná data.

### Různé úrovně pohledu na data:

- 1) Konceptuální – zabývá se modelováním reality, není (snaží se nebýt) ovlivněna budoucími prostředky řešení (E-R model, Class Diagram, ...).
- 2) Logická – vztahuje se ke konkrétnímu datovému modelu a používá jeho konstrukční dotazovací a manipulační prostředky (relační, objektová, síťová, hierarchická, XML, ...)
- 3) Fyzická - jde o fyzické uložení dat (sekvenční soubory, indexy, clustery,...). Programátor je od ní odstíněn logickou vrstvou SŘBD. [19]



Obrázek 6: Datové modely

## 4.1 Konceptuální model

Základní znalost jak interpretovat data je uložena v databázovém schématu. Konceptuální modely jsou pokusem umožnit vytvoření popisu dat v databázi nezávisle na fyzickém uložení databáze.

Obvykle se řeší:

- 1) struktura dat (datová analýza a návrh)
- 2) manipulace dat (funkční analýza a návrh)
- 3) specifikace IO (integritní omezení)

Čtyři základní principy konceptuálních modelů (popisují objekty a jejich vzájemné vztahy):

- 1) orientace na objekty - myslí se v objektech a nikoliv v identifikátorech, které je označují - vyžadována složitá struktura objektů
- 2) funkcionální podstata vztahů - do modelů se zahrnuje pojem atributu (E-R modely), dále funkce jako jediný základní konstrukt (DAPLEX, HIT), nebo systém funkcí (fragmenty v IFO)
- 3) ISA hierarchie - umožňuje práci s nadtypy a podtypy objektů
- 4) hierarchický mechanismus - pro konstrukci objektů z jiných objektů [19]

### 4.1.1 E-R model

- 1) identifikuje typy entit jako třídy objektů stejného typu (ZAKAZNIK, FILM, KNIHA)
- 2) identifikuje typy vztahů, do kterých entity identifikovaných typů mohou vstupovat ZAKAZNIK (entita) MA\_PUJCENOU (vztah) KNIHA (entita)
- 3) na základě přiměřené úrovně abstrakce přiřadí jednotlivým typům entit a vztahů atributy popisující blíže jejich vlastnosti - JMENO (jméno, příjmení)
- 4) formuluje různá integritní omezení vyjadřující soulad s větší či menší přesností schématu s modelovanou realitou.



- Entita = je objekt reálného světa, který je schopen nezávislé existence a je jednoznačně odlišitelný od ostatních objektů.
- Vztah - je vazba mezi dvěma entitami.
- Hodnota popisovaného typu - jednoduchý datový typ
- Atribut - funkce přiřazující entitám či vztahům hodnotu určující některou podstatnou vlastnost entity, nebo vztahu
- Klíč = minimální množina atributů, jejichž hodnoty jednoznačně určují každou entitu
- E-R diagram umožňuje popsat: entity (obdélníky), vztahy (kosočtverce), (ne)povinnost účasti ve vztahu (čárkovaná či plná čára), kardinalita (číslem), slabé entity (dvojitý obdélník - součástí jeho klíče jsou i atributy jiných entit), atributy (čárky z entit), klíčové atributy (jméno atributu podtrženo, příp. je na konci vyplněný kroužek), ISA hierarchie (vnoření entit, tedy obdélníků) [19]

## 4.2 Logický model

Logický model používá databáze směrem k uživateli a určuje, jak je konceptuální struktura dat implementována v konkrétním technicko-programovém prostředí.

Logické (databázové) modely:

- 1) hierarchický model
- 2) síťový model
- 3) relační model
- 4) objektově orientovaný model
- 5) objektově relační model
- 6) XML model

### 4.2.1 Hierarchický model

Základem hierarchického modelu je stromová struktura, kde výchozím prvkem je kořen (kořenový uzel, v každém systému existuje právě jeden) a na větvích jsou pak

umístěny uzly, respektive listy, pokud již tyto neobsahují žádnou další větev. Právě větve obsahují vlastní datové struktury.

Důležité je, že uzly mohou být propojeny nejen v klasickém hierarchickém vztahu rodič-syn, ale také v rámci jedné úrovně. Tato skutečnost se podílí i na usnadnění vyhledávání – není nutno prohledávat celý objem dat, ale pomocí "přesné" navigace po větvích a listech nelézt požadovanou informaci. I nadále ovšem v propojení zůstává omezení na jednosměrnost vazeb 1:N (jeden potomek má jen jednoho rodiče).

Jednou z hlavních výhod tohoto databázového modelu je vedle zefektivnění vyhledávání také odklonění se od fyzického modelu – autory aplikací vůbec nemusí zajímat, jak jsou data fyzicky uložena. Pro realizaci hierarchického modelu se obvykle využívá zřetězených seznamů.

Mezi hlavní omezení hierarchického modelu patří, v případě změny požadavků, nutnost přepracování celé struktury databáze, jinými slovy nestačí pouze ubrat či přidat jednu položku. Značným omezením je komplikovanost při realizaci vazeb N:M, která je sice řešitelná, ovšem pomocí redundantních přístupů a cyklických vztahů.

Tento databázový model je vhodný především pro aplikace, které zpracovávají data založená na hierarchické struktuře, ke kterým patří například organizační či skladové systémy (pochopitelně je lze realizovat výhodněji i pomocí novějších modelů).[1 ]

#### 4.2.2 Síťový model

Podstatou síťového modelu je použití ukazatelů vyjadřujících vztahy mezi jednotlivými databázovými položkami. Vzhledem k tomu, že tyto ukazatele mohou být lineární i cyklické a svým způsobem mohou vyjadřovat skutečné vztahy mezi objekty v databázi (tedy objekty reprezentujícími reálné objekty skutečného světa), lze poměrně snadno pomocí síťového modelu realizovat vztahy N:M, i když ve složitějších případech to znamená vysokou režii.

Vyhledávání konkrétního údaje není v rámci síťového modelu nijak složité, stejně jako v případě vyhledávání odvozených hodnot – začít lze prakticky kdekoli v rámci daného modelu. Na druhou stranu i síťový model trpí podobnou nevýhodou jako model hierarchický. Tímto handicapem jsou omezení ve změnách struktury databáze. V mnoha

případech musí autor databázové aplikace znovu vytvořit celou strukturu, i když oproti hierarchickému modelu je toto omezení méně restriktivní. [1]

Přestože se síťový model dočkal spousty vylepšení, netěší se velké oblibě pro nástup relačního modelu databází.

### 4.2.3 Relační model

Relační model je nejrozšířenějším způsobem ukládání dat do databází.

Ve své podstatě jde u relačního modelu o to, že stačí vzít množinu například osob, množinu rodných čísel a množinu dat narození a z nich vytvořit kartézský součin reprezentující možné vazby mezi jednotlivými množinami. Za relaci lze tedy považovat podmnožinu tohoto kartézského součinu odpovídajícího skutečnosti, případně kartézský součin celý (formálně je totiž relací libovolná podmnožina tohoto kartézského součinu).

Pod relací si lze představit také matici s  $m$  řádky a  $n$  sloupci, kde každému prvku relace odpovídá jeden řádek (záznam). Oproti tabulkám platí pro relace několik základních omezení, například skutečnost, že žádné dva řádky nemohou být v relaci shodné.

Relační model přináší řadu výhod, zejména mnohdy přirozenou reprezentaci zpracovávaných dat včetně snadné možnosti definovat a zpracovávat vztahy. Velkým přínosem relačního modelu je také fakt, že klade značný důraz na zachování integrity zpracovávaných dat a zavádí do použitelné praxe mnohé pojmy – referenční integritu, cizí klíče, primární klíče apod. Je dlužno poznamenat, že ne vždy platí představa o relačním modelu jako o nejrychlejším způsobu realizace databázového prostředí. Mylná představa panuje také o tom, že jazyk SQL je výhradně spojen s relačními databázemi – nemusí tomu tak být.

Objektové databázové modely lze rozdělit do dvou základních skupin – objektové a post-relační. První skupina představuje oblast od samého počátku chápanou jako čistě objektovou, druhá skupina rozšiřuje vlastnosti relačního modelu o objektové prvky. První pokusy s objektovými databázemi spadají již do přelomu 70. a 80. let minulého století. Spolu s tím, jak se objektové jazyky staly na konci tisíciletí de facto standardem pro vývoj drtivé většiny aplikací, musely na to zrychlenou evolucí reagovat i objektové databázové modely.

Základem objektové databáze je podpora perzistentního uložení objektů do datového prostoru. Objekt musí být k dispozici pro všechny přípustné činnosti i poté, co jeho "tvůrce" (obvykle databázová aplikace) již neexistuje. Každý persistentní objekt je přitom určen jednoznačným objektovým identifikátorem (zjednodušeně řečeno jde o jakýsi objektový primární klíč).

Důležitou vlastností je také schopnost objektové databázové platformy vyhodnocovat nastalé podmínky a na základě toho provádět určité akce, tedy "spouštět" odpovídající metody objektů. Stranou nezůstává ani podpora bezpečnostních omezení a transparentnost vůči fyzickému uložení dat.

Hlavní výhodou objektového přístupu je odstranění základní bariéry relačního světa při modelování složitého prostředí. Relace vedou mnohdy až k nepřipustnému zjednodušování, dokonce za využití pouze několika málo základních datových typů a vztahů. Nedochozí také ke zbytečnému oddělování datové podstaty objektů od podstaty funkční. Díky tomu je v konečném důsledku jednodušší i výsledná databázová aplikace. Jinými slovy, objektový přístup se hodí spíše pro zpracování menšího množství složitých dat, u relačního světa je tomu obvykle naopak. [2 ]

#### 4.2.4 Objektově orientovaný model

Tento model je založen na objektech, které kombinují daný kód a data.

Objektová databáze podporuje všechny vlastnosti nutné k práci s objekty:

- 1) plná podpora objektů
- 2) zapouzdření
- 3) dědičnost
- 4) polymorfismus
- 5) jednoznačná identifikace objektu
- 6) reference mezi objekty

Umožňují využitím hostitelského objektového jazyka přímo na objekty v databázi, proto není nutný žádný mezistupeň pro práci s daty.

Současné objektově orientované databázové modely podporují většinu z objektového přístupu a již v reálné praxi prokázaly, že pro mnohé aplikace jsou nemalým přínosem. Tento databázový model je výborný pro manipulaci s daty.

#### 4.2.5 Objektově relační model

Tento databázový model se snaží sjednotit rysy relačních a objektových databází. Jedná se o rozšíření relační koncepce objektovou orientací databázových systémů. Všechny trvalé informace jsou v tabulkách. Některé datové typy mohou mít bohatší datovou strukturu.

#### 4.2.6 XML model

XML databázový model se podobá hierarchickému databázovému modelu, protože XML dokument je chápán jako strom.

Výhod přináší XML ve spojení s databázovými platformami celou řadu, za hlavní přednost je považováno usnadnění tvorby otevřených aplikací využívajících například elektronickou výměnu dokumentů s rozdílnou strukturou. Podpora otevřenosti vychází z toho, že součástí každého dokumentu XML jsou nejen samotná data, ale také informace popisující jejich přesnou strukturu.

Současné databázové platformy podporují technologii XML především prostřednictvím mapování dokumentu XML na relační datové struktury, případně na struktury objektové. Výhodou zůstává obecnost takového řešení – vývojáři mohou využít hlavních předností XML. Nevýhodou zůstává ztráta výkonu – na převod mezi strukturami je nutná jistá časová reže. Navíc běžné mechanismy zvyšující výkon v relačních databázích nejsou příliš použitelné, například stromové indexové struktury jsou mnohem více efektivnější pro nalezení již konkrétní hodnoty, nikoli prohledávání XML dokumentu. I když by na první pohled mohl být pravdivý opak. [3]

### 4.3 Fyzický model

Fyzický datový model je realizací logického modelu v prostředí informačního systému. V zásadě se jedná o různé způsoby fyzické uložení dat na konkrétním médiu.

Z hlediska fyzické úrovně pohledu dělíme data na:

- 1) záznam - logická jednotka pro uložení dat
- 2) homogenní soubor - kolekce záznamů stejného typu
- 3) statická organizace souborů:
  - hromada = nehomogenní soubor záznamů obvykle proměnné délky, vkládané záznamy jsou umísťovány ve vymezeném prostoru disku fyzicky za sebou, hledání v  $O(N)$
  - sekvenční soubor = homogenní
    - neuspořádaný, řešený obvykle jako hromada
    - uspořádaný se liší tím , že je uspořádán dle klíče
    - aktualizované záznamy se umísťují do neuspořádaného sekvenčního souboru (soubor aktualizací)
    - po provedení operace REORGANIZACE se setřídí do nové verze primárního souboru
    - základní složitost  $O(N)$  lze binárním půlením zlepšit na  $O(\log N)$
  - index-sekvenční soubor = je tvořen primárním souborem (sekvenční setříděný soubor uspořádaný podle primárního klíče) s přídavnou víceúrovňovou strukturou index (obsahuje jen některá data z primárního souboru a odkaz na to, kde jsou přesně v primárním souboru - urychlí to vyhledávání)
  - indexovaný soubor = tvořen primárním souborem a indexy pro různé vyhledávací klíče. Neindexují se bloky, ale záznamy. Primární soubor tedy nemusí být setříděn, ani se nemusí nacházet v souvislé části vnější paměti. Data jsou přímo součástí indexového stromu.
  - bitová mapa = jde o velmi efektivní možnost indexace řádků tabulek relační databáze. Pro všechny hodnoty sloupce s nízkou kardinalitou sestavíme bitmapu, kde hodnota true přiřazuje řádek k hodnotě indexovaného sloupce. Použitelné jen pro atributy s dostatečně malou doménou.

- přímý přístup = pomocí hashovací funkce se rozptyluje do adresového prostoru, např. dělením modulo prvočíslo.
- 4) dynamická reprezentace souborů: B+ nebo B- stromy [19]

## 5 ŽIVOTNÍ CYKLUS INFORMAČNÍCH SYSTÉMŮ

Životní cyklus informačního systému je velmi podobný životnímu cyklu software. Jedná se o období, která jdou za sebou. Pro každé období je stanoven určitý cíl, k jehož dosažení jsou v daném období nasměrovány všechny činnosti.

Hlavní etapy životního cyklu můžeme rozdělit dle jejich obsáhlosti – jemnosti struktury rozdělení samotných etap.

### A, hrubé rozdělení:

- 1) plánování
- 2) návrh IS
- 3) zavádění IS
- 4) provoz a údržba IS

### B, jemné rozdělení:

- 1) analýza problémů
- 2) definice informačních potřeb
- 3) analýza systémových potřeb
- 4) návrh systému
- 5) vývoj a dokumentace software
- 6) testování systému
- 7) zavádění systému
- 8) údržba systému

### 5.1 Vodopádový model životního cyklu

Tento model životního cyklu patří k nejstarším. Je charakteristický ručním programováním (bez nástrojů CASE) a metodologií strukturovaného programování. Využívá databázových systémů. Aplikace jsou vyvíjené pro centrální počítač. Samotný systém je schvalován na úrovni dílčích výsledků i samotného systému jako celku. Uživatel se účastní pouze při definování požadavků na systém a při zavádění systému do provozu.



