

Informační systémy 1

texty pro distanční studium

Doc. Ing. Cyril Klimeš, CSc.

Ostravská univerzita v Ostravě, Přírodovědecká fakulta
Katedra informatiky a počítačů

OBSAH

1	INFORMATIKA, IT/IS – TECHNOLOGIE A TRENDY.....	7
1.1	CO JE TO INFORMAČNÍ SYSTÉM.....	10
1.2	TRENDY, ROZDĚLENÍ, VÝVOJ.....	11
	<i>Trendy v oblasti základního SW:.....</i>	<i>13</i>
	<i>Trendy v oblasti aplikačního SW:.....</i>	<i>13</i>
	<i>Trendy v oblasti metod a nástrojů vývoje IS/IT:.....</i>	<i>16</i>
	<i>Trendy v organizaci a řízení IS/IT:.....</i>	<i>16</i>
1.3	ZÁKLADNÍ POJMY.....	17
1.4	PROJEKTOVÁNÍ IS.....	19
2	ARCHITEKTURY IS.....	22
2.1	GLOBALNÍ ARCHITEKTURA.....	24
2.2	DÍLČÍ ARCHITEKTURY IS.....	24
2.3	TYPY ARCHITEKTUR.....	25
	<i>Hardwarová architektura.....</i>	<i>25</i>
	<i>Technologická architektura.....</i>	<i>25</i>
	<i>Softwarová architektura.....</i>	<i>26</i>
3	VRSTVENÁ ARCHITEKTURA SW PRODUKTŮ.....	29
3.1	DEFINICE A KONCEPČNÍ MODEL VRSTVENÉ ARCHITEKTURY.....	30
	<i>Uživatelská, technologická a interfacová vrstva.....</i>	<i>31</i>
	<i>Úplná struktura vrstev softwarového systému.....</i>	<i>33</i>
3.2	NÁVRH A IMPLEMENTACE SYSTÉMU S VRSTVENOU ARCHITEKTUROU.....	34
	<i>Kdo navrhuje vrstvu.....</i>	<i>34</i>
	<i>Jak navrhovat uživatelské rozhraní vrstvy.....</i>	<i>35</i>
	<i>Jak navrhnout a rozmístit jednotlivé typy vrstev ve struktuře systému.....</i>	<i>36</i>
3.3	UMÍSTĚNÍ A DOSTUPNOST FUNKCÍ V HIERARCHII VRSTEV.....	37
	<i>Co ovlivňuje stabilitu struktury vrstveného systému?.....</i>	<i>39</i>
3.4	JAKÝ ZVOLIT PŘI TVORBĚ SYSTÉMU POSTUP?.....	41
4	ŽIVOTNÍ CYKLUS VÝVOJE IS.....	43
4.1	STRUKTURA ČINNOSTÍ JEDNOTLIVÝCH FÁZÍ.....	46
	<i>Informační strategie.....</i>	<i>46</i>
	<i>Úvodní studie.....</i>	<i>46</i>
	<i>Globální analýza a návrh.....</i>	<i>47</i>
	<i>Detailní analýza a návrh.....</i>	<i>47</i>
	<i>Implementace.....</i>	<i>48</i>
	<i>Zavádění do provozu.....</i>	<i>48</i>
	<i>Provoz, údržba a rozvoj.....</i>	<i>49</i>
4.2	DIMENZE TVORBY IS.....	49
4.3	ZÁKLADNÍ PŘÍSTUPY VÝVOJE.....	50
4.4	TRENDY VE VÝVOJI METODIK.....	52
5	ANALÝZA ZADÁNÍ.....	53
5.1	STRUKTURA POŽADAVKŮ.....	55
	<i>Hlavní zásady tvorby specifikace požadavků.....</i>	<i>56</i>
	<i>Používaná slovní spojení.....</i>	<i>57</i>
	<i>Chyby při specifikaci.....</i>	<i>57</i>

Informační systémy 1

5.2	MODEL JEDNÁNÍ	57
	<i>Zobecnění a relace include, extend</i>	61
6	PRINCIPY GLOBÁLNÍ PODNIKOVÉ STRATEGIE	65
6.1	SWOT ANALÝZA	66
6.2	ANALÝZA EXTERNÍCH FAKTORŮ	67
6.3	ANALÝZA INTERNÍCH FAKTORŮ	68
	<i>Priorita faktorů</i>	69
	<i>Komentář k faktoru F1:</i>	70
	<i>Komentář k faktoru F2:</i>	71
	<i>Komentář k faktoru F3:</i>	71
	<i>Komentář k faktoru F4:</i>	71
	<i>Komentář k faktoru F5:</i>	71
	<i>Komentář k faktoru F6:</i>	71
	<i>Shrnutí SWOT faktorů</i>	71
6.4	FORMULACE POSLÁNÍ PODNIKU	72
6.5	DEFINICE GLOBÁLNÍCH PODNIKOVÝCH CÍLŮ	72
6.6	VYMEZENÍ GLOBÁLNÍCH PODNIKOVÝCH FUNKCÍ	73
6.7	TVORBA STRATEGIÍ PRO GLOBÁLNÍ PODNIKOVÉ FUNKCE	73
7	OBJEKTIVĚ ORIENTOVANÉ TECHNOLOGIE, UML	74
7.1	OBJEKTIVĚ ORIENTOVANÉ PRINCIPY	75
7.2	OBJEKTIVĚ ORIENTOVANÉ METODIKY	78
	<i>Metodika OMT – Object modeling technique</i>	79
	<i>Metodika Booch</i>	80
	<i>Metodika OOMT – Objektivě orientované metodiky a technologie</i>	81
	<i>RUP – Rational Unified Process</i>	85
7.3	UML – UNIFIED MODELING LANGUAGE	87
	<i>Krátce o historii jazyka</i>	87
	<i>Rysy UML</i>	88
	<i>Struktura jazyka</i>	88
	<i>UML diagramy v životním cyklu vývoje</i>	90
7.4	OBJEKTIVĚ ORIENTOVANÉ SYSTÉMY ŘÍZENÍ BÁZE DAT	97
	<i>Výhody OODBMS</i>	98
	<i>Požadavky na OODBMS</i>	98
	<i>Architektura OODBMS</i>	99
8	CASE SYSTÉMY	101
8.1	DRUHY CASE SYSTÉMŮ	101
8.2	KOMPONENTY CASE SYSTÉMŮ	103
	<i>Grafický interface</i>	103
	<i>Vstupní interface</i>	104
	<i>Výstupní interface</i>	104
	<i>Podpora slovníků a integrace</i>	104
	<i>Interakce přes obrazovku</i>	105
	<i>Analýza a generace zpráv</i>	105
	<i>Ostatní vlastnosti</i>	105
8.3	VOLBA A HODNOCENÍ CASE NÁSTROJŮ	106
8.4	ZKUŠENOSTI S CASE A MYLNÉ PŘEDSTAVY	106
9	PODNIKOVÉ INFORMAČNÍ SYSTÉMY	109

Informační systémy 1

9.1	ÚLOHY PODNIKOVÉHO INFORMAČNÍHO SYSTÉMU	109
	<i>Úlohy manažerské - typu EIS</i>	110
	<i>Úlohy typu datový sklad (DWH)</i>	110
	<i>Úlohy pro podporu taktického a operativního řízení - MIS</i>	110
	<i>Úlohy elektronické výměny dat - typu EDI</i>	111
	<i>Úlohy pro podporu kancelářských prací - OIS</i>	111
	<i>Úlohy výrobní - typu CAD/CAM</i>	112
	<i>Úlohy zákaznické - typu CIS</i>	112
9.2	JÁDRO INFORMAČNÍHO SYSTÉMU PODNIKU - ERP	113
9.3	MODULY INFORMAČNÍHO SYSTÉMU	129
	<i>A. Finance</i>	130
	<i>B. Obchod</i>	136
	<i>C. Zdroje</i>	142
	<i>D. Výroba</i>	145
	<i>E. Přeprava</i>	150
	<i>F. Servis</i>	150
	<i>G. Stavební výroba</i>	152
10	TVORBA PROJEKTŮ	160
10.1	PROJEKT	160
10.2	TRADIČNÍ MODEL Y PROJEKTOVÉHO ŘÍZENÍ	162
11	EKONOMIKA PODNIKŮ	163
11.1	ZÁKLADNÍ POJMY	163
11.2	ZALOŽENÍ PODNIKU	168
	<i>Živnostenské podnikání</i>	170
	<i>Obchodní společnosti</i>	171
11.3	PODNIKATELSKÉ PROSTŘEDÍ	173
	<i>Vymezení podnikatelského prostředí</i>	173
	<i>Vnitřní prostředí</i>	174
	<i>Nositelé podnikatelského prostředí</i>	175
	<i>Význam studia podnikatelského prostředí</i>	175
	<i>Strategie podniku</i>	175
	<i>Typy strategií s bližší charakteristikou</i>	177
11.4	PODNIKATELSKÝ ZÁMĚR	178
	<i>Struktura podnikatelského záměru</i>	179
12	ÚČETNICTVÍ PODNIKŮ Z POHLEDU INFORMAČNÍCH SYSTÉMŮ	181
12.1	VÝZNAM A FUNKCE ÚČETNICTVÍ	181
12.2	VYUŽITÍ ÚČETNÍCH INFORMACÍ	183
12.3	ÚČETNÍ DOKLADY, ÚČETNÍ ZÁPISY A ÚČETNÍ KNIHY	184
	<i>Faktura</i>	186
12.4	MAJETEK PODNIKU	186
	<i>Rozvaha</i>	188
	<i>Klasifikace aktiv</i>	189
	<i>Klasifikace pasiv</i>	190
	<i>Hospodářské operace</i>	190
12.5	NÁKLADY, VÝNOSY, ZISK	192
	<i>Změna základního jmění</i>	192
	<i>Osamostatnění nákladů a výnosů</i>	192

Informační systémy 1

<i>Kalkulační vzorec</i>	194
12.6 PRVKY ÚČETNICTVÍ	194
<i>Účty</i>	194
<i>Náležitosti účtů</i>	195
<i>Princip podvojnosti a souvztažnosti účtů</i>	205
<i>Účetní zápisy</i>	206
<i>Chyby a jejich opravy</i>	206
12.7 CASH FLOW	207
<i>Výpočet cash flow</i>	208
<i>Vazba na ostatní účetní výkazy</i>	210
13 INFORMAČNÍ ZDROJE	213
13.1 SLOVNÍČEK POJMŮ	219

1 Informatika, IT/IS – technologie a trendy

V této kapitole se dozvíte:

- Jak rozdělujeme informační technologie?
- Co jsou to informační systémy?
- Jaké jsou trendy posledních let v oblasti HW a SW?
- Jaké jsou trendy v oblasti metod a nástrojů vývoje IS/IT?
- Jaké jsou trendy v oblasti organizace a řízení IS/IT?
- O co jde při projektování informačních systémů?

Po jejím prostudování byste měli být schopni:

- Definovat informační systém a uvést příklady.
- Popsat trendy v oblasti HW, SW, metodách, nástrojích, organizaci a řízení IS/IT.
- Popsat proces projektování IS a vyjmenovat některé používané metodiky.

Klíčová slova této kapitoly:

Informační technologie, informační systém, projektování, metodiky.

Doba potřebná ke studiu: 3 hodiny

Průvodce studiem

Studium této kapitoly je jednoduché a popisným způsobem zde nastudujete základní rozdělení informačních technologií a informačních systémů.

Na studium této části si vyhraďte 3 hodiny. Po celkovém prostudování a vyřešení všech příkladů doporučujeme vypracovat korespondenční úkol.

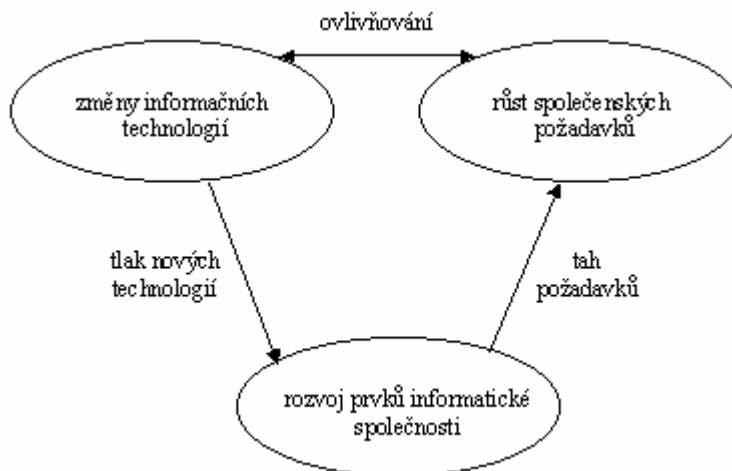


Informační systémy jsou základním nástrojem změn organizace práce a zlepšení kvalitativních charakteristik podniků a organizací [8]. Moderní prostředky komunikace (např. Internet) rozšiřují možnosti a přínosy informačních technologií/informačních systémů (IT/IS). Využívání a vznik nových IT produktů jako jsou platební karty, GSM banking, digitální přístroje v domácnosti a jejich ovládání, mobilní telefony, má také vliv na společnost. Změny se týkají osobních a mezilidských vztahů (např. využívání SMS u mobilních telefonů místo osobního jednání nebo telefonátu). Vzniká společnost nového typu. Vznikají také celé nové výroby, nastávají změny v řízení podniků a firem. Vlivy jsou patrné v různých sférách společnosti.

Příkladem přínosu a významu IS může být přechod od ručního písma až k publikaci na www. Tvorba ručního písma trvala dlouho, text četlo málo lidí, díky knihtisku a později psacím strojům se čas vytvoření podstatně zkrátil a čtenářů přibývalo. Dnes je možné s použitím nějakého editoru napsat text a publikovat ho na webu (Internet, intranet), kde si dílo mohou přečíst miliony čtenářů. Od psaní textu a publikace uplyne maximálně několik desítek minut.

Informační systémy 1

Z tohoto příkladu je také dobře vidět velké snížení nákladů (např. za tisk a distribuci).



Obrázek 1.1

Dalším příkladem může být IS v prodeji potravin. Hokynářství měla několik lokálních dodavatelů, desítky zákazníků denně a vyznačovala se osobní obsluhou každého zákazníka. Prodavač věděl co kdo nejčastěji kupuje a co navíc případně nabídnout. Později nastoupily samoobsluhy, což vedlo ke ztrátě znalosti zákazníka. V posledních desetiletích se rozmohly supermarkety, které reprezentuje: mnoho dodavatelů z celého světa, tlak na dodavatele, statisíce zákazníků. Ke znalosti zákazníka a jeho zvyků se používají zákaznické karty. Zřejmé jsou kvalitativní stupně v objednávání zboží mezi jednotlivými stupni vývoje prodeje:

- Optická kontrola regálu a ruční objednávání.
- Počítačové sledování prodeje, ale ruční objednávání.
- Počítačové sledování prodeje a automatické objednávání.
- Přímé napojení dodavatele do IS prodejce (dodavatel si sám musí zjistit, kdy má dodat zboží).

Integrace probíhá ve všech oblastech služeb, např. prodej nových i ojetých automobilů a s tím souvisejících služeb (financování, pojištění, servis) přes WWW. Oblastí, která jde příkladem je bankovníctví. Homebanking dramaticky snižuje náklady (kamenná pobočka nahrazena elektronickou) a přitom zvyšuje dostupnost (z kterékoliv místa na světě s přípojkou na Internet nebo s GSM signálem) a disponibilitu banky (24 hodin denně 7 dní v týdnu).

Případová studie: „Jak schopný management změnil jedno květinářství“

Výchozí situací je malé zahradnictví s prodejnou. Po provedení SWOT analýzy byly zjištěny následující plusy a mínusy:

- (-) malý okruh zákazníků
- (-) sezónnost prodeje – skleníky zrušeny z důvodu drahého vytápění
- (-) omezení na tuzemské květiny
- (+) kreativní management – vlastník

Hlavní myšlenkou nové globální strategie je maximální využití příležitostí současné doby. Jedná se o využití kooperace, zapojení do řetězců a využití IT.

Cílem je několikanásobné zvýšení obrátu a zisku během jednoho roku.

Informační systémy 1

Vize vedení (nový model podnikání) je:

- zapojení zákazníka do tvorby kytice (typy květin, design kytice, balení, blahopřání, místo a čas dodání),
- objednávání přes Internet,
- pro méně kreativní zákazníky nabízet hotové návrhy kytic ⇒ najmout externího designera,
- zrušit vlastní zahradnictví, nakupovat ze zahraničí (holandský velkoobchodce),
- pro dodávky zákazníka využít externího partnera (Česká pošta, FedEx, UPS, ...).

Hlavní myšlenky nové informační strategie: Firma dosud neměla žádný IS/IT, účetnictví a personalistika řešena externí firmou. Firma se zcela soustředí na hlavní předmět svého podnikání. Nový informační systém bude řešen outsourcingem a to ve třech projektech:

1. *Projekt www objednávky – komplexně řešen externě*

Interface: objednávka zákazníka v první fázi faxem, v druhé fázi elektronicky – e-mailem

Cena: -5% z ceny objednávky – motivace www poskytovatele

Náklady:	1. fáze	fax	2.000 Kč
	2. fáze	PC s připojením na Internet	25.000 Kč
Termín:	1. fáze	4 týdny	
	2. fáze	8 týdnů	

Po dobu jedné sezóny pobeží ještě provoz zahradnictví.

2. *Projekt systému pro styk s velkoobchodcem květin, stuh, ...*

3. *Projekt systému pro styk s bankou – příjem plateb platebními kartami*

Další možné projekty:

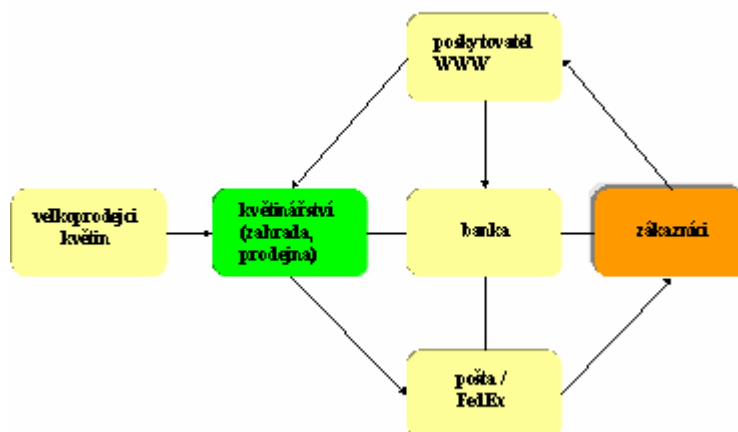
4. *Projekt systému pro styk s FedEx – dodávky do zahraničí*

Poučení vyplývající z případové studie:

Byl zvolen správný postup managementu. Nejdříve byly vytyčeny cíle, kterých chtěla firma (květinářství) dosáhnout. Poté byl definován druh a struktura poskytovaných produktů a služeb. Tyto změny byly provedeny. Vedení přistoupilo také ke spolupráci – outsourcingu – s dalšími firmami (holandský velkoobchodce, Česká pošta, FedEx, provozovatel a tvůrce IS/IT, ...), které zajišťují dovoz a dodání květin a ozdob a provoz vlastního IS. Důležitým bodem je také vytvoření a zavedení IS. Po aplikování všech těchto kroků má organizace novou strukturu, viz obrázek 1.2.

Informační systémy 1

Situace po EBE (Electronic Business Engineering)



Obrázek 1.2: Nová situace

Žijeme v globální informační společnosti, kde nejcennějším zdrojem jsou znalosti a informace (již to není půda, suroviny, pracovní síla a kapitál). Moderní firma, která chce obstát v konkurenci se neobejde bez kvalitního informačního systému vybudovaného na bázi moderních informačních technologií.

V informatice jsou zjevné dlouhodobé trendy. Nastává rozvoj informačního sektoru v ekonomice. Podniky, které se nepřizpůsobí rozvoji IT, ztrácí konkurenceschopnost a mohou i zaniknout. Výrazně se mění pracovní podmínky a pracovní prostředí. Informace budou uchovávány v elektronické podobě – zavedení elektronického podpisu a jako příklad vznik e-governmentu, možnost elektronických daňových přiznání. Bude se rozvíjet elektronická komunikace a práce doma a z domova.

1.1 Co je to informační systém

Co tedy je IS a z čeho se skládá? Podle autora Krále je *informační systém* soubor lidí, metod a technických prostředků zajišťujících sběr, uchování, analýzy a prezentace dat určených pro poskytování informací mnoha uživatelům různých profesí. IS může a nemusí být podporován výpočetní technikou. My budeme uvažovat systémy podporované PC.

IS se skládá s následujícími komponent:

- Technické prostředky (hardware) – počítačové systémy doplněné o periferní jednotky.
- Programové prostředky (software) – jsou tvořené systémovými programy, které řídí chod počítače, efektivní práci s daty, komunikaci počítačového systému s reálným světem a programy aplikačními.
- Datové zdroje – ke své práci je využívají programové prostředky.
- Organizační prostředky (orgware) – soubor nařízení a pravidel. Ty definují provozování a využívání informačních systémů a informačních technologií.
- Lidská složka (peopleware) – řeší otázky adaptace a účinného fungování člověka v počítačovém prostředí, do kterého je zasazen.

Informační systémy 1

- Reálný svět (informační zdroje, legislativa, normy) – kontext informačního systému.

Informační systém musí mít prostředky sběru, kontroly a uchování dat. Data musí být zobrazitelná ve srozumitelné formě pro uživatele. Jinak potřebujeme data zobrazit pro ředitele, jinak pro návrháře a jinak pro skladníka. Toto zobrazení bývá častým problémem budování IS.

IS je nástroj podporující jisté činnosti, proto ho není možno koupit jako obyčejný program, je třeba upravit již existující nebo vytvořit nový. K tomu je zapotřebí analýza potřeb a požadavků a z toho vyplývá spolupráce dodavatele se zákazníkem. Je potřeba vědět proč, z jakého důvodu je IS zaváděn. Základními problémy, které mohou vést až k nedokončení vývoje, bývá nejasnost nebo nekomplexnost požadavků na systém, nedostatek zájmu a podpory ze strany budoucího uživatele nebo také nedostatek zdrojů (čas, peníze).

1.2 Trendy, rozdělení, vývoj

Nyní již víme, co si pod pojmem informační systém představit. Zmíníme tedy rozdělení systému a informačních technologií podle několika kategorií a klasifikací. Dále zmíníme trendy v oblastech hardware a software. Hlavní rozdělení IT do dvou kategorií je:

- *Technické prostředky (HW)* – zařízení na pořizování, uchování, přenos, zpracování a prezentaci dat.
- *Programové prostředky (SW)* – algoritmizované postupy vyjádřené ve formě, v které jsou srozumitelné pro používaná technická zařízení.

Klasifikace systémů I.:

- Přírozené systémy – živé, neživé.
- Navrhované umělé systémy (vytvářené člověkem) – jedná se o stroje, automobily, počítače, knihy, ...
- Systémy lidských aktivit – stát, instituce, organizace, zájmové organizace – jedná se o interakci lidí.
- Transcendentální systémy – jsou systémy přesahující hranice lidského chování.

Klasifikace systému II.:

- Transcendentální systémy.
- Sociální systémy.
- Člověk se svou inteligencí.
- Živočichové.
- Genetické systémy – mají společný genetický kód.
- Otevřené systémy – spojeny s existencí života, např. buňka.
- Kybernetické systémy – systémy se zpětnou vazbou.
- Mechanické systémy – stroje a technická zařízení.
- Fyzikální systémy.

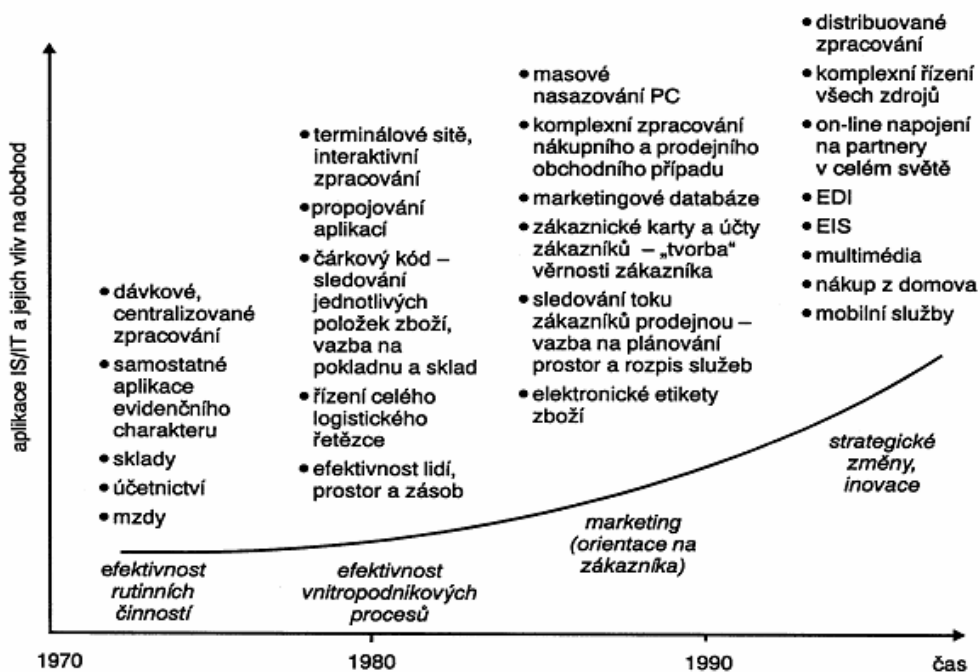
Klasifikace systémů III.:

Informační systémy 1

- *Tvrdé systémy* – mají dobře strukturované problémy – vztahy mezi vstupy a výstupy lze exaktně vyjádřit, vstupy mají převážně kvantitativní charakter, problémy lze algoritmovat.
- *Měkké systémy* – špatně strukturované problémy, vyskytují se v nich neurčitosti, rizika, nejistoty, vstupy nejsou vždy věrohodné, problémy nejsou algoritmovatelné.

Informační systémy mohou být několika druhů, pro různé oblasti nasazení. Jedná se o systémy pro přímé řízení procesů, pro řízení, pro podporu rozhodování, systémy automatizující podnikovou administrativu, expertní systémy, IS pro vrcholové řízení, strategické IS, metainformační systémy.

Velký vliv na rozvoj IT má samozřejmě rozvoj komunikací. Příkladem rozvoje může být vývoj technologií pro přenos hlasu a dat po telefonních linkách, od DSVD (Digital Simultaneous Voice Data) k ISDN (Integrated Services Digital Network). Samostatnou, ne však odtrženou, kapitolou je vznik a rozvoj Internetu, který s sebou přinesl vznik takzvané „Nové ekonomiky“. Tato dotcomová bublina později splaskla a spousta internetových firem zkrachovala¹, nicméně podíl a vliv Internetu a těchto firem na rozvoj a chápání e-commerce a celého IT je obrovský. Z výše uvedených řádků je jasné, že oblast IT má vztah k ekonomice, řízení a organizaci podniku. Na obrázku 1.3 vidíme vliv aplikace IS/IT na obchod a vývoj v čase od konce 70. let.



Obrázek 1.3: Aplikace IS/IT a jejich vliv

Organizační struktura v 70. letech byla převážně hierarchická. IS/IT bylo reprezentováno centrálním počítačem. V 80. letech vznikaly relativně nezávislé jednotky (divize a SBU) zaměřené na hlavní předmět činnosti. Struktura IT byla tvořena počítači propojenými přes síť LAN. 90. léta a první roky nového

¹ Firmy jako Amazon, Google nebo e-Bay tento „krach“ přežily a nyní díky dobré strategii a nápadům dokonce vykazují zisk.

Informační systémy 1

století jsou ve znamení flexibilních organizací, kde se struktura organizace pružně přizpůsobuje měnícím se podmínkám. Tvoří se dynamické týmy a IS/IT se vyznačuje distribuovaným a mobilním zpracováním dat.

Sílí vazba mezi IS/IT a reengineeringem podnikových procesů. Podniky přizpůsobují IS/IT dynamice světového vývoje a změnám podnikových procesů. Vznikají virtuální pracovní týmy a začínají se poskytovat mobilní služby IS/IT. Roste také význam informací o okolí a pro okolí podniku. Probíhá přesun priorit ke strategickému řízení a také posun zaměření IS/IT – snižování nákladů, zvyšování kvality při vzrůstající rychlosti reakce na kladené požadavky.

Existuje rozdílná doba morální životnosti HW, základního SW (operační systémy, SŘBD) a aplikačního SW. Tato doba je u HW asi 2-3 roky, u ZSW 4-7 let a u ASW je doba životnosti dokonce 10-15 let. Spousta z nás se alespoň jednou setkala s tím, že na počítači s instalovaným operačním systémem např. Windows 2000 vyměnila třeba grafickou kartu nebo rozšířila operační paměť RAM během dvou tří let.

Jaké jsou tedy trendy v oblasti hardware? Trvale se snižuje poměr cena/výkon a na druhé straně se zvyšují nároky SW na HW. Roste podíl PC oproti středním a velkým počítačům. V posledním pár letech prudce roste prodej přenosných počítačů – notebooků (ale také PDA a handheldů) oproti stagnujícímu prodeji PC. Technické prostředky se stávají snadněji rozšiřitelnými. Tuto rozšiřitelnost a kombinování jednotlivých technických prostředků různých výrobců umožňuje standardizace.

Trendy v oblasti základního SW:

Standardizují se funkce a uživatelské rozhraní operačních systémů (Windows, MacOS). Rozvíjí se distribuované systémy a s tím souvisí vznik platforem jako J2EE či .NET². Rozvoj komunikačního ZSW a s ním spojených služeb. Trendy v oblasti databází se vyznačují přechodem od relačních k postrelačním databázím. Vznikají a používají se objektově-relační, objektové, deduktivní, XML databáze.

Trendy v oblasti aplikačního SW:

Trendy technologicky orientovaného SW jsou office balíky – kanceláře – a zaměření na workflow. Vzniká typový aplikační software s možností parametrizace. Takový software je komplexní a lze „nastavit“ pomocí parametrů dle potřeb uživatele. Aplikační SW má stavebnicovou architekturu, lze přidávat a odebírat různé části (díly) software.

Vznikají *otevřené systémy*, které jsou standardizovány mezinárodními organizacemi a konsorcií (např. W3C, IEEE, ISO, OMG, ...), členy takových skupin jsou často velcí a významní hráči na IT trhu (IBM, Microsoft, SUN, HP, Intel, Oracle, ...). Díky těmto otevřeným standardům se upouští od proprietárních řešení výrobců. Týká se to převážně těchto oblastí:

- Počítačové sítě (OSI model, protokoly TCP/IP, IPX)

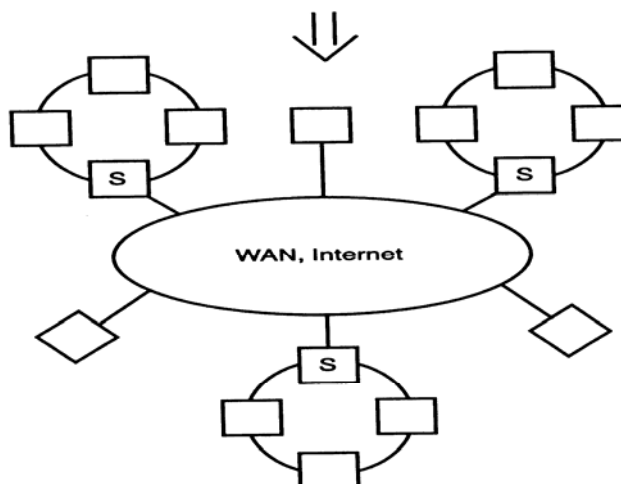
² J2EE – Java 2 Enterprise edition od firmy SUN Microsystems (<http://java.sun.com>)
.NET – platforma firmy Microsoft (<http://www.microsoft.com>)

Informační systémy 1

- Operační systémy
- Objektové prostředí (CORBA, COM/COM+, UML)
- Komunikace s databází (SQL, ODBC)
- Nezávislost uživatelského rozhraní na výpočetním rozhraní
- Možnost výměny HW a ZSW bez vlivu na aplikaci

Aplikace jsou konstruovány tak, že se přechází od jednovrstvé architektury k třívrstvé. V jednovrstvé architektuře jsou data, funkce i uživatelské rozhraní integrovány v jeden celek. V architektuře třívrstvé jsou všechny tyto části odděleny, s tím souvisí i využívání klient/server architektury. V takovém případě jsou data a funkce uloženy na serveru a na klientském počítači je pouze uživatelské rozhraní – prezentační logika, pomocí které uživatel požaduje určité funkce po serveru. Server provede danou službu a zašle odpověď zpět klientovi, kde je reprezentována pomocí uživatelského rozhraní. Díky oddělení těchto tří logických celků, lze jednoduše vyměnit určitou část systému, např. úložiště dat nebo uživatelské rozhraní bez velkých zásahů do aplikace. Změní se pouze komunikační rozhraní částí, kterých se to týká.

Přechází se od centralizovaného a decentralizovaného zpracování k distribuovanému a globálnímu zpracování. V počátcích počítačového zpracování (70. léta minulého století) bylo zpracování dat soustředěno kolem centrálního počítače. Centrální bylo rovněž řízení a kontrola. Později bylo zpracování dat decentralizováno na samostatné počítače, tím bylo dosaženo zvýšení produktivity dílčích prací. Tyto jednotky ale pracovali individuálně bez nějaké kooperace. S přechodem na distribuované zpracování dat založené na architektuře klient/server bylo dosaženo kooperace v rámci týmu (využití LAN sítě a centrálního serveru). Posledním stupněm vývoje je zpracování v rozsáhlých sítích, kde se využívá dynamické kooperace virtuálních týmů. Situace viz obrázek 1.4.

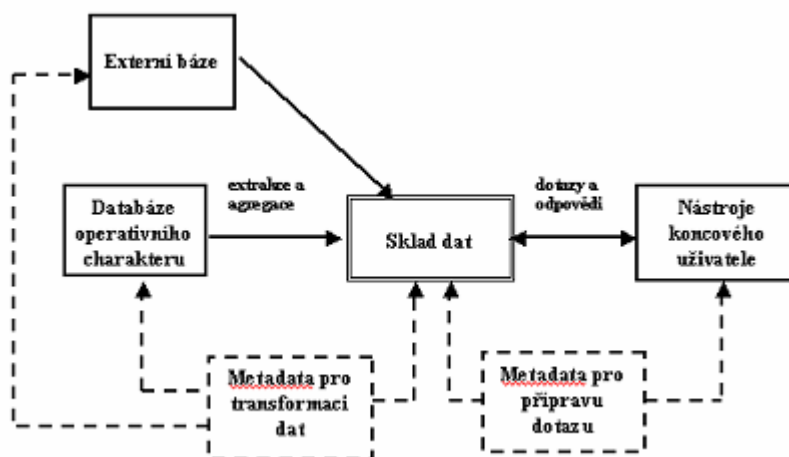


Obrázek 1.4: Zpracování v rozsáhlých sítích

Díky možnosti spojení několika virtuálních týmů lze pracovat s velkými objemy dat. Pro zpracování velkého množství dat je třeba mít také nějaké úložiště, odkud data budeme číst a také kam je budeme ukládat. Pro práci s velkými objemy dat se používají datové sklady – *Data Warehouse* (obr. 1.5).

Informační systémy 1

Takové databáze nejsou určeny k manipulaci s daty (ruční vkládání, mazání, editování), ale pouze k vytváření sestav a analyzování ukazatelů. Data jsou v nich uložena způsobem, jenž je vhodnější pro tento druh práce, a programy, které s nimi manipulují, jsou optimalizovány na čtení velkého množství dat (na úkor zápisu, který zde bývá velmi pomalý). U větších databázích není únosné vždy přenášet všechna data do datového skladu. Z tohoto důvodu je třeba navrhnout metody, často poměrně složité, které přenesou pouze nové nebo změněné záznamy. Takovými metodami se říká *transformační procedury*.



Obrázek 1.5: Data WareHouse

Při vytváření datového skladu se většinou neomezujeme jen na přenos dat z jednoho provozního informačního systému. Je vhodnější do skladu integrovat data ze všech dostupných datových zdrojů v organizaci, tedy z různých informačních systémů, účetních programů, souborů vytvořených v tabulkových procesorech nebo menších databázích a také ze zdrojů na Internetu. Provázáním dat z těchto systémů se zvýší jejich hodnota a uživatelé mohou získat informace, jenž dříve museli pracně určovat porovnáváním výpisů z několika různých programů. Datový sklad tím také plní velmi důležitou úlohu, která spočívá ve sjednocení názvosloví informačních systémů (např. sloupec CENA může znamenat něco zcela jiného ve skladovém hospodářství a něco jiného v ceníku produktů).

Nevýhodou tohoto řešení jsou vyšší náklady spojené s instalací nového počítače, vytvořením struktury datového skladu a transformačních procedur. Mezi hlavní výhody patří mnohem kratší doba vyhodnocení dotazů, tisku sestav a konzistentnost údajů ve zprávách (též reportech).

Grafické uživatelské rozhraní (zkráceně GUI) se také mění a vyvíjí. Existují dokonce obory a disciplíny³, které se zabývají vzhledem a hlavně použitelností GUI, využitím symbolů – viz koš jako místo pro vymazané soubory, apod. Uživatelské rozhraní různých aplikací se sjednotilo. Například textový editor využívá v hlavním menu nabídek jako Soubor, Úpravy, Ústroje, ..., Nápověda. Ty samé nebo podobné nabídky najdeme také u grafických editorů, webových prohlížečů, ale také u RAD, CAD nástrojů nebo prostředí pro práci s mapami.

³ Oborem, který toto vše zastřešuje je HCI – Human Computer Interaction.

Informační systémy 1

Rozšířilo se také multimediální uživatelské rozhraní a multimedia vůbec. Můžeme si všimnout velké podpory multimédií u ZSW, konkrétně u operačních systémů. Za všechny můžeme jmenovat Linux či Windows XP, jež obsahují spoustu programků a utilit pro prohlížení fotek, přehrávání videa a hudby a usnadnění práce s nimi. Nejen multimedia, ale i virtuální realita má své využití v praxi, například u prezentace zboží. Rozvoj simulace přinesl možnost simulování převážně nevratných procesů ve zdravotnictví, armádě nebo automobilismu a stavebnictví. Z rozvojem multimédií a virtuální reality souvisí také rozvoj vzdělávacího, zábavního a filmového průmyslu.

Trendy v oblasti metod a nástrojů vývoje IS/IT:

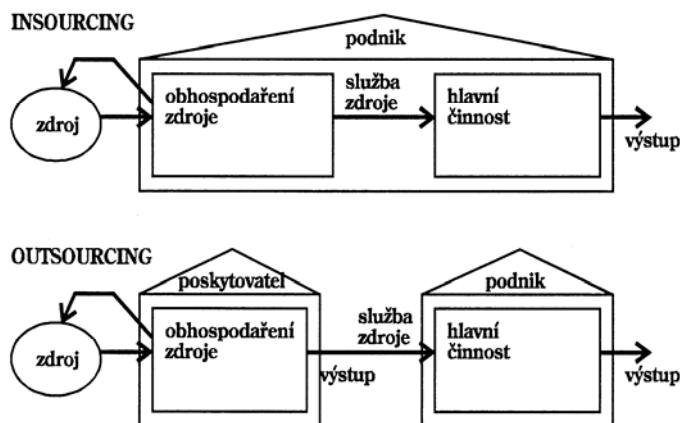
Také v této oblasti proběhla standardizace. Ta se týká také tvorby a realizace informační strategie organizací a podniků. Vývoj částí IS se provádí jedním projektem, tento projekt je řízen, přičemž práce na projektu jsou standardizovány, standardní jsou také postupy implementace a údržby ASW (různé instalátory).

Probíhá odklon od klasického sekvenčního vývoje IS/IT (etapy: specifikace požadavků, analýza, návrh, implementace a zavedení) a přechází se k inkrementálnímu vývoji IS. Odklon od inženýrských přístupů, které abstrahovaly od vlivu lidského faktoru na efektivnost užití IS. IS byl mnohdy implementován dokonale, ale lidem nesloužil. Podpora vícedimenzionálních metod vývoje IS (např. MDIS). Posun od strukturovaného k objektovému přístupu (metodiky OMT, Booch, OOSE, UP a jeho komerční varianta RUP). Objektový přístup přináší přehlednost pro tvůrce i uživatele. Vznik unifikovaného modelovacího jazyka UML („the three amigos“ Jacobson, Booch, Rumbaugh), umožňující srozumitelný pohled na systém i pro běžného uživatele. Podpora tvorby systémů pomocí CASE nástrojů.

Trendy v organizaci a řízení IS/IT:

V organizacích se využívá *outsourcingu* vývoje aplikací a také provozu HW a SW. Využívá se také *outsourcingu* programátorských týmů, ať z důvodu nedostatečné kapacity organizace, nebo z důvodů spolupráce s kvalitnějšími odborníky. Důvodů k *outsourcingu* je několik. Provoz IS/IT vlastními silami stojí čas a peníze, ubírá podniku energii, kterou by jinak mohl věnovat vlastní oblasti činnosti a zájmu. *Outsourcing* je strategický organizační nástroj. Probíhá přesun odpovědnosti za provoz funkční oblasti (činnosti) podniku na externí specializovanou firmu – poskytovatele, zpravidla včetně zaměstnanců a vlastnictví aktiv. Účelem je především zaměření se na hlavní činnost, dosažení světové úrovně kvality v oblasti, případně úspory nákladů. Aplikuje se u oblastí, které nejsou hlavními činnostmi podniku, tedy nejsou motorem dlouhodobé konkurenceschopnosti podniku.

Informační systémy 1



Obrázek 1.6: Princip outsourcingu

Outsourcing se používá i v oblasti podnikových informačních systémů. Zde je jeho aplikace složitá, protože informační systémy jsou s chodem podniku provázány a je obtížné specifikovat rozhraní (služby) mezi podnikem a poskytovatelem.

Pro vývoj a další rozvoj informačního systému organizace se využívá tzv. *systémové integrace*. Systémová integrace je proces, jehož cílem je vytvoření a další rozvoj komplexního (od ERP přes DMS až k CRM⁴ a taktéž zajištění bezpečnosti IS) a integrovaného informačního systému organizace. Tohoto cíle se dosahuje optimální kombinací a integrací vhodných hardwarových a softwarových komponent a infromatických služeb. Firma nebo organizace, která provádí systémovou integraci se nazývá *systémový integrátor*. Systémový integrátor je zákazníkem na základě smlouvy pověřen komplexním řešením IS zákazníka. Zodpovídá za kompletní a kvalitní řešení integrovaného IS. K zajištění dodávek může systémový integrátor uzavírat smlouvy s jinými dodavateli a řešiteli.

1.3 Základní pojmy

Po úvodu do problematiky IS/IT a rozdělení této oblasti do různých kategorií si uvedeme definice základních pojmů.

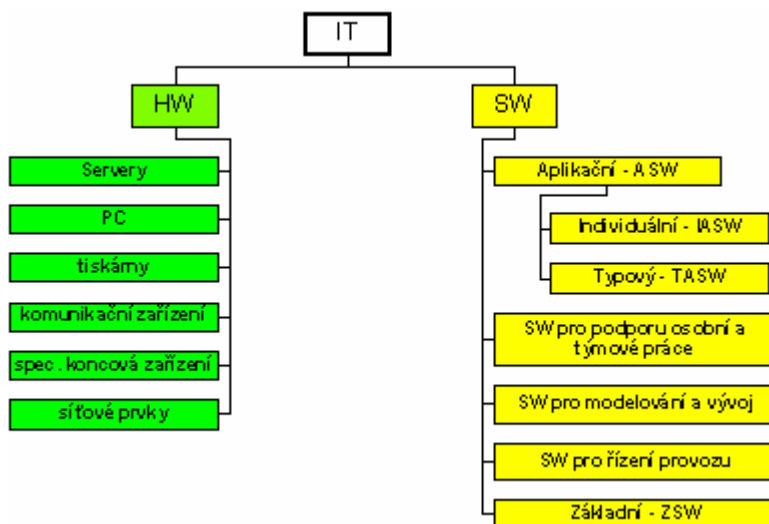
Informatika je multidisciplinární obor, jehož předmětem je tvorba a užití informačních systémů v organizacích a společnostech, a to na bázi moderních informačních technologií. Zmínili jsme spojení multidisciplinární obor. To znamená, že zahrnuje jak technické, tak také ekonomické, sociální, psychologické, právní a další aspekty.

⁴ ERP – Enterprise resource planning – srdce IS, slouží k plánování zdrojů a tudíž vlastnímu řízení podniku.

DMS – Document management system – systém pro zprávu dokumentů.

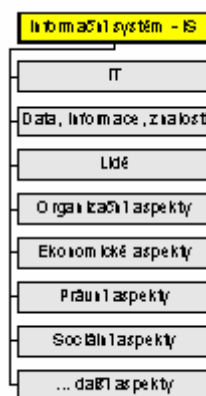
CRM – Customer relationship management – systém sloužící pro podporu styku a komunikace se zákazníky.

Informační systémy 1



Obrázek 1.7: Rozdělení IT

Na obrázku 1.7 vidíme rozdělení informačních technologií (IT) v různých kategoriích, přičemž některé z nich jsme si v textu zmínili již výše. Informační technologie jsou dle definice hardwarové a softwarové prostředky pro sběr, přenos, uchování, zpracování a distribuci informací.



Obrázek 1.8: Aspekty informačního systému

Informační systém organizace je systém informačních technologií, dat a lidí, jehož cílem je efektivní podpora informačních a rozhodovacích procesů na všech úrovních řízení organizace (firmy). Vývoj a provoz IS jsou ovlivňovány řadou aspektů. Obrázek 1.8: Aspekty informačního systému

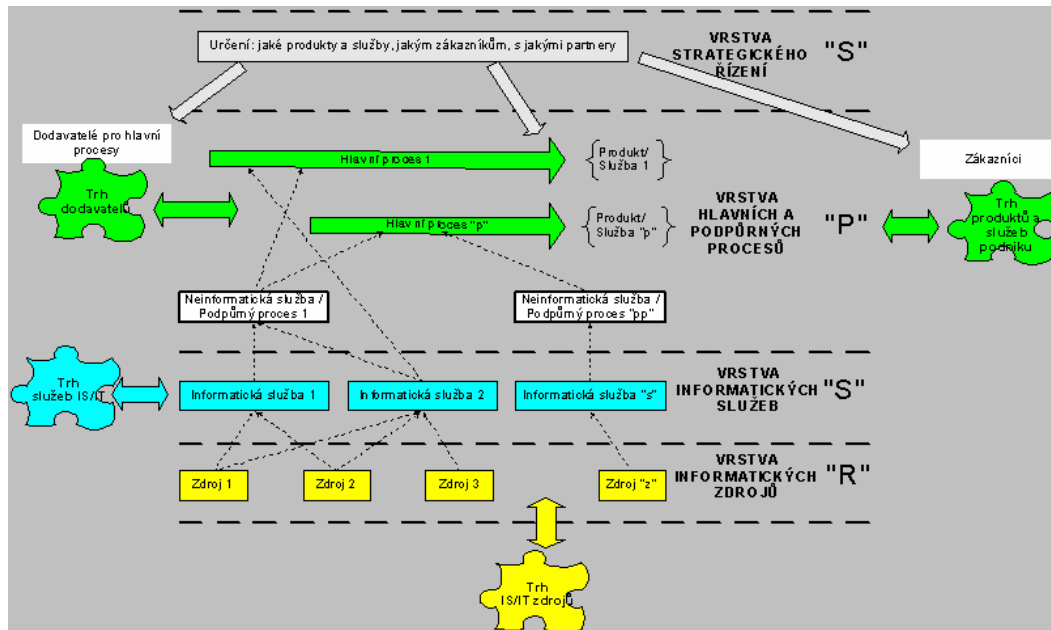
Informační aplikace je relativně samostatná část IS (zahrnující HW, SW a data), vzniklá nebo zabudovaná do IS jedním projektem (např. e-mail, správa majetku, účetnictví).

Informační služba je relativně samostatná část IS viditelná koncovému uživateli a zaměřená na podporu jednoho nebo více procesů organizace.

Zmiňujeme-li pojem aplikace, máme na mysli z čeho je IS tvořen, v případě služby k čemu příslušná část IS v organizaci slouží, kdo je jejím provozovatelem (dodavatelem) a kdo jejím uživatelem (zákazníkem).

Informační systémy 1

Informatický zdroj je komponenta (HW, SW, data-informace-znalost) nutná k tvorbě a provozu informatické aplikace nebo informatické služby.



Obrázek 1.9: Místo IS/IT v řízení organizace

1.4 Projektování IS

Projektování informačního systému je proces tvorby nového IS a jeho uvedení do provozu. Tento proces je řízen a má určitá pravidla a doporučení, kterými se při vývoji řídíme. Takový proces se nazývá *metodikou*. Metodika nám říká kdo, kdy, co a proč má dělat během vývoje a provozu IS.

Metoda nám říká, co je třeba dělat v určité fázi nebo činnosti vývoje a provozu IS. Metody používají různé techniky, každá technika nám říká, jak se dobrat požadovaného výsledku. Nakonec se zmíníme ještě o *nástrojích*, které jsou prostředkem k uskutečnění určité činnosti v procesu vývoje a provozu IS.

Definice říká, že metodika je doporučený souhrn principů, konceptů, dokumentů, metod, technik a nástrojů pro tvůrce informačních systémů, který pokrývá celý životní cyklus informačních systémů. Metodika určuje kdo, kdy, co, jak a proč má dělat během vývoje a provozu IS. Metodika napomáhá k tomu, aby byl informační systém přínosem pro uživatele a celou organizaci. Napomáhá k tomu, aby byly provedeny všechny potřebné činnosti tvorby IS, a to ve správné časové posloupnosti. Metodika také napomáhá k dobré organizaci práce na projektu a jeho dobré a srozumitelné dokumentaci a v neposlední řadě k optimalizaci spotřeby zdrojů při tvorbě a provozu IS.

Aby byla metodika efektivně využitelná, musí splňovat několik základních požadavků. Mezi ně patří:

- Jasná deklarace souboru hodnot, na kterých je metodika založena.
- Určení postupu řešení, aby bylo možné plánovat celý proces vývoje IS/IT.
- Zabývá se všemi faktory, všemi dimenzemi, které ovlivňují tvorbu a provoz IS.
- Určuje priority řešení – co je důležité a kdy.

Informační systémy 1

- Měla by doporučovat metody, techniky a nástroje, které je vhodné využít v určitých fázích vývoje.

Příklad metodik vývoje IS: SSADM, Oracle CASE method, SDM (System development methodology), MMDIS (Multidimensional management and development of information systém). Z objektově orientovaných metodik se můžeme zmínit o OMT (Object modelling techniques), Booch, OOSE nebo UP (Unified process) a RUP (Rational unified process).

Metoda nám říká, co je třeba udělat v určité fázi vývoje či provozu IS. Příkladem metod je YSM, JSD, informační analýza, řízení projektu, výběrové řízení.

Technika určuje jak dělat danou činnost, vymezuje pro činnost přesná pravidla. Technika do značné míry předurčuje způsob uvažování a vyjadřování. Technika je často spjata s konkrétním nástrojem. Příkladem technik může být vedení interview, normalizace dat, strukturované programování, datové modelování nebo prototypování.

Nástroj je vyjadřovacím prostředkem, pomůckou pro techniky a činnosti, často s automatizovanou podporou. Příklad nástrojů: DFD (data flow diagram) – diagram datových toků, ERD (entity relationship diagram) – diagram zachycující entity datové základny a zachycující jejich vztahy, SD (structure diagram) – strukturní diagram, nástroj pro návrh obrazovek, generátor programu.

Existují také programy, které podporují a automatizují tvorbu i několika takových diagramů a na jejich základě generují kód a také mohou podporovat a zjednodušovat tvorbu obrazovkových formulářů. Takové programy se nazývají CASE (computer aided system engineering), jedná se vlastně o souhrn nástrojů pro vývoj IS. CASE nástroje podporují zejména část analýzy, návrhu a implementace IS/IT, ale i dalších činností, souvisejících s vývojem IS/IT.

Aby nevznikly nejasnosti při používáním metod, technik a nástrojů, je třeba zmínit a zdůraznit, že jednotlivé metody jednoznačně nepatří určitým metodikám! Technika neslouží výhradně k podpoře nějaké své metody! Vztahy mezi metodami, technikami a nástroji i jejich náležitosti k metodikám mohou být rozmanité.

Kontrolní otázky:

1. Definiujte informační systém.
2. Vyjmenujte několik příkladů, kde se můžete setkat s informačním systémem?
3. Řekněte jaké přínosy může mít zavedení informačního systému?
4. Jaká je životnost HW, ZSW a aplikačního SW?
5. Co je to projektování IS?

Úkoly k zamyšlení:

Pokuste se podle trendů v oblasti SW a HW a podle Vašich znalostí nových technologií říct, jaké přínosy by mohlo mít zavedení některé z nich do informačního systému?



Informační systémy 1

Korespondenční úkol:

Vytvořte stručný přehled informačních systémů (české i zahraniční), které jsou dostupné na našem trhu. Zaměřte se především na podnikové (enterprise) informační systémy. Vynechejte SW typu evidence účetnictví, které lze také jako IS brát.

Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s rozdělením informačních technologií. Dozvěděli jste se, co je to informační systém a jaké jsou jeho nedílné součásti. Dále jste se seznámili s trendy které panují v oblasti HW, SW a také v oblastech metod vývoje, organizace a řízení IS/IT. V poslední části jste se dozvěděli co je to projektování IS a čeho všeho je při vývoji IS třeba.



2 Architektury IS

V této kapitole se dozvíte:

- Co je to architektura informačního systému a k čemu slouží?
- Jaká je funkce globální a dílčí architektury?
- Jaké jsou typy architektur?

Po jejím prostudování byste měli být schopni:

- Definovat účel architektury informačního systému.
- Popsat rozdíl mezi globální a dílčí architekturou a uvést příklad každé z nich.
- Vyjmenovat a popsat různé typy HW, SW či technologických architektur.

Klíčová slova této kapitoly:

Architektura informačního systému, globální, dílčí architektura, typy architektur, klient/server.

Doba potřebná ke studiu: 3 hodiny



Průvodce studiem

Studium této kapitoly přiblíží architektury informačních systémů. Těchto architektur se využívá v počátečních fázích návrhu informačních systémů a slouží jako základní „plány“ budoucího systému. Na studium této části si vyhradte alespoň 3 hodiny.

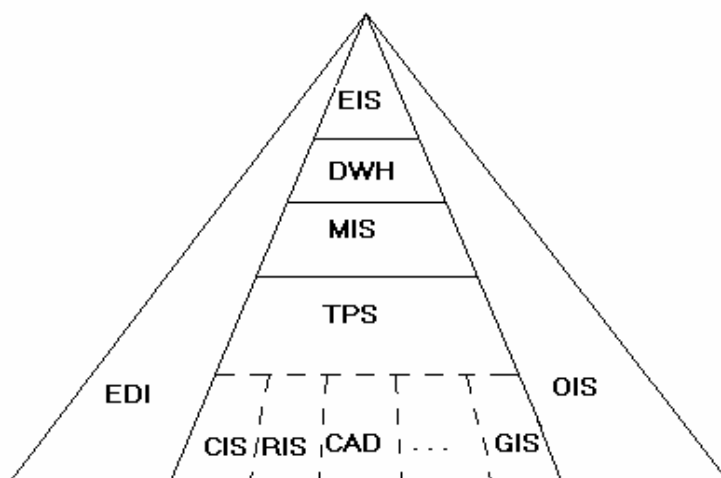
Architektura tvoří klíčový prvek řízení IS, z něhož pak vycházejí detailní analytické i plánovací charakteristiky celého IS. Architektura musí respektovat strategii podniku, podnikové cíle a cíle IS. Do architektury se musí promítat stav a rozvoj produkčních a řídicích aktivit a odpovídajících zdrojů. Podstatou a účelem architektury informačního systému je podpora následujících vlastností: strategická orientace, pokrytí uživatelských požadavků, integrovatelnost, otevřenost, jednoduchost, flexibilita, udržitelnost, efektivní provozuschopnost, viz [2]. Architektura IS vyjadřuje celkovou vizi. Je oproštěna od veškerých detailů, vychází ale z pochopení ekonomických, výrobních a obchodních cílů, které organizace sleduje. Musí být jednoduchá a srozumitelná, je to jakýsi skelet, na který se navěšují další funkce systému.

Neexistuje-li architektura IS, můžeme se setkat s těmito problémy:

- ✓ Nepokryté požadavky na funkce IS, jiné funkce jsou naopak zbytečné (jaksi navíc).
- ✓ Schází potřebné nástroje – tvorba v málo výkonném prostředí, bez CASE, bez specialistů na tvorbu IS, na počítačové sítě, na databáze, ...
- ✓ Časté přestavby z důvodů množících se požadavků uživatelů.
- ✓ Draze nakupený SW nepoužitelný z důvodů kompatibility SW nebo HW a nepoužití standardů (rozdílné uživatelské rozhraní aplikací, různé databázové systémy, ...) ⇒ poruchy, údržba bez řádné dokumentace

Informační systémy 1

Architektura musí být postavena tak, aby respektovala dynamiku změn v procesech a zdrojích a promítat je do navazujících aktivit řízení informačního systému.



Obrázek 2.1: Příklad architektury podnikového systému

Popis jednotlivých částí:

EIS (Executive IS) – podporuje vrcholové řízení organizace (strategie podniku, finanční řízení).

DWH (Data warehouse) – datový sklad, podpora řízení na základě analýz rozsáhlých dat.

MIS (Management IS) – podpora taktické a operativní úrovně řízení (účetnictví, nákup, prodej, sklad, ...).

TPS (Transaction processing system) – bezprostředně spojený s typem provozu v rámci dané organizace (systémy bezprostředně podporující dílenské, skladové, transportní operace výrobních podniků, rezervační systémy dopravních společností, zákaznické systémy energetických společností).

CIS (Customer IS) – zajišťuje bezprostřední styk se zákazníkem (odečty spotřeby energie, fakturaci na zákazníka, ...).

RIS (Reservation IS) – rezervační systémy v dopravních organizacích, cestovních kancelářích.

GIS (Geographic IS) – podpora kreslení a vyhodnocování map, tvorba územních modelů.

CAD (Computer aided design) – konstrukční a návrhářské práce v průmyslu, počítačová podpora návrhu výrobku.

CAM (Computer aided manufacturing) – automatizovaná podpora řízení výrobních provozů.

OIS (Office IS) – podpora rutinních kancelářských prací (elektronická pošta, správa a zpracování dokumentů).

EDI (Electronic data interchange) – podporuje elektronickou výměnu dat mezi obchodními partnery, bankami, ústavy, apod.

Jaké jsou odlišnosti mezi částmi EIS, TPS a MIS? EIS jsou zaměřeny na delší časový úsek, jak do minulosti, tak do budoucnosti. Pracují s historickými daty, lze tudíž vysledovat vývojové tendence. EIS uchovávají údaje o stejném objektu, vzniklé v různých časech (možnost hodnocení kvality). Často jsou založeny na technologii Data Warehouse. Naproti tomu TPS a MIS udržují

Informační systémy 1

data vypovídající o právě aktuálním stavu interních a externích podnikových procesů. Historická data jsou udržována pouze mají-li vztah k současnosti (např. dokud je zakázka v garanční době, jsou data zakázky uložena v MIS).

Architektura je také schéma (graficky vyjádřená představa) zohledňující všechny dimenze návrhu informačního systému. Architektura tvoří klíčový prvek řízení IS, z něhož pak vycházejí detailní analytické i plánovací charakteristiky celého IS.

V architektuře informačního systému existují tři vrstvy. Jde o vrstvu prostředí, aplikační a technologickou. *Vrstva prostředí* reprezentuje ekonomické prostředí, legislativu, organizační strukturu, personální kapacity a jejich kvalifikace, zkušenosti v IT a motivaci pro IT. *Vrstva aplikační* pokrývá provozované a řešené projekty, jejich dokumentace, funkční a datové specifikace, organizační pravidla jejich řešení a provozu, aplikační SW. Konečně *vrstva technologická* pokrývá návrh a provoz počítačových sítí, vymezení jednotlivých komponent IT, což představuje základní software, technické prostředky včetně jejich vazeb a vnitřní struktury.

2.1 Globální architektura

Globální architektura je hrubý návrh celého IS/IT. Je to vize budoucího stavu. Zachycuje jednotlivé komponenty IS/IT a jejich vzájemné vazby. Globální architektura je složena z tzv. bloků. *Blok* je množina informačních služeb, funkcí, které slouží k podpoře podnikových procesů (jednoho nebo více). Jsou to vlastně hlavní úlohy odpovídající optimalizovanému uspořádání procesů a zdrojů. Můžeme také říci, že jsou to množiny pro různé uživatelské skupiny – partneři, zákazníci, zaměstnanci, veřejnost, apod. Příkladem těchto bloků jsou EIS, DWH, MIS, TPS, ... (viz výše).

2.2 Dílčí architektury IS

Jedná se o detailní návrh IS z hlediska různých dimenzí. Určení obsahu těchto dimenzí IS/IT:

- Funkční – funkční struktura, náplň jednotlivých funkcí.
- Procesní – vymezení klíčových procesů a vazeb v IS/IT, (kontextový diagram, diagramy toků dat – DFD, síťové diagramy).
- Datová – určení datových objektů a zdrojů v rozlišení na interní a externí zdroje, návrh datových entit, databázových souborů a jejich uložení.
- Softwarová – rozlišení na ASW, ZSW nebo systémový SW.
- Technická – postihuje celý komplex prostředků počítačové a komunikační techniky.
- Organizační – zahrnuje organizační strukturu a vymezení organizačních jednotek.
- Personální – zahrnuje profesní a kvalifikační struktury.

Každá z těchto dimenzí je popsána svými atributy (identifikace, název, klíčové problémy, ...). Součástí modelu řízení IS/IT (v návaznosti na architekturu) je i analýza a plánování všech podstatných vazeb mezi dimenzemi.

2.3 Typy architektur

Hardwarová architektura

Určuje typy, počty a vzájemné vztahy HW komponent – personálních počítačů, serverů, přídatných zařízení (tiskárny, scannery, ...), použitých přenosových cest.

Technologická architektura

Určuje způsob zpracování aplikace nebo jejich funkcí. Dle toho, jakými podněty jsou startovány jednotlivé funkce aplikace, rozlišujeme několik metod zpracování.

Dávkové zpracování – požadavky na zpracování a vstupní data jsou shromážděna v dávce před odstartováním aplikace, ta po svém spuštění zpracuje všechny požadavky najednou.

Interaktivní zpracování – uživatel je v přímém kontaktu s počítačem, jeho požadavky jsou realizovány okamžitě s garantovanou dobou odezvy, jsou realizovány jednou transakcí.

Aplikace řízené událostmi – jsou startovány událostmi, které nastávají v reálném světě (datové, časové, mimořádné).

Aplikace pracující v reálném čase – jsou to aplikace řízené událostmi, jejich doba odezvy však přesně odpovídá podnikovým procesům, které aplikace řídí.

Čím je zpracování řízeno a jakým způsobem je prováděno je jedna stránka, druhá stránka je místo zpracování, propojení a případná kooperace při zpracování.

Decentralizované zpracování – využívá se hlavního počítače, na který jsou napojeny všechny koncové stanice. Veškeré data a programy jsou umístěny na hlavním počítači. Výhodou takového řešení je jednoduchá tvorba, řízení provozu, údržba a upgrade aplikace. Nevýhodou je přetížení hlavního počítače, ten nemůže být specializován, musí vykonávat všechny části algoritmů všech aplikací, včetně zpracování grafického rozhraní pro všechny komunikující uživatele.

Distribuované zpracování – při tomto druhu zpracování se využívá několika navzájem propojených počítačů (serverů), na které jsou napojeny koncové stanice. Každý se serverů bývá obvykle specializovaný na určitý druh úloh. Algoritmus zpracování a data jsou rozděleny na několik částí a umístěny na různých počítačích. Tyto části spolu však vzájemně komunikují – viz klient/server a vícevrstvá architektura. Výhodou je distribuovaný charakter, který více odpovídá charakteru podnikových procesů, přičemž výpadky nemají takový dopad jako při centralizovaném zpracování. Mezi nevýhody patří nutnost zajistit relativně složitou koordinaci zpracování aplikace na různých počítačích. Komplikovaná je také ochrana a zabezpečení aplikace.

Kooperativní zpracování – pracuje na principu kooperace mnoha počítačů z celosvětové počítačové sítě. Jedná se o vyšší formu distribuovaného zpracování, využívá se především u WWW aplikací.

Architektura klient/server:

Aplikace je rozdělena na dva kooperující programy – klienta a server. *Klient* je program, který požaduje provedení určité služby. *Server* danou službu na

Informační systémy 1

požádání poskytuje. Jeden a týž program může jednou vystupovat jako klient, jindy jako server.

Výhody: nižší nároky na server, volba vhodné platformy pro jednotlivé části aplikace, snadnější zabezpečení dat pomocí specializovaného serveru.

Nevýhody: vyšší počáteční investice nutná ke zprovoznění, vyšší problémy s integrací rozličných platform, vyšší nároky na kvalifikaci řešitelů.

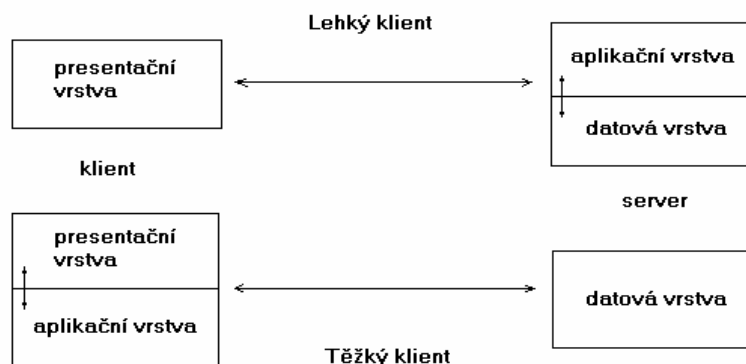
Softwarová architektura

Tato architektura určuje softwarové komponenty IS a jejich vzájemné vazby. Zahrnuje základní SW i aplikační SW. Určuje vnitřní strukturu SW komponent – určení modulů, jejich vazeb a jejich charakteristiky. Mezi charakteristiky modulů patří funkce, které model zajišťuje; vstupní, výstupní a řídicí data modulu; algoritmus popisující transformaci vstupních dat na výstupní a způsob ošetření vyjimečných stavů; vývojové prostředí modulu (programovací jazyk, CASE, ...); provozní prostředí modulu (OS, DB, prezentační systém).

Obvyklá aplikace obsahuje tři základní skupiny funkcí: datové funkce, věcně orientované funkce, které zajišťují vlastní logiku aplikace a komunikační funkce.

Monolitická architektura – všechny tři skupiny funkcí jsou realizovány jedním modulem (programem). Snadné zajištění ochrany funkcí a dat aplikace před neautorizovaným použitím, obtížná údržba a přenositelnost, obtížné zvyšování výpočetních kapacit.

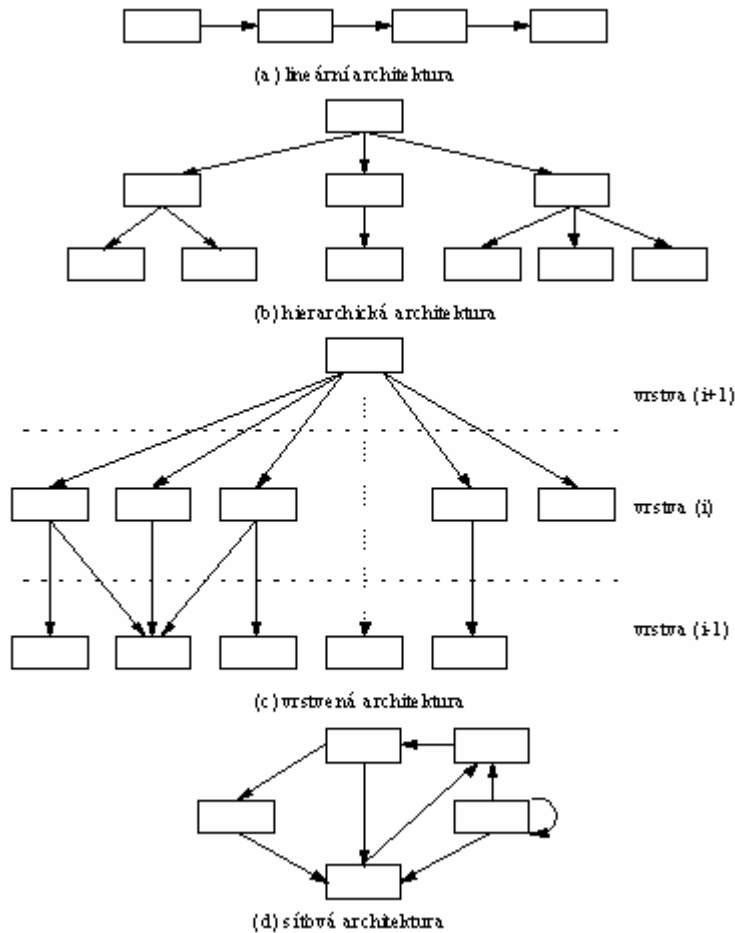
Dvouvrstvá architektura – jedna ze skupin funkcí je oddělena od ostatních. Cílem je oddělení komunikační a datové funkce aplikace. Existují dvě možnosti oddělení – lehký a těžký klient.



Obrázek 2.2: Lehký a těžký klient

Výhodou dvouvrstvé architektury je snadnější zajištění různých forem komunikace s různými koncovými stanicemi, lepší přenositelnost aplikace a snadnější doplnění výpočetních kapacit. Nevýhodou je stále propojení věcně orientovaných funkcí s jednou z dalších skupin funkcí.

Třívrstvá architektura – všechny tři skupiny funkcí jsou odděleny do samostatných programů, které mezi sebou komunikují jako klient/server. Tento způsob je ideální pro tvorbu otevřených, distribuovaných a flexibilních informačních systémů, které lze pružně přizpůsobovat změnám. Výhodou je, že lze každou ze skupin funkcí udržovat a rozvíjet zcela samostatně. Navíc pro každou ze skupin lze zvolit nejvýhodnější vývojové a provozní prostředí. Příklad třívrstvé architektury můžete vidět na obrázku 16.1.



Obrázek 2.3: SW architektury

Softwarová architektura se také týká propojení jednotlivých částí. O jaké typy tedy může jít v tomto případě? Architektura může být lineární, hierarchická, vrstvená nebo síťová. Z těchto čtyř druhů jsou univerzálně použitelné jen vrstvená a síťová architektura, zbývající dvě (lineární a hierarchická) lze použít pouze pro specifické aplikace. Síťová architektura je preferována, jestliže preferujeme nízké náklady provozu před nízkými náklady na tvorbu, údržbu a užití. V ostatních případech je vhodnější užit architekturu vrstvenou.

Kontrolní otázky:

1. Popište k čemu slouží architektura informačního systému.
2. Jaký je rozdíl mezi globální a dílčí архитектурou?
3. Vyjmenujte alespoň 3 typy SW architektury.

Úkoly k zamyšlení:

- Kde se můžete při běžné práci s počítačem a Internetem setkat s tenkým klientem?
- Běžná klient/server architektura má dvě vrstvy. Pokuste se vymyslet, co by mohlo být třetí, popřípadě čtvrtou vrstvou této architektury.



Informační systémy 1



Korespondenční úkol:

Pokuste se vypracovat návrh globální architektury finančního podniku.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s architekturou IS. Dozvěděli jste se k čemu slouží globální a také dílčí architektury. Seznámili jste se s různými typy architektur jako je hardwarová, softwarová či technologická.

3 Vrstvená architektura SW produktů

V této kapitole se dozvíte:

- Co je to vrstvená architektura SW?
- Jaký je koncepční model vrstvené architektury?
- Jaké je rozdělení a jaké jsou typy vrstev?
- Jak probíhá návrh a implementace systému s vrstvenou architekturou?
- Jaké jsou postupy při návrhu metodou shora dolů a zdola nahoru?

Po jejím prostudování byste měli být schopni:

- Definovat vrstvenou architekturu softwarového systému.
- Jaké jsou funkce jednotlivých vrstev.
- Popsat k čemu slouží agregátová a filtrující vrstva.
- Co jsou a jak se dělí uživatelské vrstvy.
- Popsat návrh systému s vrstvenou architekturou.
- Popsat postup, výhody a nevýhody návrhu shora dolů a zdola nahoru.

Klíčová slova této kapitoly:

Vrstva, uživatelská, filtrující, agregátová vrstva, postup shora dolů, postup zdola nahoru.

Doba potřebná ke studiu: 6 hodin

Průvodce studiem

Tato kapitola přiblíží vývoj softwarového systému pomocí vrstvené architektury. Na studium této části si vyhradte alespoň 6 hodin. Doporučujeme studovat s přestávkami vždy po pochopení jednotlivých podkapitol. Po celkovém prostudování a vyřešení všech příkladů doporučujeme dát si pauzu, třeba 1 den, a pak se pusťte do vypracování korespondenčních úkolů.



Podkladem pro tuto kapitolu byla publikace [6]. Vrstvená architektura je jednou z variant řešení SW architektury informačního systému. Nyní si připomeneme pojem SW architektura.

Softwarová architektura určuje softwarové komponenty informačního systému a jejich vazby. Zahrnuje jak základní software ZSW, tak i aplikační software ASW. Dále SW architektura definuje vnitřní strukturu SW komponent, jejich moduly a vazby.

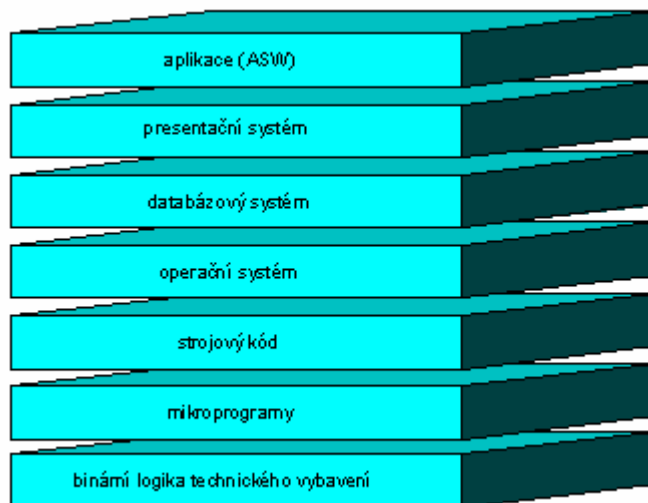
Charakteristika modulů:

- funkce, které modul zajišťuje,
- vstupní, výstupní a řídicí data modulu,
- algoritmus, který předepisuje způsob transformace vstupních dat na výstupní a způsob ošetření mimořádných stavů,
- vývojové prostředí modulu (programovací jazyk, CASE atd.),
- provozní prostředí modulu (operační systém, databázový systém, prezentační systém),

Typy softwarových architektur jsou lineární, hierarchická, vrstvená a síťová. Univerzálně použitelné jsou pouze vrstvená a síťová architektura. Lineární a

Informační systémy 1

hierarchická architektura se dá použít pouze pro specifické aplikace. Síťová architektura je použita v případech, kdy musíme preferovat nízké náklady provozu před nízkými náklady tvorby, údržby a užití (v ostatních případech je vhodnější vrstvená architektura).



Obrázek 3.1: Vrstvená architektura

Základním cílem vrstvené architektury – VA – je minimalizace nákladů na tvorbu, údržbu a užití SW systému. VA si klade za cíl minimalizaci doby potřebné na vytvoření a zavedení systému, ne však optimalizaci z hlediska nároků na kapacity počítače. Těchto cílů VA dosahuje využitím výhod strukturovaného přístupu (rozdělení problému do menších částí), umožněním paralelní práce několika týmů, použitím typových vrstev k řešení, omezením počtu duplicit, apod. Funkce systému jsou uspořádány do několika vrstev s tím, že funkce vyšší vrstvy mohou využívat jen funkce vrstev podřízených.

3.1 Definice a koncepční model vrstvené architektury

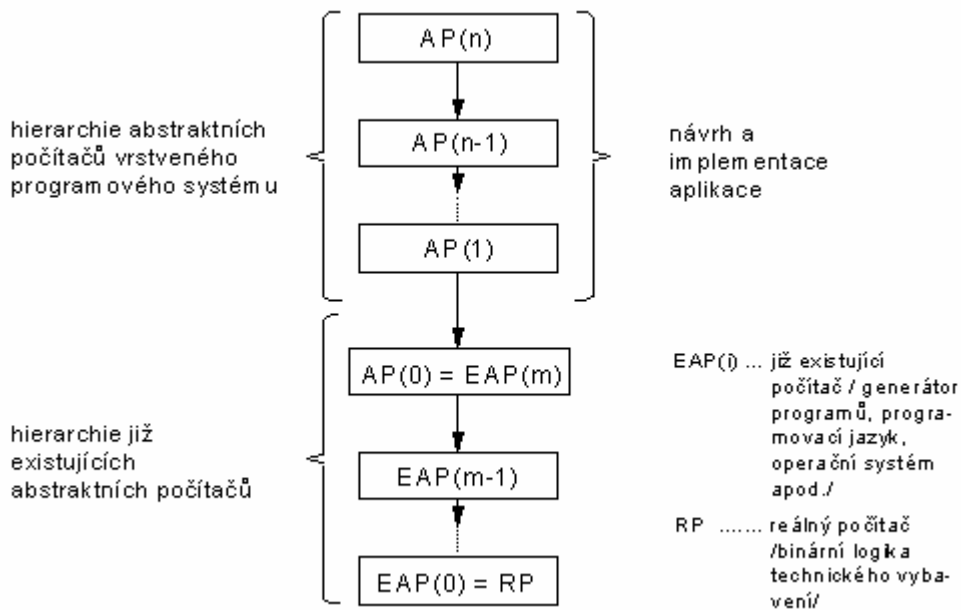
Základním principem VA je využití abstrakce. Ta je využita opakovaně tak, že na každé nižší úrovni abstrakce řešíme zadaný problém na základě vyšší (podrobnější) rozlišovací úrovně.