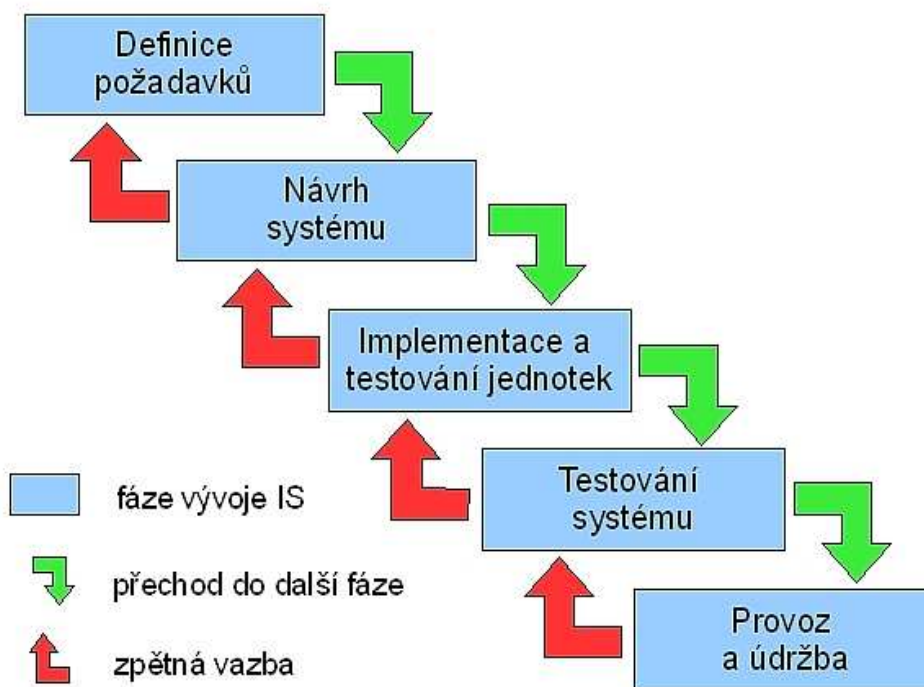
Celý proces tvorby začíná definicí požadavků, kdy zákaznická organizace, pokud možno co nejpřesněji, popíše veškeré své požadavky na systém. V této fázi nastává problém, kdy neodborná veřejnost jen stěží popisuje tvůrcům své očekávání. Mnohdy nastávají situace, kdy sami pořádně neví, co která část systému má vykonávat.

Po dokončení analýzy je vytvořen podrobný návrh systému jako celku. Následuje fáze zavádění a testování jednotlivých složek systému. Po úspěšném otestování jednotlivých složek systému je systém otestován jako funkční celek. Pokud je systém plně funkční, přechází do poslední fáze projektu, a to zavedení do provozu.

Model je složen z jednoznačně vymezených etapových posloupností, které na sebe navazují.

Největší nedostatek Vodopádového životního cyklu spočívá v dlouhé době jeho realizace, kdy až v průběhu tvorby informačního systému docházelo ke změnám původního zadání. Další nedostatek tkví v tom, že provozuschopnou verzi systému vidí zákazník až v závěrečné fázi řešení a případné nedostatky jsou odhaleny velmi pozdě. Reálné projekty zřídka sledují jednotlivé etapy v předepsaném pořadí.

Model sice je nedokonalý, ale je jednoznačně lepší než neřízený nebo dokonce náhodný přístup k řešení projektu.



Obrázek 7: Vodopádový model

## 5.2 Prototypový model životního cyklu

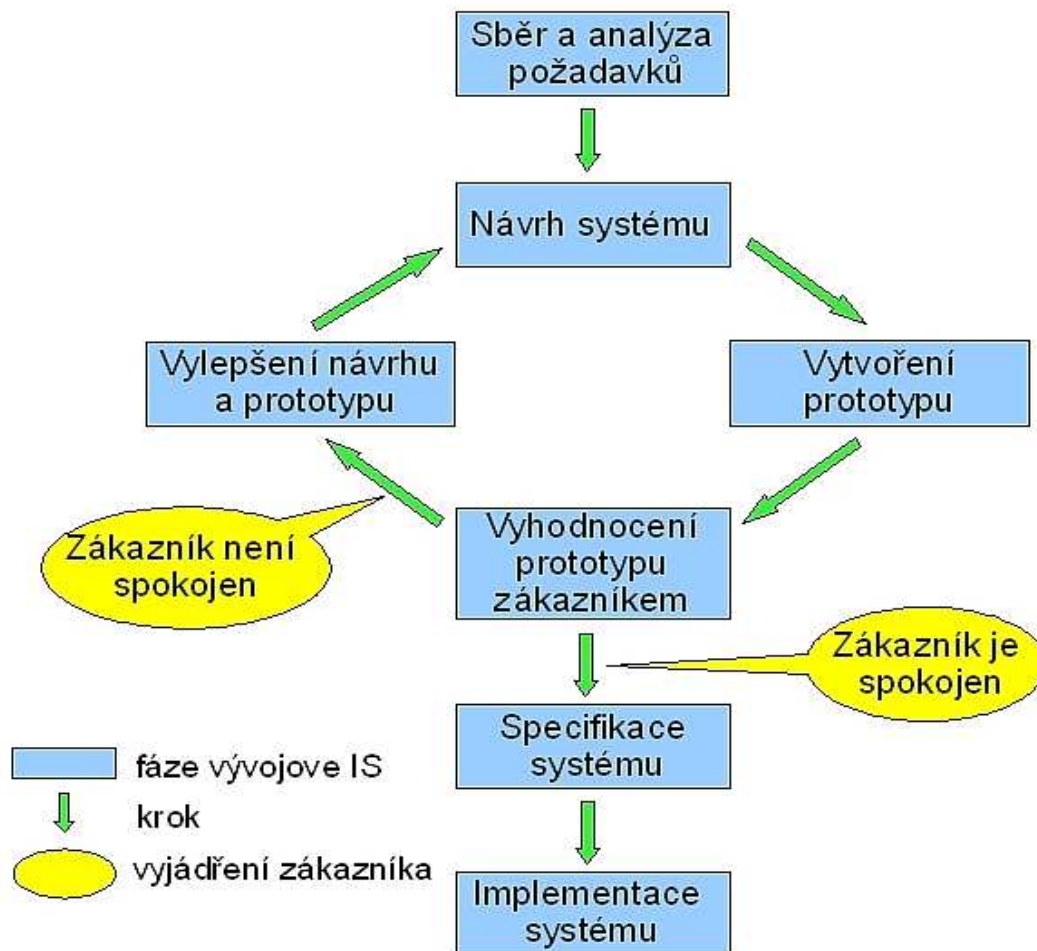
Prototyp je částečnou implementací produktu, ale může být také implementována pouze část produktu v logické nebo fyzické formě, která reprezentuje všechna vnější rozhraní.

V případě existence několika nezávislých vývojových skupin je prototypování hodně náročné na vedení.

Celý proces tvorby začíná analýzou požadavků zákaznické organizace. Následuje navržení systému a vytvoření prototypu. Když je vytvořen prototyp, dochází ke konzultaci se zákazníkem. Ten tvůrcům sdělí své stanovisko ohledně spokojenosti s navrženým systémem. Pokud zákazník není spokojen, postoupí prototyp vylepšení návrhu. Znovu je vytvořen prototyp dle nového vylepšeného návrhu a postoupen další konzultaci se zákazníkem. Tento okruh vylepšování absolvuje do té doby, než je zákazník spokojený. V konečné fázi dojde ke specifikaci systému a jeho následné implementaci.

Výhodou tohoto modelu je, že budoucí uživatelé testují a ověřují prototyp, kdy dochází k upřesnění požadavku na vytvářený systém. Zákazníci si s řešitelským týmem průběžně sledují a vyjasňují požadavky a jejich interpretaci.

Nevýhodu spatřujeme v tom, že neschválené prototypy končí nepoužitě.



Obrázek 8: Prototypování

### 5.3 Spirálový model životního cyklu

Tento model vznikl v roce 1988. Je založen na kombinaci prototypování a analýzy rizik. Při vývoji informačního systému tímto modelem dochází k stálému opakování čtyř základních kroků. Každé opakování je na vyšší – kvalitnější úrovni.

Kvalitnější úrovni rozumíme:

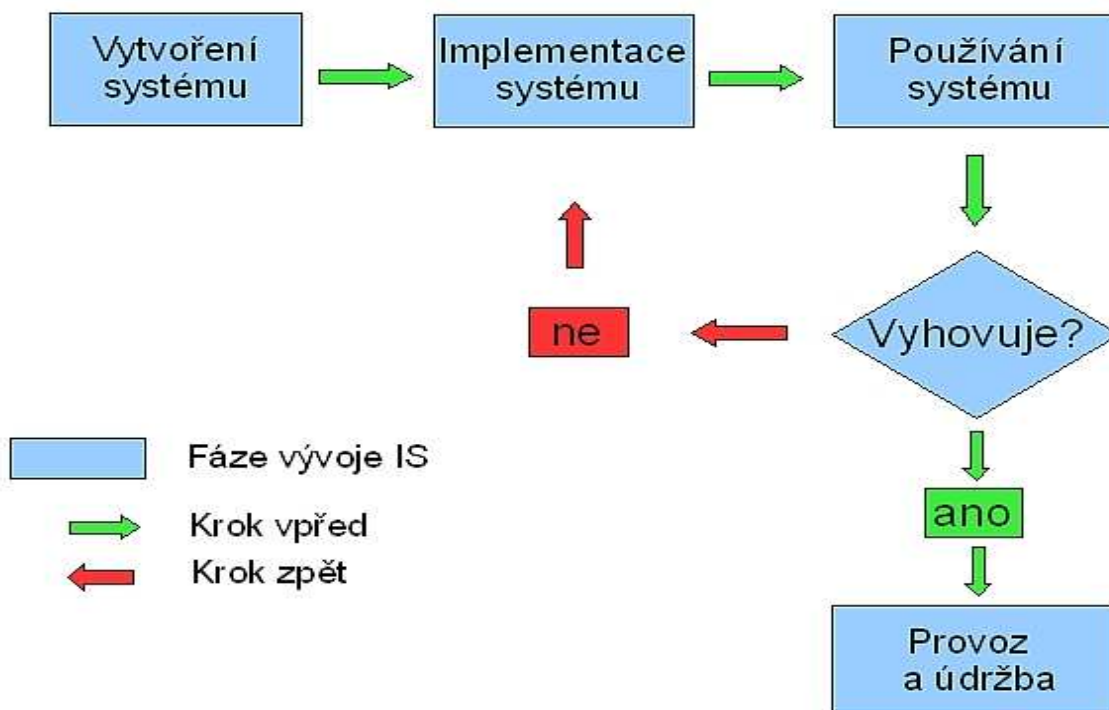
- 1) lepší určení předmětu řešení, jejich alternativ a omezení
- 2) vyhodnocení alternativ

- 3) vývoj produktu pro danou úroveň
- 4) plánování příští fáze

Samotné fungování systému spočívá v zadání požadavku od zákaznické firmy na vytvoření systému. Poté přichází na řadu jeho implementace. Systém je používán. Dochází ke konzultaci se zákazníkem. Pokud nevyhovuje, vrací se zpět k přepracování, avšak o úroveň znalostí výš. Tak se celý cyklus opakuje až do té doby, než se systém dostane na úroveň zákaznickovy spokojenosti. Pak dojde k předání systému zákaznické organizaci k užívání.

Výhodou je, že již v časných fázích modelu je pozornost věnována použití SW. Chyby a nevyhovující postupy jsou odhaleny relativně brzy a k vylepšování jsou použity jen vhodné postupy.

Nevýhodou přináší hrubé členění modelu, čímž je vhodný pro zkušené pracovníky. Pro nezkušené pracovníky je nutné model rozčlenit na podrobnější kroky a určit kontrolované výstupy.



Obrázek 9: Spirálový model

## 6 METODIKA TVORBY INFORMAČNÍCH SYSTÉMŮ

Metodika je souhrnem pravidel a doporučení, která jsou využívána v procesu tvorby informačního systému. Říká nám především kdo, kdy, co a proč má dělat během vývoje a provozu informačního systému. Cílem je vývoj a provoz IS, který optimálně využívá potenciálu dostupných informačních technologií k maximální podpoře podnikových cílů.

Příklady metodik :

- 1) CASE
- 2) OMT (Object Modelling Techniques)
- 3) SDM (Systém Development Metodology)

### 6.1 CASE (Computer Aided Software Engineering)

Jedná se o nástroj na podporu práce analytiků a programátorů při vývoji informačních systémů. Použití tohoto nástroje je jednou z cest, jak překonat problémy s výstavbou a provozem IS. Tvorba diagramů v tomto prostředí se jeví jako jednoduchá. To ale není pravda, protože vyžaduje vysokou znalost a profesionálnost tvůrce modelu. Největší předností CASE nástrojů je zajištění souvislostí, jež je člověk jen velmi těžce schopen pojmout. Nicméně ani použití tohoto prostředí nám nezaručí bezchybný a rychlý vývoj systému.

Obsah: databáze navrhovaných prvků IS

Funkce: podpora realizačního projektu IS

Základ: metodika = návod na vytváření modelů a určení závislosti mezi nimi.

Funkce CASE:

- 1) automatizovaná evidence vytvořených objektů a specifikací, dokumentace vývoje systému
- 2) grafická podpora modelování
- 3) kontrola správnosti modelů podle zvolené metodiky, zajištění integrity a konzistence návrhu (předem definovaná integritní omezení a jejich kontrola,

automatické uplatnění změn vytvořených v jedné části ve všech souvisejících částech návrhu)

- 4) podpora týmové práce (identifikace osob a týmů zodpovědných za jednotlivé modely, tvorba více modelů současně, současná práce více osob na jednom modelu)
- 5) správa verzí
- 6) automatický převod definovaných modelů do konkrétního logického a fyzického návrhu (generování programu, popisu databáze, příp. prototypu)
- 7) reverse engineering – zpětné generování konceptuálního modelu z existující aplikace [12]

## 7 POUŽITÉ WEBOVÉ TECHNOLOGIE

Pro vytvoření informačního systému je nutné použít určité vzájemně provázané technologie. Samotný výběr byl proveden s ohledem na dostupnost, oblibu a rozšířenost.

Použité webové technologie:

- 1) HTML
- 2) CSS
- 3) JavaScript
- 4) PHP
- 5) MySQL

Jednotlivé položky u tohoto bodu nebudu rozepisovat do podrobností, protože se jedná o velmi populární technologie, o kterých byla napsána spousta publikací. Zaměřím se pokud možno na výhody a nevýhody použitých technologií.

### 7.1 HTML

Jedná se o zkratku Hyper Text Markup Language a je jedním z nejpoužívanějších jazyků pro vytváření stránek v systému World Wide Web, umožňující publikaci stránek na internetu.

Jazyk HTML je charakterizován množinou značek a jejich atributů. Tyto značky jsou uzavřeny do ostrých závorek. Vzniklé uskupení se nazývá „tag“. Mezi tyto tagy se uzavírají části kódu, čímž se určuje význam „sémantika“ obsaženého textu. Takto uzavřená část dokumentu tvoří „prvek“ dokumentu.

### 7.2 CSS

Zkratku Cascading Style Sheets, což v češtině znamená tabulky kaskádových stylů. Jedná se o jazyk pro popis způsobu zobrazení stránek napsaných v jazycích XML, HTML, XHTML. Hlavním cílem kaskádových stylů je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury.