Vrstvu vrstveného SW systému definujeme jako virtuální (abstraktní) počítač (AP), který je tvořený programem realizujícím funkce dané vrstvy. Primární funkce realizujeme programem abstraktního počítače nižší úrovně. Vznikne nám tedy hierarchie abstraktních počítačů AP_0, AP_1, \dots, AP_n a jim příslušných programů P_0, P_1, \dots, P_n . AP_0 je nejprimitivnější počítač, může být reprezentovaný například kompilátorem vyššího programovacího jazyka.

Postup konstrukce SW systému s vrstvenou architekturou je postupem shora dolů. Prvním krokem tohoto postupu je vytvoření AP_n – nejvyšší úrovně na základě funkční a datové analýzy aplikační oblasti, který je s minimem počtu funkcí schopen nalézt řešení zadané úlohy. V každém dalším kroku sestrojíme AP_n nižší úrovně zjmeněním funkcí předcházejícího AP_{n-1} . Postup končí, jakmile se funkce abstraktního počítače (AP_1) podaří vyjádřit pomocí funkcí některého již existujícího abstraktního počítače (AP_0).

Informační systémy 1

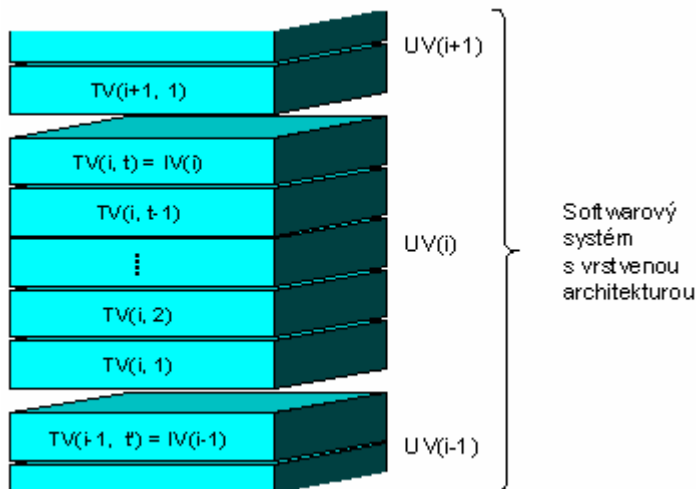
Podstata činnosti každého AP_i spočívá v překladu svých funkcí do funkcí AP_{i-1} a tím také překlady funkcí AP_{i-1} , přičemž se snažíme minimalizovat počet potřebných kroků.



Obrázek 3.2: Hierarchie abstraktních počítačů (vrstev) aplikace

Uživatelská, technologická a interfacová vrstva

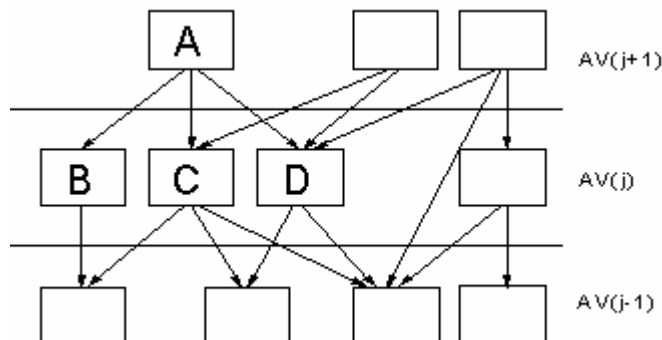
Jako uživatelskou (aplikační) vrstvu (UV) nazýváme množinu vrstev, vytvořenou jedním tvůrcem pro jiné uživatele. Jedná se o množinu vrstev, ze kterých je složena aplikace, množina je vytvořená pro určitý typ uživatele. Technologická vrstva (TV), je vrstva UV, která je “viditelná” pouze tvůrcům této vrstvy. Pouze nejvyšší technologická vrstva v rámci jedné uživatelské vrstvy je zpřístupněná nadřízenému uživateli, tuto tedy nazýváme interfacová vrstva (IV). Jaké jsou mezi těmito vrstvami vztahy, je zachyceno na následujícím obrázku 3.3. Tato struktura je zjednodušená, předpokládá že pod každou uživatelskou vrstvou je právě jedna uživatelská vrstva nižší úrovně. Zjednodušení je zavedeno z důvodu vysvětlení základních pojmů.



Obrázek 3.3: Typy vrstev softwarového systému s vrstvenou architekturou

Agregátová vrstva (AV)

Jednotlivé technologické vrstvy (v rámci jedné uživatelské vrstvy) se mohou lišit svým posláním. První typ technologických vrstev vytváří mocnější agregované funkce (stavební kameny vyššího řádu) z funkcí elementárnějších. Tyto vrstvy nazýváme agregátovými vrstvami.



Obrázek 3.4: Agregátové vrstvy

Na obrázku 3.4 jsou zachyceny agregátové vrstvy. Funkce A agregátové vrstvy AV_{j+1} vznikla účelovým uspořádáním funkcí B, C, D vrstvy AV_j . Uživatel vrstvy AV_{j+1} používá funkci A a o existenci funkcí B, C, D nemusí vědět.

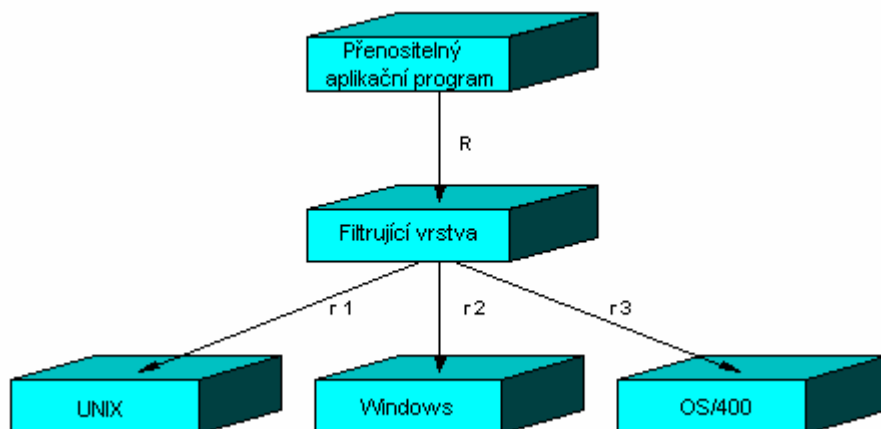
Cílem agregací je vytvořit množiny funkcí, s jejichž pomocí je možné daný problém vyřešit nejrychleji a nejefektivněji. Agregace nižších funkcí do vyšších přináší několik důsledků:

- na vyšší vrstvě se redukuje množina úloh, které lze pomocí funkcí vyšší vrstvy řešit,
- na vyšší vrstvě se redukuje počet možných cest řešení úloh realizovatelných jak funkcemi vyšší vrstvy, tak funkcemi nižší vrstvy,
- úlohy, na jejichž řešení je vyšší vrstva orientovaná se na této vrstvě řeší rychleji a efektivněji.

Redukce množiny řešitelných úloh vyšší agregátovou vrstvou neznamená, že vyšší vrstva musí mít menší počet funkcí. Redukce znamená, že na vyšší vrstvě nelze realizovat ty úlohy, které jsou z hlediska cíle vrstvy irelevantní. Příklad agregátových vrstev z jiného než počítačového světa může být člověk. Stavební kameny člověka na jednotlivých vrstvách hierarchie jsou: částice, atom, molekula, buňka, tkáň, orgán a člověk. Zřejmé je, že stavbu člověka popíšeme snadněji a přehledněji pomocí orgánů než pomocí buněk. Pomocí orgánů ale nepopíšeme rybu, kdežto pomocí buněk již ano.

Filtrující vrstva

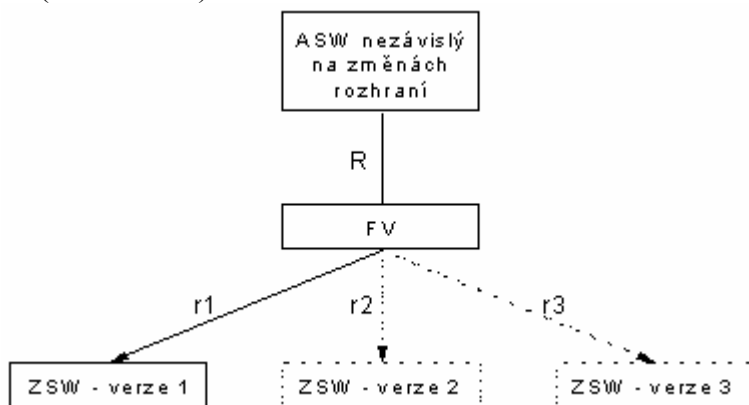
Filtrující vrstva je druhý typ technologické vrstvy, její úlohou je odstínit nepodstatné nebo nežádoucí vlastnosti vrstev na nižší úrovni hierarchie. Jedním z typických použití této vrstvy je odstínění rozdílů v uživatelském rozhraní několika HW nebo SW komponent stejného typu. Filtrující vrstvu lze použít pro zajištění přenositelnosti software (viz obr. 3.5).



Obrázek 3.5: Filtrující vrstva pro vícenásobné rozhraní

Úlohou filtrující vrstvy zakreslené na obrázku je převést tři různá uživatelské rozhraní na jedno standardizované rozhraní R. Filtrující vrstva by se skládala z tří komponent, kdy první konvertuje r1 na R, druhá r2 na R a třetí r3 na R. ASW díky tomu přistupuje k jednomu virtuálnímu operačnímu systému, místo tří různých OS.

Dalším typickým využitím je odstínění změn rozhraní v čase. V tomto případě se filtrující vrstva neskládá ze tří komponent, ale s novou verzí ZSW se celá vymění (obrázek 3.6).



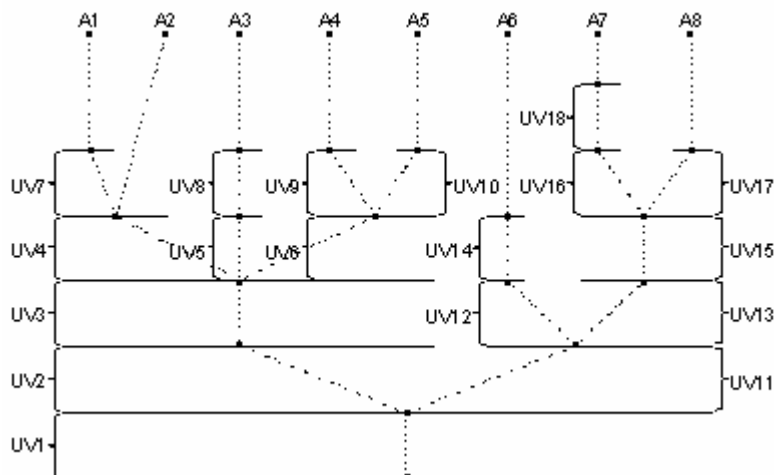
Obrázek 3.6: Filtrující vrstva pro změny rozhraní v čase

Posledním příkladem využití filtrující vrstvy je vytváření virtuálních zařízení, která odstiňují navazujícím vrstvám pro ně nepodstatné charakteristiky těchto zařízení (např. virtuální paměť, virtuální terminál). Virtuální terminál může odstínit fyzický terminál a tím určit rozměry GUI tak, jak nejlépe vyhovují aplikaci.

Úplná struktura vrstev softwarového systému

Jak již bylo řečeno, dosud jsme o vrstvách mluvili zjednodušeně tak, jako by jejich cílem byla jediná aplikace. To v praxi nestačí a tak navrhujeme co největší počet vrstev tak, aby vyhovoval širokému okruhu aplikací. Softwarový systém zachycený na obrázku 3.7 se skládá celkem z osmi různých aplikací, pro které je vytvořeno celkem osmnáct uživatelských vrstev.

Informační systémy 1



Obrázek 3.7: Úplná struktura vrstev softwarového systému

Každé větvení stromu zachycuje promítnutí specifické charakteristiky množiny aplikací do funkcí vrstev. Tak například uživatelská vrstva UV11 pokrývá požadavky aplikací A6, A7, A8, ale UV13 již pokrývá pouze požadavky aplikací A7 a A8. Cílem by mělo být, aby aplikace sdílely maximální počet vrstev.

3.2 Návrh a implementace systému s vrstvenou architekturou

Stejně jako vrstvy plní různé funkce, jde do vrstev rozdělit také tvůrce systému. Můžeme mít tedy vrstvu techniků, mikroprogramátorů, systémových programátorů, aplikačních programátorů a koncových uživatelů. Například systémový programátor využívá práci mikroprogramátora a současně tvoří pracovní nástroje pro aplikačního programátora. Každá z těchto skupin tvůrců má svůj cíl práce, typ problému, který řeší, používané metody práce a také jiný typ pohledu na počítač.

Kdo navrhuje vrstvu

Má vrstvu navrhovat její uživatel nebo její tvůrce? Pro odpověď na tuto otázku si musíme nejdříve rozdělit návrh uživatelské vrstvy do dvou etap:

1. návrh uživatelského rozhraní (návrh funkcí a jazyka vrstvy rozhraní),
2. návrh architektury uživatelské vrstvy (návrh jednotlivých technologických vrstev, jejich rozdělení na vrstvy filtrující a agregátové, specifikace funkcí technologických vrstev).

Z cílů práce, z typu problémů, které uživatel řeší a z jeho pohledu na počítač, je jasné, že se nemůže a dokonce nesmí podílet na návrhu architektury vrstvy, kterou sám používá. Koncový uživatel aplikace bude těžko radit aplikačnímu programátorovi, jako zvolit architekturu či implementační prostředí aplikace. Za návrh architektury uživatelské vrstvy je tedy zodpovědný výhradně tvůrce vrstvy. Při návrhu uživatelského rozhraní je však situace jiná. Východiskem návrhu jsou požadavky uživatele na funkci vrstvy a základní limity jako náklady tvorby, provozu, apod. (otázkou je hranice podílu uživatele).

Informační systémy 1

Výsledkem analýzy uživatelských požadavků musí být jednoduchá koncepce uživatelského rozhraní.

Jak navrhovat uživatelské rozhraní vrstvy

Toto rozhraní je hlavním faktorem, který ovlivňuje náklady a efektivnost práce uživatele vrstvy. Uživatelské rozhraní vrstvy je dáno třemi souvisejícími komponentami. Těmito komponentami jsou datové struktury, operace nad nimi a jazyk. Uživatelské rozhraní vrstvy vytváří pro uživatele prostředí, ve kterém jsou determinovány metody a nástroje řešení uživatelských problémů. Tím rozhraní předurčuje rozsah, kvalitu a náklady řešení daných problémů. Nyní si stručně popíšeme principy návrhu jednotlivých komponent uživatelského rozhraní vrstvy.

Návrh funkcí uživatelského rozhraní (datové struktury, operace vrstvy rozhraní) – vychází z datové a funkční analýzy, pro zajištění pohodlí a efektivnosti práce uživatele je nutné určit několik důležitých bodů. Patří mezi ně určení inicializace (uživatel, datová, časová nebo mimořádná událost), vazba funkcí na procesy v realitě (pořadí provádění funkcí), v jakém reálném čase musí funkce proběhnout, frekvence užití funkce, paralelní nebo sekvenční zpracování dat, apod. Funkce musí být navrženy tak, aby bylo možné minimálním počtem funkcí pokrýt všechny oprávněné požadavky uživatele.

Zde hraje důležitou roli úroveň jazyka vrstvy (tzv. Halsteadova teorie). Tvůrce musí maximálně omezit roli operátorů v jazyku. Na cestě k Halsteadově algoritmu s minimálním objemem se zmenšuje počet operátorů a zvětšuje se počet operandů. V neposlední řadě je třeba znepřístupnit v uživatelském rozhraní ty elementární funkce nižších vrstev, které nejsou nutné pro splnění uživatelských požadavků.

Návrh jazyka uživatelského rozhraní – tento jazyk je prostředek pro zpřístupnění funkcí vrstvy, špatně navržený jazyk může znehodnotit dobrý návrh ostatních komponent. Při tvorbě jazyka respektujeme tyto zásady:

1. Syntax a sémantika jazyka vrstvy musejí být maximálně jednoduché a konzistentní pro všechny zpřístupněné funkce. Nejobvyklejší chyby jsou v přílišné rozsáhlosti jazyka, jazyk vrstvy je založen na jiném národním jazyce, než jaký ovládá uživatel vrstvy, komponenty, které patří do uživatelského rozhraní, nazývají stejné objekty různými jmény.
2. Zprávy poskytované vrstvou musí být jasné a jednoznačné, na úrovni, kde se pohybuje uživatel. Typickou chybou je nepřeložení zprávy nižší úrovně do řeči uživatele, např. „hláška“ blok paměti FA005h – FFF69h není k dispozici.
3. Jazyk uživatelského rozhraní musí odstínit všechny implementační charakteristiky nižších vrstev. Typické chyby jsou: automatický předpoklad znalostí o výpočetní technice u koncového uživatele nebo nezakrytí charakteristik nižších vrstev při vzniku mimořádné události.
4. Jazyk uživatelského rozhraní by měl umožňovat uživateli vytvářet jeho vlastní uživatelské vrstvy (např. makrojazyk).
5. Jazyk by měl obsahovat prostředky pro testování vytvořených programů, důležitá je přesná identifikace místa, kde chyba nastala.

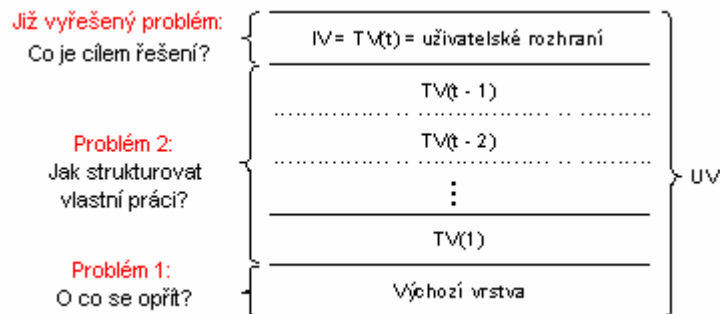
Informační systémy 1

6. Navržení příkazů jazyka s minimalizací situací, kdy jsou některé příkazy zakázané, případně fungují jinak než za standardních podmínek.
7. Vhodná volba mezi člověkem a počítačem řízenou komunikací, z důvodu odstínění uživatele od technických řešení a také samotného algoritmu úloh.
8. Jazyk interaktivního uživatelského rozhraní by měl obsahovat příkazy pro návrat do předcházejících stavů komunikace (příkazy UNDO a REDO). Umožňuje to uživateli při komunikaci experimentovat a v případě chyby tím umožní vrátit se zpět.
9. Uživatelské rozhraní musí být dokumentováno dokonalou hierarchicky uspořádanou uživatelskou příručkou. Stručná verze příručky by také měla být součástí systému.

Jak navrhnout a rozmístit jednotlivé typy vrstev ve struktuře systému

Po návrhu uživatelského rozhraní následuje návrh architektury uživatelské vrstvy. V současné době při návrhu SW systému netvoříme od začátku, ale využíváme již existujících vrstev. Návrh architektury uživatelské vrstvy (obrázek 3.8) má tři kroky:

1. Určení výchozí vrstvy (abstraktního počítače AP), která se stane základnou budované uživatelské vrstvy.
2. Určení struktury uživatelské vrstvy, tj. určení technologických vrstev, které po jednotlivých úrovních abstrakce propojí vrstvu rozhraní s výchozí vrstvou.
3. Specifikace funkcí každé technologické vrstvy.



Obrázek 3.8: První dva kroky návrhu architektury uživatelské vrstvy

Prvním krokem návrhu je tedy určení vhodné výchozí vrstvy. Výchozí vrstva (vývojové prostředí) je množina dosud vytvořených uživatelských vrstev, které budou použity jako základní stavební kameny nově budované uživatelské vrstvy vyšší úrovně. Počet jejích komponent je dán jednak požadovanými variantami provozního prostředí aplikace a jednak záměry architektů aplikace. Čím větší počet komponent, tím obtížnější je zvládnutí výchozí vrstvy tvůrcem.

Kritéria výběru komponenty výchozí vrstvy:

- maximální mocnost funkcí komponenty,
- zvládnutelnost komponenty řešitelským týmem,
- námi neovlivnitelný vývoj komponenty v čase,
- využitelnost komponenty pro tvorbu nadstavbových vrstev,
- spolehlivost komponenty a pořizovací náklady komponenty.

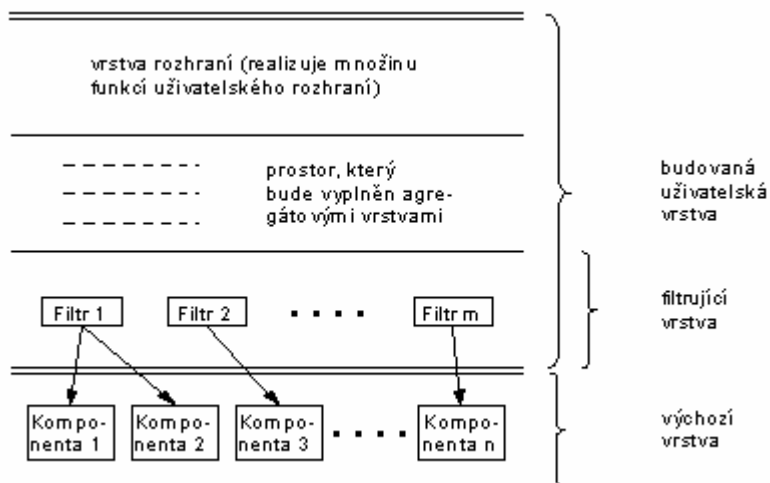
Informační systémy 1

Druhým krokem návrhu architektury je určení struktury uživatelské vrstvy. Hlavním úkolem je určení jednotlivých technologických vrstev tak, aby se minimalizovali náklady na tvorbu a údržbu systému. Musíme určit filtrující a agregátové vrstvy.

Nejdříve se budeme věnovat určení filtrujících vrstev. Při jejich určení je dobré držet se následujících doporučení nebo o nich aspoň uvažovat:

- Filtrující vrstvu je vhodné umístit nad každou komponentou výchozí vrstvy (tj. i nad fyzické zařízení) – cílem je odstínění nepodstatných specifických rysů a umožnění portability vyšších vrstev.
- Filtrující vrstvu musíme umístit všude tam, kde lze v budoucnu očekávat změny v rozhraní, např. vrstva pro komunikaci v jiných jazycích.
- Umístění alternativních filtrujících vrstev s možností vypnutí a zapnutí, např. možnost zpracování pouze textu v dokumentu nebo naopak pouze obrázků.

Potom, co rozmístíme filtrující vrstvy, přichází na řadu určení agregátových vrstev. Jejich úkolem již není nic odstínovat, ale vytvářet z elementárních funkcí (z filtrujících vrstev) agregované funkce. Chceme pomocí nich s co nejmenší námahou vytvořit most mezi filtrujícími vrstvami a určeným uživatelským rozhraním (obrázek 3.9).



Obrázek 3.9: Dosavadní stav návrhu uživatelské vrstvy

U návrhu agregátových vrstev si také zmíníme, jaká lze brát v úvahu doporučení:

- Vrstvy musí odpovídat zvoleným úrovním abstrakce.
- Vzdálenost agregátových vrstev musí být zhruba stejná, kvůli rovnoměrnému rozložení úsilí nutného k překonání vzdálenosti mezi uživatelským rozhraním a výchozí vrstvou. Vhodnou pomůckou pro určení vzdálenosti může být Halsteadova teorie.
- Při návrhu je vhodné mít na mysli i příbuzné aplikace s naší aplikací, díky tomu se mohou tvůrci jiných aplikací napojit na některé naše agregátové vrstvy.

3.3 Umístění a dostupnost funkcí v hierarchii vrstev

V podstatě můžeme říct, že jakákoliv funkce se dá realizovat libovolnou vrstvou z pěti základních vrstev výpočetního systému, kterými jsou

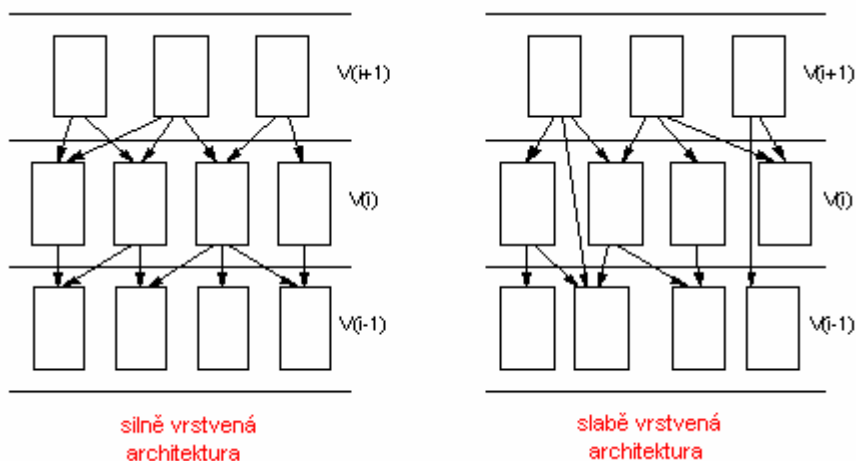
Informační systémy 1

hardware, mikroprogramy, ZSW, TASW, IASW. Víme, že v průběhu vývoje nejen HW, ale i ostatních komponent výpočetního systému, se některé funkce, které byli dříve realizovány SW vybavením, přesunuly do technického vybavení (např. dynamický překlad adres virtuální paměti). Přesuny se uskutečnily i druhým směrem (např. realizace strojového kódu se přesunula z vrstvy technického vybavení do vrstvy mikroprogramů). V principu tedy můžeme umístit každou funkci do kterékoliv z těchto pěti vrstev, nemůžeme ale umístění volit zcela náhodně. Při rozmístění funkcí uvažujeme:

- náklady na realizaci funkce v dané vrstvě, vyšší vrstva – nižší náklady díky mocnějšímu jazyku,
- požadovanou rychlost zpracování funkce, čím vyšší vrstva, tím pomalejší zpracování,
- požadovanou spolehlivost funkce, čím vyšší vrstva, tím nižší spolehlivost, na realizaci se podílí více programů,
- frekvence použití funkce, čím častější použití, tím níže by měla vrstva být (nižší náklady provozu, vyšší spolehlivost),
- frekvence změn funkce, čím častější, tím vyšší vrstva.

Každá funkce se má vyskytovat v hierarchii vrstev pouze jednou. Při umístění funkcí musíme mít na paměti také jejich dostupnost v hierarchii. Jak již bylo řečeno výše, každá funkce může využívat pouze funkcí z nižších vrstev, nemůže tedy využívat funkcí vrstev na stejné nebo vyšší úrovni v hierarchii. S tím souvisí také pojmy silně a slabě vrstvená architektura (viz obrázek 3.10):

Silně vrstvená architektura – každá funkce má dostupné pouze funkce bezprostředně nižší vrstvy.



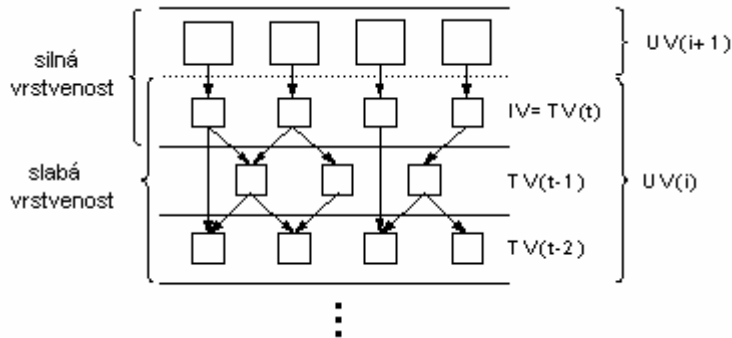
Obrázek 3.10: Silně a slabě vrstvená architektura

Ze struktury silně vrstvené architektury vyplývají její nevýhody. Každá vrstva musí vyšší vrstvě zpřístupňovat všechny potřebné funkce, tedy i ty, které by sama jinak nepotřebovala. V důsledku nutnosti zpřístupnění všech funkcí vyšší vrstvě má silně vrstvená architektura vyšší náklady na tvorbu systému. Další nevýhodou je, že silně vrstvená architektura předpokládá jen minimální změnu množiny funkcí. Přidání nové funkce do vrstvy může znamenat nutnost úpravy několika podřízených vrstev, což znamená vyšší náklady na údržbu. Silně vrstvená architektura má také vyšší provozní náklady z důvodu nutnosti

Informační systémy 1

mnohonásobného překladu funkcí z jazyka jedné vrstvy do druhé před samotným spuštěním funkce.

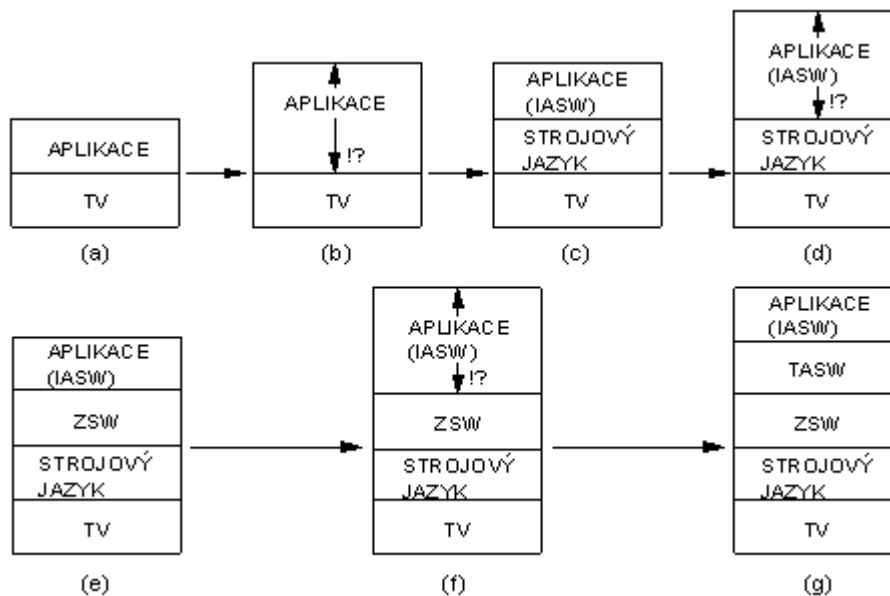
Z výše uvedených skutečností, výhod a nevýhod je vhodné volit kombinaci obou typů vrstvené architektury. Silně vrstvená architektura by se měla použít pro vrstvu rozhraní (zpřístupňuje všechny funkce) a slabě vrstvená architektura by se měla použít pro ostatní technologické vrstvy uživatelské vrstvy.



Obrázek 3.11: Kombinace silně a slabě vrstvené architektury

Co ovlivňuje stabilitu struktury vrstveného systému?

Zde si zmíníme, jaké jsou zákonitosti vývoje struktury vrstveného systému. Uvedené úvahy by měli sloužit k lepšímu návrhu architektury. Na obrázku 3.12 je zachyceno schéma dosavadního vývoje aplikací počítačů a základních vrstev výpočetního systému.



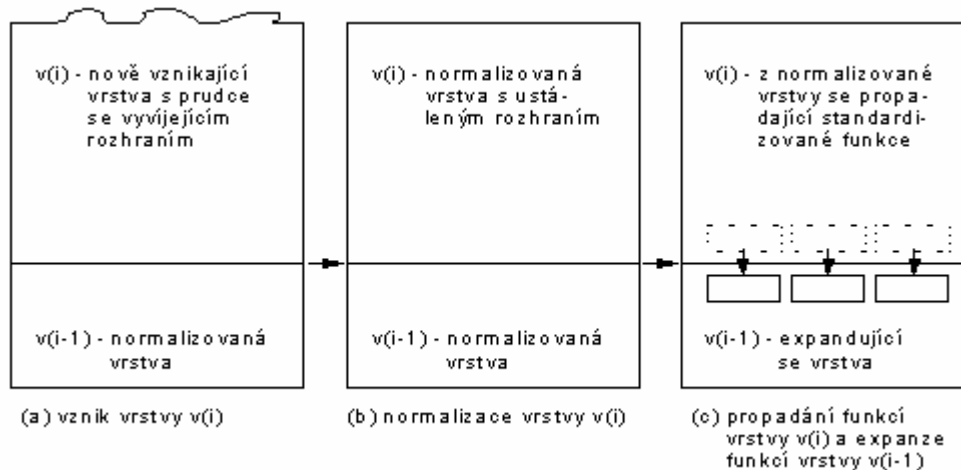
Obrázek 3.12: Vztah expanze aplikací a vývoje vrstev výpočetního systému

Případ (a) zachycuje ranné aplikace v počátcích výpočetní techniky, ovládání počítače se dělo manuálně, propojováním vodičů či nastavováním kolíčků. Aplikace se rozšiřovaly i přes komplikovanost zadávání příkazů, apod. Situace (b) ukazuje stádium, kdy došlo k rozporu mezi požadavky aplikace a schopnostmi člověka tyto požadavky řešit pomocí dosavadních nástrojů. Vznik strojového jazyka (c) tuto bariéru prolomil. Zanedlouho se požadavky aplikací opět dostaly do rozporu s dosavadními nástroji řešení – situace(d). Vývoj

Informační systémy 1

pokračoval dál až k současnému TASW. Z tohoto vývoje můžeme zobecnit tři charakteristická stádia vývoje vrstvy:

1. *Vznik vrstvy* – podmínky jsou vytvořeny tehdy, když nižší vrstva, na které má být nová vrstva postavena, je ustálená (tj. je ve stadiu normalizace).
2. *Normalizace vrstvy* – relativní ustálení funkcí vrstvy a uživatelského rozhraní vrstvy, ze zvláštního krystalizuje obecné. Příkladem krystalizace obecných funkcí v oblasti ZSW, konkrétně operačních systémů, jsou řízení procesoru a operační paměti, I/O operací, spotřeby zdrojů, apod.
3. *Propadání a expanze vrstvy* – souvisí s prohlubujícím se poznáním problémů, které se řeší na dané vrstvě, možnost tvorby standardizovaného řešení, které je možno předat do kompetence nižší vrstvy, hovoříme o propadání funkce z vyšší do nižší vrstvy. Příkladem může být přesun třídících algoritmů z IASW do ZSW na počátku 70. let. Z bližší historie lze uvést příklad procesoru Pentium MMX od společnosti Intel, který byl rozšířen o desítky nových instrukcí pro rychlejší zpracování multimediálních aplikací.



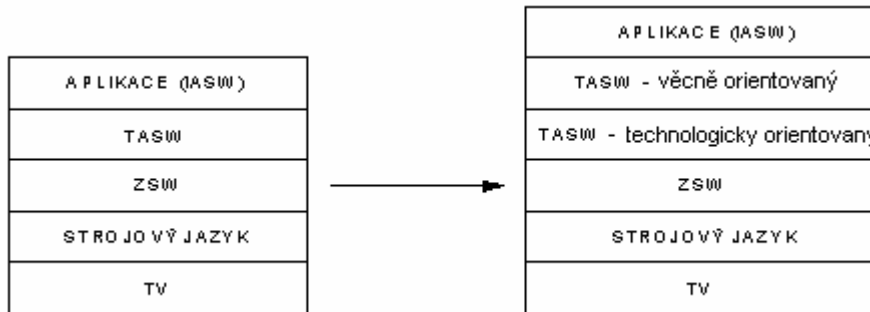
Obrázek 3.13: Stádia vývoje vrstev

Vývoj poznání a automatizace určité oblasti prochází obvykle těmito kroky:

1. Problémy oblasti reality řešeny intuitivně bez předem připraveného plánu.
2. Postupné poznání vede k vytvoření různých metodik řešení předmětných problémů. Aplikace metodiky vyžaduje tvůrčí přístup.
3. Další prohloubení poznání předmětné oblasti a zkušenosti s metodikami vedou k vytvoření konsensuálního postupu řešení. Přibližně vymezíme kroky a jejich činnosti včetně návazností.
4. Konsensuální postup se mění v algoritmus řešení, jedná se už o detailně určený postup, není třeba kreativního přístupu. Algoritmus se stává součástí IASW.
5. Postupná standardizace původně odlišných algoritmů řešení téhož, resp. podobných problémů vede k přesunu (k propadnutí) algoritmu do nižší vrstvy.
6. Ty algoritmy, které se nejvíce standardizovaly a na jejichž rychlosti závisí výkonnost výpočetního systému se propadají až do hardwaru.

Informační systémy 1

Pokud shrneme to, co bylo napsáno výše, lze říci že pro třetí stádium vývoje vrstvy je typické propadání standardizovaných funkcí této vrstvy do vrstvy nižší. To má za následek expanzi počtu funkcí nižší vrstvy, zvýšení mocnosti jazyka nižší vrstvy, ulehčení a standardizaci řešení úloh na vyšší vrstvě a také omezení variety úloh na vyšší vrstvě.



Obrázek 3.14: Rozdělení vrstvy TASW

Propadneme-li na nižší úroveň příliš mnoho funkcí, může se tato vrstva stát pro její tvůrce nezvládnutelnou. Na obrázku 3.14 můžeme vidět příklad rozdělení vrstvy TASW na vrstvu věcně orientovanou (SAP R/3, Oracle Financials) a na vrstvu technologicky orientovanou (aplikace OIS). Struktura vrstev a rozmístění funkcí do jednotlivých vrstev nejsou napevno dané jednou provždy. Řešení, které je optimální dnes se může stát bariérou rozvoje aplikací za pár let – viz historický vývoj na obrázku 3.12.

3.4 Jaký zvolit při tvorbě systému postup?

Při tvorbě systému využívá vrstvená architektura vhodnou kombinaci postupů shora dolů (top down) a zdola nahoru (bottom top).

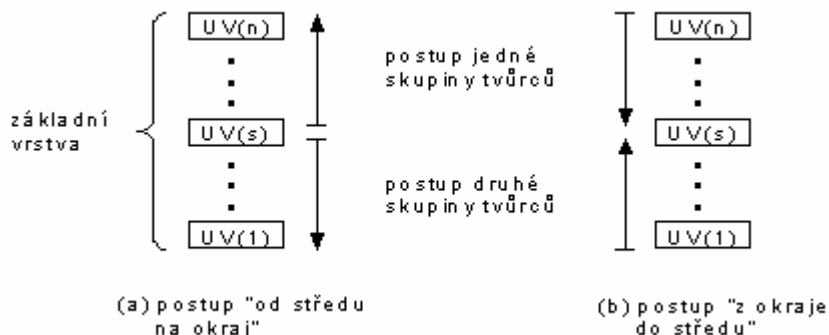
Při postupu shora dolů dělíme problémy na řadu menších až k dosažení množiny problémů, z nichž každý je přiměřeně rozsáhlý a tím snadněji řešitelný. Tento postup vychází primárně z požadavků uživatele a klade důraz na to, co se řeší. Výhodou tohoto řešení je záruka, že v každém kroku řešení je respektován globální cíl řešení, každá navržená funkce je nutnou podmínkou pro vyřešení zadaného problému. Nevýhodou tohoto postupu je nejistota, že navržená funkce je realizovatelná v rámci omezení daných technickým a programovým vybavením. To může být až příčinou ztroskotání projektu.