Výhody:

- 1) rozsáhlejší možnosti – rozšiřuje možnosti formátování HTML
- 2) konzistentní styl – všechny stránky webové aplikace by měly mít nadpisy stejné úrovně, seznamy a zdůrazněné texty, což je pomocí samotného HTML obtížné.
- 3) oddělení struktury stylu
- 4) dynamická práce se styly – provedení změny stylu pomocí HTML znamená vyhledat a změnit všechny adekvátní znaky, což u CSS znamená pouze přepsání jednoho souboru
- 5) větší kompatibilita alternativních webových prohlížečů
- 6) kratší doba načítání – veškerý design a formátování stránky se načítá z jednoho souboru pro celý web
- 7) podpora zobrazení pro různá zařízení – při existenci různých stylů pro různá zařízení( Webdesigner, PDA, atd)
- 8) možnost úpravy formátování dle prohlížečů – dojde k vytvoření více souborů se styly .css

Nevýhody:

- 1) špatná podpora prohlížečů, kdy různé prohlížeče interpretují stejný kód CSS jinak. Tím vzniká problém napsání kódu tak, aby se výsledek zobrazil vždy stejně.

### 7.3 JavaScript

Jedná se o interpretovaný multiplatformní programovací jazyk se základními objektově orientovanými vlastnostmi. Program v JavaScriptu se obvykle spouští po stažení WWW stránky na klientské straně. Na rozdíl od jiných interpretovaných programovacích jazyků jako jsou PHP atd, které se spouští na straně serveru. Pro účely webového programování byl JavaScript rozšířen přidáním objektu reprezentující okno webového prohlížeče a jeho obsah. Klientská verze JavaScriptu přidává možnost vložit do webových stránek proveditelný obsah, čímž vznikne fakt, že webová stránka již může obsahovat



dynamické programy komunikující s uživatelem, které ovládají prohlížeč a dynamicky vytvářejí obsah HTML.

JavaScript svými programovými konstrukcemi připomíná C,C++ a Javu. Slovo Java je součástí jeho názvu jen s marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript je programovací jazyk bez typové kontroly, to znamená, že proměnné nemusí být specifikované typy. [5]

## 7.4 PHP

PHP (Hypertext Preprocesor) je skriptovací jazyk, který je instalován na straně serveru. Tím se liší od jazyků pracujících na straně uživatele, jakým je např. JavaScript. Tento jazyk lze zapouzdřit do dokumentu HTML, kde umožňuje generování požadovaného obsahu. Webové sídlo se tak může převést na webovou aplikaci, která není jen sadou statických stránek, jejichž informace nemohou být jednoduše aktualizovány [6]

Výhody:

- 1) jednoduchost
- 2) přirozená práce s databázemi
- 3) nezávislost na platformě
- 4) cena
- 5) upotřebitelnost

### 7.4.1 PHP 5

Verze 5 zdokonaluje existující funkcionalitu a přidává také několik nových schopností, které běžně provázejí architektury vyzrálého programovacího jazyka [7]

Bylo zdokonaleno:

- 1) výbava objektově orientovaného programování
- 2) zpracování výjimek ve stylu try/catch
- 3) podpora databázového serveru SQLite
- 4) práce s řetězci

- 5) podpora XML a webových služeb

## 7.5 MySQL

MySQL je multiplatformní. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru. MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet. [8]

Podporované vlastnosti:

- 1) cizí klíče
- 2) transakce
- 3) poddotazy
- 4) pohledy
- 5) uložené procedury
- 6) práce s metadaty
- 7) trigger
- 8) podpora různých znakových sad a časových pásem

### 7.5.1 phpMyAdmin

phpMyAdmin je nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní. V současné době umožňuje vytvářet/rušit databáze, vytvářet/upravovat/rušit tabulky, provádět SQL příkazy a spravovat

klíče. Jedná se o jeden z nejpopulárnějších nástrojů pro správu databáze. Je k dispozici v 52 jazycích. [9]

## 8 EFEKTIVNOST ZAVÁDĚNÍ IS

Efektivností zavádění informačního systému rozumíme účinnost prostředků, které byly vloženy do činnosti zavedení IS z hlediska užitečného výsledku této činnosti. Je zcela evidentní, že výdaje, které zákazník vynaložil na zavedení požadovaného produktu jsou viditelné, zatím co přínosy jsou neviditelné. Nelze prokázat konzistentní vztah mezi výdaji a příjmy se zavedení informačního systému.

### Náklady:

#### 1) Jednorázové

- Náklady na vývoj IS (dočasné vybavení pracovišť, konzultace, cestovné..)
- Náklady na investice (HW, SW, realizace sítě...)
- Náklady na vlastní implementaci IS (konverze dat, školení pracovníků...).

#### 2) Pravidelné roční

- Stálé (odpisy, mzdy, energie, údržba SW, HW..)
- Variabilní ( cestovné, penále, doprava, kurzy...).

### Přínosy:

#### 1) Kvantifikovatelné

- Jednorázové (prodej projektu IS)
- Roční (zvýšení výroby, zvýšení odbytu, snížení nákladů, úspora pracovníků, nižší skladové zásoby..).

#### 2) Nekvantifikovatelné

- Zjednodušení a urychlení přístupu k informacím
- Kvalitnější evidence
- Přesnější zpracování informací
- Zlepšení jména firmy, efektivnější práce, kvalitnější vztah se zákazníkem...[11]

## 9 ZABEZPEČENÍ PODNIKOVÉHO IS

Hlavní důvod pro realizaci zabezpečení podnikového systému je ochrana informačních aktiv firmy. Hodnota informací, které zpracovává podnikový informační systém je vysoká, mnohdy finančně těžce vyčíslitelná až nenahraditelná v samotných důsledcích, jenž by mohlo zneužití dat přinést. Také výpadky ve fungování, které může podnikový informační systém mít, mají závažné dopady na chod a prosperitu celé organizace.

Samotné informace jsou uloženy v souborech a databázích, což umožňuje snadné kopírování, tedy neoprávněné čerpání dat podnikového informačního systému.

Dokonalé zvládnutí informační bezpečnosti je mnohdy limitním požadavkem zadavatele na tvůrce IS.

### 9.1 Bezpečnostní hrozby a rizika

Samotný informační systém obsahuje zranitelná místa. Existence zranitelných míst způsobuje, že některé vlivy prostředí, ve kterém se IS provozuje, představují hrozbu. Hrozbou označujeme jakoukoliv okolnost působící na zranitelné místo, kdy může způsobit potencionální škodu.

Hrozby dle hledisek:

#### A, Objektivní:

- 1) přírodní (požár, povodeň, výpadek proudu)
- 2) fyzikální (elektromagnetické záření)
- 3) technické (porucha, nebo zničení paměti)

#### B, Subjektivní (lidský faktor):

- 1) neúmyslné (nezaškolený uživatel)
- 2) úmyslné vnitřní (útok na data v IS z vnitřku)
- 3) úmyslné vnější (útok na data v IS z vnějšku)

## 9.2 Zajištění bezpečnosti dat

Obecné zajištění bezpečnosti dat lze provést s ohledem na výše uvedená hlediska následujícím způsobem:

**A,Objektivní příčiny ztráty dat** lze odstranit zvolením vhodné zálohovací metodiky. Jako velmi účinné se ukázalo pravidelné zálohování dat na dvou fyzicky nezávislých místech.

**B,Subjektivní příčiny ztráty dat** lze ošetřit aplikací základních atributů zabezpečení přístupu.

- 1) identifikace – každý uživatel je jednoznačně identifikován
- 2) autentizace – uživatel prokáže svoji totožnost (heslem, otiskem prstu,..)
- 3) autorizace – každý uživatel je oprávněn k úkonům odpovídajícím roli, kterou zastává. [13]

## 9.3 Zajištění bezpečnosti služeb

Dostupnost služeb – nutnost zajistit ochranu proti výpadkům elektrické energie, výpadkům sítě a také duplikací software a hardware pro případ poruchy. Dále ochrana proti útokům (firewally, atd)

Důvěrnost služeb – shodné se zajištěním bezpečnosti dat z hlediska subjektivních příčin s nutností zabezpečení bezpečnosti přenosu dat mezi zdrojem a cílem.

## 9.4 Zajištění ochrany technických programových prostředků

Ochrana fyzických prostředků:

- 1) umístěním do míst, kde nevznikají rizika prostředí
- 2) ochrana telekomunikačních prostředků (nastavení switchů, routerů, ...)
- 3) ochranu proti poškození a přetížení.
- 4) Ochrana programových prostředků:
- 5) ochrana proti vkládání chybných dat (SQL injection,...)

- 6) ochrana proti parazitním kódům
- 7) ochrana proti podvržení identity
- 8) ochrana proti nežádoucímu smazání či modifikaci

## 9.5 Zajištění ochrany pře zneužitím či odcizením dat

### Riziko vnitřní:

- 1) pracovníci si nesmí sdělovat přihlašovací heslo ani si je nikam psát
- 2) heslo nesmí být snadno odhadnutelné (jméno, rodné číslo,...)
- 3) nenavštěvovat nebezpečné stránky
- 4) nestahovat samo spustitelné aplikace
- 5) neotvírat nebezpečné přílohy

Ochrana: - je nutné správně nastavit Firewall. Dále nastavit přístupová práva a zavést antivirovou kontrolu.

### Riziko vnější:

- 1) prolomení hesla hrubou silou (uhodnutí hesla softwarem, který během krátké doby vyzkouší velké množství kombinací znaků)
- 2) prolomení hesla sociálním útokem (heslo je zjištěno neoprávněnou osobou z důvodu nedbalosti pracovníka)
- 3) hardwarový útok (využívá nedokonalosti hardware – odposlouchávání paketů, dále také odcizení harddisku, nebo celého počítače)

Ochrana: - nastavení omezeného počtu vložení hesla, volba hesla s vysokým ochranným faktorem (kombinace znakových sad), zamezení přístupu k místům vhodným pro umístění odposlechů.

## **II. PRAKTICKÁ ČÁST**



## 10 ANALÝZA

### 10.1 Zadání

Požadavkem zadavatele je vytvoření sdruženého informačního portálu pro servisní a obchodní oddělení, pro rychlou vzájemnou výměnu důležitých informací. Těžiště využití směřuje na servisní oddělení s tím, že využívá podstatné informace prezentované obchodním oddělením. Vytvořený informační portál musí být dostupný z webového rozhraní.

### 10.2 Blížší specifikace

Požadovaný IS musí sloužit jako portál pro výměnu informací. Informace budou vkládat a čerpat jak zaměstnanci obchodního oddělení, tak i zaměstnanci servisního oddělení. Pro přístup k aplikaci se využívá webové rozhraní z důvodu možnosti vzdáleného přístupu k aktuálním datům servisních i obchodních pracovníků. Přístupem k aplikaci přes webové rozhraní, samozřejmě za předpokladu možnosti připojení k internetu, docílíme schopnosti získání potřebných informací přímo na místě servisního zásahu.

Samotná struktura portálu bude realizována pro 3 oblasti.

#### 10.2.1 Obchodní oddělení

Obchodní oddělení bude do systému vkládat především:

- 1) smlouvy na zařízení
  - dodavatel
  - zákazník
- 2) smlouvy na servis a technickou podporu
- 3) plán instalací nových zařízení
- 4) plán dalších akcí s předpokládanou účastí servisu
- 5) seznam neuhrazených faktur

Z dat vložených obchodním oddělením můžou čerpat potřebné informace jak samotné obchodní oddělení - mají přehled při tvorbě plánu servisních prací a instalací nových zařízení, tak i servisní oddělení - přehled nad plánovanými úkony tak potřebné smlouvy.

### 10.2.2 Servisní oddělení

Servisní oddělení bude do systému vkládat:

- 1) servisní zprávy
- 2) upravovat popis umístění speciálních a kalibračních přípravků.

Z dat vložených servisním oddělením budou čerpat informace jak obchodní, tak servisní oddělení - získají možnost náhledu na splněné a nesplněné úkoly, které mají následný vliv na plánování dalších úkolů a hlavně fakturaci.

### 10.2.3 Seznam zákazníků a zařízení

Tato část portálu bude představovat seznamy zákazníků a zařízení. Tyto seznamy můžou být různě strukturované od možnosti jednoho dokumentu až po členění pomocí podkategorií.

## 10.3 Důležité faktory

Během provádění analýzy požadavků zadavatele docházelo k postupnému upřesnění funkčních a obsahových detailů navrhovaného systému. Tím vyšly na povrch důležité faktory a omezení.

Jednalo se především o tato omezení:

- 1) forma požadovaného systému
- 2) velikost požadovaného systému
- 3) správa dat

### 10.3.1 Forma požadovaného systému

Prioritním faktorem, který značně omezuje navrhovaný systém, je jeho forma a tedy funkčnost. Celý systém má být realizován jako „úřední deska“, sloužící pro jednoduchou výměnu informací.

### 10.3.2 Velikost požadovaného systému

Dalším požadavkem je co nejmenší velikost systému. Celý systém má za úkol zabírat co nejmenší paměťovou kapacitu. Tímto jsou samozřejmě limitovány dokumenty, které budou v systému uchovávány z pohledu velikosti, kdy např. ze servisních listů budou vybrány jen nejdůležitější položky, zobrazující se jen zájemci o tyto informace. Zbývající informace budou cíleně opomenuty.

### 10.3.3 Správa dat

Jedním z nejhlavnějších požadavků zadavatele byl fakt, že všechny informace budou zobrazovány v tabulkách, jež budou obsahovat nejdůležitější informace o požadovaných nebo provedených úkonech a veškeré smlouvy budou uloženy formou dokumentů ve Wordu. Seznam zákazníků a zařízení měl být taktéž organizován formou tabulky.

## 10.4 Analýza procesů ve firmě pomocí objektově orientované metody

### 10.4.1 Vytvoření funkčního modelu

Hlavním účelem funkčního modelu je identifikace architektury (kostry) podnikového procesu, jeho zákazníků a produktů.

Byly vytvořeny dva modely:

- 1) model prodeje stroje
- 2) model servisního zásahu

Rozborem funkčního modelu zjistíme, zda zákazník, požadující stroj nebo servisní zásah na vlastním stroji, je s procesem uspokojen nebo neuspokojen. Tedy produktem procesu je buď předaný stroj nebo zákazník odchází jako neuspokojen. Uspokojený zákazník odejde pokud splní firma všechny jeho požadavky a odejde neuspokojen v případě, kdy firma není schopna zareagovat na jeho požadavky (cena, termín, ...).

Další zjištění, které můžeme z modelů vypožorovat je, že vlastní proces obsahuje dva podprocesy, jenž spolu spolupracují.