Při postupu zdola nahoru skládáme elementární operace do vyšších úrovní až k vytvoření prostředku, se kterým je řešení snadnější. Tento postup klade důraz na to, s čím se řeší. Výhodou tohoto postupu je jistota v každém kroku řešení, že navržená funkce je realizovatelná v rámci omezení daných technickým a programovým vybavením. Další výhodou postupu zdola nahoru je lepší využití pro tvorbu systémů, které mají splnit požadavky širokého spektra různých aplikací, umožňuje také rychleji dokončit a předat do užití částečné řešení. Nevýhodou pak může být vznik funkcí, které nejsou v souladu s řešením zadaného problému, tj. funkce neodpovídají potřebám vyšších vrstev, tím se komplikuje uživatelské rozhraní vrstev.

Z uvedených postupů tvorby, jejich výhod a nevýhod můžeme pro vrstvenou architekturu vyvodit následující závěry. Pro postupné vytváření základních

Informační systémy 1

vrstev je použitelný pouze postup zdola nahoru (pro vznik nové vrstvy je nutná existence normalizované nižší vrstvy). Při tvorbě uživatelských vrstev v rámci základních vrstev je možné použít obou postupů, či jejich modifikací s tím, že bude přesně definováno uživatelské rozhraní středové vrstvy (viz obrázek 3.15).



Obrázek 3.15: Modifikace postupů shora dolů a zdola nahoru

Postup shora dolů bychom měli použít, máme-li nejasnosti v oblasti, co řešit nebo jaká je podstata řešeného problému. Postup zdola nahoru přichází v úvahu v případě nejasností v oblasti, jak řešit a nebo na čem řešit.

V jednotlivých etapách tvorby uživatelské vrstvy je vhodné používat kombinace obou postupů. V první etapě (návrh uživatelského rozhraní vrstvy) dominuje postup shora dolů, tím zajistíme splnění všech uživatelských požadavků. V druhé etapě (návrh uživatelské architektury) dominuje v prvních krocích zejména postup zdola nahoru (převládá zde totiž problém na čem řešit) a v následujících krocích je možné použít kteréhokoli z obou postupů. Ve třetí etapě (implementace funkcí technologické vrstvy) je vhodnější postup zdola nahoru, umožňuje to snadnější testování programů a postupné testování celého SW systému.



Kontrolní otázky:

1. Popište k čemu slouží agregační a filtrační vrstva.
2. Jaká jsou stádia vývoje vrstvy?
3. Jaký je rozdíl mezi slabě a silně vrstvenou architekturou?
4. Jaký je rozdíl mezi vývojem shora dolů a zdola nahoru?



Úkoly k zamyšlení:

Který z postupů (zdola nahoru nebo shora dolů) je podle Vás využívám při návrhu a implementaci kancelářských balíků typu Office, Open Office a proč?



Korespondenční úkol:

Pokuste se navrhnout několik vrstev a jejich základní funkce pro funkce kalkulatoru.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s vrstvenou architekturou software. Dozvěděli jste se, jak se navrhují a implementují jednotlivé vrstvy. Poznali jste rozdělení vrstev a jejich úkoly.

4 Životní cyklus vývoje IS

V této kapitole se dozvíte:

- Co je to životní cyklus vývoje IS?
- Co je to metodika vývoje a k čemu slouží?
- Z jakých etap se skládá životní cyklus vývoje IS a co je jejich úkolem?
- Kdo metodiky využívá?
- Jaké jsou dimenze tvorby IS?
- Jaké jsou základní přístupy vývoje SW?

Po jejím prostudování byste měli být schopni:

- Definovat životní cyklus vývoje SW.
- Rozdělit vývoj IS do etap, jednotlivé etapy popsat včetně jejich vstupů a výstupů.
- Popsat různé dimenze vývoje IS.
- Popsat základní přístupy vývoje IS.

Klíčová slova této kapitoly:

Životní cyklus, metodika, fáze vývoje, dimenze, přístupy vývoje, vodopádový přístup, přírůstkový přístup.

Doba potřebná ke studiu: 4 hodiny

Průvodce studiem

Tato kapitola přiblíží proces vzniku a existence softwarového systému. Tento proces (životní cyklus) je řízen metodikou. Je důležité pochopit, co za činnosti se v jednotlivých fázích vývoje provádí.

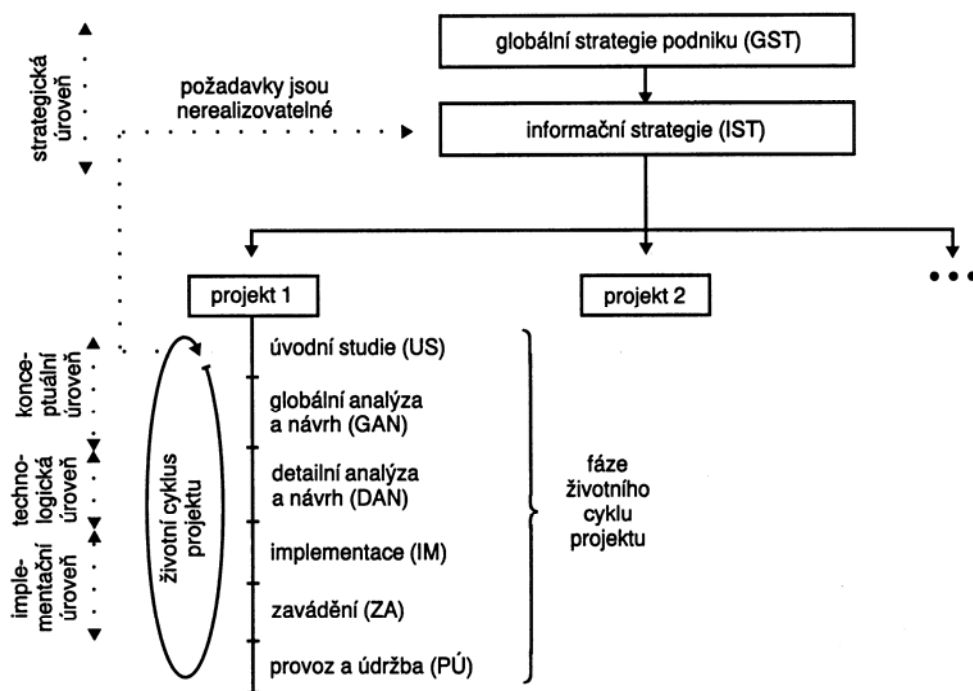
Na studium této části si vyhraďte alespoň 4 hodiny.



Metodika vývoje informačního systému zahrnuje celkový pohled na proces vzniku a existence informačního systému – tj. životní cyklus systému. Dále zahrnuje a rozpracovává všechny dimenze všech fází životního cyklu a také zahrnuje příslušné metody, techniky a nástroje, jež se využívají v určitých fázích vývoje IS. Metodika tvorby IS je souhrnem etap, přístupů, zásad, postupů, pravidel, dokumentů, řízení, metod, technik a nástrojů, který pokrývá celý životní cyklus IS. Metodiky vývoje IS nejsou určeny jen pracovníkům zabývajícím se tvorbou a provozem IS, jsou určeny také pracovníkům a vedení organizace, pro kterou je systém vyvíjen či zaváděn. Metodiky totiž určují kdo, kdy, co a proč má dělat během vývoje a provozu IS, tudíž nejen dodavatel, ale i zákazník-odběratel.

Základem pro metodiky je představa o struktuře a náplni životního cyklu informačního systému. Příklad životního cyklu můžeme vidět na obrázku 4.1. K jednotlivým etapám životního cyklu se potom váží jednotlivé dokumenty, cíle, metody, techniky, nástroje i způsob řízení.

Informační systémy 1



Obrázek 4.1: Životní cyklus projektu

Základními (klíčovými) body postupu jsou začátky a konce etap, fází a kroků vývoje IS. Metodika by měla u každé etapy stanovit:

- Cíl (co je výsledkem a proč má být provedena), účel a obsah etapy (role etapy a shrnutí prováděných činností).
- Předpoklady zahájení a ukončení etapy.
- Klíčové dokumenty etapy – nejen vytvořené, ale i upravené, změněné.
- Kritické faktory etapy – faktory, které mohou způsobit nějaké problémy při vývoji.
- Seznam a popis činností v etapě a jejich návaznost.

Pro každou činnost by měl být určen také cíl, postup, vstupní (podklady), výstupní (produkty) a klíčové dokumenty. Nezbytné je také určit profese, které se jednotlivých činností účastní, a které za co odpovídají. Součástí popisu činností může být také doporučení k použití nástrojů a technik.

Postup řešení v jednotlivých fázích vývoje IS:

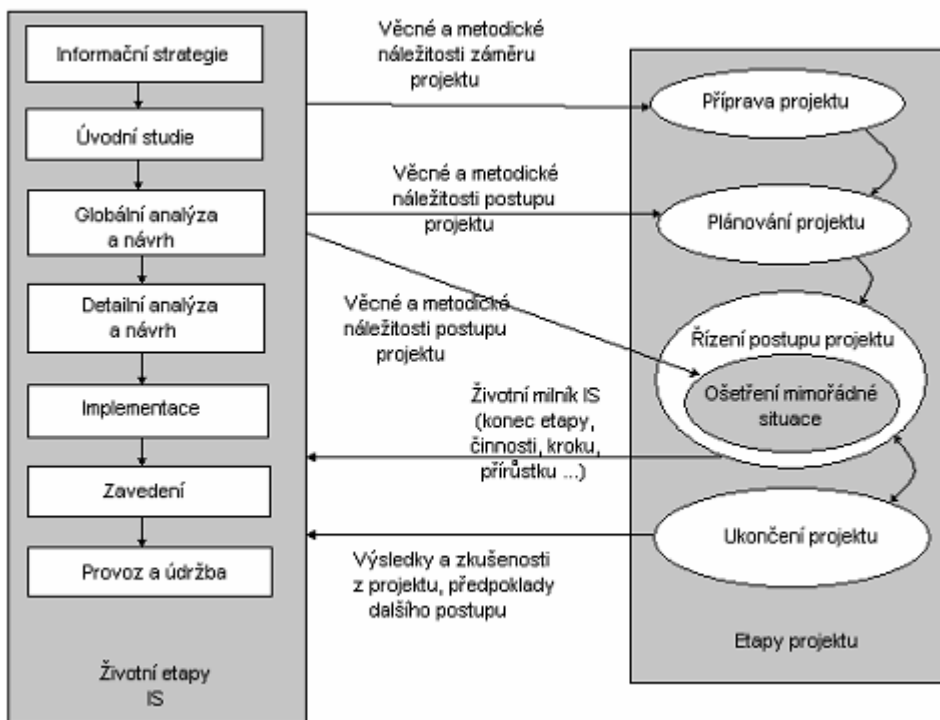
- Stanovení cílů.
- Vypracování/úprava plánu prací.
- Porozumění řešení oblasti a formulace hlavních problémů řešení.
- Formulace vize řešení.
- Analýza dosavadního stavu IS/IT podle platné vize. Jsou-li výsledky analýzy ve shodě s vizí, pokračujeme dalším krokem, pokud ne, vracíme se k vypracování a tvorbě plánu prací.
- Návrh IS/IT podle jednotlivých obsahových dimenzí.
- Řešení vzájemných vazeb mezi dimenzemi.
- Ověřování a testování návrhu.
- Prezentace, oponentura a předání výsledků.

Informační systémy 1

Smysl metodiky není v přesném popisu postupu vývoje IS krok po kroku, spíše jde o zahrnutí veškerých podstatných aspektů vývoje a také působnost od samého počátku až do případného úplného konce vývoje IS. Úspěšný proces vývoje nezáleží pouze na správném pochopení metodiky a provedení všech fází a činností. Velmi důležitým aspektem je také způsob řízení takového procesu. Je třeba si uvědomit, že vývoj informačního systému je projektem. Každý projekt je neopakovanou a časově omezenou činností. Má definovaný nějaký cíl, začátek a samozřejmě konec a má přidělené omezené zdroje a rozpočet. Neprobíhá do nekonečna, v určitém okamžiku je nasazen a začíná sloužit u zákazníka. Projekt má také nějaký životní cyklus.

Mezi základní činnosti patří:

- Příprava a plánování projektu
- Zahájení a operativní řízení postupu projektu
- Ukončení projektu



Obrázek 4.2: Vztah řízení procesu a vývoje IS

Na obrázku 4.2 je zachycen vztah mezi metodikou řízení projektu a metodikou vývoje IS. Je třeba rozlišovat mezi věcnými činnostmi projektu (stanovené metodikou vývoje IS) a činnostmi jejich řízení (stanovené metodikou řízení projektů). Metodika vývoje IS stanovuje obsahové náležitosti celého života IS. Konkrétní způsob jeho naplnění je záležitostí konkrétních projektů. Jeden projekt většinou pokrývá ucelenou část životního cyklu projektu (projekt analýzy, projekt návrhu, projekt implementace, ...). Pokud však budeme vyvíjet systém ve funkčních celcích – modulech, může být etapa analýzy, návrhu i implementace pokryta více projekty. Ne všechno je však projekt, například provoz a údržba informačního systému je běžnou rutinní záležitostí a tudíž se nejedná o projekt. Nemá přesně stanovený začátek a konec, celkovou sumu finančních prostředků, narozdíl třeba od implementace.

Informační systémy 1

Ještě stručně k obrázku 4.2. K prvnímu kontaktu obou metodik dojde hned na začátku. K přípravě projektu je třeba formulovat věcný obsah, k tomu potřebuji znát činnosti vývoje IS. V etapě plánování projektu je třeba znát konkrétní časovou a věcnou náplň. Metodika vývoje IS dodá následnost činností a požadavky na zdroje (potřebné profese a jejich kvalifikace, nástroje). Podobné je to i v samotném postupu projektu. Ukončením projektu vždy uzavřeme určitou část vývoje IS (etapu, několik etap, krok). Hlavním smyslem etapy Ukončení projektu je posouzení průběhu a výsledku projektu.

4.1 Struktura činností jednotlivých fází

Jako základ, pro popis jednotlivých etap a činností v nich obsažených a zasazení do celkového kontextu problematiky vývoje IS, nám poslouží metodika MDIS. Tato metodika podporující vícerozměrový (multidimenzionální) přístup vznikla na katedře Informačních technologií VŠE v Praze. Její popis můžeme najít například v [5].

Informační strategie

Informační strategie vychází z globální podnikové strategie, plánů a zjištěných nedostatků v informační podpoře. IS podporují strategické cíle a záměry organizace a měli by být vzájemně konzistentní. V etapě se provádí analýzy stavu informačních systémů, stavu IT v organizaci a analýza vývojových trendů v oblasti IT. Určují se architektury IS, a to funkční datová a technologická. Analyzují se dopady IS na organizační strukturu firmy, na zaměstnance a jejich kvalifikaci, ekonomické a technologické dopady nákupu/vytvoření IS. Probíhá vypracování projektů – určuje se jejich obsah, harmonogram, způsob řešení, přiřazení zodpovědnosti jednotlivým pracovníkům, ekonomická stránka (plánování a přidělení finančních prostředků) a v neposlední řadě se určuje pracovní náročnost jednotlivých činností.

Úvodní studie

V této etapě posuzujeme realizovatelnost jednoho vybraného systému. Určujeme, zda lze dosáhnout očekávaných výsledků. Učiníme rozhodnutí, zda ve vývoji pokračovat, či nikoliv. Odhadujeme náklady a přínosy návrhu systému. Východiskem jsou cíle projektu, analýza současného stavu a specifikace požadavků. Probíhá hrubý návrh funkcí, vstupů, výstupů, datového modelu systému, vymezení procesů a hranic systému. Dále probíhá hrubý návrh technologického řešení – výběr ASW, ZSW a HW.

Náplní řízení projektu je organizace činností, stanovení pracovních procedur, použití standardů a tvorba dokumentace. Dopady projektu jsou organizační a personální – výběr a složení týmu pracující na projektu – a také metodické. Je také potřeba zajistit harmonogram činností, naplánovat a zajistit finanční prostředky a také kontrolovat migraci pracovníků.

Vstupy etapy:

- Informační strategie podniku,
- Základní podnikové materiály, organizační řád, popisy funkčních míst,

Informační systémy 1

- Původní projekty včetně jejich dokumentace (jsou-li dostupné a vztahují-li se k řešení),
- Výstupy – přehledy a sestavy zpracovávané až dosud v rámci řešené oblasti.

Výstupy etapy:

- Dokument úvodní studie obsahující výše zmíněné,
- Návrh smlouvy na řešení celého projektu nebo jeho částí.

Globální analýza a návrh

V této etapě provádíme zpodrobnění základních požadavků, rozdělení na podsystémy a vymezení subprojektů, návrh hrubého modelu funkcí a dat pro každý subsystém a návrh rozhraní systému. Probíhá zde úplná specifikace všech hlavních funkčních požadavků, datových, prováděcích a dalších požadavků, snažíme se nalézt také odvozené požadavky. Určujeme prioritu všech požadavků a také strukturu systému. Výchoziskem pro tuto etapu jsou cíle IS/IT organizace a schválený plán jejich vývoje. Je třeba zajistit školící a instalační materiály, popř. zřídit školící středisko. Provádí se také návrh a úpravy harmonogramu projektu.

Vstupy etapy:

- Úvodní studie,
- Smlouva na projekt,
- Směrnice zákazníka (organizační a podpisový řád, ...),
- Dokumentace vlastního aplikačního software (ASW) zákazníka,
- Dokumentace ASW zákazníka, na který mají být realizovány vazby.

Výstupy etapy:

- Specifikace a analýza funkčních požadavků vzhledem k možnostem ASW,
- Školící materiály pro pracovníky zákazníka (případně předávací protokol o zřízení školícího pracoviště),
- Detailní organizační pravidla projektu a zpřesněný harmonogram projektu.

Detailní analýza a návrh

V této fázi návrhu se provádí analýza, definice požadavků a návrh systému až na úroveň, kdy je možné začít navržený systém implementovat. Již z názvu vyplývá, že zpodrobňujeme funkce, požadavky a modely z předchozí fáze Globální analýzy a návrhu. Pokud existuje více podsystémů, jsou činnosti prováděné v této etapě stejné pro každý z nich. Detailní návrh se týká technologické architektury, datové základny, výstupů systému a také organizační struktury. Vytváříme také prototypy systému (např. návrh uživatelského rozhraní). To zahrnuje určení dat, kterých se to týká, samotnou realizaci a nezbytné ověření prototypu. Odhadují se také náklady potřebné na dokončení systému nebo jednotlivých subsystémů.

Vstupy etapy:

- Specifikace a analýza funkčních požadavků,
- Organizační směrnice projektu.

Informační systémy 1

Výstupy etapy:

- Prototypová řešení a jejich dokumentace,
- Protokoly z verifikace (ověření) prototypových řešení uživateli,
- Souhrnná specifikace a analýza nároků na úpravy,
- Návrh datové základny,
- Návrh výstupů, jejich obsahů i formy,
- Souhrnné návrhy technologické architektury,
- Návrh organizačních úprav.

Implementace

Cílem je vytvořit fungující systém, který realizuje návrh vytvořený v předchozích etapách v daném implementačním prostředí. Provádí se také testování systému dle dokumentace. Systém musí bezchybně fungovat a musí mít implementovány všechny stanovené požadavky. Důležité je také vytvoření uživatelské a provozní dokumentace a popisu pracovních procesů.

Vstupy etapy:

- Analýza a návrh nových modulů (celého systému) a protokoly a analýza návrhů úprav jednotlivých modulů,
- Organizační směrnice projektu.

Výstupy etapy:

- Upravené (customizované) moduly ASW,
- Řešení specializovaných funkcí a modulů, které nebyly obsaženy v ASW,
- Upravené či nová uživatelské dokumentace, eventuelně i změny v provozní dokumentaci,
- Dokumentace datových rozhraní na ostatní ASW,
- Zpráva o průběhu testování,
- Plán školení uživatelů a zavádění systému.

Zavádění do provozu

Tato etapa se může na první pohled zdát jako nejméně obsáhlá, ale není to pravda. Zahrnuje instalace technického a programového vybavení, školení uživatelů – vedoucích pracovníků, administrátorů systému a běžných uživatelů, vytvoření a úpravy databáze, integrační a zátěžové testy. Je třeba aby přechod na nový systém v organizaci neomezoval běžný pracovní režim uživatelů a je také třeba zajistit, aby měli uživatelé čas si na nový systém zvyknout. Dodavatel zajistí počáteční podporu systému, která zahrnuje pomoc uživatelům, sledování zkušebního provozu a opravy chyb.

Vstupy etapy:

- Upravené a nové moduly ASW a jejich dokumentace,
- Dokumentace existujících databází a dokumentace návrhu datových rozhraní k ostatnímu ASW,
- Organizační směrnice projektu.

Výstupy etapy:

- Instalované moduly, předávací protokoly o instalaci modulů ASW,

Informační systémy 1

- Vytvořené nové (příp. změněné) databáze a předávací protokoly o migraci databází,
- Předávací protokoly o instalaci potřebné techniky, základního SW a komunikací.

Provoz, údržba a rozvoj

Cílem této etapy je zajištění provozu systému, jeho údržbu a rozvoj vzhledem k novým uživatelským požadavkům, které jsou v souladu se záměry a cíly organizace. Probíhá také monitorování provozu z důvodu optimalizace procesů, zjištění využití aplikací a provozních chyb. Návrhy změn a úprav mohou být funkčního, provozního nebo organizačního charakteru.

Vstupy etapy:

- Uživatelská a provozní dokumentace,
- Předávací protokoly k ASW, výpočetní technice a ZSW.

Výstupy etapy:

- Analýzy a monitorování provozu IS (využití funkcí, doba odezvy, provozní chyby a poruchy, ...),
- Návrhy úprav s určením jejich závažnosti,
- Dílčí úpravy ASW.

Pozn.: Na začátku každé etapy probíhá stanovení plánu prací v etapě. Ve všech etapách je důležitá a nezbytná podpora vedení, účast budoucích uživatelů na vývoji a podpora z jejich strany.

4.2 Dimenze tvorby IS

Cílem multidimenzionálního řešení je neopomenout žádný z faktorů, které mají vliv na zdárné dokončení, zavedení a provozování systému. Nelze opomenout ani vzájemné ovlivňování a vazby jednotlivých faktorů. Rozlišujeme dimenze ve dvou skupinách:

1. Reprezentuje úroveň integrace IS/IT, použitou úroveň abstrakce a časovou dimenzi řešení. Promítají se do jednotlivých fází vývoje IS – globální podniková strategie, informační strategie, úvodní studie, ...
2. Tato skupina reprezentuje dimenze, které se aplikují v každé fázi vývoje IS. Jde o dimenze informační/datová, procesní/funkční, ekonomická/finanční, softwarová, ...

Obsahové dimenze reprezentující druhou skupinu si nyní zmíníme a stručně popíšeme:

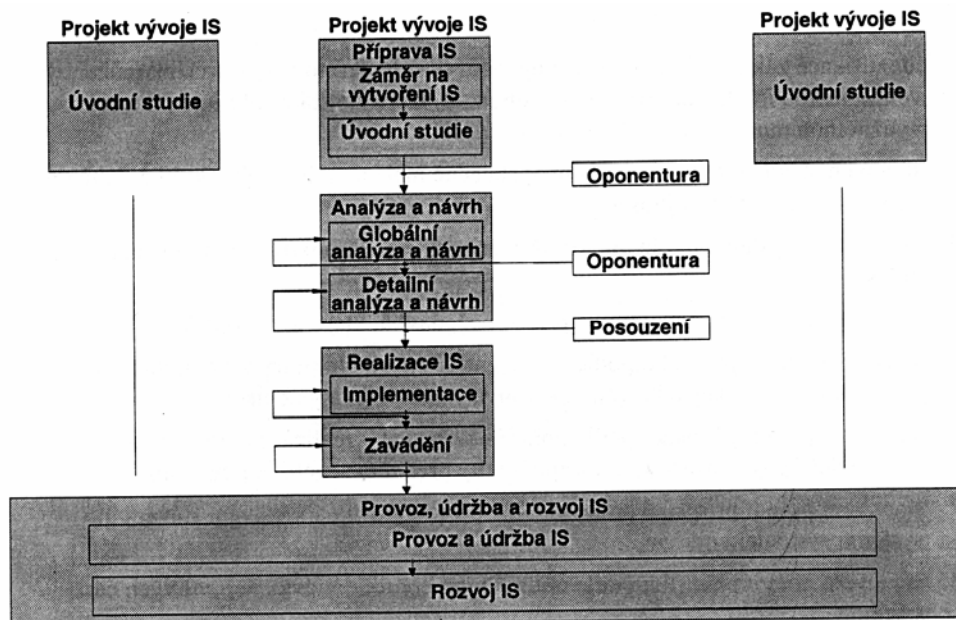
- *Data/informace* – typy informací, které jsou potřebné při jednotlivých podnikových aktivitách, obsah a struktura datové základny podniku.
- *Funkce/procesy* – procesy, které probíhají v podniku a možnost jejich podpory informačním systémem.
- *Organizační a legislativní aspekty* – legislativa země, ve které podnik působí, podnikové normy a směrnice, platné obecné standardy a normy – ČSN, ISO, vliv zavedení nového IS/IT na organizaci podniku.

Informační systémy 1

- *Pracovní, sociální a etické aspekty* – struktura pracovníků podniku (počet, kvalifikace, věk, pohlaví) pro stav po zavedení nového IS/IT, možné sociální a etické dopady přechodu na novou verzi, potřebná opatření (nábor/propouštění, rekvalifikace a školení) zavedení nové verze.
- *Software* – určení architektury programového systému (typy, parametry, vazby), určení zda nakoupit nebo vyvinout vlastními silami.
- *Hardware* – HW architektura IS/IT (typy, parametry a počty PC, periférie, komunikační sítě, ...).
- *Ekonomické a finanční aspekty* – finanční náklady na tvorbu a provoz IS/IT a přínosy podniku z využití IS/IT. Z jakých zdrojů budou náklady kryty a v jakých obdobích.
Mezi metodicko-organizační dimenze patří:
- *Metody* – metody a s nimi související techniky a nástroje využívané v jednotlivých fázích vývoje IS.
- *Dokumenty* – jaké dokumenty vznikají v průběhu vývoje a provozu IS, jejich obsah a vzájemné návaznosti.
- *Řízení prací dané fáze* – podklady k optimalizaci využití zdrojů (lidských, finančních, materiálových, ...) a díky tomu dosažení požadovaných cílů v plánovaném čase.

4.3 Základní přístupy vývoje

Životní cyklus vývoje informačního systému vymezuje základná etapy a jejich obsah. Existuje několik způsobů, jak realizovat vývoj IS. Základním přístupem je tzv. Vodopádový postup vývoje IS. Ten reprezentuje jednorázový průchod životním cyklem od první etapy až po poslední.



Obrázek 4.3: Základní – vodopádový – postup vývoje IS

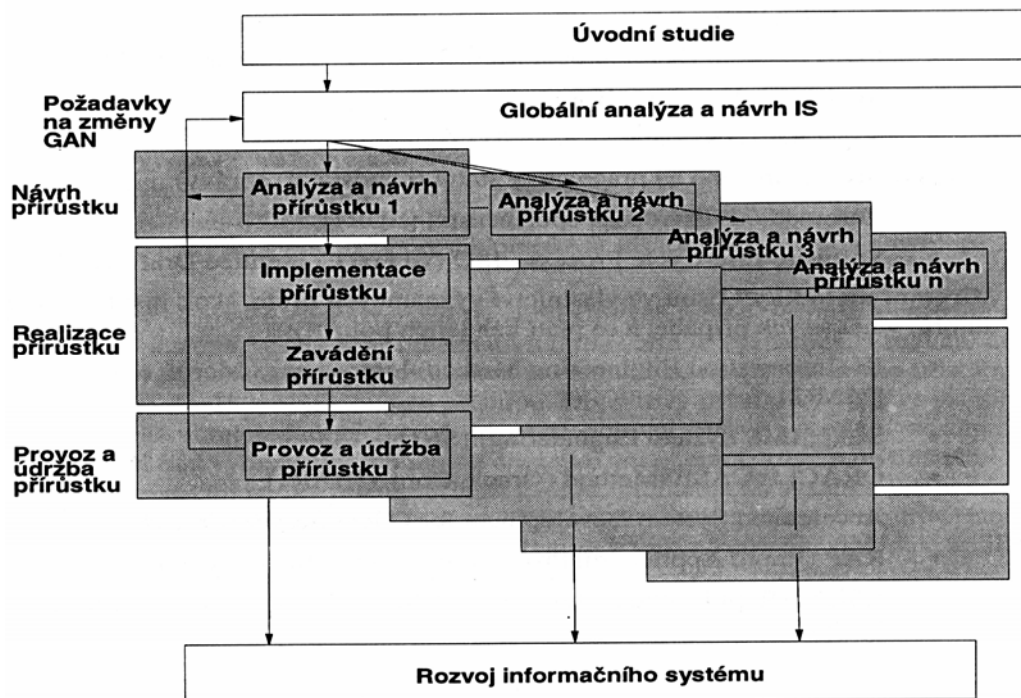
Tento přístup není možné většinou použít z několika logických důvodů. Zmíníme si některé z nich:

- Požadavky na informační systém nejsou dobře definovány nebo se často mění.

Informační systémy 1

- Oblast, kterou má pokrývat IS je natolik složitá, že je třeba ji poznávat po částech.
- Pokud bychom nerozdělili systém na části, které postupně realizujeme, vývoj by trval velmi dlouhou dobu, navíc by se mohlo stát, že na konci vývoje (např. za 5 let) by byly požadavky na systém velmi odlišné.
- Vývoj celého systému by stál značnou sumu peněz.

V takovém případě je běžné využít jiný přístup. Nabízí se nám tzv. Přírůstkový postup. Přírůstek je ucelená část systému – nějaký subsystém nebo modul, kterou lze samostatně navrhnout, implementovat a uvést do provozu. Funkčnost dříve dokončených a zavedených částí však zůstane zachována. Systém je pak vyvíjen postupně, kdy jeden přírůstek může být ve fázi implementace a druhý například ve fázi úvodní studie. Většinou se nejdříve zanalyzují, implementují a zavedou do provozu klíčové, snadno realizovatelné nebo jasně dané části systému. Po ustálení a upřesnění uživatelských požadavků se pokračuje v práci na ostatních přírůstcích, které se mezitím mohou dostat do fáze úvodní studie či globální analýzy a návrhu.



Obrázek 4.4: Přírůstkový postup vývoje IS

Činnosti potřebné k rozdělení systému na přírůstky:

1. Identifikace základních jednotek fungování.
2. Stanovení částí systému, které jsou malé a mohou být v krátké době viditelným přínosem.
3. Určení pořadí částí podle významnosti pro organizaci, věcnou oblast.
4. Stanovení priorit přírůstků podle priorit požadavků.
5. Stanovení pořadí přírůstků z hlediska rizik.
6. Stanovení pořadí přírůstků z podle pořadí návrhu datových entit

Pozn.: Zdrojem obrázků 4.3, 4.4 je standard SIS.

4.4 Trendy ve vývoji metodik

Stejně jako se vyvíjí a zlepšují se parametry hardware a software, tak i metodiky procházejí vývojem. Musí být aktualizovány nebo přepracovány, aby ctily tyto změny a vývoj HW, SW či nástrojů pro programování. Mezi nové rysy v posledních letech patří především již zmíněné nahrazování vodopádového přístupu iterativními (přírůstkovými) přístupy tvorby IS. Nové metodiky podporují inkrementální a rychlý vývoj s použitím prototypů – RAD (Rapid Application Development) nástroje. Dalším rysem je nástup objektových metodik a pronikání objektových principů do metodik. Tento přístup spojuje datový a funkční přístup a přináší větší přehlednost systému jak pro tvůrce, tak pro uživatele. Díky objektovému přístupu lze uživateli zpřístupnit vnitřně složitější aplikace a ten se může také více zapojit do návrhu informačního systému.



Kontrolní otázky:

1. Popište co je to metodika vývoje SW a k čemu slouží?
2. Jaké jsou základní fáze vývoje SW?
3. Popište činnosti fáze globální analýza a návrh.
4. Jaké jsou nevýhody vodopádového přístupu k tvorbě SW?



Úkoly k zamyšlení:

- Ve které fázi vývoje by podle Vás probíhalo rozdělení jednotlivých funkcí systému do vrstev (v případě vrstvené architektury)?
- Pokuste se zamyslet ve skupině s kolegy a slovně popište, jak byste řešili projekt informačního systému knihovny.



Korespondenční úkol:

Zamyslete se nad tím, která (které) fáze vývoje SW je v praxi nejčastěji opomíjena nebo je jí věnována velmi malá pozornost a proč? Výsledky úvahy zapište.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s životním cyklem vývoje IS. Dozvěděli jste se z jakých fází se tento proces skládá a jaké činnosti jsou náplní jednotlivých fází. Seznámili jste se s různými přístupy vývoje SW a také jste poznali dimenze tvorby IS.

5 Analýza zadání

V této kapitole se dozvíte:

- Co za činnosti se provádí v analýze zadání?
- Co je a jak se správně tvoří specifikace požadavků?
- Co popisují a jak se tvoří modely jednání?
- K čemu slouží a co popisují slovní scénáře?

Po jejím prostudování byste měli být schopni:

- Vytvořit specifikaci požadavků SW systému.
- Porozumět modelům jednání a strukturovat hranice a aktory SW systému.
- Pomocí modelů jednání popsat funkce budoucího IS.
- Popsat jednotlivé funkce budoucího systému slovními scénáři.

Klíčová slova této kapitoly:

Požadavek, specifikace požadavků, aktor, model jednání, use case, slovní scénář.

Doba potřebná ke studiu: 4 hodiny

Průvodce studiem

Tato kapitola přiblíží proces tvorby a analýzy požadavků na softwarový systém. K tomu slouží grafické modely jednání, které posléze podrobněji rozepisují slovní scénáře.

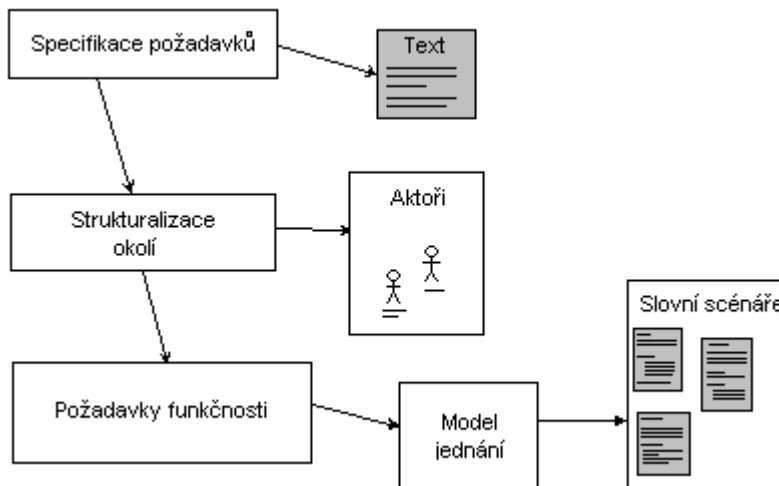
Na studium této části si vyhrad'te alespoň 4 hodiny.



Jak již víme, vývoj jakéhokoliv systému nebo aplikace není jen samotné programování, ale zahrnuje také sestavení požadavků na systém, analýzu systému a jeho návrh. Podle návrhu systém implementujeme, poté otestujeme a jde-li vše zdárně, tak i nasadíme a provozuje. Tato kapitola se zabývá úvodní částí vývoje, tj. analýzou zadání.

Analýza zadání je úvodní etapou projektu vývoje, která přesně specifikuje požadavky na vytvářený systém. K formulaci požadavků se používají tři modely: model odpovědnosti, požadavků a jednání.

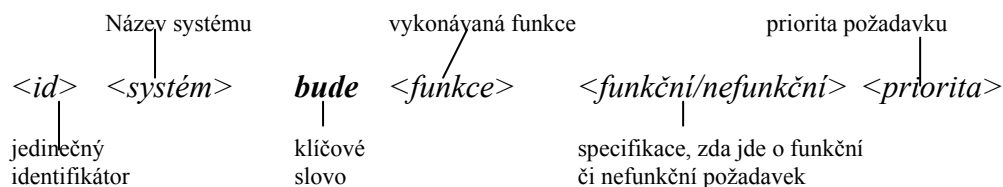
Analýza tedy obsahuje *Specifikaci požadavků*, což je vyjádření toho, co od systému čekáme. K zápisu se používají jasné a stručné věty. Musíme také určit, co nebo kdo z okolí bude jakýmkoliv způsobem ovlivňovat náš systém – strukturalizace okolí. Při tvorbě požadavků funkčnosti si systém představíme jako hotový a popíšeme, jak se bude používat. K tomuto popisu slouží *model jednání* a *slovní scénáře*. Slovní scénáře jsou podrobným popisem interakce mezi budoucím systémem a uživatelem. Jak z názvu vyplývá, jedná se obdobu scénářů používaných při divadelním představení nebo ve filmu. Výsledkem analýzy zadání jsou dva dokumenty – seznam požadavků a model jednání (včetně slovních scénářů).



Obrázek 5.1: Postup tvorby modelu požadavků a modelu jednání

Ve fázi *Úvodní studie* životního cyklu informačního systému je obsaženo zadání systému. Toto zadání je vlastně model požadavků. Model požadavků musí být úplný, musí tedy obsahovat úplně všechny požadavky, které jsou na systém kladeny. Požadavky mohou být funkční nebo nefunkční a mohou mít různou prioritu. Na systém pohlížíme jako na černou skříňku. Představujeme si co má dělat, ale již ne jak to má dělat! To je nutné proto, abychom ve fázi úvodní studie systému nezačali přemýšlet nad implementačními záležitostmi a omezeními (např. jak bude vypadat tabulka s daty, jaké rozhraní využijeme ke komunikaci se subsystémem, ...).

Jak již bylo výše zmíněno, mají být požadavky úplné, ale jednoduché. Vše potřebné musí být v zadání uvedeno, nelze něco vynechat s tím, že je to zřejmé z kontextu. Zadání musí být srozumitelné pro každého pracovníka v příslušném oboru. Někdy se spolu se zadáním vytváří tzv. slovník domény, který obsahuje a popisuje různé specifické názvy a objekty pro danou aplikační (obchodní nebo business) doménu. Nesmí se používat ani programátorský slang. Zadání slouží ke kontrole (správnosti a úplnosti požadavků) a konzultaci s uživateli a zadavatelem. Projektanti používají zadání k tvorbě projektu. Pro zápis požadavků se používá jednoduchý předpis:



Příkladem funkčních požadavků pro přihlášení do systému elektronického obchodu může být:

1. Systém ověří platnost uživatelského jména.
<nutnost>
2. Systém ověří platnost zadaného heslo pro daného uživatele.
<nutnost>

Informační systémy 1

3. Systém umožní vkládat položky do košíku.
<nutnost>

Nefunkční požadavky jsou většinou nějaké omezující podmínky, které jsou závislé na prostředí, použitých technologiích, apod.

4. Systém bude provozován v prostředí WWW.
5. Systém bude napsán pomocí technologie J2EE.
6. Pro zadání správného uživatelského jména a hesla budou maximálně 3 pokusy.

5.1 Struktura požadavků

Model požadavků by měl obsahovat účel a cíl systému, proč ho chce zákazník vůbec dělat. Z účelu systému potom odvodíme odpovědnosti a z nich odvodíme požadované funkčnosti. Zadavatel musí tedy jasně formulovat účel systému. Nedělejte systém, jehož účel není jasně formulován! Je třeba také odlišit účel a cíl (např. účelem školního kursu je se něco naučit, kdežto cílem je získat zápočet; účelem prodejního systému v knihkupectví je obsloužit prodej knih, jeho cílem je (pravděpodobně) zamezit chybným účtováním a zpronevěrám).

Z účelu se vyvíjejí odpovědnosti. Ty specifikují to, co chce uživatel. Je-li v knihkupectví účelem prodej knih, plynou z něho následující odpovědnosti: odpovědnost za přehled prodávaných knih (včetně jejich cen), odpovědnost za vystavení účtu prodaných knih, odpovědnost za evidenci všech prodejů.

K jedné odpovědnosti patří obvykle několik funkčností. Odpovědnost je souhrn logicky provázaných činností, funkčnost je jedna komplexní činnost.

Součástí modelu by měli být také požadavky na vyhledávání a jednotlivé přehledy. V zásadě není rozdíl mezi vyhledáváním a přehledem. Přehled může mít dvoje nasměrování – vnější (pro informaci uživateli) a vnitřní (pro potřeby systému). Výsledkem hledání je dílčí seznam.

Důležité je také specifikovat bezpečnostní stránku systému. Ta zahrnuje ochranu dat proti lidem a přírodním živlům. Na ochranu dat proti lidem se vyžaduje například autorizace přístupu (přístupovými hesly) a kódováním přenosu dat. Proti přírodě se používá archivace dat. Bezpečnostní stránku nejde stanovit bez toho, aniž bychom věděli jaké bude rozmístění systému (komponent na kterých poběží). Může jít o jeden nebo více počítačů zapojených v síti. Systém může být uzavřený – intranet, nebo otevřený a všem přístupný na Internetu. Z rozmístění plynou další požadavky, a to provozní vlastnosti systému. Je třeba specifikovat dobu odezvy systému, přizpůsobení organizačním požadavkům (poučený, běžný „hloupý“ uživatel), způsob přístupu (speciální klient, webový prohlížeč).

Jednoduchý příklad Půjčovny videokazet:

Účel a cíl systému:

Účelem informačního systému v půjčovně kazet je mít přehled o půjčených kazetách, tj. kdo je má půjčené a jak dlouho. Cílem je poslat upomínky těm, kteří kasety nevracejí.

Odpovědnosti:

Informační systémy 1

Evidence půjčených kazet u jednotlivých zákazníků. Z toho plynou dílčí odpovědnosti: evidence kazet a evidence zákazníků.

Funkčnosti:

Odpovědnost evidence kazet má tři funkčnosti:

- Vložení nového záznamu (kazety do evidence),
- Editace (oprava, změna) existujícího záznamu (kazety),
- Výmaz (vyřazení) z evidence.

Odpovědnost evidence zákazníků:

- Vložení nového zákazníka do evidence,
- Editace (oprava, změna) údajů existujícího zákazníka,
- Výmaz zákazníka z evidence.

Popis funkčnosti Vložení nové kazety do evidence:

Z papírové faktury či z propagačních materiálů se opíše údaje o nové kazetě. Faktura nebo nabídka může mít formu emailu, takže místo ručního zadávání můžeme údaje přepokopírovat pomocí schránky nebo k tomu může sloužit pomocný program. Dodavatel má na Internetu server s nabídkou, kde jsou všechny potřebné informace.

Obligátní a analytické přehledy:

Analytické přehledy: seznam nevrácených kazet, seznam půjčovnosti kazet agregovaný podle tématických skupin.

Bezpečnost:

Systém poběží na platformě Windows 2000 (XP), přístup bude chráněn přístupovým heslem jak do Windows, tak do samotného systému. Každý den ve večerních hodinách (po zavírací době) se bude provádět záloha databáze jednoduchými nástroji, které nabízí aplikace správce databáze.

Rozmístění a provozní vlastnosti:

Systém poběží na 2 počítačích – ve skladu a za přepážkou obsluhovanou zaměstnancem. Již z popisu umístění počítačů vyplývá, že nebudou přístupné zákazníkům, ale pouze zaměstnancům. Oba počítače budou propojeny sítí. K přihlášení do systému se bude používat jednoduchého klienta. Systém bude připojen do sítě Internet z důvodů přijímání nabídek kazet e-mailem a vystavení aktuálního seznamu kazet na internetové prezentaci.

Hlavní zásady tvorby specifikace požadavků

Při sepisování specifikace požadavků se popisuje systém pouze z hlediska dané problematiky a pomocí pojmů dané problematiky. K tomu používáme slovník domény (viz výše). Postupuje se systematicky podle jednotlivých požadovaných funkcností budoucí aplikace. Nesmí se používat pojmy z oblasti programování, informačních technologií (např. zakázané pojmy jsou tabulka, SELECT apod.), ale musíte používat pouze a jenom pojmy z dané problémové oblasti. Používají se takové výrazy, že uvedený systém by bylo možné vést i na papíře v písemné kartotéce. Věty musí být stručné, ale přesné a výstižné. Výčet požadavků musí být úplný (tj. neměl by chybět žádný funkční ani nefunkční).

Informační systémy 1

Obsah každého požadavku musí být také úplný, tj. popis požadavku by měl obsahovat hranice požadavku.

Může se stát, že při vývoji systému v nějaké z dalších fází narazíme na nějaké nedostatky nebo tedy na nutné změny ve specifikaci požadavků. Určitě je třeba je provést zpětně a promítnout je do všech následných fází. Musíme zaznamenat jak původní stav, tak nový stav po změně. Pokud pracujeme v týmu, musíme o změnách informovat všechny zúčastněné členy týmu a řádně archivovat (například pomocí Visual Source Safe, ...).

Používaná slovní spojení

“vyžaduje se evidovat <co>”

“systém bude podporovat přidání, editaci, vymazání<čeho>”

”vyžaduje se ochrana proti nepovolanému uživateli...”

“systém bude číst data <čeho odkud> a bude je zpracovávat za účelem <čeho>”

”systém pošle <komu> zprávu o <čem>”

“data přenosu budou nečitelná pro jiného uživatele”

Chyby při specifikaci

Předbíhání požadavků nabízejícím se řešením. Je třeba psát požadavek a nikoliv řešení. Nevynechat žádný z požadavků. V každém požadavku vyjmenovat sice stručně, ale vše, co se v něm požaduje. Nepoužívat programátorský žargon a výrazy!

5.2 Model jednání

Cílem modelu jednání (nebo také model případů užití) je popsat chování a komunikaci systému jako černou skříňku a také vymežit a strukturalizovat okolí systému. Účelem tohoto modelu je upřesnit zadání, je to také východisko pro dynamické modely a v neposlední řadě jde o východisko pro modelování tříd a tvorbu objektového modelu.

Při tvorbě modelu se tváříme jako by byl již systém hotov. Nevíme a ani nás nezajímá, jak systém uvnitř funguje a vypadá. Zajímá nás pouze to, jak se systém chová a projevuje navenek, jak komunikuje s okolím. Tomuto se říká princip černé skříňky.

Základními prvky modelu jednání jsou 4 komponenty:

- Účastníci (actors) – jedná se o role, přidělené osobám nebo předmětům používajícím daný systém.
- Případy užití (use cases) – jedná se o činnosti, které mohou účastníci provádět.
- Relace (relationships) – smysluplné vztahy mezi účastníky a případy užití.
- Hranice systému (systém boundary) – ohraničení zobrazené kolem případů užití, jedná se o vyznačené hranice modelovaného systému.

Modelování případů užití je doplňujícím způsobem specifikace a dokumentace požadavků. Modelování případů užití probíhá v následujících krocích:

Informační systémy 1

- *Nalezení hranic systému* – určíme, co je součástí systému, a co už není. Hranice systému specifikuje účastník, který systém používá.
- *Vyhledání účastníků (actors)* – aktor je ten, kdo se systémem přímo komunikuje, může se jednat o studenta v knihovně OU, zákazníka nakupujícího přes e-shop nebo o čidlo v elektrárně. Aktor je role, nejedná se o člověka nebo věc, jeden člověk může hrát více různých rolí, může být jak student OU, tak zákazník v e-shopu.
- *Nalezení případů užití* – případ užití je něco, co účastník očekává od systému ⇒ jsou vždy iniciovány účastníkem. Případ užití popisuje nějakou akci.
- *Specifikace případů užití* – zpodrobnění případů užití
- *Tvorba scénářů* – viz obr. 5.3

Prohlédneme-li si pořádně uvedené akce, zjistíme, že výše zmíněné čtyři základní prvky jsou výstupy jednotlivých akcí. Jak již bylo zmíněno výše, daný projekt může obsahovat slovníček pojmů. V něm jsou zachyceny klíčové obchodní pojmy, termíny a definice. Kromě těchto definic by měl slovníček by měl také řešit otázku synonym a homonym.

Nyní se vraťme k modelování případů užití. K nalezení jednotlivých účastníků (aktorů, actors) systému a k nalezení všech případů užití nám mohou pomoci následující otázky.

K označení účastníků systému napomohou tyto otázky:

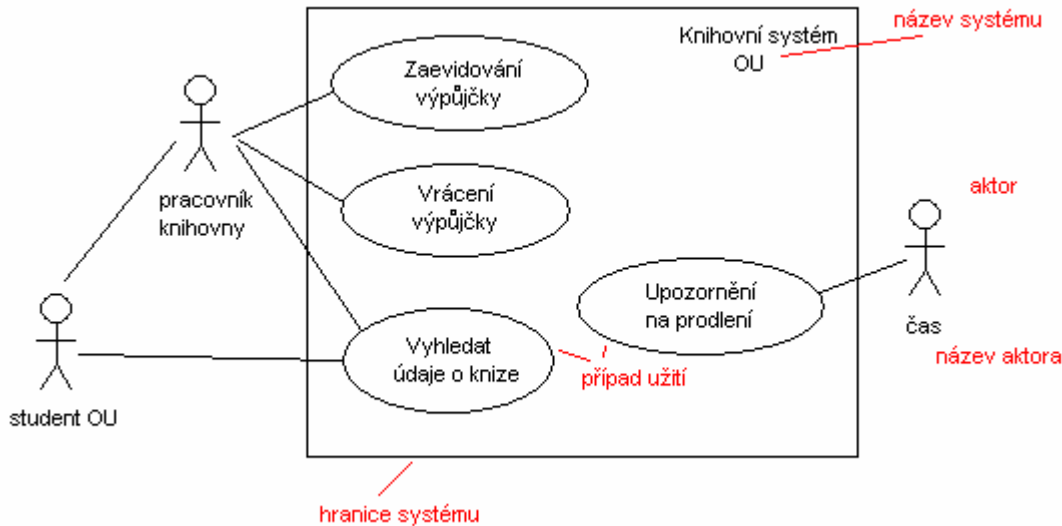
- Kdo nebo co daný systém používá?
- Jakou roli v této interakci hraje?
- Kdo instaluje systém?
- Kdo spouští a vypíná systém?
- Kdo se stará o jeho údržbu?
- Jaké další systémy spolupracují s naším systémem?
- Kdo systému zadává informace a kdo je používá?
- Děje se něco v určitou dobu?

Je třeba mít na paměti, že účastníci jsou vždy externí – jsou mimo naši kontrolu. Účastníci komunikují přímo se systémem – to napomáhá vhodné definici hranic systému. Účastníci představují role, které hrají, nejsou to konkrétní osoby ani předměty.

K nalezení případů užití napomohou tyto otázky:

- Jaké funkce jednotliví účastníci od systému očekávají?
- Bude systém uchovávat a poskytovat informace? Pokud ano, kteří účastníci budou tyto činnosti aktivovat?
- Jací účastníci budou upozorněni na změnu stavu systému?
- Existují nějaké vnější události, které ovlivňují systém? Co upozorní systém na tyto události?

Informační systémy 1



Obr 5.2: Příklad modelu jednání knihovny OU

Na obrázku 5.2 vidíme systém, který popisuje část funkčnosti knihovního systému ostravské univerzity. Aktory systému jsou pracovník knihovny, student OU a také čas. Student je aktorem systému pouze pokud vyhledává údaje o knihách (název, ISBN, autor, signatura, ...), chce-li knihu vypůjčit nebo vrátit účastníkem není. V takém případě komunikuje s pracovníkem knihovny a až ten je aktorem tohoto systému. Z obrázku také vidíme, že i pracovník knihovny má možnost vyhledat údaje o knize. Účastník čas je zde z důvodu evidování doby výpůjčky, respektive upozornění na překročení doby výpůjčky. Po překročení doby jednoho měsíce systém vystaví pokutu (resp. aktor čas) a tu může poté pracovník vytisknout a zaslat. Tato funkčnost již ale není zachycena v našem modelu.

Následující obrázek 5.3 ukazuje scénář neboli slovní popis případu užití. Každý případ užití má svůj název a specifikaci. Specifikace se skládá ze vstupní podmínky, z toku událostí a z následné podmínky. Pro specifikaci případů užití, stejně jako pro specifikaci požadavků, neexistuje žádný standard. Existuje spousta jednoduchých i složitých (ve smyslu komplexních) šablon, je však dobré držet se těch jednodušších.

Případ užití: Zaevidování výpůjčky	- Název případu užití
ID: UC1	- Jedinečný identifikátor
Účastníci: pracovník knihovny	- Účastníci případu užití
Vstupní podmínky: 1. Student si vybral knihu a dal si žádanku. 2. Kniha je k dispozici. <<nebo>> Student má na knihu rezervaci. 3. Student se prokázal průkazem ISIC. 4. Pracovník je přihlášen k systému.	- Stav systému před spuštěním případu užití
Tok událostí: 1. Pracovník vybere studenta v databázi knihovny pomocí karty ISIC. 2. Zaeviduje výpůjčku pomocí signatury knihy. 3. Systém doplní ostatní údaje a datum vrácení knihy.	- Skutečné kroky případu užití

Následné podmínky:

1. Student podepíše výpůjčku.
2. Student převezme knihu a kartu ISIC.

Obr 5.3: Struktura scénáře

Vstupní podmínky musí být splněny ještě předtím, než je možné případ užití spustit. Jsou to omezení stavu systému. Tok událostí popisuje jednotlivé kroky v případě užití. Následné podmínky jsou kritéria, která je nutno splnit na konci případu užití. V toku událostí je možné mít také podmíněčné vykonání akce, např. u nákupního košíku v e-shopu:

1. Případ užití začíná volbou „Zobrazit obsah košíku“.
2. KDYŽ je košík prázdný:
 - 2.1 Systém oznámí zákazníkovi, že košík neobsahuje žádné položky.
 - 2.2 Případ užití skončí.
3. Systém zobrazí seznam všech položek v košíku včetně jejich ID, názvu, krátkého popisu, množství a ceny.

Pokud je košík prázdný, provede se činnost v bodech 2.1 a 2.2, pokud není košík prázdný, bod 2. se přeskočí, jelikož není splněna podmínka, a pokračuje se prováděním bodu 3. Následné podmínky mohou mít alternativní toky. Znamená to, že po provedení případu užití lze provést jednu z činností uvedenou v alternativních tocích. U výše vypsaneho nákupního košíku v e-shopu to mohou být:

Alternativní tok 1: Zákazník může kdykoliv opustit obrazovku košíku.

Alternativní tok 2: Zákazník může kdykoliv opustit systém.

Stejně jako jsme použili klíčové slovo KDYŽ (anglicky IF), je možné použít i další klíčová slova FOR a WHILE pro opakování uvnitř toku. Použití FOR (pro) se používá s celým číslem určujícím počet opakování. U nákupního košíku bychom například prováděli nějaké akce PRO všechny položky nákupního košíku.

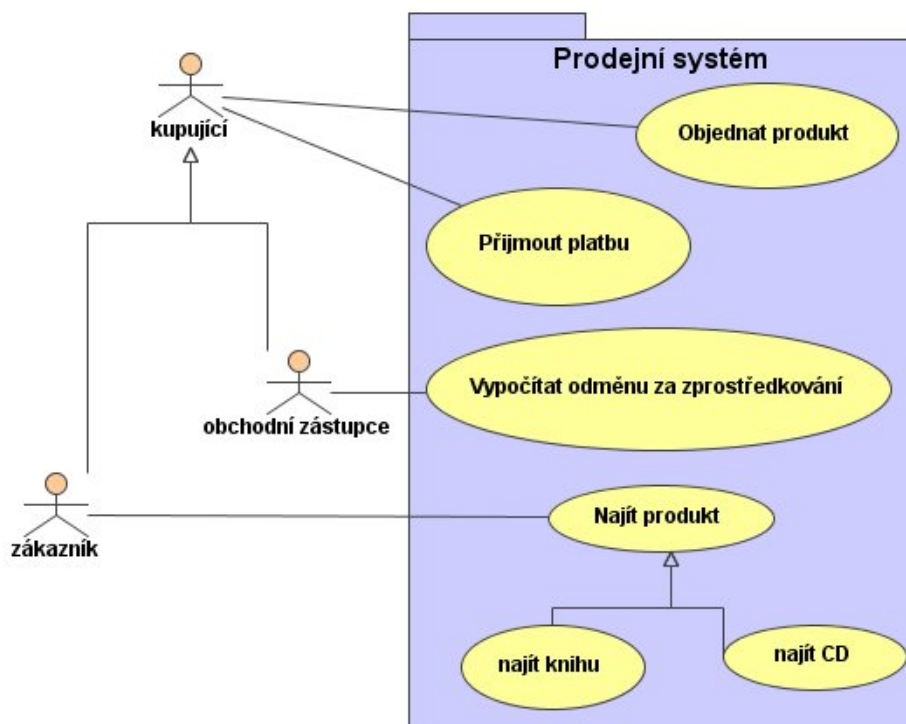
Dosud jsme mluvili o hlavních scénářích, popisujících nějaký případ užití systému. Existují však také vedlejší scénáře, které popisují různé chybové nebo vyjimečné stavy, které mohou v systému nastat. Může se jednat o chybné přihlášení do systému vinou špatného přihlašovacího jména nebo hesla, neplatné číslo kreditní karty při platbě, neplatné ID zákazníka, překročení doby platnosti karty, apod. Takové případy je třeba ošetřit a aplikaci buď ukončit (u špatného přihlášení po třech pokusech) nebo znovu zobrazit stránku pro zadání údajů nebo zobrazit chybovou stránku s dalšími instrukcemi.

Při tvorbě modelu jednání vytvoříme nejdříve jednoduchý model obsahující obecnější typy jednání, které pak dále upřesňujeme, zpodobňujeme a doplňujeme. Tak můžeme například obecný typ jednání Obsluha zákazníka v půjčovně videokazet zpodobnit na podtypy Půjčení kazet, vrácení kazet, vložení (změna) údajů o zákazníkovi.

Informační systémy 1

Zobecnění a relace include, extend

Mezi účastníky a také mezi případy užití mohou vzniknout různé vztahy, jedním z nich je generalizace neboli zobecnění. Způsob zakreslení vidíme na následujícím obrázku 5.4.



Obrázek 5.4: Zobecnění (generalizace) v modelu jednání

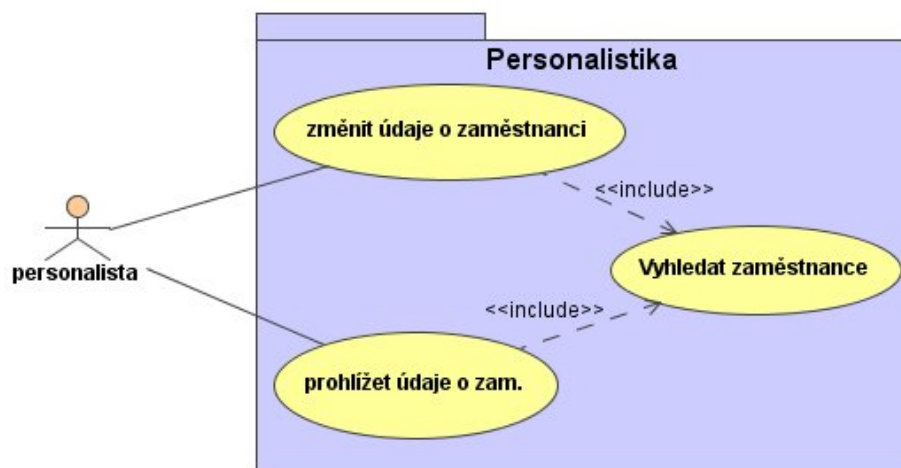
O zobecnění účastníka můžeme uvažovat, když mají dva nebo více účastníků hodně společného. Jedná se především o způsob interakce se systémem, tj. spouštějí-li účastníci stejné případy užití. Pro zobecnění účastníků platí jedna důležitá podmínka: Potomka můžeme dosadit všude tam, kde lze očekávat přítomnost (kde se může vyskytovat) jeho předka. Rodičovský účastník může být buď abstraktní nebo konkrétní role. Potomci dědí od svých předků nejen role, ale i relace, toho důvodu musí platit zmíněná podmínka. Na našem obrázku je zřejmé, že můžeme dosadit účastníka zákazník nebo obchodní zástupce všude tam, kde je očekáván účastník kupující.

Zobecnění případů užití se používá pouze v případě, máme-li několik typů jednání, které jsou jistou specifikací obecnějšího případu. Odvozené případy užití mohou:

- dědit funkce a vlastnosti od svých předků,
- přidávat nové funkce a vlastnosti,
- překrývat zděděné funkce a vlastnosti.

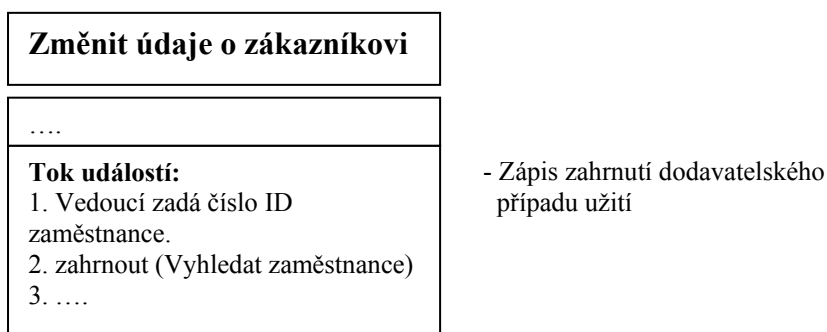
Odvozený případ užití automaticky dědí všechny funkce a vlastnosti od svého předka. Obě techniky generalizace používáme pouze tehdy, přinese-li to viditelné zjednodušení diagramů případů užití.

Typy jednání (případy užití), které jsou přímo spojené s nějakým účastníkem jsou *hlavní typy jednání*. Typy jednání používané v jiných případech jsou *vkládáné typy jednání*. Tyto vkládané relace mohou být typu *include* nebo *extend*. Jaký je mezi nimi rozdíl a kdy a jak je můžeme použít si řekneme v následujícím textu.

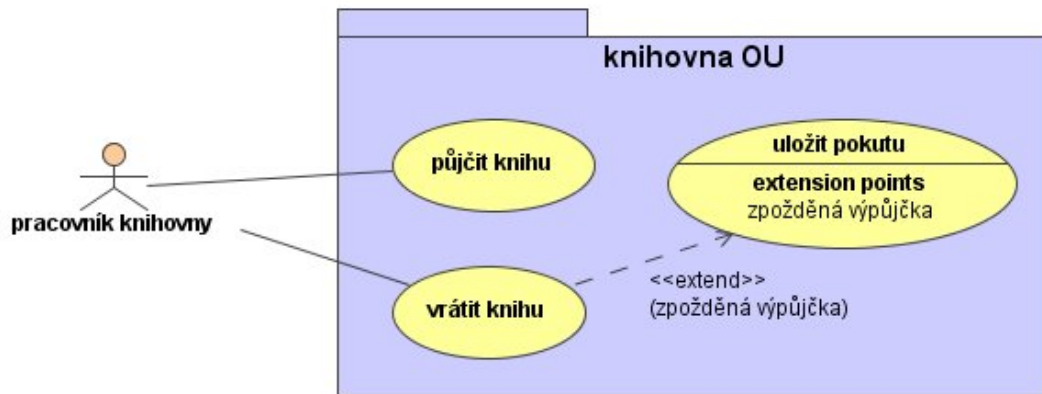


Obrázek 5.5: Relace include

Relace *include* (syntaxe zápisu viz obr. 5.5) umožňuje zahrnout chování dodavatelského případu užití do toku klientského případu. Důležité je si uvědomit, že klientský případ užití není bez zahrnutí dodavatelských případů úplný! Dodavatelské případy tvoří nezbytnou část klientského případu, a právě v tom, jak za chvíli uvidíte, je zásadní rozdíl mezi druhou relací *extend*. V klientském případě užití musíme určit přesný bod, kdy dodavatelský případ zahrnout. Syntaxe zahrnutí se podobá volání funkce. Krátký příklad scénáře k případu užití z obrázku 5.5 nám ji pomůže objasnit:



Relace *extend* (syntaxe zakreslení viz obr. 5.6) poskytuje způsob rozšíření chování stávajícího případu užití o chování nové. Bázový případ užití poskytuje určitou množinu bodů, kterým říkáme *body rozšíření* (anglicky extension points). K nim lze připojit rozšíření v podobě nového chování. Bázový případ užití je úplný i bez rozšiřujících případů. Pro ně má připraveny body rozšíření a vůbec o rozšiřujících případech nemusí vědět. Tok událostí bázového případu užití neví, v jakém místě bude rozšířen, ani ho to nezajímá. Body rozšíření totiž existují ve vrstvě nad hlavním tokem událostí. Tuto vrstvu si lze představit jako tenký povlak nad hlavním tokem, který obsahuje body rozšíření.



Obrázek 5.6: Relace extend

Způsob zápisu bodů rozšíření ve scénářích vidíme na následujícím scénáři, který popisuje případ užití vrátit knihu z obrázku 5.6:

	Vrátit knihu
	ID: UC5
	Účastníci: Pracovník knihovny
	Vstupní podmínky: 1. Pracovník je přihlášen k systému knihovny
	Tok událostí: 1. Knihovnik zadá ID čtenáře. 2. Systém zobrazí údaje čtenáře včetně seznamu půjčených knih. 3. Knihovnik vyhledá knihu k vrácení. <zpožděná kniha> 4. ...
	Následné podmínky: Kniha byla vrácena.
Bod rozšíření	

Je důležité si uvědomit, že zásadní rozdíl mezi relacemi extend a include je v úplnosti případu užití. Při použití relace include není klientský případ užití úplný bez zahrnutí dodavatelských případů. V případě relace extend je rozšiřující případ užití pouze rozšířením stávajícího chování a základní případ užití je kompletní i bez něj.

Zadání vytvářeného systému máme ve formě specifikace požadavků a modelu jednání, nyní musíme oba modely porovnat a sladit.

Diagramy případů užití jsou jednoduché a srozumitelné, a proto je možné je použít i pro konzultaci a upřesnění zadání u zadavatele (zákazníka) systému. Diagramy se pro tento účel také používají, protože jim jsou schopni porozumět i lidé, kteří nejsou odborníci v IT. Díky tomu je potom zadání přesnější a neobsahuje nepřesnosti.

Diagramy případů užití jsou diagramy modelovacího jazyka UML. Více o tomto modelovacím jazyce pojednává poslední kapitola těchto skript.

Informační systémy 1



Kontrolní otázky:

1. Jaký je rozdíl mezi funkčním a nefunkčním požadavkem na systém?
2. Co nebo kdo je aktor?
3. Jaké by měly být základní body slovního scénáře?



Úkoly k zamyšlení:

Zamyslete se, jestli může mít tvorba modelu jednání (Use case diagramu) zásadní vliv na řízení a plánování prací na vývoji SW? Jestli ano, tak proč? Jestli ne, tak proč ne?



Korespondenční úkol:

V minulé kapitole jste v úkolu k zamyšlení popisovali, jak byste řešili projekt informačního systému knihovny. Pokuste se nyní popsat požadavky na tento systém ve formě specifikace požadavků.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s tvorbou specifikace požadavků, která popisuje chování budoucího systému. Dozvěděli jste se, jak tyto požadavky zapisovat v textové a grafické formě srozumitelné i pro zadavatele systému. Dále jste se naučili, jak zapsané požadavky podrobně rozpracovat analyzovat.

6 Principy globální podnikové strategie

V této kapitole se dozvíte:

- Co definuje globální podniková strategie?
- Jak se tato strategie tvoří?
- Co je a k čemu se využívá SWOT analýza?
- Jak se definují cíle, poslání a funkce podniku?

Po jejím prostudování byste měli být schopni:

- Definovat pojem globální podniková strategie.
- Porozumět, jak se taková strategie tvoří a co všechno obsahuje.
- Umět vypracovat a porozumět SWOT analýze.

Klíčová slova této kapitoly:

Globální strategie, SWOT analýza, interní, externí faktor.

Doba potřebná ke studiu: 3 hodiny

Průvodce studiem

Tato kapitola se zabývá náplní a vypracováním globální strategie podniku. To zahrnuje vypracování SWOT analýzy a definování poslání a funkcí podniku.

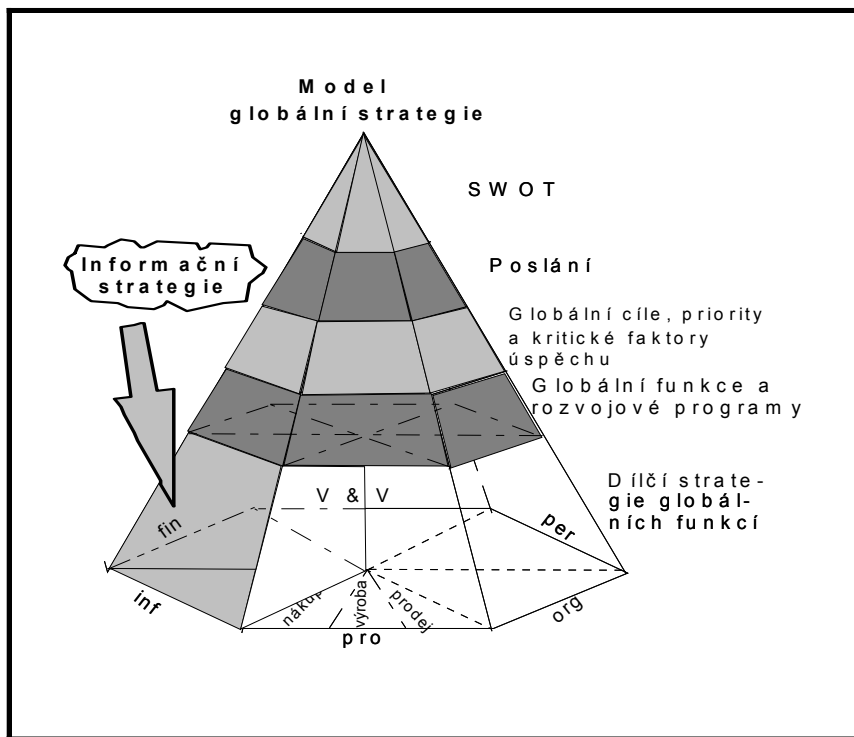
Na studium této části si vyhradte alespoň 3 hodiny. Doporučujeme nastudovat zvláště část o SWOT analýze a následně se pokusit vypracovat SWOT analýzu nějakého, Vám známého, problému.



Tato strategie určuje poslání firmy, celopodnikové cíle, priority, podmínky a zdroje pro dosažení těchto cílů. Zejména musí strategie určit následující:

- hlavní předmět podnikání,
- skupiny zákazníků, na které je podnik orientován,
- nabízené produkty a služby,
- hlavní obchodní partnery (zejména ve smyslu určení místa podniku v dodavatelsko odběratelských sítích),
- zdroje (lidé, znalosti, finance, technologie,...) nutné pro dosažení stanovených cílů.

Na následujícím obrázku 6.1 si můžeme prohlédnout, jak vypadá model globální strategie organizace. Z obrázku je zřejmé, že globální strategie organizace nebo podniku má několik dimenzí.



Obrázek 6.1: Model globální strategie

Při tvorbě globální strategie organizace slouží konceptuální model tvorby globální strategie. Tento model definuje hlavní zaměření podniku, dále podnikové cíle a jejich priority, definuje také zdroje pro realizaci cílů, způsob ověřování jejich naplňování a osoby odpovědné za jejich dosažení. Konceptuální model se skládá z několika bodů, jsou jimi:

1. SWOT analýza
2. Formulace poslání podniku
3. Definice globálních podnikových cílů
4. Vymezení globálních podnikových funkcí
5. Tvorba strategií pro globální podnikové funkce
6. Vyhodnocení a změny

Tyto body si nyní podrobněji popíšeme.

6.1 SWOT analýza

Jednotlivá písmena S, W, O, T jsou úvodní písmena z anglických slov Strengths, Weaknesses, Opportunities, Threats, což v tomto případě volně přeloženo znamená silné stránky, slabé stránky podniku, příležitosti a možné hrozby prostředí. Na následujícím jednoduchém příkladu si můžeme ukázat, jak vlastně taková SWOT analýza vypadá.

SWOT faktor	Příčina	Priorita	Navrhovaná akce
Externí			
(-) vysoká cena materiálu	monopolní dodavatel	1	substituční materiál
(- +) naši velcí odběratelé si vynucují od dodavatelů	možnosti komunikačních	2	zavedení EDI

Informační systémy 1

komunikaci pomocí EDI	technologií, zrychlení obchodního cyklu a snížení nákladů.		
Interní			
(-) dlouhá reakce na objednávky zákazníků	nevhodně navržený proces „vyřízení objednávky“	3	BPR procesu „vyřízení objednávky“
(+) vnitropodnikové účetnictví dává věrný obraz o nákladovosti a ziskovosti jednotlivých profit center, výrobků a zakázek	dobře navržený a dobře využívaný controlling	2	udržet a dále zpružnit vazbu controllingu na ostatní oblasti (prodej, výrobu, marketing,...)
další faktory ...			

Tabulka 6.1: Příklad SWOT analýzy

6.2 Analýza externích faktorů

Při analýze externích faktorů uvažujeme o věcech jako jsou zájmy vlastníků firmy, obchodní partneři, konkurenti, technologie, ekonomika, legislativa a politika země. Jedná se tedy o faktory z okolí firmy, z prostředí, v kterém firma existuje a s kterým komunikuje a spolupracuje. Při zkoumání zájmů vlastníků firmy zjišťujeme zájmy majoritních vlastníků – těch, kteří vlastní rozhodující kapitál. Neopomeneme analyzovat možný vývoj struktury vlastníků (prodej nebo sloučení firmy, apod.) a jejich zájmů.

Partneři, vlastníci a zaměstnanci mohou tvořit koalice, kde předmětem společného zájmu a tím i možné dlouhodobé spolupráce je průnik jejich společných zájmů. U odběratelů – zákazníků sledujeme hlavní odběratele a jejich specifika, očekávaný vývoj těchto odběratelů a jejich potřeb, závislost odběratelů na firmách XX, závislost firem XX na odběratelích, další možnosti rozšíření odběratelů a proč nejsou. Analyzujeme také, je-li možná užší kooperace firmy XX s odběratelem (propojení). U dodavatelů je důležitá především jejich spolehlivost a kvalita, očekávaný vývoj hlavních dodavatelů, závislost firmy XX na dodavatelích, možnosti substituce dodavatelů a zda lze rozšířit okruhy dodavatelů a pokud na, tak proč. Co se týká formálních či neformálních aliancí či sdružení, zajímá nás výhodnost a perspektiva dosavadních spojení, zda je složení aliance optimální, jestli existuje možnost nových aliancí a kdo jsou možní kandidáti na nové aliance.

U konkurence sledujeme její podíl na trhu, silné a slabé stránky, typ a stav IS. Dále nás zajímá očekávaný vývoj hlavních konkurentů a také nebezpečí vstupu dalších konkurentů na trh, sledujeme i hrozby náhradních komodit a služeb.

Důležité je také sledovat vznik a rozvoj technologií v oblasti hlavní činnosti firmy, jaká je standardní světová úroveň, jaké jsou nejnovější trendy v dané oblasti. Neopomeneme uvést možnosti zkvalitňování technologie v organizaci a případně se snažíme objasnit proč je zkvalitnit nelze. Analýza zdrojů pracovních sil a jejich potenciálu obnáší specifikaci požadovaného vzdělání pracovníků, požadavky na jejich praktické znalosti, zkušenosti, případné reference, věkové omezení.

Informační systémy 1

Pro podnik je také důležité sledovat ekonomické podmínky v zemi, jedná se o hlavní ekonomické ukazatele jako je makroekonomická stabilita země a její inflace, dále je důležitá situace v bankovním sektoru a dostupnost finančních služeb. Firma musí také ctít zákony a předpisy vztahující se k její činnosti v dané zemi. Ekonomická a legislativní situace je provázána s politickými faktory jako je stabilita politické situace v zemi a síla jednotlivých lobbystických skupin. Z důvodu přístupu k zaměstnancům, obchodním partnerům, institucím a při jednání s nimi je více než dobré znát sociální a kulturní vlivy ve společnosti. Mezi tyto vlivy patří obecně uznávaný hodnotový systém, kulturní zvyklosti v daném teritoriu, síla a zájmy odborové organizace. V neposlední řadě analyzujeme faktory geografické a informační infrastrukturu v místě působení firmy. U geografických faktorů se zajímáme o podnebí (možná zemětřesení, velké výkyvy teplot, výskyt častých bouří a hurikánů, ...), možnost dopravy, vzdálenost od ekonomických center, od hlavních dodavatelů i odběratelů. Hlavní faktory informační infrastruktury jsou stav telekomunikací, podmínky jejich využití a připravenost partnerů na efektivní způsob komunikace.

6.3 Analýza interních faktorů

Interní faktory jsou ty, které se přímo týkají vlastní firmy, ne jejího okolí. Jde o zaměstnance a jejich vzdělání, organizační strukturu, management a vedení. Dalšími faktory jsou nákup a prodej, výroba, vývoj, skladování a také vlastní informační systém firmy. Nyní si opět tyto faktory probereme podrobněji.