### 10.4.1.1 Funkční model prodeje stroje

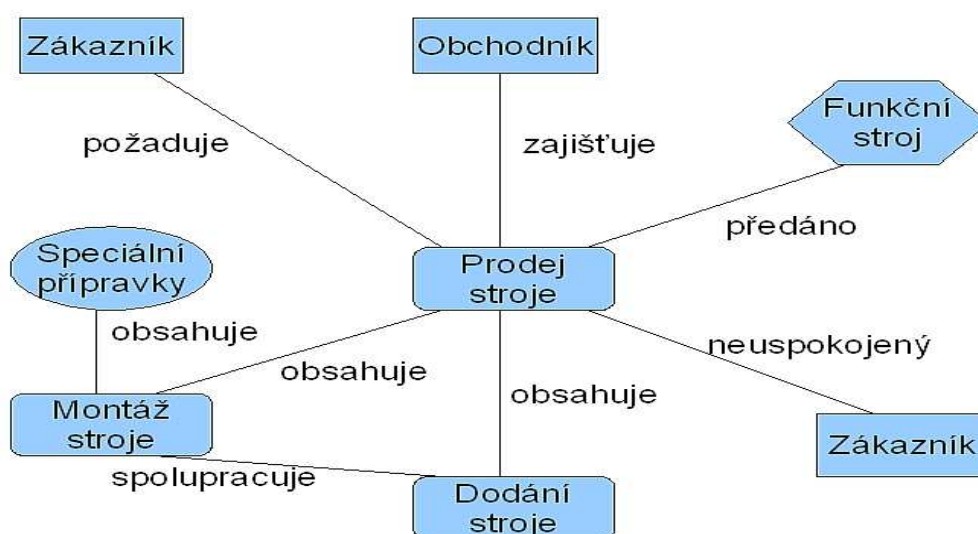
Rozborem funkčního modelu prodeje stroje zjistíme, zda zákazník požadující stroj, je procesem uspokojen nebo neuspokojen. Tedy produktem procesu je buď předaný stroj, nebo zákazník odchází jako neuspokojen. Uspokojený zákazník odejde, pokud splní firma všechny jeho požadavky a odejde neuspokojen v případě, kdy firma není schopna zareagovat na jeho požadavky (cena, termín, ...).

Další zjištění které můžeme z modelů vypožorovat je, že vlastní proces obsahuje dva podprocesy, které spolu spolupracují a jsou na sobě závislé.

Jedná se o:

- 1) dodání stroje
- 2) montáž stroje

Pokud není požadovaný stroj zajištěn, nelze provést montáž a na druhou stranu, pokud montáž není naplánována, nelze ji provést. Dále můžeme sledovat ještě jednu velmi důležitou závislost, a to, že k montáži stroje mohou být požadovány speciální přípravky, bez kterých montáž nelze povést. Tato fakta kladou velký důraz na plánování montáží s ohledem jednak na požadovaný termín dodání od zákazníka, dále na včasné dodání požadovaného stroje od výrobce, a také na volný termín montážní skupiny s ohledem na momentální dostupnost speciálních montážních přípravků.



Obrázek 10: Funkční model prodeje stroje

#### *10.4.1.2 Funkční model servisního zásahu*

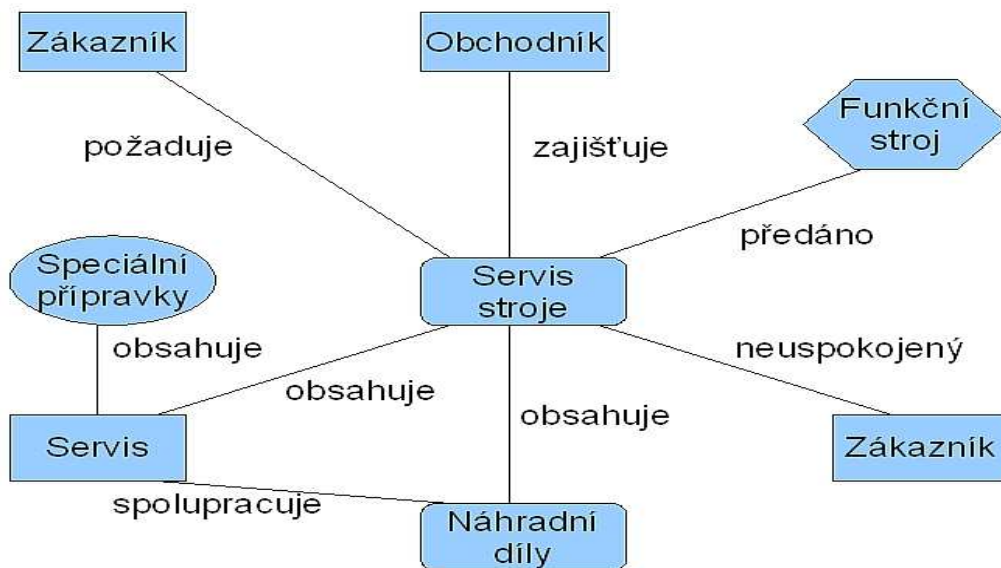
Rozborem funkčního modelu servisního zásahu zjistíme, zda zákazník požadující servisní zásah, je procesem uspokojen nebo neuspokojen. Tedy produktem procesu je buď předaný stroj, nebo zákazník odchází jako neuspokojen. Uspokojený zákazník odejde, pokud splní firma všechny jeho požadavky a odejde neuspokojen v případě, kdy firma není schopna zareagovat na jeho servisní požadavky (cena, termín, atd. ...).

Další zjištění, které můžeme z modelů vypožorovat, nám ukazuje, že vlastní proces obsahuje dva podprocesy, které spolu spolupracují a jsou na sobě závislé.

Jedná se o:

- 1) náhradní díly
- 2) servis

V případě nutnosti dodání náhradních dílů musí být požadovaný díl buď na skladě nebo objednan u dodavatele. Servisní zákrok musí být naplánován. Dále můžeme sledovat ještě jednu velmi důležitou závislost a to, že k servisu stroje mohou být požadovány speciální přípravky, bez nichž servisní zákrok nelze povést. Tato fakta kladou velký důraz na plánování servisních zákroků s ohledem jednak na požadovaný termín dodání od zákazníka, dále na dostupnost náhradních dílů servisovaného stroje a také na volný termín servisní skupiny s ohledem na momentální dostupnost speciálních montážních přípravků.



Obrázek 11: Funkční model servisního zásahu

#### 10.4.2 Vytvoření objektového modelu

Objektový model identifikuje statickou strukturu všech objektů (aktivních i pasivních) nutných pro realizaci procesu.

Byly vytvořeny dva modely:

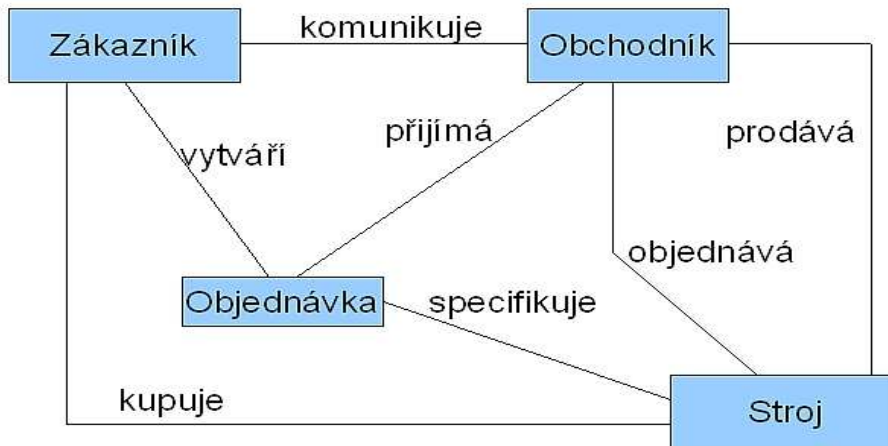
- 1) model prodeje stroje
- 2) model servisního zásahu

Z pohledu na oba modely je zřejmé, že zákazník musí komunikovat s obchodníkem a opačně. Výsledkem jejich společného úsilí může být objednávka obsahující všechny potřebné informace k úspěšnému ukončení celé transakce. Samozřejmě může nastat situace, kdy i přes usilovná jednání není objednávka vytvořena, ale pak nedojde na následné činnosti spojené s prodejem nebo servisem stroje.

##### 10.4.2.1 Objektový model prodeje stroje

Zákazník komunikuje s obchodníkem za účelem koupě stroje. Vytvoří objednávku, která specifikuje stroj a předá obchodníkovi. Na základě objednávky dojde k podepsání

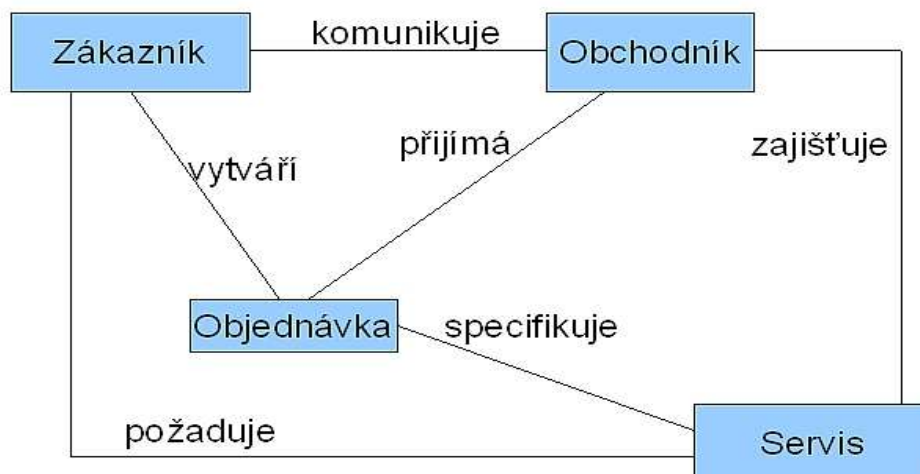
patříčných smluv. Obchodník objedná stroj u dodavatele, který je předmětem kupní smlouvy. Samotný stroj pak může, ale také nemusí být předmětem montáže stroje.



Obrázek 12: Objektový model prodeje stroje

#### 10.4.2.2 Objektový model servisního zásahu stroje

Zákazník komunikuje se zákazníkem za účelem servisu stroje. Vytvoří objednávku, specifikující servisní zásah a předá obchodníkovi. Obchodník přijme objednávku a na její základě vystaví potřebné smlouvy, které nechá podepsat zákazníkovi. Po podpisu smluv předá informace servisu zajišťující servisní zásah.



Obrázek 13: Objektový model servisního zásahu

### 10.4.3 Vytvoření dynamického modelu

Tvorba dynamického modelu slouží k přípravě klasických scénářů pro typické průchody aplikací. Dále identifikuje jednotlivé události od objektů a slouží k přípravě toku událostí pro systém.

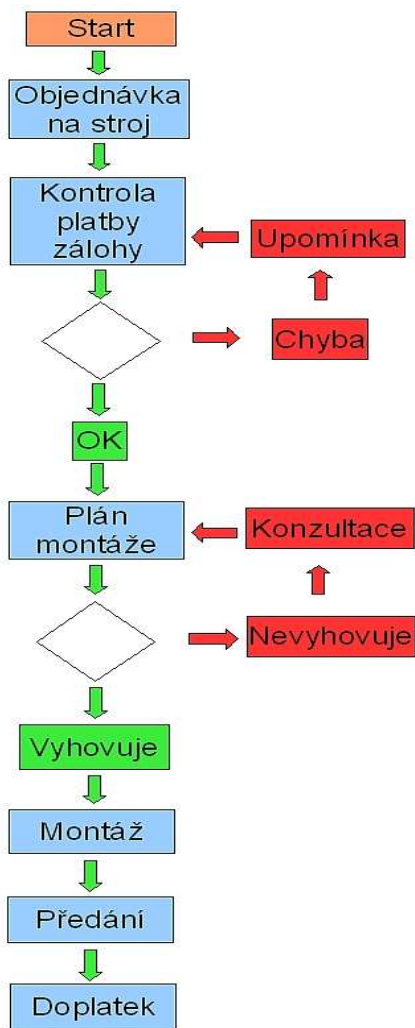
#### 10.4.3.1 Dynamický model prodeje stroje

Prodej stroje začíná přijetím objednávky a podpisem smlouvy na dodání a montáž stroje, kdy zákazník souhlasí s cenovou nabídkou – označeno jako políčko „START“. Následuje kontrola platby zálohy. V případě, že záloha nebyla uhrazena, je zákazník na tuto skutečnost upomenut. V případě, že byla záloha zaplácena, je montáž stroje zařazena do plánu montáží. Výše zálohy je předmětem smluvního ujednání při předání objednávky a může být i 100% ceny stroje i s montáží. Termín montáže je konzultován se zákazníkem i přes skutečnost, že termín byl obsahem objednávky. Konzultace plánu montáže je nutná, protože v době přijímání objednávky, která je vystavena v nutném předstihu, mnohdy probíhají u zákazníka práce na stavebních úpravách nebo instalacích souvisejících částí celého systému (přívod energetického zdroje, přívod vzduchu, atd.). Z vlastních zkušeností vím, že obvykle bývají přípravné práce před montáží ve skluzu.

Dalším nutným aspektem zůstává fakt, že k samotné montáži mohou být zapotřebí určité přípravky nebo speciální vybavení montážní skupiny, jež není duplicitní a může nastat situace, kdy tento přípravek je v den plánované montáže použit na jiné montáži nebo servisním zásahu.

Nesmíme opomenout ani předpoklad, že montáže provádí servisní skupina, která může mít nečekaný servisní zásah s větší prioritou než je montáž nového zařízení. Pokud termín stanovený plánem montáže vyhovuje jak zákazníkovi, tak montážní skupině dochází k montáži a následnému předání. Doplatek je uplatňován v případě, že nebyla celá cena uhrazena jako záloha (záloha mohla činit 100% smluvené ceny – platba před montáží).



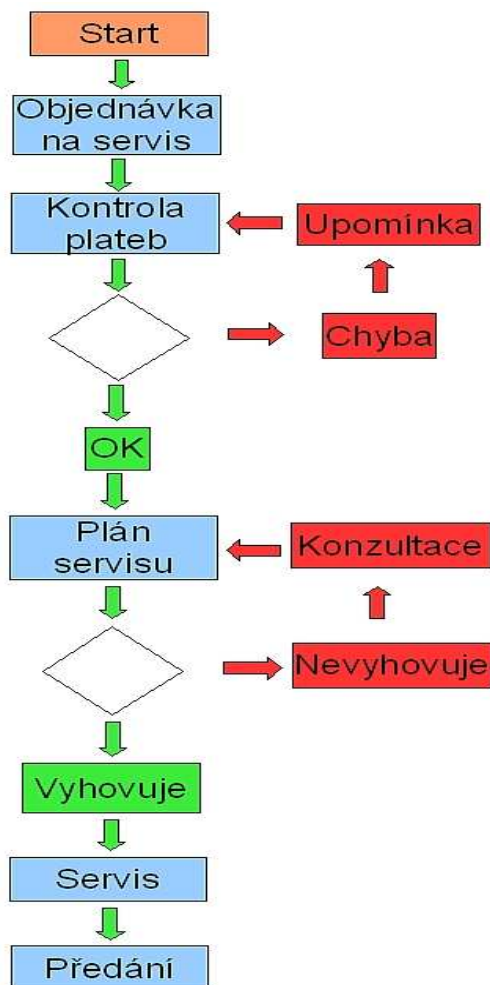


Obrázek 14: Dynamický model  
prodeje stroje

#### 10.4.3.2 Dynamický model servisního zásahu

Proces servisního zásahu začíná přijetím objednávky a podpisem smlouvy na servis – označeno jako políčko „START“. Následuje kontrola plateb za předešlá období. Pokud je zjištěn nedoplatek, odchází upomínka zákazníkovi na zjištěný fakt. Pokud jsou platby v pořádku, je servisní zákrok zařazen do plánu servisu. Termín servisního zásahu je konzultován se zákazníkem, protože se může stát, že navrhovaný termín nevyhovuje. Důvodů může být několik. Mezi takové hlavní patří ze strany zákazníka příprava stroje na servisní zásah, zákazník má potřebu bližšího termínu, nutnost souběhu servisních prací s jinou organizací, která opravuje jinou část stejného stroje, atd. Pokud stanovený termín

vyhovuje, proběhne servis dle stanovených smluvních podmínek a následné předání s fakturací.

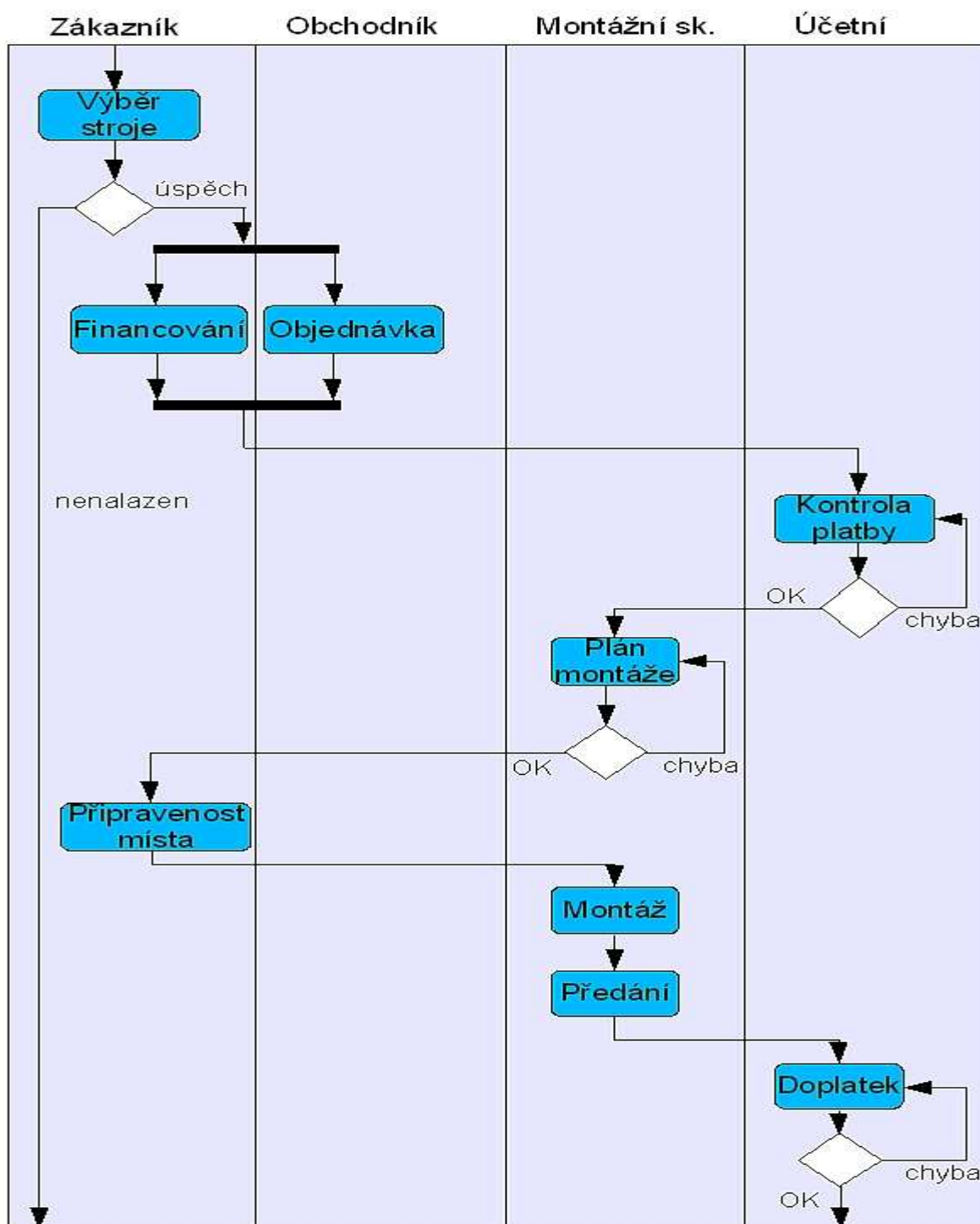


Obrázek 15: Dynamický model servisního zásahu

#### 10.4.3.3 Zodpovědnosti pracovníků

Diagram aktivit znázorňuje, kdo jaké činnosti vykonává a za vykonané činnosti má také příslušnou odpovědnost.

##### 10.4.3.3.1 Zodpovědnosti pracovníků při prodeji stroje



Obrázek 16: Zodpovědnosti účastníků při prodeji stroje

Z obrázku vyplývají jednotlivé zodpovědnosti:

**Zákazník:**

- 1) výběr stroje a vystavení objednávky
- 2) financování stroje

- 3) připravenost místa k montáži

**Obchodník:**

- 1) objednávka stroje u dodavatele
- 2) přijetí objednávky od zákazníka a podepsání smluv

**Montážní skupina:** - plnění plánu montáže, má možnost vyjádřit se k jednotlivým termínům

- 1) montáž stroje
- 2) předání stroje

**Účetní:**

- 1) platba zálohy
- 2) doplatek v případě, že záloha nebyla na 100% smluvené ceny

#### 10.4.3.3.2 Zodpovědnosti pracovníků při servisním zásahu

Z obrázku vyplývají jednotlivé zodpovědnosti:

**Zákazník:**

- 1) objednávka servisu
- 2) financování servisu
- 3) připravenost stroje k servisnímu zásahu

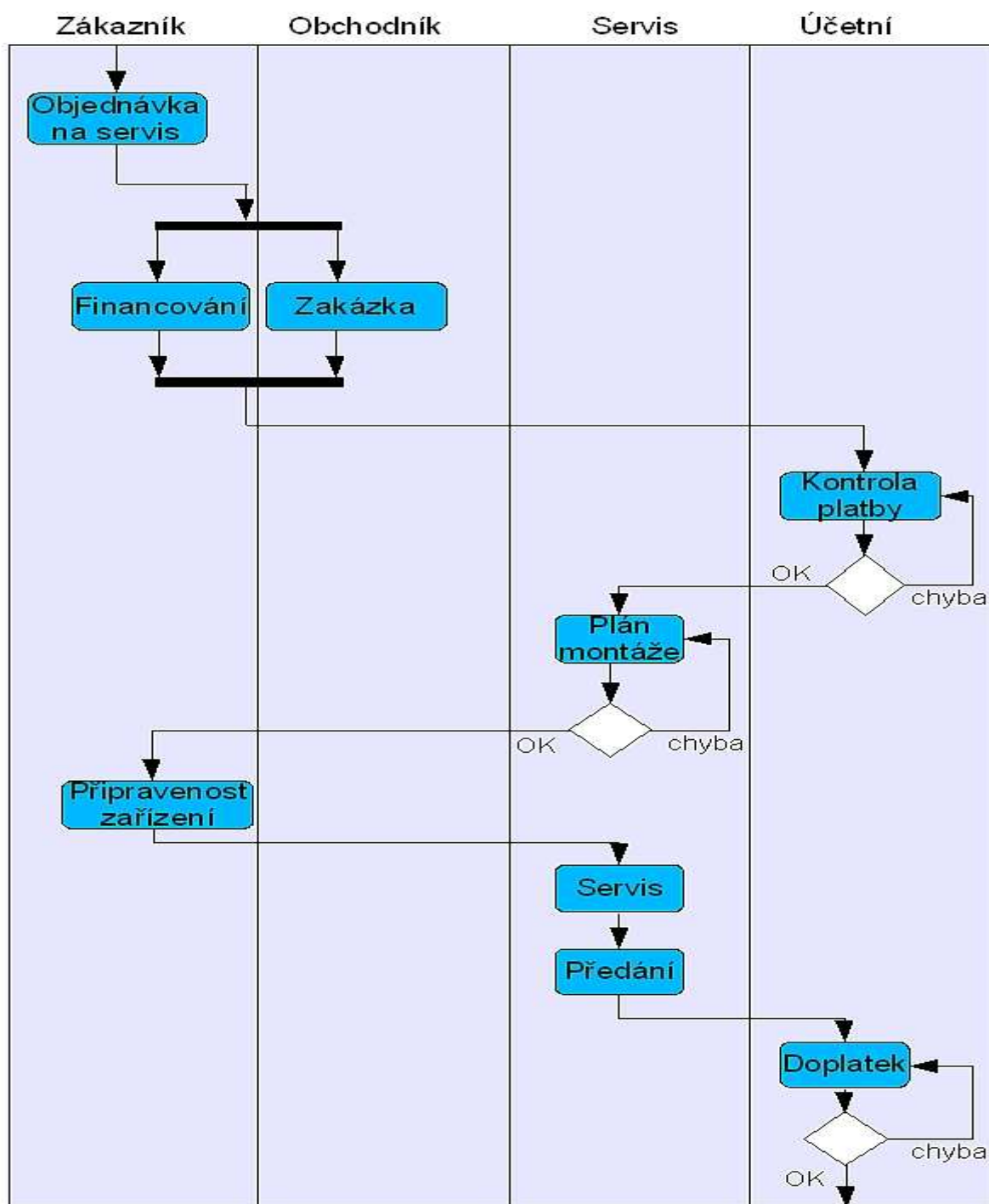
**Obchodník:** - přijetí zakázky na servisní činnost od zákazníka a podepsání smluv

**Montážní skupina:** - plnění plánu servisu, má možnost vyjádřit se k jednotlivým termínům

- 1) servis stroje
- 2) předání stroje

**Účetní:**

- 1) platba zálohy
- 2) doplatek v případě, že záloha nebyla na 100% smluvené ceny

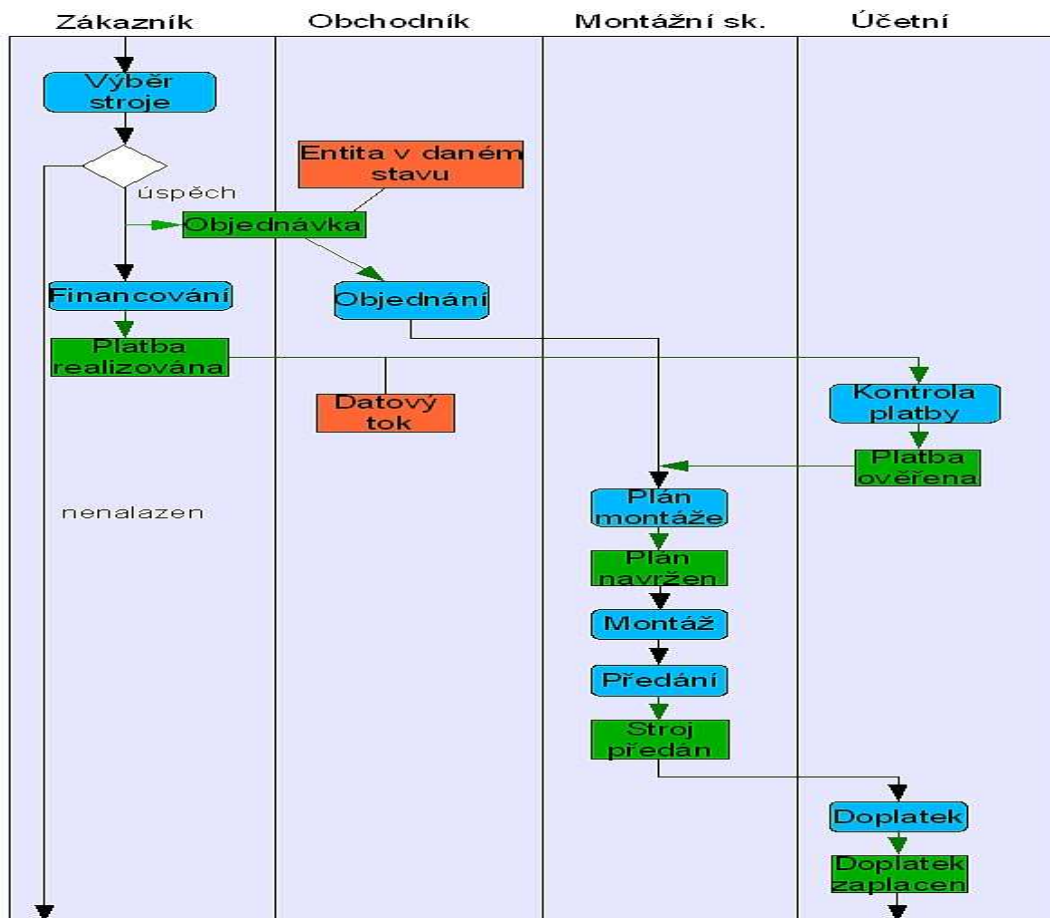


Obrázek 17: Odpovědnost účastníků při servisním zásahu

## 10.5 Modelování entit

### 10.5.1 Modelování entit a jejich stavů při prodeji stroje

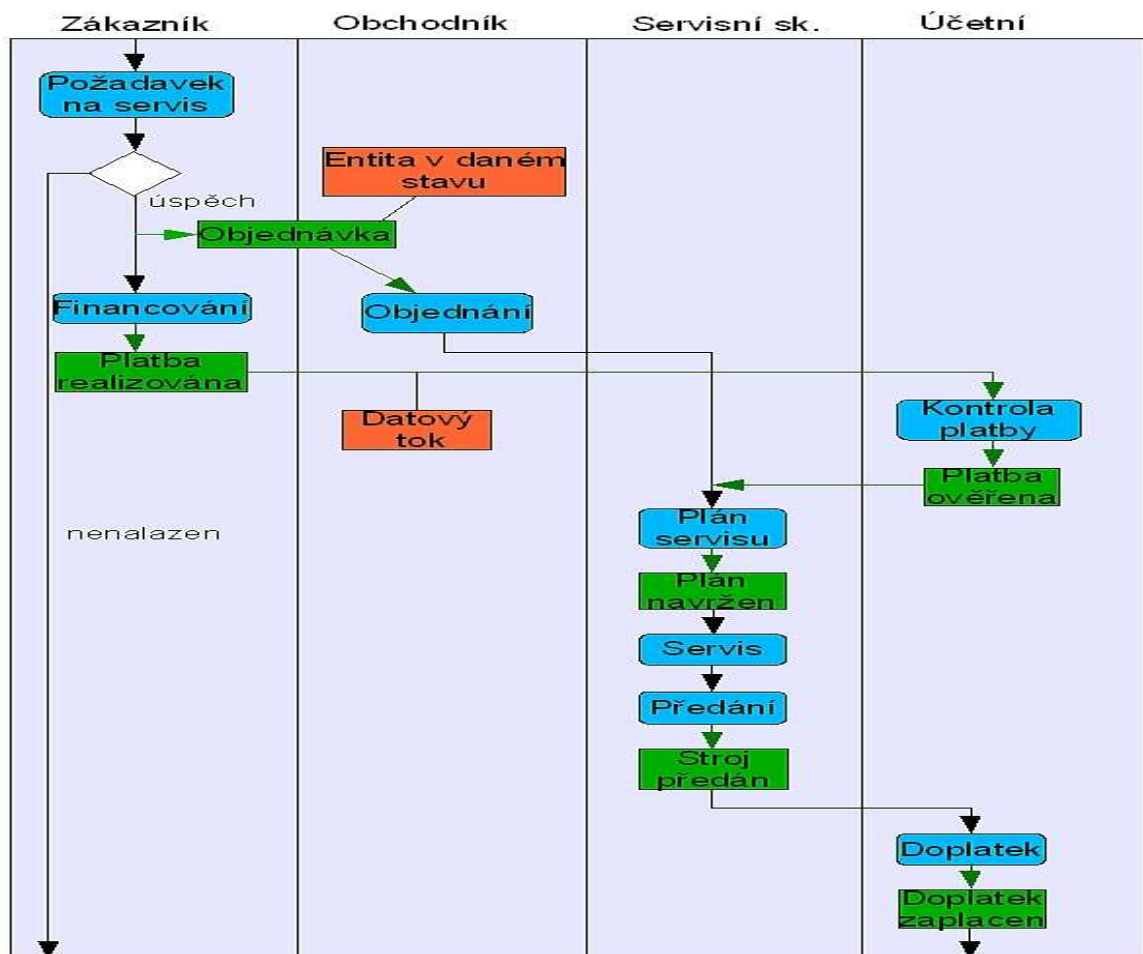
Prodej stroje je zahájen aktivitou jeho výběru. V případě, že firma požadovaný stroj není schopna zajistit, proces končí. V případě, že firma je schopna stroj zajistit je tok rozdělen do dvou souběžných vláken týkajících se „Financování“ produkující entitu platby a vlákna entity „Objednávky“, které zajistí objednání stroje u dodavatele. Aktivita „Kontrola platby“ produkuje entitu „Platba ověřena“. Spuštění aktivity „Plán montáže“ je možné v okamžiku ověření platby. Aktivita „Montáž“ je spuštěna po upřesnění entity „Plán navržen“. Následují aktivity „Montáž“ a „Předání“ produkuje entitu „Stroj předán“ na základě, které vzniká aktivita „Doplatek“. Aktivita „Doplatek je platná jen v případě, že nebyla vybrána záloha ve výši 100% smluvní ceny na dodání stroje nebo zařízení. Aktivita „Doplatek“ produkuje entitu „Doplatek zaplacen“. Zaplacením doplatku končí proces prodeje stroje.