U vrcholového řízení sledujeme výsledky vedení podniku za minulé období, tvorbu cílů a vizí, úspěšnost dosahování cílů, vývojové trendy. Sledujeme formu a organizaci vrcholového vedení, efektivnost podnikového řízení a také stupeň centralizace řízení jednotlivých činností.

Marketing se zabývá například mapováním trhu, tvorbou marketingové strategie, a tvorbou nabídek. U marketingu proto analyzujeme dosavadní metody a výsledky, úroveň mapování trhu, úroveň informací o zákaznících (získávání, vedení a využívání těchto informací), úroveň vyhodnocení reakcí podniku na potřeby zákazníků. Zajímá nás také do jaké míry se marketingu daří určovat segmenty trhu, akviziční činnost nebo cenovou strategii.

U nákupu analyzujeme úroveň a metody zjišťování nákupních potřeb, metody výběru zákazníka pro nákup, kvalitu, včasnost a cenu nákupu. V oblasti výroby, skladování a služeb analyzujeme výrobní technologie a s tím spojené náklady na výrobu, rychlost reakce na požadavky zákazníka, možnosti operativních změn. Důležité je také sledovat kvalitu výrobků a služeb, zanalyzovat strukturu plánovacích ukazatelů a zhodnotit investiční aktivity firmy. V oblasti prodeje analyzujeme efektivnost informací z této oblasti, vnitřní vlivy ohrožující tuto oblast, vnitřní silné stránky této oblasti a stabilizující faktory.

Jedním z interních faktorů je také vlastní výzkum a vývoj. Je tedy třeba provést analýzu potřebnosti a také výsledků existujícího podnikového výzkumu.

Důležitým interním faktorem je ekonomická situace a hospodaření firmy. V této oblasti je třeba si všimnout vývoje nákladů a výnosů a s tím spojený vývoj cash flow (toku peněz). Nezanedbatelná je kontrola úrovně vnitropodnikového účetnictví, to závisí na úrovni a existenci controllingu. Pro SWOT analýzu je

Informační systémy 1

dále důležité si všimnout struktury interních finančních ukazatelů, identifikovat přímé a nepřímé náklady na jednotku produkce, všimnout si kvality (či vůbec její existence) platební a úvěrové politiky a investiční politiky.

V oblasti organizace a řízení firmy definujeme a sledujeme stav řízení podnikových procesů, vhodnost organizační struktury podniku. U pracovníků analyzujeme shodu míry odpovědnosti a kompetencí a s tím spojenou efektivnost motivací pracovníků. Je také třeba zanalyzovat opatření pro řízení kvality. U pracovníků dále sledujeme jak věkovou, tak kvalifikační strukturu, jejich fluktuaci na jednotlivých místech. Všimáme si také existence a kvality kvalifikačních a rekvalifikačních programů. V analýze bychom neměli opomenout celkové klima v podniku, podnikové hodnoty a vztahy mezi nadřízenými a podřízenými.

V oblasti IT/IS je třeba se ptát minimálně na to, jestli je provozovaný IS nástrojem zvyšování konkurenceschopnosti podniku, jaké jsou slabé a silné stránky IS pro vedení podniku.

K vyhodnocení SWOT analýzy (viz tabulka 6.1) se používá tabulka obsahující daný faktor včetně upřesnění, zda se jedná o faktor interní nebo externí. Znaménkem + a – určujeme vliv tohoto faktoru, který může být kladný nebo záporný nebo obojí (v něčem přínos, v něčem přítěž). Sloupec příčina obsahuje popis příčiny, proč se tak děje a v posledním sloupci popisujeme navrhovanou akci na odstranění, udržení nebo zlepšení působení daného faktoru. Je možné také určit prioritu analyzovaných faktorů. Rozlišujeme, které faktory jsou kritické a které jsou málo významné nebo bezvýznamné.

Priorita faktorů

Každý faktor má nebo může mít definovanou svoji prioritu. Tato priorita určuje důležitost a nutnost reakce na vypořádání se s působením faktoru. Následující výčet ukazuje stupnici priorit a jejich popisy:

- 0** – faktor je nevýznamný,
- 1** – málo významný faktor, který nemá přímý vliv na konkurenceschopnost a ekonomické výsledky podniku,
- 2** – středně významný faktor, který má přímý vliv,
- 3** – velmi významný (kritický) faktor, nereagování na takový faktor výrazně ohrožuje ekonomickou situaci a pozici vůči zákazníkům firmy nebo podniku.

Část SWOT analýzy si nyní pro lepší pochopení ukážeme (tabulka 6.2) na příkladu teplárenského podniku. V podniku je používán systém MARKET, analýza se týká modulu Odběratelé.

SWOT faktor F1	+/-	Příčina	Navrhovaná akce
Neznalost vývoje hlavních odběratelů tepla z hlediska budoucích potřeb odběru.	–	<ul style="list-style-type: none">➤ Nedostatečné (většinou statické) údaje o odběratelích tepla.➤ Pasivní přístup k získávání informací.➤ Malá informovanost o případné restrukturalizaci.➤ Neexistence soustavného	<ul style="list-style-type: none">➤ Úprava marketingových procesů ve firmě.➤ Restrukturalizace útvaru marketingu.➤ Modifikace využívání systému MARKET.

Informační systémy 1

SWOT faktor F2	+/-	Příčina	Navrhovaná akce
Neadekvátní struktura údajů o hlavních odběratelích.	–	<ul style="list-style-type: none"> ➤ Neúplná specifikace potřebných informací. ➤ Obtížnost získání těchto údajů. ➤ Preferování statických dat. 	<ul style="list-style-type: none"> ➤ Upřesnění struktury požadovaných údajů. ➤ Posouzení výhodnosti systému MARKET z hlediska těchto údajů. ➤ Reorganizace práce marketingového útvaru.
SWOT faktor F3	+/-	Příčina	Navrhovaná akce
Nedostatečná zpětná vazba na odběratele.	–	<ul style="list-style-type: none"> ➤ Nedostatečná činnost odbytu a marketingu. ➤ Neochota získávat i jiné než oficiální informace. ➤ Malá vlastní iniciativa. ➤ Neexistence aktivní obchodní politiky. 	<ul style="list-style-type: none"> ➤ Vytvoření Hot line pro odběratele. ➤ Aktivní přístup ve styku se zákazníkem. ➤ Průběžné hodnocení spokojenosti zákazníka.
SWOT faktor F4	+/-	Příčina	Navrhovaná akce
Operativní informace o odběratelích převažují nad informacemi strategickými.	–	<ul style="list-style-type: none"> ➤ Snazší získávání operativních informací. ➤ Charakter stávajících informačních systémů. 	<ul style="list-style-type: none"> ➤ Reengineering marketingových procesů. ➤ Posouzení IS MARKET z hlediska jeho podpory strategických informací. ➤ Zavedení systému „hlášenek o jednání“ a jejich zpracování.
SWOT faktor F5	+/-	Příčina	Navrhovaná akce
Hrozba ztráty odběratelů.	–	Nedostatečné informace o odběrateli, jeho vývoji, spokojenosti a záměrech.	<ul style="list-style-type: none"> ➤ Reengineering marketingových procesů. ➤ Aktivní zpětná vazba na odběratele. ➤ Zavedení systému „hlášenek o jednání“ a jejich zpracování.
SWOT faktor F6	+/-	Příčina	Navrhovaná akce
Nepodchycení všech potenciálních zákazníků.	–	Nedostatečně vypracovaný interní systém získávání neformálních informací o možných zákaznících.	<ul style="list-style-type: none"> ➤ Vytvoření stimulací pro získávání i neformálních informací, které rozšíří možnosti odbytu. ➤ Zavedení systému „hlášenek o jednání“ a jejich zpracování.

Tabulka 6.2: SWOT analýza teplárenské firmy

Komentář k faktoru F1:

Marketingový systém MARKET není využíván tak, aby podchytil hlavní vývojové trendy hlavních odběratelů tepla a případné změny jak jimi používané technologie ve vlastní výrobě (která tak může mít vliv na velikost

Informační systémy 1

odběru tepla), tak i ekonomické tendence. V teplárně se neprovádí důsledné sledování a vyhodnocování možných vývojových trendů technologií používaných hlavními odběrateli s cílem upozornit na případné možné změny, které by mohly vést ke změně velikosti odebíraného tepla. O těchto změnách se teplárna dovídá většinou až po jejich realizaci, kdy se jen obtížně hledá náhradní odběratel.

Komentář k faktoru F2:

Ve stávající situaci se teplárna dozvídá pozdě např. o organizačních změnách uvnitř jednotlivých odběratelů, které mohou vést k zastavení určité výroby a tedy ke změnám v odběru tepla. Na tuto situaci lze pak jen obtížně reagovat až v okamžiku výskytu a vzniklý výpadek tak eliminovat. Marketinkové útvary vyvíjejí nedostatečnou aktivitu v této oblasti a orientují se na snazší způsoby získávání oficiálních údajů o jednotlivých odběratelích. Tuto strukturu údajů podporuje ale také systém MARKET.

Komentář k faktoru F3:

Teplárna se někdy se zpožděním dovídá o přání zákazníka přejít na jiný způsob zásobování teplem (tedy vlastně o jeho nespokojenosti s teplárnou, danou např. cenovou politikou teplárny). O těchto informacích se dovídá až např. v okamžiku, kdy zákazník žádá o stavební povolení na vybudování jiného způsobu vytápění. Nebyl tedy včas učiněn pokus o dosažení dohody se zákazníkem např. formou aktivní cenové politiky.

Komentář k faktoru F4:

IS MARKET, jakožto jediný využívaný IS o odběratelích je naplňován daty převážně operativní povahy, chybí data týkající se strategie rozvoje odběratelů. Tyto údaje lze mnohem obtížněji získat, doposud nebyly takto důsledně vyžadovány. Bez těchto údajů lze jen obtížně odhadovat další vývoj odběratelů a jejich vztah k teplárně. Mezi tato data je nutno zařadit i data neformální.

Komentář k faktoru F5:

Výše uvedené interní faktory mají za důsledek výskyt tohoto externího faktoru, který může být způsoben vzájemnou neinformovaností jak ze strany teplárny směrem k odběrateli tak i nedostatečnou úrovní znalostí o potřebách odběratele.

Komentář k faktoru F6:

Stejně problémy, které se týkají nebezpečí ztráty existujících zákazníků existují i v oblasti získávání nových možných zákazníků. Příčinou jsou mj. nedostatečné nebo neúplné informace o možných zákaznících, které je však nutno získávat především z neformálních zdrojů.

Shrnutí SWOT faktorů

V tomto shrnutí uvádíme pro jednotlivé SWOT faktory jejich působení (tj. zda se jedná o slabé/silné stránky interní nebo externí ohrožení/příležitosti) a

Informační systémy 1

jejich odhadovaný celkový vliv na výsledky firmy. Celkový vliv byl stanoven expertním posouzením jednotlivých faktorů a jejich složek.

Faktor	Název faktoru	Klasifikace SWOT faktoru	Působení faktoru
F1	Neznalost vývoje hlavních odběratelů tepla z hlediska budoucích potřeb odběru tepla.	Slabá	2
F2	Neadekvátní struktura údajů o hlavních odběratelích.	Slabá	1
F3	Nedostatečná zpětná vazba na odběratele.	Slabá	2
F4	Operativní informace o odběratelích převažují nad strategickými.	Ohrožení, slabá	1
F5	Hrozba ztráty odběratelů.	Ohrožení, slabá	3
F6	Nepodchycení všech potencionálních zákazníků.	Slabá	1

Tabulka 6.3: Shrnutí SWOT faktorů

Z tabulky můžeme vyčíst, že zřejmě nejdůležitějším faktorem je faktor F5, jelikož se jedná o kritický faktor (nejvyšší priorita 3). Ztráta odběratelů by měla přímý vliv na chod firmy, je tedy jasné, že tento faktor je velmi důležitý a navrhované akce by měli být provedeny.

6.4 Formulace poslání podniku

Dalším bodem konceptuálního modelu globální podnikové strategie, který následuje po provedení SWOT analýzy je formulace poslání podniku. Je třeba stanovit kritické faktory rozvoje podniku – ty jsou vymezeny ve SWOT analýze. Následně je třeba na identifikované kritické faktory adekvátně reagovat. To znamená začít uskutečňovat navrhované akce ze SWOT analýzy z důvodu odstranění negativně působících faktorů či udržení a zlepšení pozitivních faktorů.

Vedení firmy by si mělo položit tyto otázky:

- Jaké potřeby chce firma uspokojit?
- Pro jaké skupiny zákazníků?
- V jakém teritoriu?
- Jakou technologií?

Je nezbytné na tyto otázky odpovědět.

6.5 Definice globálních podnikových cílů

Třetím bodem tvorby modelu je definice podnikových cílů. Globálních cílů podniku může být několik, záleží na tom, z jakého hlediska je bereme. Může to být například z hlediska:

- vlastníků podniku,
- vrcholového vedení,
- pracovníků podniku,
- společnosti.

Cílem vlastníků podniku může být zatraktivnit firmu za účelem jejího prodeje. Vrcholové vedení podniku se může snažit zlepšit chod procesů a cash flow. Ve spojení s lepším cash flow pak zlepšit finanční sílu a možnosti podniku. U pracovníků společnosti může být cílem dobré pracovní prostředí, adekvátní odměna za odvedenou práci a dobrý systém motivace.

6.6 Vymezení globálních podnikových funkcí

Má-li podnik stanovené cíle, lze přikročit k vymezení jeho funkcí, což je také dalším bodem návrhu konceptuálního modelu globální podnikové strategie. Stejně jako SWOT analýzu provádíme pro všechny faktory (interní i externí, organizační jednotky, apod.), tak i funkce podniku definujeme pro všechny jeho části – organizační jednotky. Počínaje vrcholovým řízením organizace a marketingem přes nákup, prodej, výrobu, skladování, služby až po výzkum a vývoj, ekonomiku či informační systém organizace.

6.7 Tvorba strategií pro globální podnikové funkce

Předposledním bodem modelu je tvorba strategií pro různé funkce podniku. Tak například v marketingové strategii budeme definovat jak sbírat, skladovat a využít data o našich zákaznících v souladu se zákonem, jakou mají mít strukturu a obsah, způsob a rozsah reklamy výrobků, prezentaci firmy veřejnosti, apod. Finanční strategie může obsahovat způsob využití volných finančních prostředků, nastavení úvěrové politiky, způsob výpočtu penále pohledávek nebo definici vzájemných zápočtů mezi firmami. Personální strategie se zabývá stavem zaměstnanců, jak početním, tak znalostním. Definujeme systém hierarchického postupu, odpovědností pracovníků, odměn a výhod. Navrhujeme školení a další kvalifikační kurzy, také rekvalifikační kurzy s možnostmi spolupráce s pracovními úřady, apod. Informační strategie definuje potřebu a možný rozvoj informačních zdrojů v organizaci.

Kontrolní otázky:

1. Co definuje globální strategie podniku?
2. K čemu slouží SWOT analýza?
3. Jaký je rozdíl mezi interním a externím faktorem?



Úkoly k zamyšlení:

Představte si, že máte za úkol vytvořit nový systém pro celorepublikového dopravce osob. Tento dopravce provozuje stávající systém, který naleznete na adrese <http://www.vlak-bus.cz>. Zamyslete se a napište, jak byste formulovali poslání tohoto podniku (dopravce)?



Korespondenční úkol:

V úkolu k zamyšlení jste se měli pokusit formulovat poslání podniku. Nyní se zamyslete a podle Vašich zkušeností se stávajícím systémem (viz opět <http://www.vlak-bus.cz>) se pokuste vytvořit SWOT analýzu tohoto systému pro potřeby návrhu nového systému.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s globální strategií organizace. Poznali jste, že konceptuální model tvorby globální strategie definuje hlavní zaměření podniku, podnikové cíle a jejich priority, definuje také zdroje pro realizaci cílů, způsob ověřování jejich naplňování a osoby odpovědné za jejich dosažení. Seznámili jste se se SWOT analýzou a jejím použitím.



7 Objektově orientované technologie, UML

V této kapitole se dozvíte:

- Jaké jsou základní pojmy a principy v objektově orientovaných technologiích?
- Základní informace o několika objektově orientovaných metodikách.
- Co je UML? Jakých využívá diagramů?
- Jaké jsou znaky, požadavky a architektury objektově orientovaných systémů řízení báze dat?

Po jejím prostudování byste měli být schopni:

- Popsat a porozumět pojmům z OOP.
- Vyjmenovat a popsat některé objektově orientované metodiky.
- Porozumět modelovacímu jazyku UML a jeho diagramům.
- Popsat architekturu, požadavky a principy OOSŘBD.

Klíčová slova této kapitoly:

Objekt, třída, zpráva, generalizace, specializace, stav, událost, agregace, kompozice, zapouzdření, objektově orientovaná metodika, OMT, Booch, OOMT, UML, OOSŘBD.

Doba potřebná ke studiu: 8 hodin

Průvodce studiem

Tato kapitola se zabývá objektově orientovanými technologiemi. V první části si připomeneme objektově orientované principy a pojmy. Druhá část se zabývá objektově orientovanými metodikami a modelovacím jazykem UML. Poslední část pojednává o architektuře a požadavcích na objektově orientované databáze.

Při studiu podkapitoly o UML, doporučujeme veškeré diagramy vytvořit a modifikovat, pro základní pochopení smyslu a konstrukce prvků těchto diagramů. Na studium této části si vyhraďte alespoň 8 hodin.

Objektové technologie zahrnují databázové technologie, objektově orientované programování, objektově orientovanou analýzu a návrh, modelování procesů, implementaci a testování. V tomto textu se zmíníme o metodikách, analýze a návrhu, částečně i o implementaci a testování a také o objektově orientovaných systémech řízení báze dat.

Strukturované metodiky využívají funkčního a datového modelování, které má své výhody i nevýhody. Při snaze tyto dva modely sjednotit se nepřišlo na nic lepšího než, že se oba modely budou dál vyvíjet paralelně a pořád se budou porovnávat. Podstatou použití modelů je to, že jeden aspekt systému model rozpracovává, ale další opomíjí. Druhý model rozpracovává tyto opomíjené aspekty ale opomíjí aspekty rozpracované v prvním modelu. Aby změna jednoho modelu ovlivnila model druhý, je nutné, aby měly oba modely společný alespoň jeden rozpracováváný aspekt. Modely, které se vyskytují v UML nejsou odtrženy jeden od druhého, ukazují systém z různých úhlů a vzájemně se doplňují. U některých je dokonce možnost přímo transformovat

Informační systémy 1

jeden model do jiného. Strukturované metodiky mají většinou dobře propracovanou fázi analýzy, ale hůře jsou na tom ve fázi designu. Snaží se utíkat k různým nemetodickým postupům.

Koncem 80.let nastala krize ve strukturovaném programování. Software byl značně nespolehlivý a měl malou užitnou hodnotu, navíc se neúnosně zvyšoval podíl údržby takového SW (přes 60% celkových nákladů). Náklady na analýzu, návrh a programování činily pouhých 20% nákladů. Přes 70% software nebylo nikdy dokončeno nebo nebylo nikdy použito. Navíc zde existovala sémantická mezera v pohledu na systém. Z pohledu vývojáře existovaly databáze, programovací nástroje 4 generace, grafické uživatelské rozhraní, client/server architektura, apod. Z pohledu uživatele existovaly zase věčné oblasti jako sklady a zásobování, nákup, prodej a finance, personalistika. Mezi těmito pohledy existovala sémantická mezera, neexistoval společný jazyk, kterým by mluvil vývojář a uživatel mu rozuměl. Všechny výše zmíněné důvody vedly k zavedení a využívání objektových principů jak v programování, tak později i v analýze a návrhu systémů.

Objektově orientované programování (OOP) umožňuje lepší využití kódu, než knihovny procedur. Navíc knihovny tříd zvyšují znovupoužitelnost již jednou napsaného kódu. Použitím principů OOP změnili úlohy svůj charakter. Z převážně dávkových úloh se staly úlohy iterační. Funkce má jeden vstup a jeden výstup, to znamená, že reaguje na stejný podnět vždy stejně. U objektů je tomu jinak. Objekt může reagovat na více různých podnětů a to na každý jinak. Objekt dokonce může reagovat jinak i na stejný podnět.

Metodicky bylo ve strukturovaném programování vymyšleno mnoho různých zásad a doporučení pro správné programování (oddělení programových modulů od definice rozhraní, utajování informací, abstraktní data). Jednalo se však o pouhá doporučení a tak neměla velký vliv na zmenšení výskytu chyb v programech. Většina těchto principů a doporučení je přímo obsažena v použití objektů a objektových přístupů, navíc jsou implementovány v definici objektově orientovaných jazyků. Porušení takových zásad tedy způsobí chybu již při kompilaci programu a tvůrce programu ji musí odstranit.

Velký úspěch OOP vedl k zavedení objektově orientovaných principů i do analýzy a návrhu informačních systémů. Objektově orientovaná analýza a návrh (OOAN) není pouze použití objektů a objektově orientovaných principů, ale zahrnuje i osvědčené postupy strukturovaných metodik. OOAN používá také nové myšlenky, které nebyly ve strukturovaných metodikách zahrnuty.

V této kapitole se zmíníme o objektových principech a metodikách a také o vizuálním modelovacím jazyku UML. Řekneme si, kdy vznikl, k čemu se používá, jaké využívá diagramy a modely k popisu systému a také si ukážeme použití a zařazení diagramů ve spojení s metodikou vývoje, tj. použití diagramů v jednotlivých etapách životního cyklu vývoje informačního systému. K demonstraci použijeme metodiku RUP – Rational Unified Process.

7.1 Objektově orientované principy

Dříve než přejdeme k výkladu UML a popisu OO metodik, zopakujeme si základní objektově orientované pojmy a principy jako jsou třída, objekt, asociace, abstrakce, dědičnost, zapouzdření či skládání objektů.

Objekt obsahuje jak data, tak i funkce. Datům se obvykle říká vlastnosti a funkcím metody nebo operace. Objekt je cokoli, u čeho má smysl popisovat

Informační systémy 1

vlastnosti. Objekt má svou identitu, lze jej odlišit od jiných objektů. Objekt má svůj obraz v reálném světě. Objektem může být Petr Novák, který je studentem Ostravské univerzity. Tento objekt má své vlastnosti (data): věk, výšku, váhu, pohlaví, dosažené vzdělání a také chování (metody): mluví, vidí, jí, studuje, apod.

Třída popisuje skupinu objektů obdobných vlastností (atributů), společného chování (operací) a shodných vztahů k jiným objektům. Petr Novák, student Ostravské univerzity je objektem. Pojem student Ostravské univerzity je třída objektů s podobnými vlastnostmi a chováním. Jedná se o zobecnění konkrétních studentů. Objekt je instancí (prvkem) třídy. Petr Novák je tedy prvkem třídy studentů Ostravské univerzity.

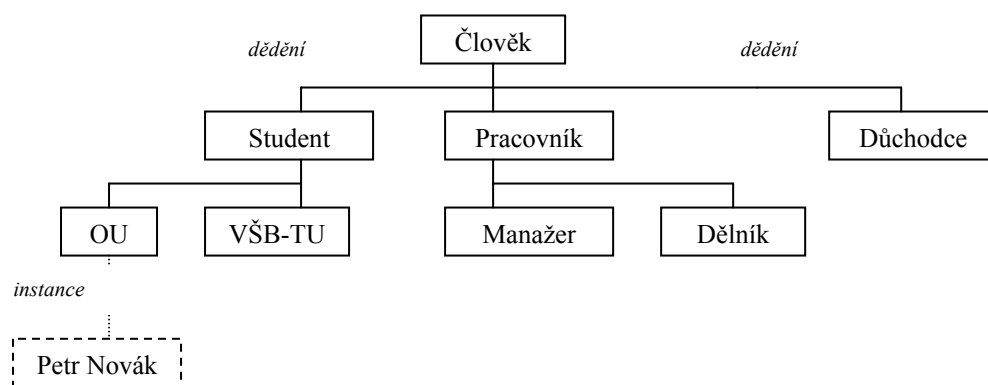
Atribut je datový údaj spojený s objektem např. věk, váha, barva, výkon, číslo účtu, IČO. Na rozdíl od objektu atribut nemá svou identitu. Atributy mohou být soukromé, z venku objektu nepřístupné nebo veřejné, které jsou z venku objektu všem přístupné.

Operace je akce nebo transformace prováděná objektem nebo na objekt aplikovatelná. Např. otevři, zavři jsou operace objektu Ventil; vlož, vyber, zobraz zůstatek jsou operace objektu Číslo Účtu.

Metoda je implementací operace. Například objekt Soubor má operaci tisk. Operace tisk musí být implementována různě pro odlišné typy souborů. Tisk souboru, který obsahuje bitovou mapu probíhá jinak než tisk ASCII souboru.

Polymorfismus je vlastnost, kdy tatáž operace může být v různých objektech realizována různým způsobem. Jinak budeme sčítat přirozená čísla, jinak komplexní čísla a jinak vektory. Tudíž pro třídy Přirozená čísla, Komplexní čísla a Vektory bude existovat metoda se stejným názvem sečti, ale každá bude obsahovat jinou implementaci, budou mít jiné vstupní a výstupní parametry.

Dědičnost umožňuje sdílení atributů a metod mezi třídami v rámci jejich hierarchie. Rodičovská, nadřazená třída je obecnější (*generalizace*), potomek je specializovanější případ svého rodiče (*specializace*), odlišuje se dalším atributem (atributy) a/nebo metodou (metodami), které přidává k existující množině atributů a metod zděděných od rodiče. Vytvářením potomků více konkretizujeme výslednou třídu.



Obrázek 7.1.: Hierarchie tříd

Na obrázku je znázorněn příklad jednoduché hierarchie tříd a jednoho objektu. Nejvýše je třída člověk, která je nejobecnější, může obsahovat pouze atribut jméno, váha, výška a metody jíst, spát, jít. Třída student je již speciálnější, může obsahovat navíc atributy dosažené vzdělání, studovaný ročník a metody studovat, dělat zkoušku. Stejně tak třída pracovník může obsahovat k atributům

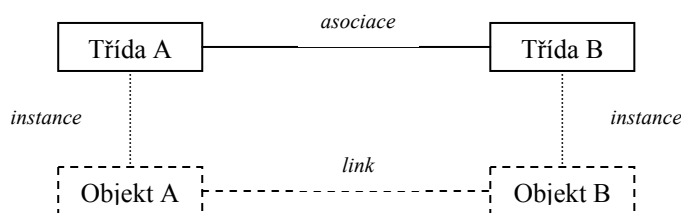
Informační systémy 1

zdeděným ze třídy člověk atributy jako dosavadní délka praxe, plat a metody pracuj, vezmi si dovolenou. Z tohoto popisu je zřejmé, že na čím výše třída stojí, tím je obecnější a naopak čím níže se v hierarchii nachází, tím je speciálnější. Objekt Petr Novák je instancí třídy studenti Ostravské univerzity, není tedy abstraktním typem studenta OU, ale je jedním konkrétním studentem, který má jedinečné číslo studenta a konkrétní hodnoty atributů jméno, příjmení, věk, studovaný ročník, stipendium nebo adresa.

Zpráva je požadavek, který posílá jeden objekt druhému. Zpráva vyžaduje, aby přijímací objekt provedl určitou operaci nebo poskytl určitá data. Mluvíme o zasílání zpráv mezi objekty.

Link je spojení mezi objekty, po kterém může být předána zpráva. Nemá-li objekt link na jiný objekt, nemůže mu zaslat zprávu.

Asociace je abstrakce linku, je to spojení mezi třídami, které může být realizováno například linkem.



Obrázek 7.2: Vztah mezi třídami, objekty a třídou a objektem

Zapouzdření neboli skrývání informací odlišuje vnější chování od vnitřní organizace. Objekty dané třídy mohou být zpřístupněny pouze přes dané rozhraní. Hodnoty atributů objektů nejsou obecně přístupné, uživatel smí používat jen nabídnuté operace. Jedno z typických použití zapouzdření u objektů je to, že data jsou soukromá. Jakoukoliv změnu dat nebo přístup na data lze provést pouze pomocí veřejné operace, která vstupní data kontroluje a tím zajišťuje, že objekt vždy obsahuje správná data. Tvůrce objektu rozhodne, která data jsou veřejná a která jsou soukromá, všichni ostatní, kteří používají tento objekt (třídu) musí jeho rozhodnutí respektovat. Nelze jít nějakou jinou cestou, která by zapouzdření obešla.

Kompozice a agregace (skládání) – jedná se o speciální formy relace mezi objekty. Agregace je volná vazba mezi objekty. Nejlépe si tyto formy relace vysvětlíme na příkladu. Počítač a jeho periférie (klávesnice, tiskárna) mezi sebou mají volnou vazbu. Periférie lze odpojovat a připojovat, může je sdílet více počítačů. Nejsou vlastněny žádným konkrétním počítačem. Kompozice je naproti tomu příkladem velmi pevné vazby mezi objekty. Příkladem může být strom a jeho listy. Listy jsou se stromem pevně spjaty. Listy jsou vlastněny právě jedním stromem, stromy si je nemohou navzájem půjčovat, ani měnit. Jestliže strom zemře, zemřou s ním zároveň i jeho listy. Toto je pevná vazba – kompozice. Agregace je vztah tranzitivní a antisymetrický. Je-li objekt 1 částí objektu 2 a ten částí objektu 3, pak je objekt 1 částí objektu 3 (tranzitivita). Je-li objekt 1 částí objektu 2, pak objekt 2 není částí objektu 1 (antisymetrie).

Událost reprezentuje vnější stimul objektu. Události mohou být na sobě závislé nebo mohou nastat nezávisle. Událost je jednosměrný přenos informace mezi objekty. Odezva objektu je také událost, události můžeme popsat pomocí

Informační systémy 1

atributů. Příkladem události může být: odjezd_vlaku (typ_vlaku, čas_odjezdu, odkud, směr).

Stav je abstrakcí zachycující hodnoty všech vnitřních proměnných objektu, stav objektu se může změnit jen příchodem nějaké události.

Každý objekt má *odpovědnost* za svou činnost za svůj svěřený úkol. V OOAN se můžeme ve fázi analýzy setkat s metodou CRC štítků, kde definujeme činnost, odpovědnost dané třídy. Ke každé třídě vytvoříme štítek se třemi oddíly, které obsahují název třídy, zodpovědnost (svěřený úkol) třídy a spolupracující třídy.

7.2 Objektově orientované metodiky

Strukturované metodiky analýzy a návrhu rozdělují modelování systému na:

- tvorbu datového modelu,
- tvorbu funkčního modelu,
- tvorbu modelu uživatelského rozhraní.

Naproti tomu objektový přístup neodděluje data a funkce, chápe je jako neoddělitelnou součást objektu. Tento přístup zavádí také nový způsob myšlení, technologickou kázeň a větší podporu počítačů ve fázi analýzy a návrhu použitím CASE nástrojů. V metodikách se využívá iteračního přístupu. Zadaný problém se řeší v přírůstcích, to umožňuje postupně upřesňovat cílový produkt cestou jeho inkrementálního rozšiřování z původního hrubého řešení až do výsledné podoby. Systém je vyvíjen průběžně v jakýchsi verzích, které lze navíc konzultovat se zadavatelem systému. Dalším rysem objektově orientovaných metodik je vizualizace modelu softwarového systému a vývoj s použitím softwarových komponent. Základem pro vizualizaci – průmyslovým standardem – je jazyk UML, který je primárně určený pro modelování softwarových systémů.

Tři základní modely objektové metodologie jsou:

- statický model – popisuje statickou strukturu, jednotlivé třídy objektů, jejich vzájemné vazby, třídy a operace,
- dynamický model – jedná se o model chování systému v čase, zachycuje tok řídicích informací,
- funkční model – popisuje transformace dat.

Modely jsou vzájemně propojeny a existují mezi nimi souvislosti.

Mezi hlavní přínosy OO technologií patří: členění do logicky definovaných etap, široký výběr modelů pro různé typy úloh, použití grafických schémat a s tím spojené použití CASE nástrojů, obsažené prostředky k sebekontrolě, přímý vztah ke konceptuálnímu modelování (objekty reálného světa lze reprezentovat jako složité objekty), část funkcí specifikovaných dříve v aplikačních programech se přenáší do definice objektu (zapouzdření a znovupoužitelnost). OOSRBD mohou navíc pracovat s bohatě strukturovanými objekty vhodnými pro modelování nejkomplicovanější reality.

Nyní se krátce zmíníme o několika objektově orientovaných metodikách. Od prvních, více používaných, až k dnešním metodikám přizpůsobeným pro nejlepší použití s vizuálním modelovacím jazykem UML. Cílem výčtu není graficky znázornit všechny používané modely, zmíníme se o nich pouze

Informační systémy 1

slovně. Většina z nich tvořila výchozí bod pro diagramy jazyka UML, tudíž bude jejich obdoba popsána při popisu UML.

Metodika OMT – Object modeling technique

Tvůrcem této metody byl James Rumbaugh. Základ vychází ze strukturovaného modelování, konkrétně z ER diagramu, který byl rozšířen o důležité objektové rysy. Později byl z této metodiky vypuštěn diagram DFD. Naopak byl do tohoto modelování vložen Use Case model, který pochází z metodiky Objectory/Object-Oriented Software Engineering a používá se ve fázi analýzy k vymezení hranic systému a interakci systému s vnějším okolím.

Díky tomu se tato metodika stává použitelnější a využitelnější ve všech fázích životního cyklu, na rozdíl od metody BOOCH (popis také v této kapitole). Životní cyklus je rozdělen do čtyř fází: analýzy, systémového návrhu, objektového návrhu a implementace. Během toho cyklu používá tři základní modely: objektový, dynamický a funkční.

Protože OMT má kořeny ve strukturovaném modelování je vhodný i pro datové modelování. Díky výše uvedeným vlastnostem je OMT jednou z nejpoužívanějších metod, zaujímá největší podíl mezi vývojáři CASE nástrojů a ovlivnila spoustu dalších metod objektově orientovaného modelování. Stala se také základem pro jednu z českých metod OOMT, vyvíjenou pod vedením P. Drbala na VŠE.

Výše již bylo řečeno, že analýza a design se dělí do tří částí. V následující tabulce je zobrazen životní cyklus této metody (bez implementace a testování) a využití jednotlivých modelů v dané fázi.

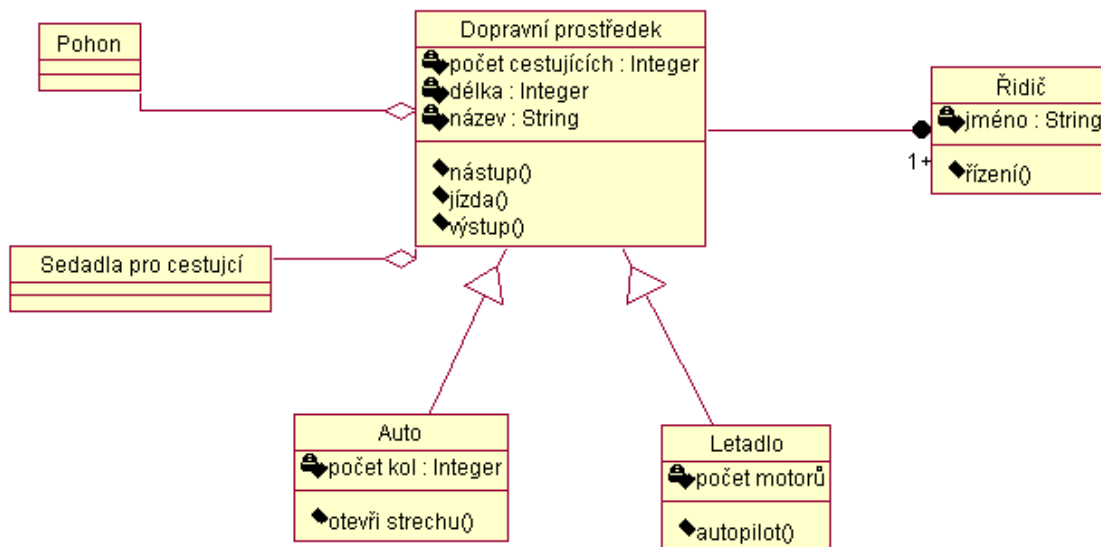
Co se týče části rozboru zadání, pak se v této fázi vytváří model jednání (Use Case), pojmový objektový model a část funkčního modelu, který je v této fázi reprezentován kontextovým diagramem. V některých publikacích se tato fáze zahrnuje do fáze analýzy.

Fáze návrhu	Model jednání	Model spolupráce	Objektový model	Dynamický model	Funkční model
Rozbor zadání	ANO		ANO		Zčásti
Analýza		ANO	ANO	ANO	ANO
Systémový design			ANO	ANO	
Objektový design			ANO	ANO	

Tabulka 7.1: Životní cyklus metody OMT

Na následujícím obrázku si můžeme prohlédnout notaci diagramu tříd, která je použita v metodice OMT. Tato notace je také převzata v české metodice OOMT, která je na OMT částečně postavena. Srovnej notaci s diagramem tříd v UML (dále v textu).

Informační systémy 1



Obrázek 7.3: Znázornění diagramu tříd v metodice OMT

Další objektové metodiky, které můžeme zmínit patří například Objectory od pana Jacobsona. Objectory se snaží podpořit celý životní cyklus vývoje softwarového produktu. Jednodušší verzí Objectory je OOSE (Object-Oriented Software Engineering), kterou Jacobson vytvořil jakožto metodologii pro tvorbu aplikací.

Metodika Booch

Autorem této metody je Grady Booch. Byla to jedna z prvních metodik objektově orientované analýzy a designu, která vznikla. V době svého vzniku byla spolu s OMT jednou z nejpoužívanějších objektových metodik. Některé z prostředků pro implementační fázi se později objevili mezi nástroji jazyka UML v pozměněné podobě. Tato metodika neobsahuje některé stěžejní prostředky pro fázi analýzy, čímž se stává použitelnou pouze pro oblast SW systémů.

Skládá se ze čtyř hlavních aktivit a šesti notací. Pokrývá oblasti analýzy požadavků a analýzu reality, avšak hlavní důraz je kladen na design. Z metodiky vyzařuje i silný vztah přímo k programování.

Booch nabízí různé pohledy pro popsání systému:

- *Fyzický model* – zahrnuje diagramy modulů a procesní diagramy, znázorňuje hardwarovou strukturu (procesorů, zařízení, síťových propojení) v kombinaci se softwarovými komponentami systému.
- *Logický model* – reprezentován strukturou tříd a objektů.
- statický a
- dynamický model.

Boochova metodika zahrnuje i popis postupu návrhu a tvorby informačního systému, neomezuje se pouze na popis nástrojů a diagramů. Definuje dva druhy procesů popisujících objektově orientovaný vývoj systému, a to makroproces a mikroproces.

Informační systémy 1

Mikroproces popisuje spíše každodenní aktivitu vývojářů. Zaměřuje se na zachycení obecně uznávaných, nutných kroků pro vytvoření diagramu tříd a diagramu objektů.