Obrázek 18: Model entit a jejich stavů při prodeji stroje

### 10.5.2 Modelování entit a jejich stavů při servisu stroje

Servis stroje je zahájen aktivitou „Požadavek na servis“. V případě, že firma požadovaný servis neprovádí, je celý proces ukončen. V případě, že je firma schopna požadovaný servis provést, je tok rozdělen do dvou souběžných vláken týkajících se aktivity „Financování“ produkující entitu platby a vlákna entity „Objednávky“ která zajistí objednání servisního zákroku. Aktivita „Kontrola platby“ produkuje entitu „Platba ověřena“. Spuštění aktivity „Plán servisu“ je možné v okamžiku ověření platby. Aktivita „Servis“ je spuštěna po upřesnění entity „Plán navržen“. Následují aktivity „Servis“ a „Předání“ produkuje entitu „Stroj předán“, na základě které vzniká aktivita „Doplatek“. Aktivita „Doplatek je platná jen v případě, že nebyla vybrána záloha ve výši 100% smluvní ceny na servis stroje. Aktivita „Doplatek“ produkuje entitu „Doplatek zaplacen“. Zaplacením doplatku končí proces servisu stroje.



Obrázek 19: Model entit a jejich stavů při servisu stroje

## 10.6 Diagram tříd

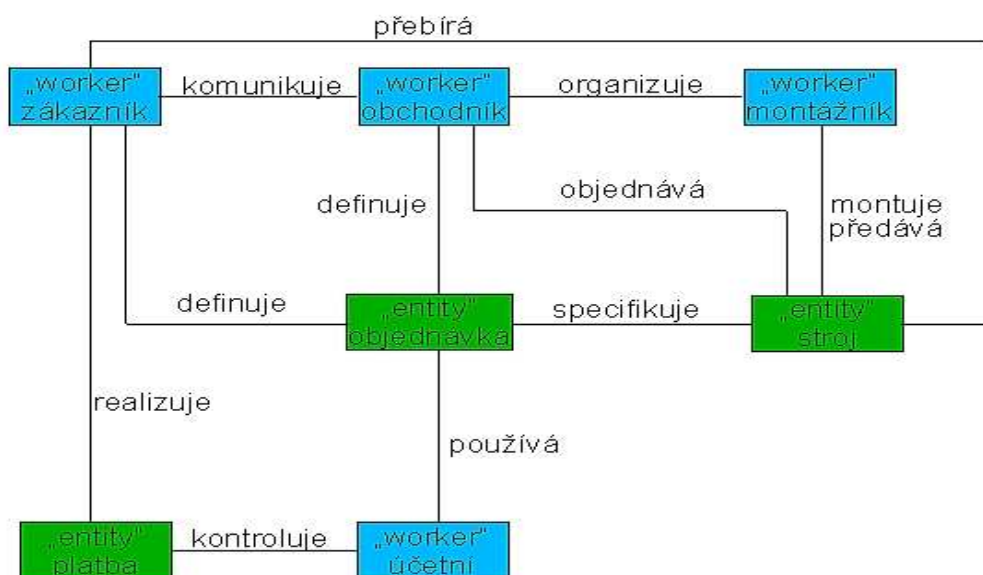
Předchozí diagramy se zaměřují především na popis dynamiky funkčních procesů ve firmě. Prostřednictvím drah zodpovědností a entit byly zavedeny do těchto grafů strukturální elementy. Pro lepší porozumění funkčních závislostí je vhodné zavést diagram tříd.

Diagram tříd je tvořen dvěma typy elementů reprezentujícími aktivně působící pracovníky, kteří budou v grafu označeni klíčovým slovem „worker“ a pasivními entitami, které budou v grafu označeny klíčovým slovem „entity“. Tyto elementy jsou vzájemně propojeny vazbami „asociacemi“, které nám vyjadřují konceptuální nebo fyzické spojení těchto elementů.

**Pracovník** „worker“ definuje chování a zodpovědnost odpovídajícího jednotlivce.

### 10.6.1 Diagram tříd prodeje stroje

K prodeji stroje potřebujeme čtyři aktivní aktéry procesu (zákazník, obchodník, účetní, montážník), kteří jsou označeni v grafu jako „worker“. V případě, že stroj není montován, odpadá aktér (montážník). Zákazník musí komunikovat s prodejcem, aby mohl vzniknout pasivní element „entity“ objednávka. Objednávka specifikuje stroj, který požaduje zákazník. V případě, že je realizována platba ať zálohy, nebo celé smluvní částky, kterou kontroluje účetní následuje montáž stroje a následné předání zákazníkovi.

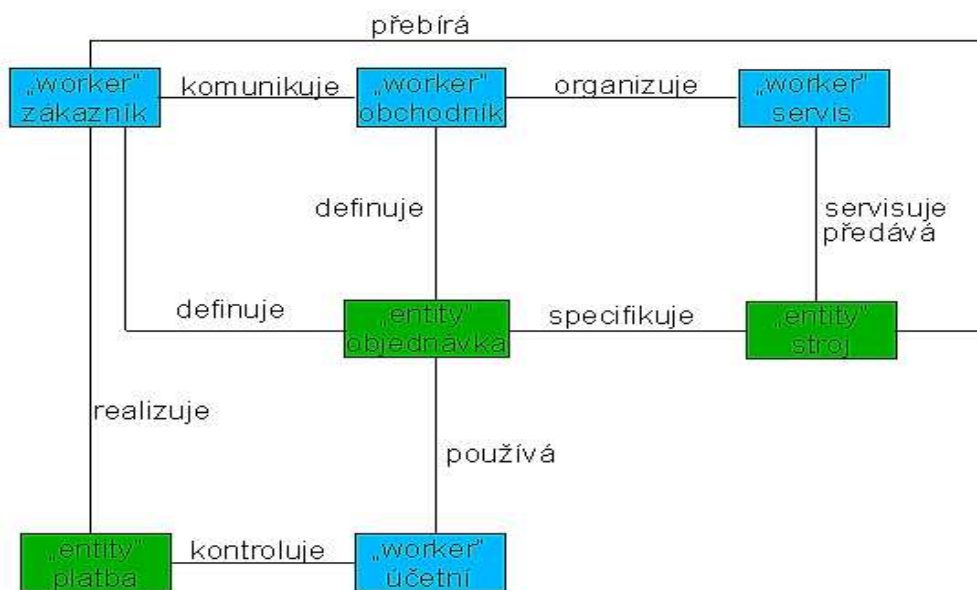


Obrázek 20: Diagram tříd prodeje stroje



### 10.6.2 Diagram tříd servisu stroje

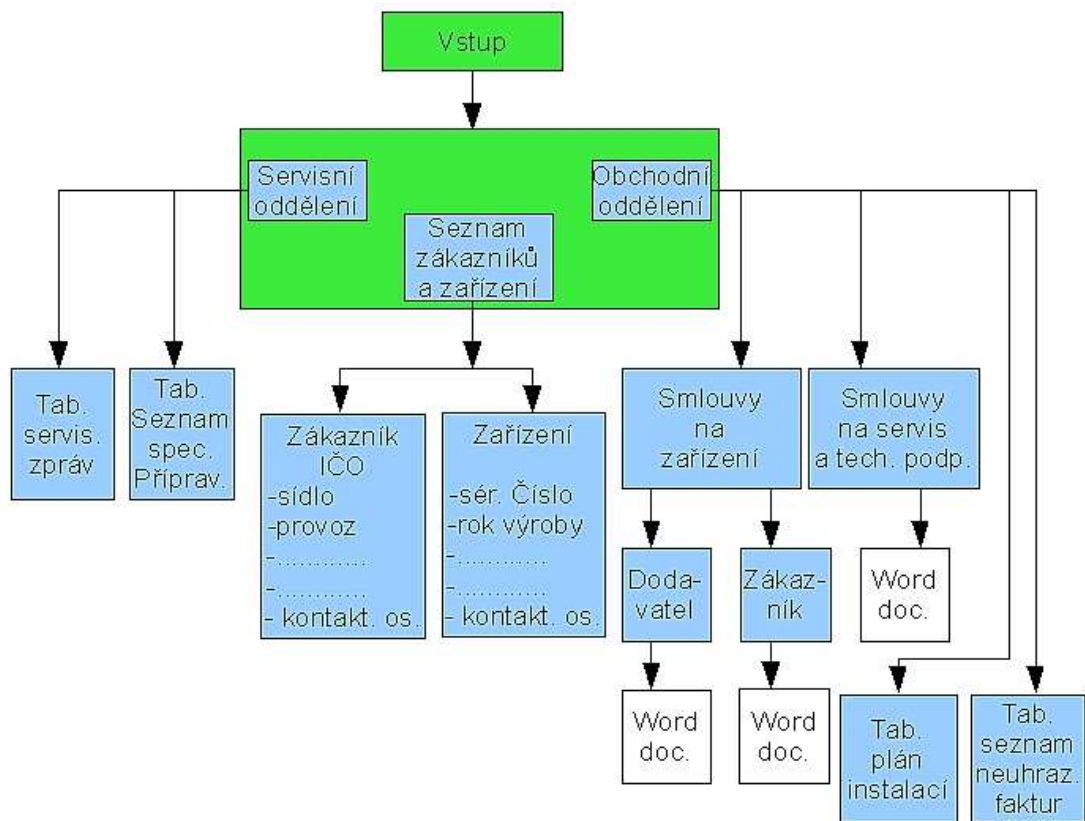
K servisu stroje potřebujeme čtyři aktivní aktéry procesu (zákazník, obchodník, účetní, servis), jenž jsou označeni v grafu jako „worker“. Zákazník musí komunikovat s prodejcem, aby mohl vzniknout pasivní element „entity“ objednávka. Objednávka specifikuje stroj, na který požaduje zákazník servisní zásah. V případě, že je realizována platba ať zálohy, nebo celé smluvní částky, kterou kontroluje účetní, následuje servis stroje a následné předání zákazníkovi.



Obrázek 21: Diagram tříd servisu stroje

### 10.7 Výsledek analýzy

Bylo podrobně analyzováno zadání a principy procesů ve firmě, pro niž má být požadovaný systém navržen. Výsledkem analýzy je model systému. Navržený model systému byl konzultován se zadavatelem, což mělo za úkol dosáhnoutí maximální shody mezi představou zadavatele a pochopenou realitou tvůrce. Dokladem o probíhajících konzultacích je fakt, že až třetí navržený systém vyhovoval všem požadavkům zadavatele, kdy bylo dosaženo maximální shody mezi zadavatelem a realizační stranou.



Obrázek 22: Schéma návrhu systému, u něhož byla dosažena shoda

## 11 NÁVRH SYSTÉMU

Po důkladné analýze potřeb a principů servisního týmu v oblasti údržby, oprav a instalací technologického zařízení v elektronické výrobě požadovaného systému, přichází na řadu jeho návrh. Při návrhu byly brány v úvahu požadavky na funkci celého systému. Tato kategorie tvorby IS má za úkol provést návrh pro programátory.

Pro samotnou realizaci považuji za nejvhodnější třívrstvou architekturu.

Tato architektura se skládá z:

- 1) klientské vrstvy
- 2) aplikační vrstvy
- 3) databázové vrstvy

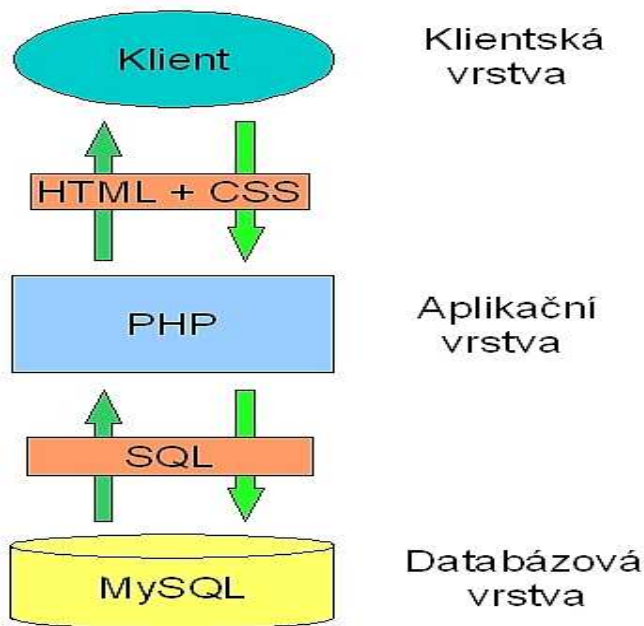
### 11.1 Systémový návrh systému

Pro samotnou realizaci považuji za nejvhodnější třívrstvou architekturu.

Tato architektura se skládá z:

- 1) klientské vrstvy
- 2) aplikační vrstvy
- 3) databázové vrstvy

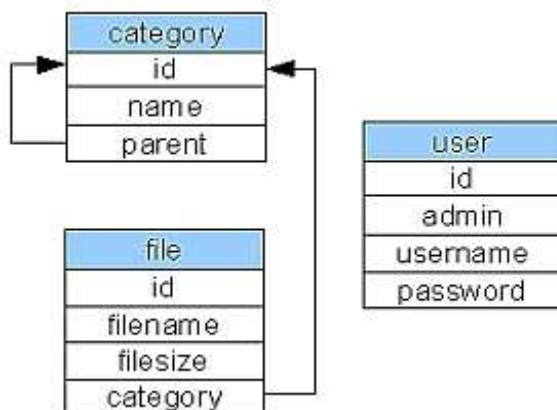
Na klientské vrstvě je využit internetový prohlížeč (Explorer, Opera, Mozilla Firefox,...). Tato vrstva komunikuje s aplikační vrstvou pomocí jazyka HTML a CSS na které je umístěno PHP. Aplikační vrstva komunikuje s databázovou vrstvou MySQL pomocí SQL dotazů.



Obrázek 23: Třívrstvá architektura

### 11.1.1 Struktura databáze

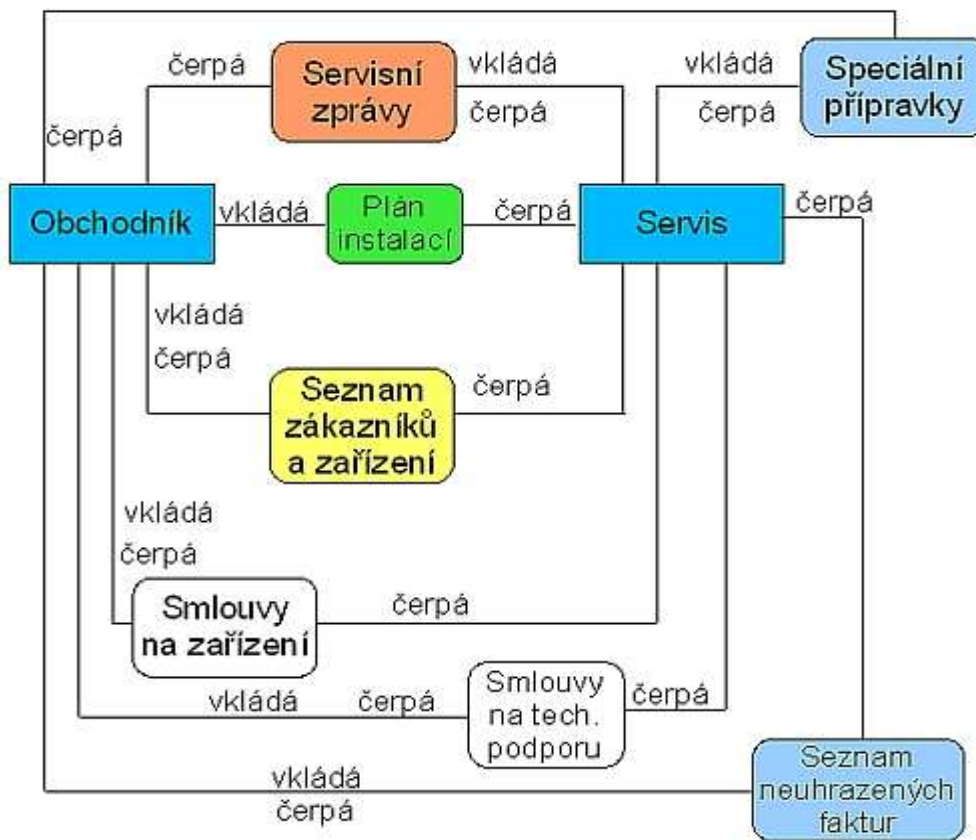
Výsledkem analýzy je struktura databáze. Následující obrázek vyjadřuje vztahy mezi tabulkami.



Obrázek 24: Vztahy v databázi

### 11.2 Objektový návrh systému

Tento diagram má za úkol popsat, jaké budou v systému realizovány objekty a jaké činnosti můžou vykonávat s dokumenty, které má systém za úkol spravovat. Tato fakta jsou důležitá pro správné přidělení přístupových práv, což nám ukazuje, kdo bude mít oprávnění konkrétní dokument vložit, editovat, číst a smazat.



Obrázek 25: Diagram objektů v navrhovaném systému

Po rozboru na jednotlivé účastníky nám vyplynou tyto skutečnosti.

### Servisní zprávy

Obchodník – může číst

Servis – může vystavit, editovat, mazat a číst

Ostatní – můžou číst

### Plán instalací

Obchodník – může vkládat, editovat, mazat a číst jednotlivé servisní zásahy

Servis – může číst

Ostatní – můžou číst

**Seznam zákazníků a zařízení**

Obchodník – může vkládat, editovat, mazat a číst

Servis – může číst

Ostatní – můžou číst

**Smlouvy na zařízení i technickou podporu**

Obchodník – může vkládat, editovat, mazat a číst

Servis – může číst

Ostatní – můžou číst

**Seznam neuhrazených faktur**

Obchodník – může vkládat, editovat, mazat a číst

Servis – může číst

Ostatní – můžou číst

**Speciální přípravky**

Obchodník – může číst

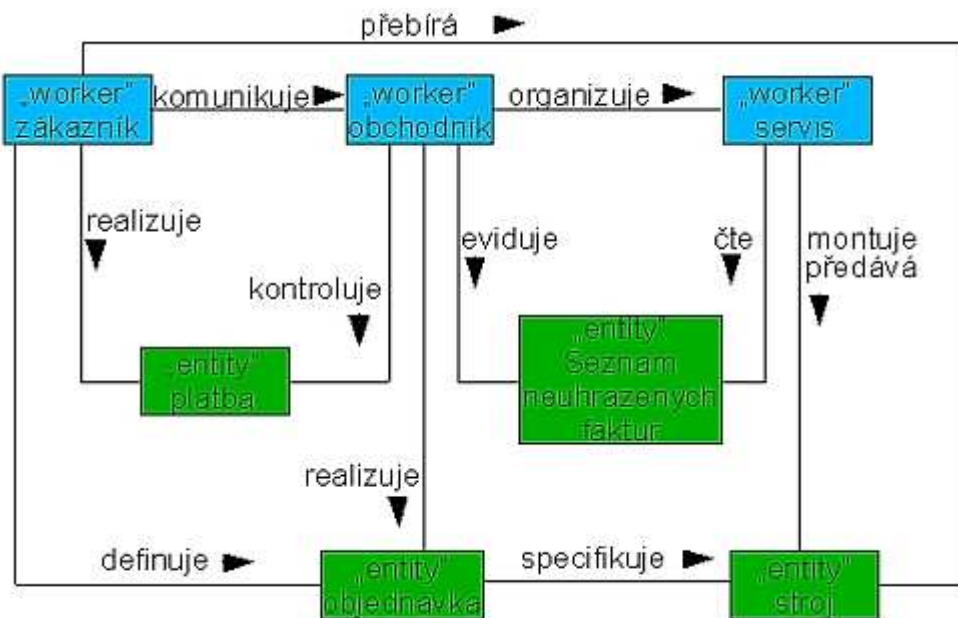
Servis – může vkládat, editovat, mazat a číst

Ostatní – můžou číst

**11.2.1 Diagram tříd v systému**

V navrhovaném systému budou tři aktivní aktéři procesu, v obrázku označeni jako „worker“ (Zákazník, Obchodník, Servis). Do samotného systému budou mít přístup pouze aktéři dva (Obchodník a Servis). V Obrázku je uveden i aktér zákazník pro lepší představu vzniku asociací. Zákazník musí komunikovat s prodejcem, aby mohl vzniknout pasivní element „entity“ objednávka. Objednávka specifikuje stroj, na který požaduje zákazník

servisní zásah. V případě, že je realizována platba ať zálohy, nebo celé smluvní částky, kterou kontroluje v grafu uveden obchodník (informace získá od účetní), následuje servis stroje a následné předání zákazníkovi.



Obrázek 26: Diagram tříd v navrhovaném systému

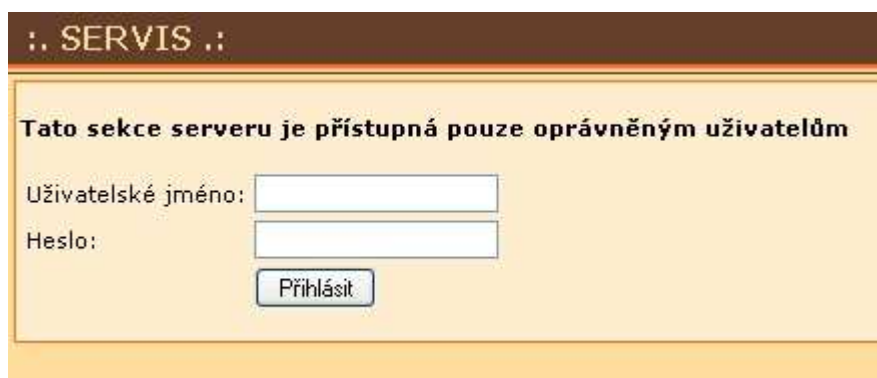
## 12 IMPLEMENTACE

Implementací je proces uskutečňování teoreticky stanovené myšlenky v předchozí fázi návrhu systému. Vstupem pro implementaci je výstup z návrhu systému.

Systém z pohledu administrátora je členěn na „Materiály a kategorie“ a „Spravovat uživatele“.

### 12.1 Ověřování a správa uživatelů

Pro přihlášení do systému je použit formulář, který po správném vyplnění umožní přístup do informačního systému.



.. SERVIS ..

Tato sekce serveru je přístupná pouze oprávněným uživatelům

Uživatelské jméno:

Heslo:

Obrázek 27: Přihlašovací menu

Uživatelé jsou rozděleni do tří skupin:

#### 1) Admin

- může vytvářet podkategorie v části „Materiály a kategorie“
- vkládat, editovat, mazat a číst dokumenty
- vytvářet a měnit účty administrátor, uživatel, čtenář
- vkládat, mazat, měnit hesla administrátorům, uživatelům a čtenářům



## 2) Uživatel

- může vkládat a číst dokumenty
- měnit své heslo

## 3) Čtenář

- může jen číst
- měnit své heslo

Nejvhodnější počet administrátorů je jeden. Na požadavek zadavatele není však počet administrátorů limitován. Nicméně je nutné, aby jeden administrátor, kterému bude systém předán k užívání, vytvořil účet administrátorovi druhému a jeden ze dvou existujících vytvořil účet případným dalším administrátorům. Z pohledu na diagram tříd v části návrh systému je jasné, že budou vytvořeni minimálně dva administrátoři a to minimálně jeden z obchodníků a minimálně jeden ze servisních pracovníků. Tato nutnost vyplývá z možnosti vkládání, editace a mazání podkategorií. Aby systém plnil svou funkci a nedošlo k jeho zhroucení, může mít právo administrátora jen osoba, ve kterou má zřizovatel maximální důvěru a je dostatečně prověřená a zainteresovaná ve vedení. Prvotní předpoklad, ať už bude konečný počet administrátorů jakýkoliv, je v kolektivní práci a v nulovém zneužívání svěřené pravomoci vůči jiným kolegům administrátorům.

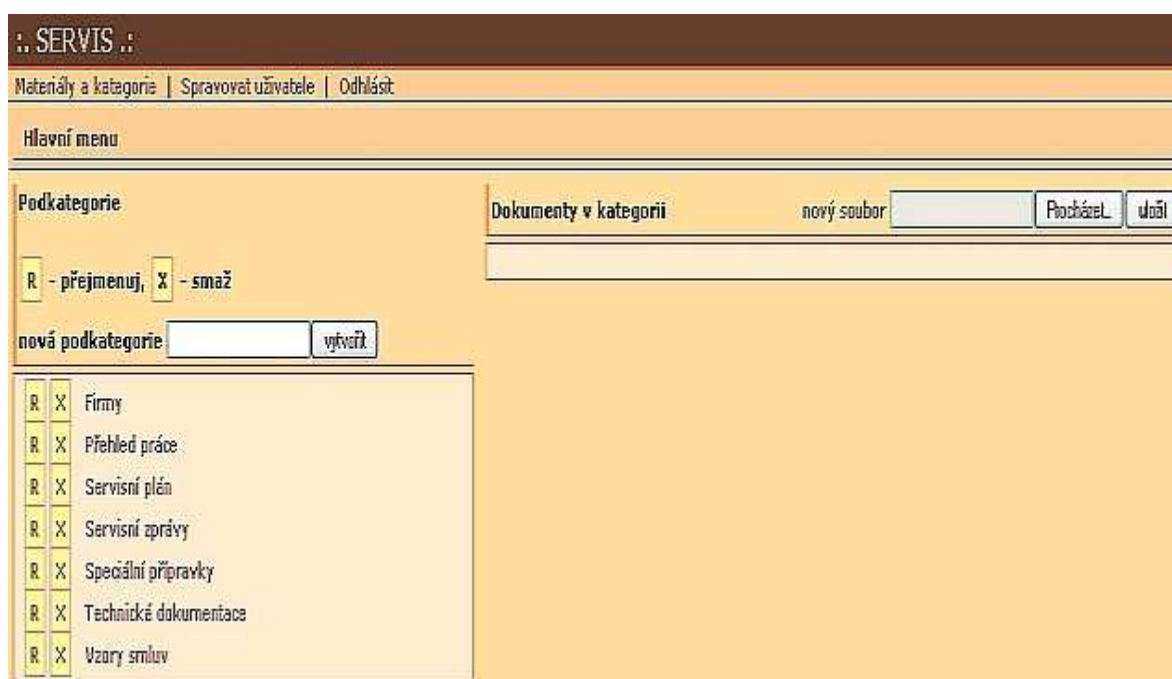
Uživatele představují jednotliví servisní pracovníci, kteří budou vkládat servisní zprávy od servisních událostí, jež vykonaly. Čerpat mohou veškeré informace usnadňující jim práci jak v kanceláři, tak na místě zásahu.

Čtenáři budou všichni, kdo mohou jen číst a není potřeba, aby do systému vkládali jakékoliv dokumenty.

U všech tří kategorií účastníků se na provozu systému dle dodatečných informací zadavatele není nutné vedení jakýchkoliv personálií. Hlavním důvodem je dostatečná vzájemná znalost kolektivu spolupracovníků.

## 12.2 Materiály a kategorie

Správa této části systému spočívá ve vytváření, editaci a mazání podkategorií, které se mohou členit na další a další podkategorie. Množství podkategorií není nijak limitováno. Tento fakt umožňuje mnohonásobné rozčlenění každé zvolené podkategorie. Každá podkategorie může obsahovat vložené soubory, které považují administrátoři za důležité k dané podkategorii. Tím je splněn hlavní požadavek zadavatele, a to vytvoření libovolně rozšiřovatelného systému.



Obrázek 28: Materiály a kategorie

## 12.3 Bezpečnost systému

Základní bezpečnostní prvky informačních systémů jsou podrobněji popsány v teoretické části. V této části se zaměřím na bezpečnostní prvky přímo aplikované na implementovaném systému.

Nezbytnou samozřejmostí je, že propuštěným zaměstnancům bude v čas zamezen přístup do informačního systému. Jsou známy případy, kdy zaměstnanec před odchodem záměrně poškodil systém dle možností vzhledem k právům, které měl.

### 12.3.1 Přihlášení do systému

K úspěšnému přihlášení do systému potřebuje kdokoliv zadat správné „Uživatelské jméno“ a „Heslo“. Při přihlašování do systému je jednou z nejaktuálnějších hrozeb vniknutí do systému tzv. „prolomení hesla hrubou silou“. Pro ochranu systému proti této hrozbě je použita metodika třech přihlášení a stop. Tedy po třech neúspěšných přihlášeních je další možnost přihlášení blokována na určitou dobu. V době testování systému byl tento čas nastaven na 10 minut. Po uplynutí stanoveného času je možné podniknout opět tři pokusy k přihlášení do systému.