Prvky mikroprocesu: určení tříd a objektů na dané úrovni abstrakce, určení sémantiky těchto tříd a objektů (zachycení jejich vlastností), určení vztahů mezi těmito třídami a objekty, určení rozhraní a implementace těchto tříd a objektů.

Makroproces je kostrou shrnující kroky nutné při inkrementálním vývoji systému. Systém jako celek je vytvářen v postupných krocích, následující krok inkrementálního vývoje vždy o něco rozšíří funkčnost a strukturu vytvořenou v předcházejícím kroku. Je možné si všimnout, že kroky makroprocesu jsou klasicky uznávanou posloupností kroků při současném objektově orientovaném vývoji informačních systémů.

Prvky makroprocesu: určit základní požadavky (konceptuální část), vytvořit model požadovaného chování (analýza), vytvořit architekturu (design), implementovat, poimplementační údržba.

Kroky metodologie Booch:

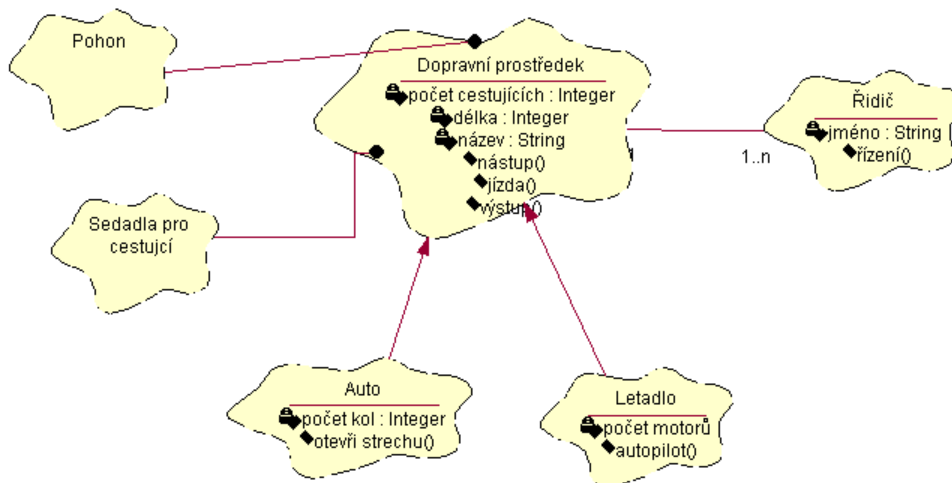
Logická struktura: diagram tříd, objektový diagram

Fyzická struktura: diagram modulů

Dynamika tříd: stavový diagram

Dynamika instancí: diagram interakcí

Na následujícím obrázku 7.4 je notace diagramu tříd v Boochově pojetí. Na první pohled se sice liší od pojetí pana Rumbaugh – obrázek 7.3, ale při bližším prozkoumání můžeme zjistit, že obsahuje podobnou notaci. Necháme na čtenáři samotném, aby posoudil, zda se tvůrci jazyka UML rozhodli správně a zvolili pro diagram tříd lepší a přehlednější grafickou formu vyjádření.



Obrázek 7.4: Příklad diagramu tříd v metodice Booch

Metodika OOMT – Objektově orientované metodiky a technologie

OOMT vznikla na katedře informačních technologií VŠE v Praze v rámci řešení grantu 402/95/1428 „Objektově orientovaná analýza a design“ GAČR. Výzkum probíhal v letech 1994-1996, poznatky z tohoto výzkumu byly také

Informační systémy 1

ověřeny v praxi. Popis metodiky je čerpán z [], obrázky byly z tohoto zdroje převzaty s drobnými změnami. OOMT se zaměřuje hlavně na fáze analýzy a designu. Autoři si kladli za cíl co největší univerzálnost a co nejmenší originalitu (netvořit nic dalšího nového, spíše využít dobré, již existující přístupy, modely) metody. Autoři dospěli k závěru, že klíčovým problémem návrhu informačních systémů je mentální zvládnutí složitosti. Proto se problém dělí na části a to pomocí 3 hlavních způsobů: etapizace, modely, rozklad na dílčí problémy.

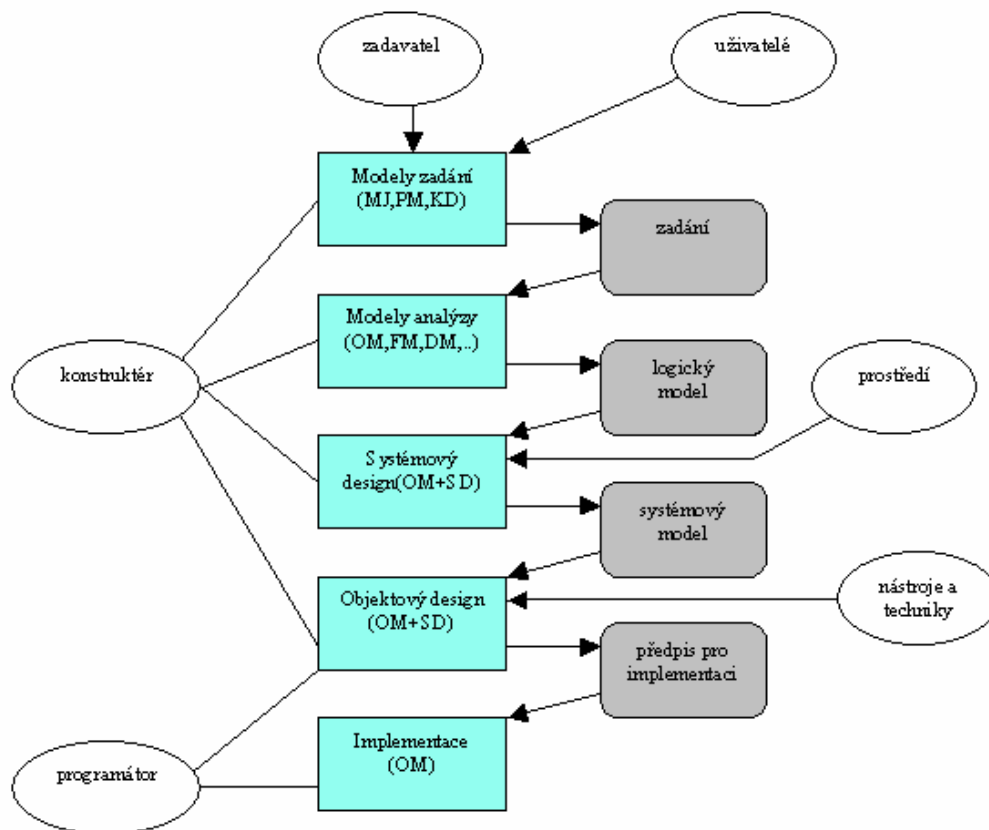
Etapizace je běžně používaný postup pro řešení složitějších problémů. Výsledkem je určení do jakých etap má smysl návrh systému dělit, jaké jsou vstupy a výstupy etap a jaké činnosti má smysl v nich provádět. Obrázek 7.5 znázorňuje etapy metodiky OOMT. Etapy napsané tučným písmem jsou etapy, na které se metodika OOMT zaměřuje především.

Modely jsou nepostradatelnou částí všech metodik (strukturovaných nebo objektově orientovaných). Modely se zabývají pouze některými aspekty řešeného problému, ostatní zanedbávají. Podstatou metody je vzájemné porovnávání a ovlivňování jednotlivých modelů. Na obrázku 7.6 jsou znázorněny jednotlivé modely včetně jejich tvůrců a zdrojů.



Obrázek 7.5: Etapy OOAN

Informační systémy 1



Obrázek 7.6: Použité modely v jednotlivých etapách návrhu

Legenda k obrázku 7.6:

Ovály značí zdroje informací nebo zpracovatele postupu, obdélníky jsou jednotlivé etapy návrhu, obdélníky s kulatými rohy znamenají výsledek etapy a/nebo vstup do etapy.

Zkratky v jednotlivých etapách jsou označení modelů:

- MJ** – model jednání
- PM** – pojmový objektový model
- KD** – kontextový diagram
- OM** – objektový model
- FM** – funkční model
- DM** – dynamický model
- SD** – stavové diagram

Rozklad na dílčí problémy je třetím způsobem dělení problému v metodice OOMT. Někdy se označuje také jako hierarchický rozklad. Je podstatnou částí při řešení složitých problémů. V metodice jsou použity dva typy rozkladu a to čistě funkční hierarchický rozklad ve funkčním modelu a druhý tzv. synergický rozklad, kde je problém rozložen na systém objektových tříd – na síť objektů. Při vývoji složitějších systémů je hierarchicky rozkládán celý postup. Takže například fáze návrhu obsahuje také rozbor zadání, analýzu, systémový design a objektový design. Podobá se to přírůstkovému způsobu vývoje IS.

Z výše uvedeného je tedy zřejmé, že metoda se skládá ze tří stavebních kamenů:

- modely,

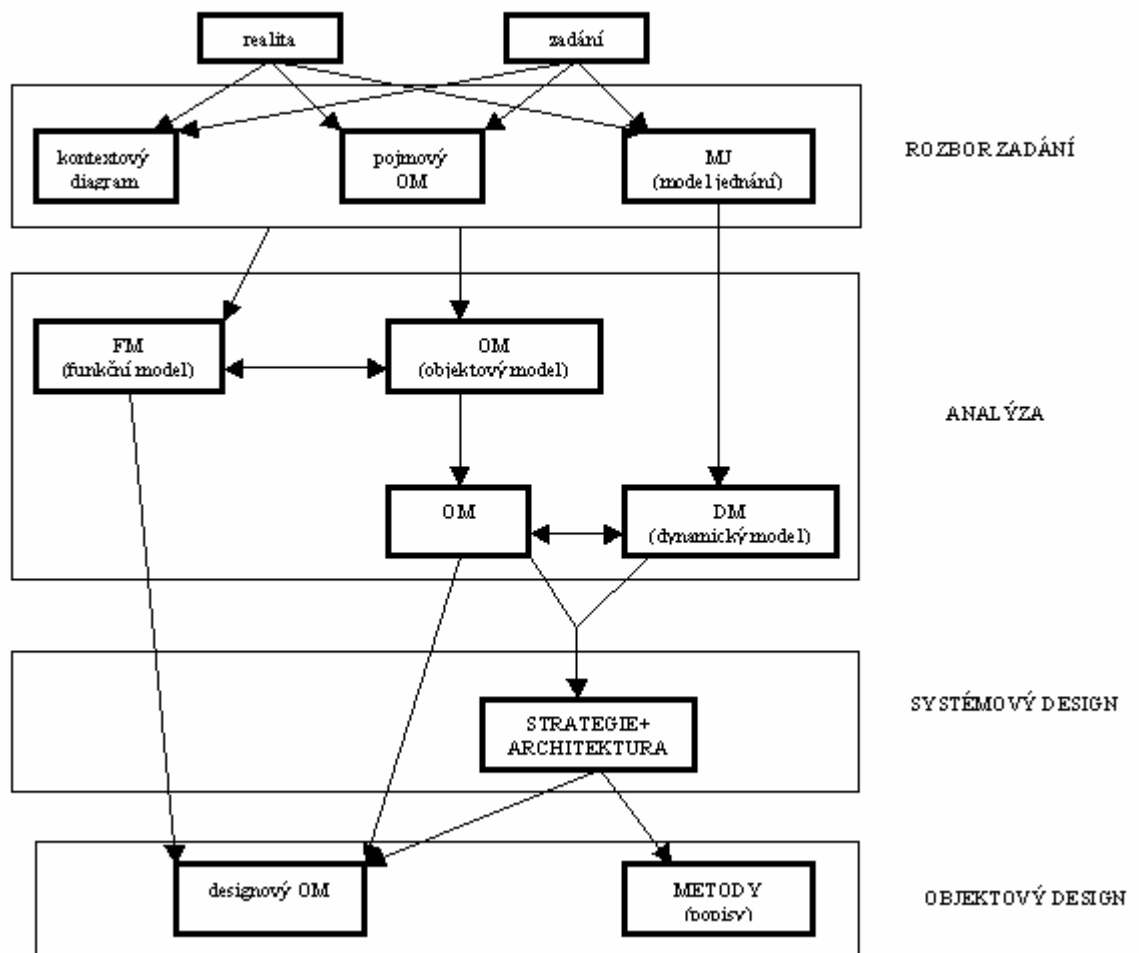
Informační systémy 1

- postupy pro tvorbu a vývoj jednotlivých modelů,
- postupy pro navázání dvou modelů.

OOMT rozlišuje 4 základní druhy modelů:

1. *Objektový model* – popisuje statickou stránku systému.
2. *Funkční model* – diagram datových toků, popisuje funkční požadavky na systém, rozkládá je až na úroveň jednotlivých operací.
3. *Dynamické modely* – popisují dynamickou stránku systému, kvůli složitosti je rozložen do více etap a používá se několik druhů modelů.
4. *Modely zadání* – popis zadání systému pro jeho lepší pochopení.

Vazby mezi jednotlivými modely a jejich využití v etapách návrhu systému znázorňuje obrázek 7.7. Dynamický model je zde znázorněn modelem jedním, jak však bylo napsáno výše skládá se z několika druhů modelů.



Obrázek 7.7: Hlavní modely metody a jejich použití v etapách

OOMT je síť prostředků, ze kterých má projektant na výběr a ze kterých si vytváří svou vlastní metodu. K dispozici má jednotlivé modely a jejich postupy, postupy jednotlivých etap řešení, návaznosti modelů a prostředky ke kontrole a synchronizaci modelů, doporučené postupy pro výběr metody.

Informační systémy 1

Je třeba také dodat, že metodika vznikla a začala se používat dříve, než byla schválena první verze UML, tudíž některé modely se zdají podobné modelům UML, ale notace není stejná. To je způsobeno tím, že autoři nevytvářeli své vlastní nové notace – viz snaha o co nejmenší originalitu jako cíl OOMT, ale přebrali některé osvědčené modely a metody z jiných metodik (např. OMT, Booch), ty pak byly také zapracovány a mírně upraveny v UML. Z tohoto důvodu je možné použít místo modelů v metodice OOMT zmíněných modely z UML (místo MJ model případů užití, místo OM diagram objektů, místo grafických scénářů sekvenční diagramy, apod.).

RUP – Rational Unified Process

Popis je čerpán z [white paper]. Rational Unified Process (RUP) je přímým následovníkem Rational Objektority Process (ROP). RUP zahrnuje správu dat, modelování business procesů a řízení projektu. Cílem platformy RUP je poskytnout snadno přístupný nástroj pro odhad rizik, usnadnění komunikace mezi všemi zainteresovanými osobami (podpora standardizovaných procesů), zlepšení přístupu k informacím (přes www rozhraní na stránkách www.rational.net - Rational Developer Network) atd. Samozřejmostí je implementovaná podpora běžných procesů a nástrojů, snadná rozšiřitelnost a síťová podpora.

Základní východiska RUP:

Iterativní vývoj softwaru (Develop Software Iteratively). V každé fázi vývoje se pozornost obrací na klíčové problémy, výsledkem fáze jsou snadno testovatelné produkty (zřetelné cíle, spustitelný software). Iterativnost celého procesu dává možnost snadno promítnout změny do systému.

Snadná správa požadavků (Manage Requirements). RUP jasně definuje prostředky a způsob dokumentace požadované funkcionality - model jednání a modely objektové interakce jsou zde důležitým prostředkem.

Využití komponent (Use Component-Based Architectures). Využití stávajících komponent (části s jasně definovaným účelem a rozhraním) je značným urychlením procesu vývoje, popřípadě znamená urychlení vývoje v budoucnu, pokud komponentu vyvíjíme sami.

Důraz na vizualizaci vyvíjeného softwaru (Visually Model Software). Moderní nástroje poskytují vhodný (vizuální) pohled na systém a dovolují využít další, snadněji ovladatelné produkty (různá grafická IDE). UML je ideální jazyk pro zachycení modelů a následné počáteční generování kódu.

Testování (Verify Software Quality). Testování je nedílnou součástí vývoje softwaru a RUP se snaží objektivními technikami podporovat testování během celého procesu vývoje.

Snadné zapracování změn (Control Changes to Software). Iterativní vývoj vyvolává množství změn, které je nutné ihned promítnout do systému - RUP poskytuje vhodné řídicí nástroje (od obecných premis až po konkrétní kroky), které pomohou udržet vše v přijatelných mezích a pod kontrolou.

V procesu RUP lze rozpoznat dvě roviny - dynamickou a statickou. Dynamická složka zachycuje, jak proces probíhá v čase a postupně prochází fázemi, cykly a dosahuje milníků. Bývá označována jako časová dimenze. Statickou složku představují činnosti, artefakty a stále se opakující činnosti probíhající v každé fázi (workflows).

Informační systémy 1

Časová dimenze

Časovou dimenzi tvoří čtyři fáze, na konci každé fáze je jeden ze čtyř hlavních milníků.

Inception phase – cílem je věcně vymezit projekt a na tomto základě definovat jeho cíle. K věcnému vymezení slouží model jednání, kde jsou zachyceny externí entity systému (aktoři) a jejich interakce se systémem (typy jednání). Složitější typy jednání se blíže popisují. Nedílnou součástí je nalezení kritických rozhodnutí (snaha učinit je co nejdříve) a odhad požadovaných zdrojů. Výsledkem je Vize (dokument obsahující základní vlastnosti, hlavní omezení a požadavky projektu), počáteční model jednání (hotov z 10% - 20%), definice základních pojmů (možno vyjádřit doménovým modelem), zpráva s úvodním rozpočtem, faktory úspěchu, upozornění na kritická rozhodnutí, plán projektu, věcný model systému.

Elaboration phase - cílem je pochopit věcnou oblast, rozhodnout o architektuře systému a podrobněji rozpracovat celý plán projektu. Snahou celé metodiky je přesunout kritická rozhodnutí do brzkých stádií projektu, dokud je ještě čas na změnu. Právě proto, že jsou prováděna klíčová rozhodnutí, jde o nejdůležitější fázi projektu. Výsledkem by měl být spustitelný (a testovatelný) prototyp systému. Dílčí cíle: model jednání (hotov z 80%), doplňující (ne tak kritické) požadavky na funkcionalitu, budoucí architektura systému, prověřený plán projektu (podrobnější, upravený z předchozí fáze), předběžný uživatelský manuál (není podmínkou).

Construction phase - fáze je zaměřena na doplnění a implementování veškeré funkcionality (opět v několika iteracích podle složitosti projektu), včetně optimalizací. Je rozmyšleno nasazení systému a jsou napsány uživatelské manuály. Třetí hlavní milník znovu prověřuje systém, zejména jeho použitelnost a stabilitu.

Transition phase - zaměřuje se na dodávku systému uživateli, jeho drobné úpravy (beta testování): naplnění daty, školení koncových uživatelů, správců, konfigurace. Vše zakončuje poslední milník, na jehož konci stojí (v ideálním případě) použitelný a používaný systém, vytvořený s plánovanými zdroji.

Každá fáze RUP může být rozdělena do několika iterací (v závislosti na požadavcích a složitosti systému). Výsledkem každé iterace by měl být spustitelný a testovatelný systém (ať již určený pro vnitřní či vnější využití).

Statická struktura

Statická rovina RUP popisuje kdo (pracovníci), co (artefakty), jak (činnosti) a kdy (pracovní postupy) dělá. Pracovník představuje osobu nebo skupinu osob (tj. tým), která má odpovědnost za určitou odvedenou práci tj. činnost. Do češtiny je nejhůře převeditelné slovo artefakt (artifact). Artefakt představuje libovolnou informaci, jež se během projektu využívá a během projektu vzniká, popř. to, co tuto informaci nese: model, prvek modelu, zdrojový kód, apod.

Existuje několik hlavních pracovních postupů, jež by měly proběhnout vždy (paralelně, s různou intenzitou během fází): modelování procesů (Business Modelling), tvorba požadavků (Requirements), analýza a design (Analysis & Design), implementace (Implementation), testování (Test), nasazení (Deployment).

Podpůrné pracovní postupy jsou: konfigurace a změnové řízení (Configuration & Change Management), správa projektu (Project

Informační systémy 1

Management), vývojové prostředí (Environment) - zahrnuje podpůrné nástroje pro postup v procesu, jako nástroje modelování a tvorby systému.



Obrázek 7.8: Struktura RUP (zdroj – Rational Software White paper)

7.3 UML – Unified Modeling Language

UML není metodikou, je to pouze vizuální modelovací nástroj. S žádnou konkrétní metodikou také není svázán. Lze jej použít se všemi existujícími metodami. Některé jsou pro jeho použití více vhodné, jiné zas méně. Nejlépe adaptovaná metodika pro použití UML je metodika UP – Unified process a z ní vycházející RUP (popis viz výše). Záměrem pro vznik UP a UML byla podpora nejlepších postupů používaných v metodikách a v softwarovém inženýrství. V UML a UP byly sjednoceny (unifikovány, jak již vypovídají názvy) předchozí pokusy o tvorbu vizuálních modelovacích jazyků. UP a RUP nejsou jedinými metodikami, které jazyk UML využívají, další je například OPEN (Object-oriented Process, Environment and Notation⁵). V této kapitole zmíníme pouze stručný popis UML a jeho modelů. Pro podrobnější seznámení s tímto jazykem doporučím [8] nebo anglické knihy přímo od autorů tohoto jazyka [9], [10].

Krátce o historii jazyka

Do roku 1994 byl ve světě objektově orientovaných metod chaos. Existovalo několik jazyků pro vizuální modelování a také několik metodik. Každá přinesla něco nového, většinou v oblasti notace. Více než polovinu trhu používala pro vizuální modelování metody Booch a OMT, mezi metodikami si vedla nejlépe Objectory od pana Jacobsona. Jedním z prvních pokusů o sjednocení byla metodka Fusion (1994). Byla sice poměrně hodně medializována, ale do její přípravy nebyli zapojeni tvůrci výše zmíněných metod s největším podílem na trhu (Booch, Jacobson, Rumbaugh). Fusion se neujala. Autoři Booch a Rumbaugh se spojili ve firmě Rational Corporation a začali pracovat na tvorbě jazyka UML. Byly určité obavy ohledně komercializace, ale jazyk UML se stal otevřeným standardem.

⁵ Více lze najít na adrese www.open.org.au

Informační systémy 1

V roce 1996 navrhlo sdružení OMG⁶ UML jako standard objektově orientovaného jazyka pro vizuální modelování. Standard byl v roce 1997 sdružením OMG přijat. Bez námitek ho přijala také veřejnost. Zatím poslední uvolněná verze je 1.4.

Jazyk UML přebírá to nejlepší, co doposud vzniklo a integruje to dohromady.

Rysy UML

Jazyk se neoznačuje unifikovaný jen proto, že integroval modely a postupy z jednotlivých, historicky starších, metod. Snaží se také o unifikaci různých domén.

- UML nabízí vizuální syntaxi pro modelování během celého vývojového cyklu, od analýzy až po implementaci.
- UML slouží pro modelování čehokoliv, podporuje různé aplikační domény (od real-time systémů až po expertní systémy).
- UML je nezávislý na programovacím jazyku. Nejlepší použití je samozřejmě s OO jazyky jako je Smalltalk, Java nebo C#. Je dobrý ale i pro hybridní jazyky jako je C++ nebo Visual Basic.
- O vnitřní jednotu a konsistenci se UML snaží svými vlastními interními pojmy.

Jelikož UML modeluje software a také obecně svět nebo systém jako kolekci vzájemně spolupracujících a ovlivňujících se objektů, je možné ho využít také pro modelování obchodních a podnikových procesů.

UML má 2 aspekty, které se vzájemně doplňují:

- *statická struktura* – popisuje typy objektů a jak spolu tyto objekty souvisejí,
- *dynamické chování* – popis životního cyklu objektů, vzájemná spolupráce s cílem dosáhnout požadované funkce navrhovaného systému.

Struktura jazyka

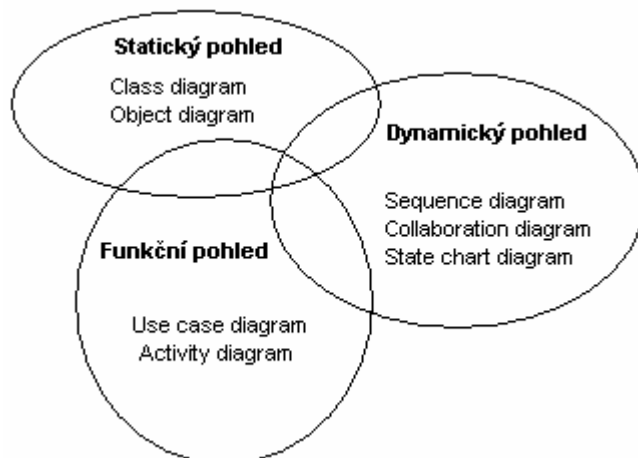
Struktura jazyka obsahuje tyto součásti:

- *Stavební bloky* – jsou to základní prvky modelu, pojítka mezi předměty (relace) a také diagramy, které ukazují pohledy na systém.
- *Společné mechanismy* – jedná se o obecné mechanismy, jimiž v UML dosáhneme specifických cílů, jsou to specifikace (textový popis sémantiky jednotlivých elementů), ozdoby (znázornění elementu symbolem), podskupiny (klasifikátory, instance a rozhraní, implementace) a mechanismy rozšiřitelnosti (mechanismy, které lze použít pro rozšíření jazyka).
- *Architektura* – pohled UML na architekturu navrhovaného systému. Definice organizační struktury systému, rozklad na části, propojitelnost, interakce, zásady.

⁶ Object Management Group je organizace, starající se o rozvoj myšlenek objektové orientace i mimo tradiční hranice programování (do oblastí modelování a metamodelování).

Informační systémy 1

Jedním způsobem, jak pohlížet na organizaci UML diagramů je využití pohledů. Pohled je kolekce diagramů popisující společné aspekty projektu. Tyto tři, vzájemně se doplňující, pohledy poskytují kompletní a přehledný obrázek systému nebo předmětu který modelujeme. Pohledy si můžeme prohlédnout na obrázku 7.9.



Obrázek 7.9: 3 pohledy na UML a jeho diagramy

Funkční pohled

Při vývoji nového systému nejdříve definujeme, jak by měl systém pracovat. K tomuto účelu mohou posloužit Use case diagram (diagram případů užití) a Activity diagram (diagram aktivit, činností). Diagram případů užití může doplňovat jak slovní popis – scénáře, tak popis problémové domény a funkcí pomocí diagramu aktivit. Diagram aktivit můžeme také použít pro modelování procesů a k popisu workflow.

Statický pohled

Tento pohled nám ukazuje elementy systému, ale neříká nám, jak se budou jednotlivé elementy chovat. Primárním nástrojem statického pohledu je diagram tříd. Poskytuje fixní pohled na každý zdroj – třídu a na její rysy. Tento diagram se používá pro generování kódu přímo z UML. Jako podpora nebo doplněk diagramu tříd lze využít objektový diagram (Object diagram). Tento diagram může odkrýt relace a informace, které by mohli v obecnějším popisu diagramu tříd zůstat skryté. Lze ho také použít pro kontrolu správného zakreslení diagramu tříd.

Dynamický pohled

Statický pohled znázorňuje, co se kde vyskytuje, jaké jsou mezi objekty vazby. Co z tohoto pohledu nemůžeme vyčíst je, jak tyto objekty spolupracují. Použití si přiblížíme na příkladu. Statický pohled nám pomůže popsat dům, kde vidíme vchodové dveře, kamna, termostat na stěně, schody a psa v bytě. Model interakcí nám ozřejmí závislosti mezi zmíněnými objekty. Co se například stane, když necháme otevřené vchodové dveře v zimním období. Manželka na nás může křičet, že pes uteče na zahradu, termostat pošle signál kamnům a kamna se mohou zapnout. Pro popis interakcí mezi objekty používáme

Informační systémy 1

Sequence a Collaboration diagram (diagram sekvencí a spolupráce). Dynamický pohled zahrnuje ještě State chart diagram (stavový diagram), který popisuje jak a proč se objekt a jeho reakce na prostředí mění v čase.

Existuje ještě jeden pohled, můžeme ho nazvat implementační. Ten popisuje výskyt hardware a rozmístění softwarových komponent (včetně vzájemných vazeb) na tomto hardware. Tyto diagramy se dají použít v etapách implementace a nasazení, ale také v etapě analýzy, kde pomohou upřesnit hrubou architekturu systému a vybrat vhodné řešení (lokální, síťová či distribuovaná aplikace, ...). Jedná se o diagramy komponent a nasazení (component diagram, deployment diagram).

UML diagramy v životním cyklu vývoje

Nyní si stručně popíšeme jednotlivé diagramy UML a ukážeme si jedno z možných použití diagramů ve fázích vývoje softwarového systému.

Business modelování

Smyslem business modelování je poskytnout společný jazyk pro softwarové inženýry a odborníky z oblasti podnikání. Jazyk UML pro potřeby specifikace business procesů využívá dvou typů diagramů:

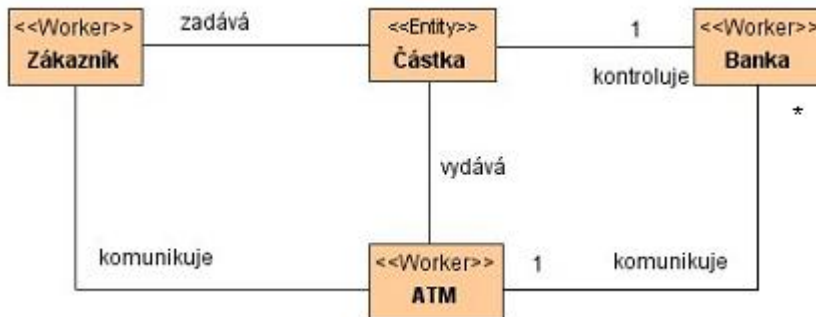
- diagram aktivit – jeho účelem je blíže popsat tok činností v dané oblasti řešení,
- diagram tříd – jeho účelem je zde zachytit statický aspekt modelované domény.

Diagram aktivit je zobrazen na obrázku 7.11. Diagram je rozdělen do tří pruhů, kterým se říká dráhy (swimlanes). Každá dráha definuje aktivity, které vykonává jednotlivý pracovník nebo určitý subjekt, stroj, systém, v našem případě jde o zákazníka, banku a bankomat ATM. Pokud jsou činnosti vykonávány paralelně, lze je rozdělit do paralelních (souběžných) vláken. Rozdělení a posléze i sjednocení probíhá pomocí rozdělovače toku – vodorovné silné čáry. Tok aktivit je možné rozdělit rozhodovacím blokem. Potom tok následuje jednou či druhou větví, podle toho, kterou podmínku splňuje (viz správně nebo špatně zadaný PIN). Podmínka se píše do lomených závorek. Dá se říci, že tento rozhodovací blok je synonymem pro příkaz if-else. Tento diagram popisuje chování určité jednotky.

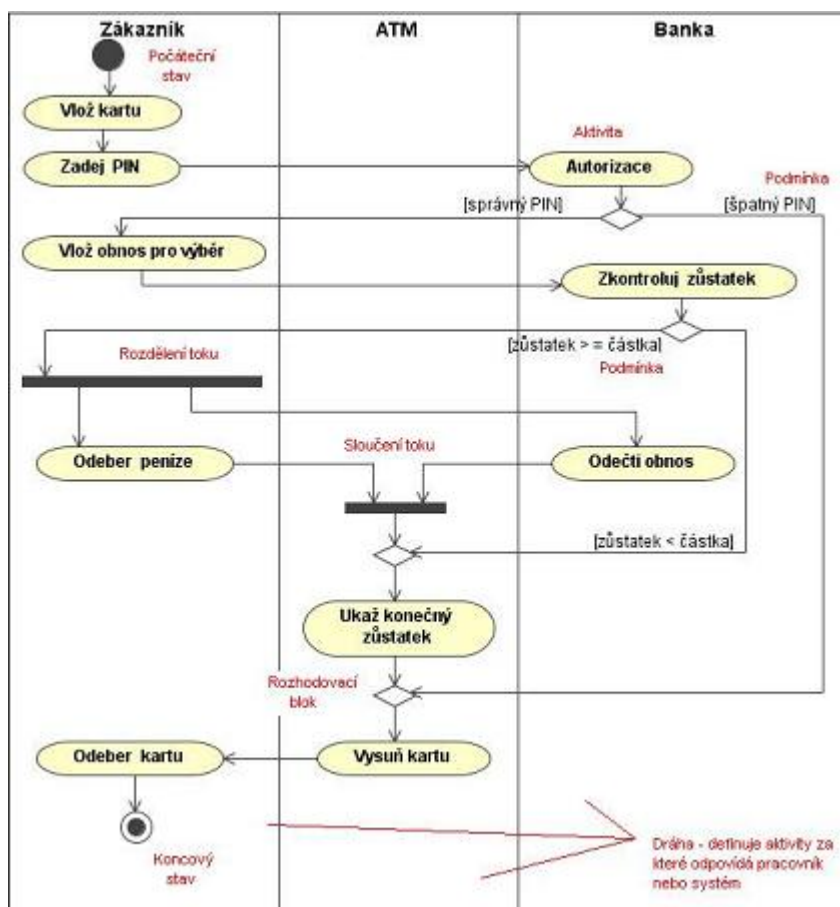
Druhým diagramem, který se v business modelování dá použít, je diagram tříd. Pomocí drah jsme definovali strukturální elementy. Diagram tříd znázorňuje statické vazby mezi jednotlivými elementy. Tento diagram obsahuje entity, které jsou označeny jako <<entity>> nebo <<worker>>. První entita je pasivní, druhá označuje aktivně působící pracovníky. Elementy jsou vzájemně propojeny statickými vazbami, které reprezentují jejich skutečné nebo konceptuální spojení. Tyto vazby mají přiřazenou násobnost, která definuje četnost výskytů. Příklad jednoduchého diagramu tříd využitého k business modelování je na obrázku 7.10. Slovo uzavřené do takovýchto <<slovo>> dvojitých znaků nazýváme stereotypem. Stereotyp umožňuje definovat nový element modelu UML, jenž je založen na existujícím elementu. V tomto případě jsme definovali třídy jako <<worker>> – aktivní třída a <<entity>> – pasivní třída. Dále je, pro rozšíření jazyka UML, možné použít

Informační systémy 1

omezení (constrains) a označené hodnoty (tagged values). Zapisují se do složených závorek, např. {Banka = Komerční banka, a.s.}.



Obrázek 7.10: Diagram tříd pro business modelování



Obrázek 7.11: Diagram aktivit

V diagramu aktivit lze znázornit vytvořené objekty (např. dokumenty) a také komunikaci mezi objekty – odesílání a přijímání zpráv. K tomuto účelu slouží speciální značky, o kterých se zde nezmiňujeme.

Specifikace požadavků

K této fázi jen krátce, jelikož se jí podrobně zabývá kapitola 5. Pro popis funkčnosti systému používáme diagram případů užití. Dále bychom měli mít podrobný popis specifikace požadavků včetně jejich priorit. Jednotlivé případy

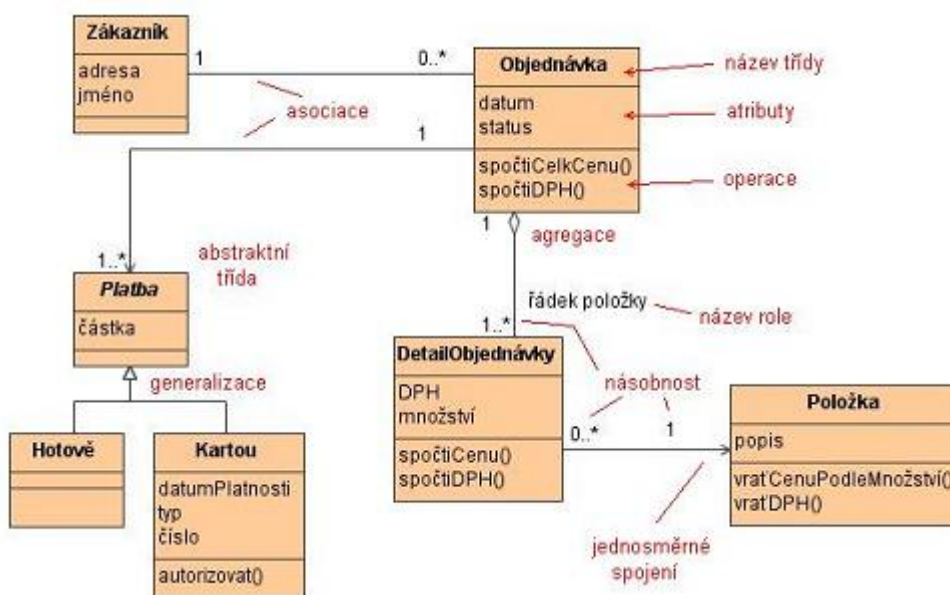
užití dále rozepisují slovní scénáře. Příklady diagramů, výčtu požadavků a slovních scénářů jsou obsaženy také v kapitole 5.

Analýza

Ve fázi analýzy je našim cílem hledání analytických tříd. Tyto třídy by měly mapovat pojmy skutečného světa. Podle toho by měly být také pečlivě voleny jejich názvy. Analytické třídy můžeme ve fázi podrobného návrhu upřesnit do 1 nebo naopak do více tříd. Analytické třídy mají většinou pouze klíčové atributy (tzv. adepty na atributy) a klíčové služby, z nich se ve fázi návrhu stanou skutečné atributy a metody. Analytické třídy by měly obsahovat minimálně název, nejdůležitější atributy, obecné operace (argumenty a návratové typy použít jen pokud je to nezbytně nutné k pochopení modelu). Obvykle nespecifikujeme typ viditelnosti – viz obrázek 7.12 (zda je atribut či metoda privátní nebo veřejná), stereotypy a označené hodnoty se používají pouze k vylepšení modelu (nutně se použít nemusí).

Pro hledání analytických tříd můžeme použít několik metod. První je analýza podstatných jmen a sloves. Podstatná jména a fráze jsou v podstatě vyjádřením tříd a atributů. Slovesa a slovené fráze jsou vyjádřením odpovědností nebo operací třídy. Další metoda je pomocí CRC štítků, její popis viz výše. Dalším způsobem hledání analytických tříd může být inspirace ve skutečném světě. Můžeme použít skutečné fyzické objekty (člověk nebo hotel budou vyjádřeni jako třída), kancelář a její vybavení (dokumenty – objednávky, nabídky, ...), známá rozhraní (klávesnice, obrazovky, ...) a to zvláště u vložených systémů.

V popisu jazyka UML jsme zmínili, že se mimo jiné skládá z ozdob. Pro pochopení pojmu ozdoba můžeme porovnat diagramy tříd na obrázku 7.10 a 7.12. Diagram na obrázku 7.12 je vyzdoben více než diagram na obrázku 7.10, má bohatší grafickou notaci. Toto vyjádření není úplně přesné, jelikož oba diagramy jsou použity v jiných souvislostech (business modelování vs. analýza), ale pro nástin pojmu postačí.