Celá „finta“ je dána principem pokusu prolomení hesla hrubou silou, kdy speciální software generuje a aplikuje na přihlašovací menu během doby útoku obrovské množství hesel. Začíná od hesel, které výrobci software používají jako standard pro počáteční spuštění aplikace. Dále vyzkouší sadu nejpobulárnějších hesel, až dojde na řadu cílené generování méně užívaných hesel za účelem prolomení hesla.

Tím, že umožníme zadání pouze tří hesel značně omezíme možnosti útočného software. Navíc samotný software neví, jak dlouho bude čekat na další možnost k použití útoku.

Uživatel systému má možnost nastavit libovolný časový úsek jako prodlevu mezi jednotlivými sadami pokusů k přihlášení.

Samozřejmě nastává otázka, co účastník, kterému se nepodařilo přihlásit z důvodu zadání chybných hesel. Může se jednat o obchodníka na jednání, který nutně potřebuje smlouvy pro zákazníka a nemůže si dovolit čekat ani deset minut. Dál se může jednat o servisního technika, který potřebuje vystavit servisní list nebo se jen podívat na podmínky záruky. Díky aplikaci informačního systému nemusí podmínky studovat, ale stačí mu

nahlédnout na daná fakta až při předávání servisní zakázky zákazníkovi. Zkrácením časového úseku jakožto prostoje mezi sadami pokusů o přihlášení zvýšíme možnosti útočného software. Časový interval deseti minut se dá elegantně obejít vypnutím a znovu spuštěním internetového prohlížeče. Pak nemusíme čekat žádný čas a máme hned tři nové pokusy k přihlášení. Tento fakt působí lidsky na uživatele, ovšem útočný software o této skutečnosti neví, a tak mu nezbyvá než čekat, až uplyne daný časový interval, který může být nastaven i jeden den. Nehledě na fakt, že když se nepodaří uživateli přihlásit na poprvé, bere vše v klidu. Když se nepodaří úspěšné přihlášení ani na druhý pokus, zapůsobí psychika a zvyšuje se pravděpodobnost chybného třetího přihlašovacího pokusu.

O pokus vniknout do systému se nemusí snažit jen speciální software, ale také osoby, jež nemají k dispozici žádný speciální software, a přesto se pokouší o vniknutí do informačního systému firmy. Nutno podotknout, že se stejným výsledkem jako speciální software pokud nedošlo k odhalení hesla jinou cestou.

Princip možnosti použití nových třech pokusů po znovuspuštění internetového prohlížeče je dán session proměnnou, vázanou na konkrétní spojení prohlížeč – server. Tím, že vypneme internetový prohlížeč, ukončíme spojení prohlížeč – server a novým spuštěním dojde k novému spojení s novým ID session. Dále se nabízí možnost pojištění blokad IP adresy, ale mohlo by dojít k zablokování celé sítě za jednou veřejnou adresou, a to bez ohledu na skutečného pachatele, který měl tři neúspěšné pokusy přihlášení. Další možnost pojištění je zadání cookie proměnné, ale hrozí že kdokoliv s trochou znalostí stejně vymaže důvěrná data.

Celé zajištění je primárně obráceno proti automatům, tedy software s účelem prolomit heslo a namátkovým testerům.

### 12.3.2 SQL injection

SQL injection je technika napadení databázové vrstvy programu vsunutím (odtud „injection“) kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu. Toto nechtěné chování vzniká při propojení aplikační vrstvy s databázovou vrstvou (téměř vždy se totiž jedná o dva různé programy) a zabraňuje se mu pomocí jednoduchého escapování potenciálně nebezpečných znaků. [20]

Nejjednodušší obrana spočívá ve vhodné kontrole a úpravě vstupních dat. Prakticky každý skriptovací program s podporou databáze má nějakou vestavěnou funkci pro převedení potenciálně nebezpečných znaků na bezpečnou sekvenci. Tuto funkci je možné použít před vytvořením dotazu. [20]

Implementovaný informační systém má aplikovanou funkci SQL Clar, která má za úkol automaticky převádět nežádoucí znaky, jenž by mohly vést k úspěšnému útoku na neškodné a zlikviduje pokusy o napadení databáze nebo kódu.

### 12.3.3 Uložení přihlašovacích hesel v databázi

Jedná se o prvek ochrany hesel uložených v databázi. V implementaci systému je využita funkce SQL password, mající za úkol neukládat do databáze jednotlivá hesla, ale jejich hashe. Tímto neumožníme získat jednotlivá hesla případnému útočníkovi

## 12.4 Instalovaný software nutný pro běh aplikace

Použitý software byl volen s ohledem na jeho kvalitu, funkcionalitu a také dostupnost. Veškeré softwarové vybavení nutné pro provoz aplikace je volně stažitelné což je hlavní výhodou použití zvolených programů.

Pro úspěšný rozjezd a užívání aplikace je nutné nainstalovat:

- 1) MySQL 5.0.4
- 2) Apache server 2.2
- 3) PHP 5.1.2

### 12.4.1 Instalace

Aby systém správně fungoval je nutné zajistit jeho umístění na server.

- 1) do rootu serveru umístěte stránky (elektronická příloha) a ověřte, že se (byť s chybovým hlášením o neexistenci databáze) stránky zobrazují.
- 2) přes PHPmyAdmin vytvořte databázi kouril a importujte sql soubor (elektronická příloha).
- 3) upravte v sql-init.php v řádku

```
$mysqlConnection = mysql_connect ('localhost', 'XXXXXXXX', 'YYYYYYY') or  
die('<html><head><title>Database</title></head><body><h4>Obnovuji spojeni s  
databazovym serverem. Prosim o strpeni</h4></body></html>');
```

XXXXXXXX za Vámi vytvořeného uživatele MySQL

YYYYYYY za příslušné heslo

A mělo by vše fungovat.

## 12.5 Požadavky na Hardware

Tato aplikace nemá speciální požadavky na hardware. Postačí standardní kancelářský počítač s instalovaným webovým serverem Apache, PHP a MySQL. Pro zajištění přístupu v každém okamžiku je nutné, aby byl stále zapnut. Kapacita nutná pro ukládání dat je dána množstvím dat, která bude aplikace obsahovat. Pro software, který musí být nainstalován se jedná o stovky MB. Samotné vkládané dokumenty z pravidla nejsou moc obsáhlé, proto odhaduji při běžné údržbě cca15 GB jako postačující.

## ZÁVĚR

Úkolem této diplomové práce je vytvořit informační systém, jenž má plnit funkci portálu s možností vzdáleného přístupu k informacím, které obsahuje. Obsahové zaměření je cíleno na práci servisního týmu s informační podporou obchodního střediska. Servisní tým se zabývá údržbou, opravami a instalací zařízení v elektronické výrobě. Informační systém je zaměřen na práci s dokumenty a databázemi zákazníků včetně instalovaného strojního zařízení.

Pro vypracování je využita třívrstvá architektura, kdy na nejvyšší prezenční vrstvě běží internetový prohlížeč, který komunikuje s nižší střední aplikační vrstvou pomocí protokolu HTML a CSS. Na aplikační vrstvě komunikuje webový server Apache se skriptovacím jazykem PHP. Na nejnižší databázové vrstvě běží systém MySQL, komunikující se střední vrstvou pomocí SQL dotazů.

Na prezentovaném výsledku se do značné míry podepsal vývoj při samotné tvorbě informačního portálu, kdy počáteční myšlenka nakonec nebyla uskutečněna buď vůbec, anebo jen částečně. Během tvorby byly mnohé myšlenky nahrazeny novými, lepšími a účelnějšími. Tento fakt je dán vzájemným prolínáním analýzy, návrhu systému a implementací, kdy byli pomocí konzultací během tvorby doladovány funkční detaily.

Z bezpečnostního hlediska je informační systém plně vyhovující pro provoz v reálném světě.

Samotný systém je připraven jako dále rozšiřitelný jak pro podkategorie a dokumenty, které bude obsahovat, tak pro další uživatele firmy, čímž byl splněn jeden se základních požadavků na informační systém jako celek.

**SUMMARY:**

The goal of this diploma thesis was to create an information system able to serve as a portal with remote access to information stored. The content is targeted to work of service team with an information support of sales centre. The service team deals with installation, maintainance and repairs of equipment in electronic production. Information system is focused on work with customer documents and databases including installed facility equipment.

The system was created with three-layer architecture, where the top layer holds the internet browser that communicates with a middle application layer via HTML and CCS protocol. On the application layer runs web server Apache with PHP script. On the bottom database later runs MySQL system communicating with the mid layer via SQL questions.

The presented results are pretty much influenced by the development during process of creation of the information portal itself, where the initial thoughts either haven't been realized at all, or just partially. Along the way many thoughts were replaced by new ones, which were considered better and more effective. This fact is given by mutual permeation of analysis, system draft and implementation, where the functional details were tuned during the creation process by consultations.

From safety point of view is the information system fully compliant to work in real world.



**SEZNAM POUŽITÉ LITERATURY**

- [1] *Databáze dneška popáté* [online]. 2003 [cit. 2008-04-14]. Dostupný z WWW: <<http://www.dbsvet.cz/view.php?cisloclanku=2008041402>>.
- [2] *Databáze dneška šesté* [online]. 2003 [cit. 2008-06-13]. Dostupný z WWW: <<http://www.dbsvet.cz/view.php?cisloclanku=2008061302>>.
- [3] *Databáze dneška sedmé* [online]. 2003 [cit. 2008-08-25]. Dostupný z WWW: <<http://www.dbsvet.cz/view.php?cisloclanku=2008082502>>.
- [4] TVRDÍKOVÁ, Milena. *Aplikace moderních informačních technologií v řízení firmy*. [s.l.] : Grada Publishing a.s., 2008. 176 s. ISBN 978-80-247-2728-8.
- [5] FLANAGAN, David, MAGERA, Ivo, VORÁČEK, Karel. *JavaScript : Kompletní průvodce*. Praha : Computer Press, 2002. 825 s. ISBN 80-7226- 626-8.
- [6] CASTAGNETTO, Jesus. *Programujeme PHP profesionálně*. Ivo Magera, Ludvík Roubíček. 1. vyd. Praha : Computer Press, 2002. 656 s. ISBN 80-7226- 310-2.
- [7] GILMORE, W. , Jason. *Velká kniha PHP5 & MySQL : Kompendium znalostí pro začátečníky i .* Miroslav Kučera, RNDr. Jan Pokorný. 1. vyd. Brno : Zoner Press, 2005. 711 s. ISBN 80-86815-20-X.
- [8] *MySQL* [online]. 1995 , 19.5.2009 [cit. 2009-05-19]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [9] *phpMyAdmin* [online]. 1995 , 11.5.2009 [cit. 2009-05-16]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/PhpMyAdmin>>.
- [10] *Aplikovaná informatika* [online]., 3.11.2009 [cit. 2009-05-15]. Dostupný z WWW: <[kst.uniza.sk/predmety/irsd/Aplikovana%20informatika.pps](http://kst.uniza.sk/predmety/irsd/Aplikovana%20informatika.pps)>.
- [11] *Informační systémy* [online]., [cit. 2009-04-10]. Dostupný z WWW: <<https://akela.mendelu.cz/~rybicka/prez/infysyst07.ppt>>.
- [12] KUČEROVÁ, Helena. *Projektování informačních systémů : sylaby ke kurzu*. 2007. vyd. [s.l.] : [s.n.], 2007. 111 s.
- [13] *Metody zabezpečení informačních systémů* [online]., 11.2.2009 [cit. 2009-04-14]. Dostupný z WWW:

<[http://www.grapeservices.cz/\(wa3fffeltsu1ry45pfdp0tzb\)/default.aspx?id=69&ido=5&sh=220527480](http://www.grapeservices.cz/(wa3fffeltsu1ry45pfdp0tzb)/default.aspx?id=69&ido=5&sh=220527480)>.

[14] *Objektově orientované metody analýzy* [online]., 6.2.2001 [cit. 2009-04-19]. Dostupný z WWW: <<http://reboot.cz/howto/programovani/objektove-orientovane-metody-analyzy/articles.html?id=102>>.

[15] *Implementace* [online]. 1995 , 10.4.2009 [cit. 2009-04-12]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Implementace>>.

[16] *Enterprise resource planning* [online]. 1995 , 18.5.2009 [cit. 2009-05-20]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Enterprise\\_resource\\_planning](http://cs.wikipedia.org/wiki/Enterprise_resource_planning)>.

[17] *Webhosting* [online]. 1995 , 28.4.2009 [cit. 2009-05-12]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Webhosting>>.

[18] *Jak si vybrat informační systém* [online]., [cit. 2005-05-15]. Dostupný z WWW: <<http://informacni-system-erp.cz/vyber-informacniho-systemu.html>>.

[19] *Datové modely* [online]., 7.6.2008 [cit. 2009-05-13]. Dostupný z WWW: <[http://www.student.cvut.cz/cwut/index.php/Datov%C3%A9\\_modely](http://www.student.cvut.cz/cwut/index.php/Datov%C3%A9_modely)>.

[20] *SQL injection* [online]. 1995 , 14.2.2008 [cit. 2009-03-10]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/SQL\\_injection](http://cs.wikipedia.org/wiki/SQL_injection)>.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

PHP	Personal Home Page.
MySQL	Databázový systém
ERP	Celopodnikový informační systém
CSS	Cascade style sheets
XML	Databázový model
CASE	Computer Aided Software Engineering
HTML	HyperText Markup Language

**SEZNAM OBRÁZKŮ**

Obrázek 1: Prvky informačního systému.....	13
Obrázek 2: Strukturovaná metodika .....	18
Obrázek 3: Objektový model .....	23
Obrázek 4: Dynamický model .....	24
Obrázek 5: Funkční model.....	25
Obrázek 6: Datové modely .....	31
Obrázek 7: Vodopádový model .....	42
Obrázek 8: Prototypování .....	43
Obrázek 9: Spirálový model .....	44
Obrázek 10: Funkční model prodeje stroje.....	60
Obrázek 11: Funkční model servisního zásahu .....	62
Obrázek 12: Objektový model prodeje stroje .....	63
Obrázek 13: Objektový model servisního zásahu.....	63
Obrázek 14: Dynamický model prodeje stroje.....	65
Obrázek 15: Dynamický model servisního zásahu .....	66
Obrázek 16: Zodpovědnosti účastníků při prodeji stroje.....	67
Obrázek 17: Odpovědnost účastníků při servisním zásahu .....	69
Obrázek 18: Model entit a jejich stavů při prodeji stroje.....	70
Obrázek 19: Model entit a jejich stavů při servisu stroje.....	71
Obrázek 20: Diagram tříd prodeje stroje .....	72
Obrázek 21: Diagram tříd servisu stroje .....	73
Obrázek 22: Schéma návrhu systému, u něhož byla dosažena shoda.....	74
Obrázek 23: Třívrstvá architektura .....	76
Obrázek 24: Vztahy v databázi .....	76
Obrázek 25: Diagram objektů v navrhovaném systému .....	77
Obrázek 26: Diagram tříd v navrhovaném systému.....	79
Obrázek 27: Přihlašovací menu .....	80
Obrázek 28: Materiály a kategorie.....	82