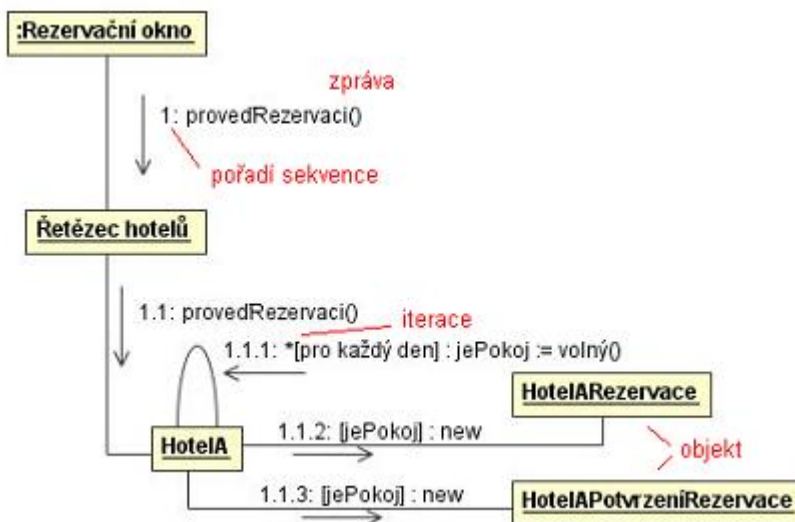
Obrázek 7.12: Diagram tříd

Informační systémy 1

Na obrázku 7.12 je také zachyceno několik druhů relací. Chceme-li vytvořit diagram tříd (objektový diagram), který popisuje realitu, musí spolu třídy nějak spolupracovat. K propojení slouží asociace (u objektového diagramu spojení). Spojení je cesta, po které si objekty navzájem posílají zprávy.

Pokud z našeho diagramu tříd vytvoříme instance objektů, bude mezi zákazníkem a objednávkou obousměrné spojení (oba objekty budou mít odkazy na sebe navzájem), kdežto mezi platbou a objednávkou existuje pouze jednosměrné spojení. Pouze objekt objednávka bude mít odkaz na objekt platba. Vztah mezi třídami platba, hotově a kartou nazýváme generalizace, opačným směrem specializace. Vyjadřuje dědičnost mezi rodičovskou třídou platba (ta je abstraktní – nelze vytvářet její instance) a potomky hotově a kartou. Na diagramu je ještě znázorněn příklad použití agregace. Třídy můžeme radit do tzv. balíčků (packages). Ty třídy seskupují do logických celků.

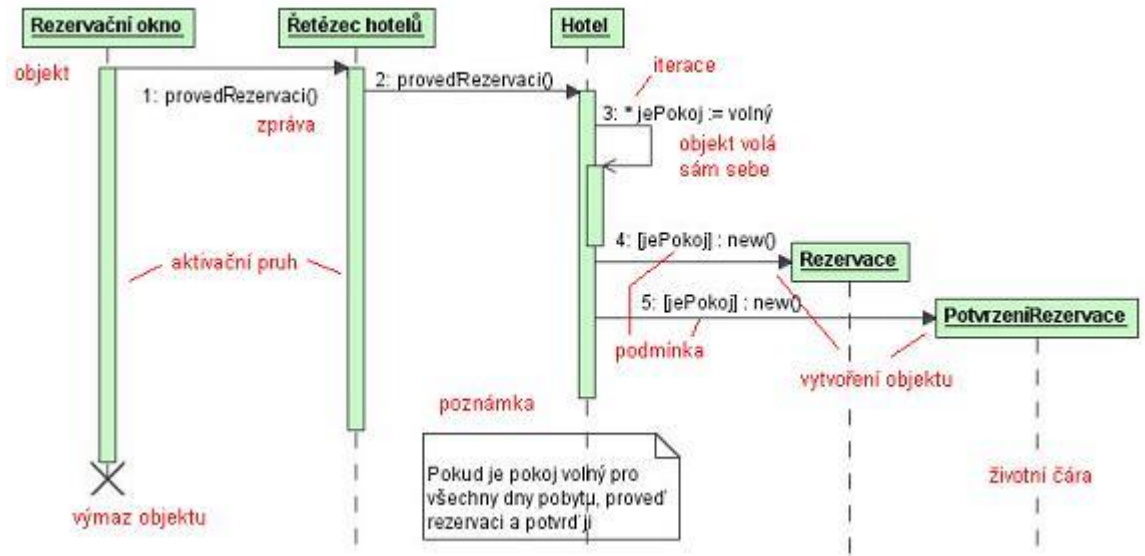
Ve fázi analýzy je další činností realizace případů užití. K tomu se používají diagramy interakce, což jsou diagram spolupráce a sekvenční diagram a podrobnější diagramy případů užití. Diagramy spolupráce ukazují jak spolu jednotlivé instance spolupracují, včetně časové následnosti jednotlivých zpráv. Na obrázku 7.13 můžeme vidět notaci diagramu.



Obrázek 7.13: Diagram spolupráce

Sekvenční diagramy zdůrazňují časově orientovanou posloupnost zpráv předávaných mezi objekty. Sekvenční diagramy jsou přehlednější než diagramy spolupráce a tak jsou většinou lépe pochopitelné. Tyto dva druhy diagramů jsou ve skutečnosti pouze dvěma různými pohledy na jeden model UML. Říkáme, že jsou izomorfní. Většina CASE nástrojů umožňuje automaticky převádět diagramy jednoho typu na druhý. Notace diagramu sekvencí je zachycena na obrázku 7.14. V české literatuře je možné se setkat také z názvem grafické scénáře, jedná se totiž v podstatě o graficky znázorněné slovní scénáře.

Informační systémy 1

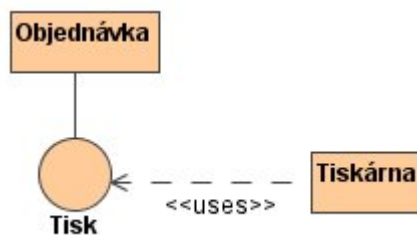


Obrázek 7.14: Diagram sekvencí

Zasílání zpráv v diagramu sekvencí může probíhat třemi způsoby. První je znázorněn na obrázku. Jedná se o volání procedury, odesílatel čeká, dokud příjemce nedokončí úlohu. Toto je nejčastější volba zasílání zpráv. Značí se plnou šipkou. Druhou možností je asynchronní komunikace. Odesílatel pokračuje ve své činnosti bezprostředně po odeslání zprávy, nečeká na odpověď příjemce. Značí se prázdnou šipkou → a používá se k modelování asynchronně probíhajících souběžných procesů. Poslední možností je návrat z volání procedury. Ten je vždy implicitní, lze jej však explicitně znázornit pomocí přerušované prázdné šipky.

Návrh

V této etapě pracujeme s návrhovými třídami. Ty získáme buď upřesňováním analytických tříd nebo z domény řešení (knihovny tříd – Date, String, kolekce; komponenty; ...). Návrhový diagram tříd obsahuje třídy, jejichž specifikace je na takovém stupni, že je lze implementovat. Návrhová třída by měla obsahovat úplný výčet atributů včetně jejich viditelnosti, názvu a datového typu (nepovinně lze znázornit i implicitní hodnoty), dále všechny metody, jejich název, viditelnost, návratové typy a argumenty. Návrhový diagram tříd má již také upřesněny všechny typy relace (agregace, kompozice, generalizace, reflexivní relace, ...), násobnosti, omezení, říditelnost. V této fázi návrhu upřesňujeme a specifikujeme rozhraní systému. Jeho notaci vidíme na následujícím obrázku.



Obrázek 7.15: Notace rozhraní

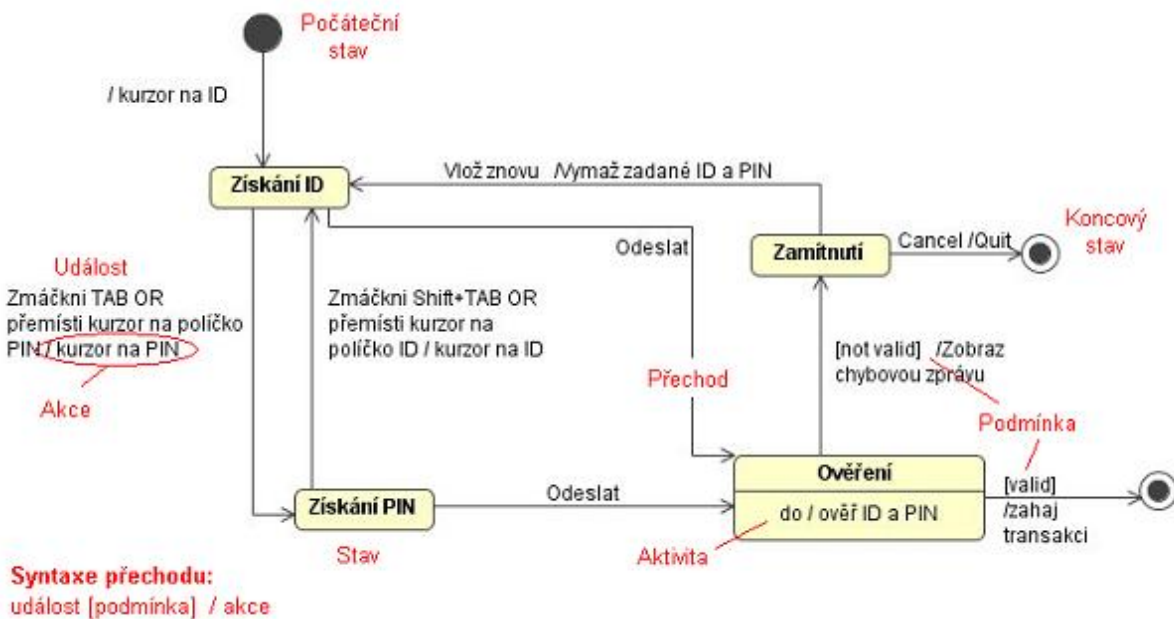
Informační systémy 1

Pro usnadnění návrhu se používají tzv. *návrhové vzory (Design Patterns)*. Návrhové vzory by měly napomoci sdílení a znovupoužití znalostí na vyšší úrovni mezi projektanty a návrháři. Jde v podstatě o zobecněná, již úspěšně použitá řešení. Definice vzoru obsahuje jeho název a použití, účel, scénář popisující použití, strukturu, omezení a další důležité popisy. Návrhových vzorů existuje velké množství, pro doplnění si zmíníme pár často používaných vzorů:

- továrna (factory),
- kompozit (composite),
- pozorovatel (observer),
- ...

Pro podrobnější studium lze doporučit knihu *E. Gamma a kolektiv – Design Patterns*, která existuje také v českém překladu.

V etapě návrhu se opět používají diagramy iterací. Můžeme buď upřesňovat klíčové analytické diagramy nebo vytvářet diagramy nové. Diagramy na tomto vývojovém stupni by měly objasnit mechanismy jako například perzistence objektů. Dalším druhem využívaných diagramů jsou diagramy stavové. Jedná se o důležitý diagram pro modelování dynamického chování objektů, které reagují na nějaké vnější události, tedy mění svůj stav. Stavový diagram obsahuje přesně jeden stavový automat pro jeden takový objekt. Syntaxi tohoto diagramu vidíme na obrázku 7.16. Popisuje proces zadání a ověření identifikačního čísla (ID) a PIN zákazníka u on-line bankovního systému.



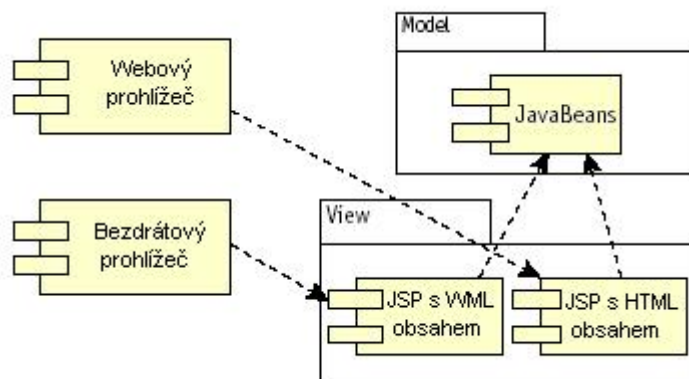
Obrázek 7.16: Stavový diagram

Implementace a nasazení

Implementace spočívá v převodu návrhového modelu do spustitelného kódu. Tvořit kód můžeme buď ručním programováním nebo generováním kódu z CASE nástroje. Je třeba vytvořit implementační model, který zahrnuje rozdělení návrhových tříd do komponent. Způsob provedení většinou závisí na volbě programovacího jazyka. Komponenta je fyzicky nahraditelná součást systému, která obaluje implementaci a poskytuje funkčnost pomocí specifikovaných rozhraní. Komponentou jsou například objekty EJB

Informační systémy 1

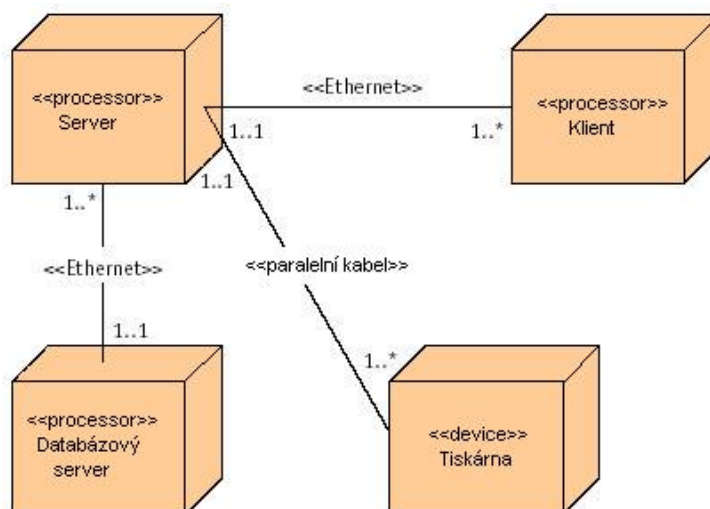
(Enterprise JavaBeans) nebo ovládací prvky ActiveX. Diagram komponent (Component diagram) znázorňuje komponenty a jejich rozmístění a vazby. Pomocí stereotypu můžeme definovat několik druhů komponent: například `<<executable>>` vykonávající kód, `<<table>>` reprezentující databázovou komponentu ke které přistupuje běžící komponenta, `<<file>>` reprezentující data nebo zdrojový kód.



Obrázek 7.17: Diagram komponent obsahující 2 pohledy a zohledňující MVC design

Diagram nasazení (Deployment diagram) ukazuje nejen fyzický hardware na němž bude systém spuštěn, ale i způsob, jímž je software na tomto hardware nasazen. Diagram nasazení se může vyskytovat ve dvou verzích:

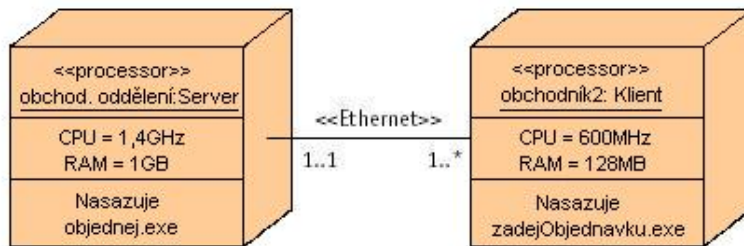
- *diagram nasazení* – obsahuje komponenty (typ softwaru – Internet Explorer, MS Word, http server Apache, ...), uzly (typ hardware) a vztahy mezi jednotlivými uzly.
- *diagram konkrétního nasazení* – obsahuje instance uzlů, relace mezi nimi a instance komponent. Instance uzlu zastupuje specifickou část HW, např. počítač číslo 6 v učebně A36. Instance komponenty zastupuje specifickou instanci SW, např. konkrétní kopie programu Poseidon CE použitá k tvorbě UML diagramů v této kapitole. Pokud neznáme podrobnosti nebo nejsou tolik zajímavé, můžeme použít anonymní instanci.



Obrázek 7.18: Diagram nasazení

Informační systémy 1

Syntaxe diagramu nasazení je jednoduchá (viz obrázek 7.18), uzly jsou zobrazeny jako krychle. Každý uzel nese označení typu fyzického HW, který reprezentují. Relace mezi dvěma uzly naznačuje existující spojení. V našem případě je mezi serverem a klientem linka Ethernet, mezi serverem a tiskárnou je spojení realizováno paralelním kabelem. Na libovolné uzly můžeme nasazovat komponenty, které na nich mají být instalovány nebo spouštěny. Tak můžeme diagram nasazení a komponent spojovat dohromady.



Obrázek 7.19 Diagram konkrétního nasazení

Diagram konkrétního nasazení se používá z toho důvodu, že je často třeba modelovat specifický hardware na straně klienta. Na obrázku 7.19 vidíme dva konkrétní uzly. Jeden reprezentuje server obchodního oddělení, obsahuje komponentu `objednej.exe` a má zobrazen výčet rychlosti procesoru a velikosti operační paměti RAM. Druhý uzel reprezentuje počítač konkrétního obchodníka, na kterém spouštíme komponentu `zadejObjednavku.exe`. Druhý uzel obsahuje taktéž specifikaci rychlosti procesoru a velikosti paměti RAM. Uzly jsou spojeny relací, která reprezentuje ethernetovskou síť.

7.4 Objektově orientované systémy řízení báze dat

Již jsme zmínili, že objektově orientované technologie zahrnují jak OOP (objektově orientované programování), OOAN (objektově orientovanou analýzu a návrh) a také OODB. Aby byl výčet kompletní, zmíníme se nyní také o OODB.

S rozvojem objektově orientovaných programovacích jazyků (počátkem devadesátých let) a se vzrůstající složitostí zpracovávaných dat, bylo třeba najít jiné způsoby prezentování a ukládání dat, než nabízí relační přístup. V 90. letech se staly OO jazyky standardem pro vývoj aplikací. Velký rozmach nastal také v oblasti GIS a stále více se začaly používat technologie pro podporu rozhodování, správu dokumentů, multimediální data a také CASE a CAD. Z těchto důvodů dochází k rozvoji a použití post-relačních, převážně objektově orientovaných, DBMS (DataBase Management System) ve specifických oblastech. Pro zajímavost, výzkum týkající se OODBMS, začal již na přelomu 70. a 80. let. Relační struktura databáze je totiž výrazně odlišná od objektové struktury aplikací, proto je třeba podnikat celou řadu mezikroků podporujících spolupráci relačních DB s OO aplikacemi. Je jasné, že tyto mezikroky nejen dělají aplikaci složitější, ale také snižují výkonnost a zvyšují riziko výskytu chyb. Se vzrůstající složitostí aplikace se prodlužuje i délka vývoje a testování a rostou s tím spojené náklady.

Vývoj a použití OODBMS lze chápat jako alternativu a rozšíření k relačním DBMS. Nástup OODBMS neznamena ústup nebo dokonce zánik relačních databází. Každý z těchto způsobů má své opodstatněné využití ve specifické

Informační systémy 1

oblasti nasazení. Výhodou RDBMS je jejich jednoduchost a pochopitelnost, mají základ v matematické teorii, jsou široce rozšířeny, mají pevné a známé standardy a implementují standardizované dotazovací jazyky (SQL, QBE). Na druhou stranu jednoduchost relačních databází znamená malou modelovací sílu. Pro složitější objekty nepřipustně zjednodušují realitu, znají pouze jednoduché datové typy, oddělují objekty od chování. Z těchto bodů je jasné, že RDBMS jsou dobré pro správu velkého objemu jednoduchých dat. Obsahují dobré prostředky pro selekci dat, ale komplikovaná je manipulace s nimi. OODBMS dobře vystihují složité vztahy mezi objekty, umožňují flexibilní manipulaci s daty, ale dotazovací možnosti jsou pod úrovní standardizovaného SQL. Oblast využití OODBMS je hlavně v CAD a GIS aplikacích, CASE systémech, systémech pro podporu rozhodování a pro správu dokumentů. Jde tedy o oblasti, kde pracujeme s persistentními objekty nebo s hustými grafy objektů.

Výhody OODBMS

Hlavní výhodou je možnost přímého vyjádření složitosti modelované reality v databázi. Díky tomu, že součástí uložených objektů je také jejich chování, zjednodušuje se struktura aplikace. Odpadají mezikroky převodu objektů do normalizovaných tabulek relační databáze před uložením každého objektu. Stejně tak je zjednodušen i opačný krok načítání objektů z databáze do aplikačního serveru. U RDBMS bychom museli jednoduchou strukturu uloženou v DB poskládat a vytvořit objekt, s kterým umí aplikace pracovat.

Objektové třídy, se kterými aplikace (OO programovací jazyk) pracuje jsou třídy používané v OODBMS. Modely databáze a programovacího jazyka jsou konzistentní, proto není třeba transformace programového objektového modelu do modelu specifického databázového systému. Z výše zmíněného plyne, že design objektové databáze tvoří velkou část designu celé aplikace. Objektový model OODBMS je postaven na běžném objektovém modelu, který je používán v OO programových jazycích či v OO metodikách návrhu a tvorby IS.

Požadavky na OODBMS

Hlavním požadavkem na OO databáze je perzistence. Objekt musí existovat i poté, co zanikl (byl ukončen běh) program, který ho vytvořil. Existují dva druhy objektů volativní a perzistentní. Práce s oběma druhy musí probíhat stejným způsobem, nesmí mezi nimi být rozdíl.

Volativní objekt je uložen v operační paměti. Perzistentní objekt je identifikován OID (objektovou identitou), což je ukazatel na perzistentní objekt. Perzistence je vlastnost instance, nikoliv třídy.

Pro jednoznačnou identifikaci záznamu v relačních databázích se používá primárního klíče. Pro jednoznačnou identifikaci objektu používáme OID – objektovou identitu, která nemá přímou obdobu v RDBMS.

- OID je nezávislá na místě (kvůli využití fragmentace, replikace databáze).
- OID je nezávislá na obsahu, je stejná i po změně hodnot objektu.
- OID je nezávislá na struktuře objektu, zůstává stejná i po změně struktury objektu.

Informační systémy 1

Identifikátor objektu OID není odvozován od atributů objektu ani od paměťové adresy, ale je systémově generován ve chvíli, kdy je persistentní objekt zapsán do databáze. Tento klíč je neměnný, unikátní a není recyklovatelný. S využitím OID lze jednoduše konstruovat vztahy mezi objekty.

Dalším požadavkem kladeným na OODBMS je *uchovávání verzí objektů*. Databáze by měla být schopna uchovávat historii vývoje objektu v čase.

Třetím požadavkem jsou *podmínky a akce*. Systém musí umět reagovat na splnění určitých podmínek vyvoláním příslušných metod objektů.

Požadavkem čtvrtým je *omezení*. Pro udržení konzistence databáze je nutné mít možnost omezit přístup k některým objektům

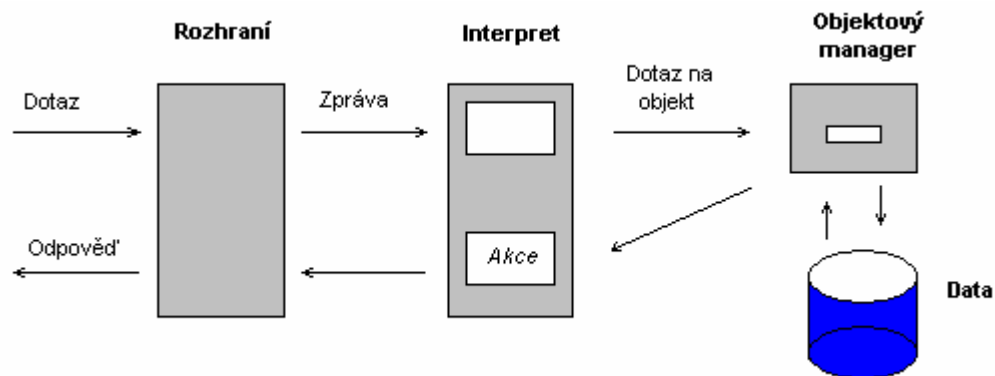
V databázi je třeba zachovávat konzistenci dat, proto existuje referenční integrita. Základním bodem referenční integrity je zajistit, aby po vymazání objektu byly odstraněny všechny reference na daný objekt. Toto odstranění může probíhat na aplikační úrovni prostřednictvím metod. Pomocí čítače linků udržujeme u každého objektu počet aktivních referencí na daný objekt. Tohoto způsobu využívá například databázový systém POET.

Existují 2 směry vývoje OODBMS. První směr používá již výše zmíněného objektového modelu a přidává k němu databázové prvky. Druhý směr používá relační DBMS, ke kterému přidává objektově orientované prvky. Pro objektově orientované databázové systémy, které využívají prvního směru, existují již také standardy, které definuje skupina ODMG (Object Database Management Group). Jde o pracovní skupinu, která je součástí OMG (Object Management Group). Produktem práce této skupiny je OQL a ODL. OQL (Object Query Language) je standard pro definici dotazů nad objektovou databází. ODL (Object Definition Language) je standard pro specifikaci a tvorbu objektově orientovaných databázových schémat.

Architektura OODBMS

Architektura objektově orientovaných DBMS je zachycena na obrázku 7.20. Skládá se ze tří podsystémů:

- *uživatelské rozhraní* – poskytuje prostředky pro formulování dotazů a jejich transformaci na posloupnost zpráv, které je třeba poslat, abychom dostali odpověď,
- *interpret zasílaných zpráv* – zprostředkuje doručení zprávy příslušnému objektu a provedení metody. Testuje syntaktickou a sémantickou správnost přijaté zprávy,
- *objekt manager* – najde a dodá požadovaný objekt, ať už v paměti nebo na disku. Stará se o uložení a údržbu dat na disku, práci s vyrovnávacími paměťmi.



Obrázek 7.20: Architektura OODBMS

Kontrolní otázky:

1. Jaký je rozdíl mezi agregací a kompozicí?
2. Co je to objekt, čím je definován? Co popisuje?
3. Jaké jsou 4 základní druhy modelů metodiky OOMT?
4. Jaké jsou tři pohledy na UML?
5. Co popisuje diagram sekvencí (sequence diagram) v UML?
6. Co je to OID, k čemu je potřeba?

Úkoly k zamyšlení:

V této kapitole jste se seznámili s objektově orientovanými principy a metodikami. Zamyslete se, jaký je rozdíl mezi strukturovanými a objektovými metodikami a proč se nyní používají převážně již metodiky objektově orientované?

Korespondenční úkol:

Pokuste se sestavit diagram tříd pro celou (většinu) datovou základnu systému, který eviduje zaměstnance a obchodní partnery, dále veškeré doklady a také bankovní účty a pokladnu. Důraz kladte na entity a jejich vztahy, úplnost atributů a metod není důležitá, stačí jen několik základních nebo vůbec žádné. Důležité doporučení: nesnažte se vycházet z ERD! Objekty a jejich vazby se modelují jiným způsobem! Viz principy a pojmy v úvodu kapitoly.

Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s objektově orientovanými technologiemi. Připomenuli jsme si pojmy jako objekt, třída, událost, stav, apod. V druhé části jsme si představili některé objektově orientované metodiky a také modelovací jazyk UML. Každému diagramu UML jsme věnovali několik vět a také vždy ukázali příklad diagramu. Poslední část pojednávala o architektuře a požadavcích na objektově orientované databáze.

8 CASE systémy

V této kapitole se dozvíte:

- Co jsou a k čemu slouží CASE nástroje?
- Jaké jsou druhy CASE?
- Jaké jsou komponenty CASE?
- Podle čeho CASE nástroj vybírat?

Po jejím prostudování byste měli být schopni:

- Popsat k čemu CASE slouží.
- Říct, jaké jsou jednotlivé druhy CASE.
- Popsat, jak CASE pokrývají vývojový cyklus IS.
- Vyjmenovat několik vlastností, kterých si při výběru CASE všimnout.

Klíčová slova této kapitoly:

CASE, automatizace vývoje, pre, upper, middle, lower, post CASE, komponenta, ISO.

Doba potřebná ke studiu: 2 hodiny

Průvodce studiem

Tato kapitola se zabývá podpůrnými softwarovými nástroji využívanými při vývoji IS. Popíšeme si jejich rozdělení a komponenty, z kterých se skládají. Na konci kapitoly zmíníme některé vlastnosti, kterých si je třeba všimnout při výběru CASE nástroje.

Na studium této části si vyhrad'te alespoň 2 hodin.

Vznik metodik vývoje software a diagramů pro znázornění systému z různých úhlů pohledu vedl k požadavku automatizace vývoje SW pomocí počítačů. Odpovědí se staly CASE – Computer Aided Software Engineering. CASE nástrojů existuje celá řada, a to jak pro strukturované, tak i pro objektově orientované metody vývoje. Některé CASE nástroje jsou integrovány do moderních prostředí pro vývoj software (např. firma Borland nebo Oracle). Přestože vypadá, že tvorba diagramů v těchto nástrojích je jednoduchá, vyžaduje vysokou znalost a profesionálnost tvůrce a těch, kdo je používají. Druhým předpokladem úspěchu je vhodnost použitých metod, na kterých je CASE založen. Podle obratu na trhu CASE je jasné, že o CASE nástroje je velký zájem. Použití CASE neznámá jen kreslení různých modelů, jde o silné nástroje pro zajištění souvislostí, které člověk mentálně neumí pojmout.

8.1 Druhy CASE systémů

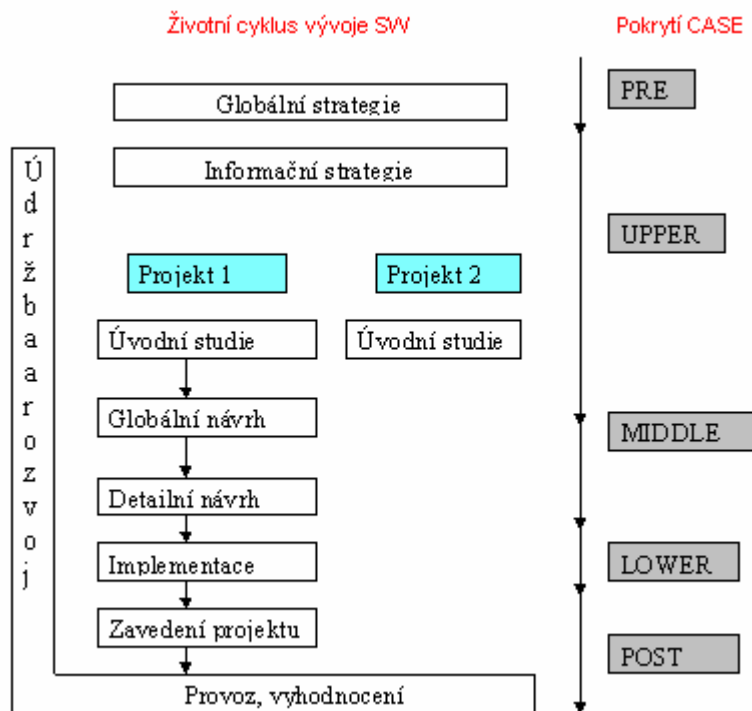
CASE systémy se využívají ve fázích specifikace požadavků, návrhu, kódování a údržbě. Nástroje použité v různých etapách se liší a je obvyklé, že pokrývají jen určité činnosti. Hranice mezi CASE a integrovanými vývojovými nástroji se postupně stírají (viz například CASE a také vývojový nástroj QI



Informační systémy 1

Builder informačního systému (QI). Podle životního cyklu vývoje software se CASE nástroje dělí do skupin:

- *Pre CASE* – podporuje tvorbu globální strategie.
- *Upper CASE* – podporuje plánování, specifikace požadavků, modelování organizace podniku a globální analýzu IS. Hlavním úkolem nástroje je analýza organizace, zobrazení procesů v organizaci, definice klíčových informačních toků a dokumentace zjištěných požadavků. Z těchto údajů je jasné použití při specifikaci cílů, počáteční specifikaci požadavků a řízení projektů. Cílem je pochopit danou oblast a specifikovat systém jako celek. Hlavní nástroje jsou DFD a jejich varianty, ERD bez podrobných atributů, prostředky pro řízení projektů a sledování ekonomických skutečností, popis základních vlastností systému prostředky OO modelování.
- *Middle CASE* – podporuje podrobnou specifikaci požadavků a vlastní návrh systému. Tato třída CASE nástrojů je neúspěšnější. Používají se pro podrobnou specifikaci požadavků, návrh systému, dokumentaci a vizualizaci systému. Obsahem středního CASE jsou metody a nástroje popisované v předchozích kapitolách: DFD včetně podrobného popisu procesů, datových úložišť, podrobné ERD, pro OOAN – diagramy tříd, instancí, přechodové diagramy, apod. Dále middle CASE obsahují systém správy dokumentů a konfigurace, systém pro vyhodnocování metrik, vývoj prototypů, návrh rozhraní, generátory obrazovek a sestav, generátory (kostry) definic dat. Hlavním cílem je formalizace specifikace a návrhu s možností snadných změn a komunikace se zákazníkem a také vytvoření modelů umožňujících generování návrhu. Tento druh CASE je jádrem komerčně dodávaných CASE systémů.
- *Lower CASE* – obsahují nástroje pro podporu kódování, testování a údržby, reverzního inženýrství. Integrované jsou nástroje jako generátory kódu (generují jen kostru nebo až $\frac{3}{4}$ výsledného kódu, programátor doplňuje většinou jen detaily), prostředky pro reverse engineering (rekonstrukce dokumentace a modelů z existujícího SW), prostředky pro sledování a vyhodnocení metrik, prostředky plánování a zjištění kvality SW (sběr informací o průběhu testování, vyhodnocení výsledků testů, řízení testování), správa konfigurace, prostředky sledování a vyhodnocování práce systému. Funkce dolních CASE se často překrývají s funkcemi obecných vývojových prostředí.
- *Post CASE* – podporuje organizační činnosti (zavedení, údržbu a rozvoj IS).



Obrázek 8.1: Pokrytí fází životního cyklu druhů CASE

8.2 Komponenty CASE systémů

Z toho, jaké jsou obecné funkce a vlastnosti CASE systémů vyplývá také z jakých komponent se tyto systémy skládají. Mezi důležité funkce a vlastnosti CASE systémů patří:

- *konzistentní grafické ovládací prostředí* (podle zásad tvorby GUI) – jednotný vzhled obrazovek, popisků, tlačítek, jednotné ovládání, použití symbolických ikon, apod.
- *centrální databáze* pro uchování informací o všech objektech IS (tímto způsobem se zaručí, že informace je použitelná v libovolném dalším kroku projektování),
- *prostředky verifikace konzistentnosti dat* a podpora normalizace dat,
- *textový editor* pro popis jednotlivých objektů – pro účely technické a uživatelské dokumentace systému, možnost jejího přímého generování ze systému,
- *možnost rychlého návrhu uživatelských obrazovek* včetně simulace vstupů a výstupů (je vyžadováno pro prototyping),
- *generátor zdrojových programů* (pro případy častého znovupoužití daného kódu až $\frac{3}{4}$ výsledného kódu),
- *export / import dat* – pro práci s modely a dokumentací, které byly vytvořeny v jiných programech nebo jsou v jiných programech dále využívány a zpracovávány.

Po vyjmenování základních vlastností a funkcí CASE systémů se zmíníme o jejich základních komponentách.

Grafický interface

Grafický interface se skládá z obrazových primitiv, která jsou předdefinována (kružnice, čtverce, přímky, křivky, šipky), existuje možnost

Informační systémy 1

jejich definování s minimální námahou a s možností uchování. Jedná se v podstatě o elektronickou tabuli, na kterou analytik konstruuje grafy a diagramy. Další požadavky na tyto primitiva jsou podpora kontrolních procesů, kontrola toků.

Grafický interface umožňuje vytváření grafů z těchto primitiv, přiřazování jmen grafickým objektům, sdružování jmen s objekty, kontrolu pozic jmen v diagramu. Umožňuje také editování vytvořených grafů (schopnost mazat, přepisovat, modifikovat grafické objekty). Úsilí jaké je třeba k vytvoření diagramu může být měřeno počtem stisků klávesy nebo kliknutí myši.

Další vlastností grafického interface je tzv. trickle down efekt při mazání, kdy systém automaticky vymaže odkazy na mazaný objekt, aby byla zachována konzistence modelu. Grafický interface by měl také umožňovat přemístění jednoho nebo více grafických objektů v tomtéž diagramu nebo za hranice diagramu.

Z dalších vlastností vyjmenujme přejmenování grafických objektů, obnovu objektů do jejich předchozího stavu (po výmazu, i více kroků zpět) – funkce „UNDO“ nebo změnu měřítka objektů.

Prohlížení grafů zahrnuje rotace grafů nebo jejich částí, podporu pozitivního a negativního zoomu (pokud jsou grafy větší než obrazovka), vytváření stránkových diagramů.

Vstupní interface

Pokud jsme zmínili grafický interface, je jasné, že musí existovat nějaké vstupy, kterými bude možno s danými grafickými primitivy či s grafy pracovat, manipulovat a také pomocí nich celý systém ovládat. Mezi vstupní zařízení řadíme jak standardní vstupní periferie počítače (klávesnice, myš), tak také další technická zařízení jako scanner, světelné pero a speciální hardware. Vstupní interface zajišťuje přenos vstupní informace do systému a grafickou interpretaci vstupních operací. Jakýkoliv pohyb myši, stisk klávesy, pohyb světelným perem musí být promítnut do systému a následně na obrazovku počítače (ve spolupráci s výstupním interface).

Výstupní interface

Existuje-li nějaké vstupní rozhraní (interface), mělo by existovat také výstupní. Výstupní interface se stará o provedení výstupů ze systému. Mezi výstupní technická zařízení může patřit monitor, tiskárna nebo plotter. Výstupní interface zahrnuje také definici výstupního formátu tisků – formát papíru, kvalita tisku, fonty a velikost písma, okraje, tituly, apod.

Podpora slovníků a integrace

Důležitou komponentou CASE systémů je slovník, jeho přítomnost v podstatě klasifikuje systém do rodiny CASE nástrojů. V některých nástrojích je slovník automatický (jakmile vytvoří uživatel objekt diagramu, je ve slovníku automaticky o tomto objektu vytvořen záznam). Velikost slovníku definuje maximální počet procesů, toků, datových objektů. U některých systémů může existovat omezení velikosti některých modelovaných komponent. Existují různé metody navigace (přístup k položkám databáze CASE) ve slovníku, přístup pomocí diagramu, přímí přístup do slovníku, apod.

Informační systémy 1

Slovník funguje většinou také jako textový procesor pro účely dokumentace. Ve slovníku je obsažena definice datových struktur (použitých datových entit), definice vztahů v hierarchii procesů (rodič-potomek) a v neposlední řadě také relace mezi daty a procesy.

Interakce přes obrazovku

Modelování systémů je doprovázeno možností vytvářet vstupně-výstupní obrazovky. Vývojová prostředí čtvrté generace tyto prostředky obsahují, u CASE nástrojů to však obvyklé není. Každý CASE, který generování obrazovek podporuje může mít jinou úroveň použití grafiky při definici obrazovek, jiný stupeň sjednocení mezi použitou grafikou a textem na obrazovce, jinou úroveň integrace slovníku dat s výstupem.

Analýza a generace zpráv

Důležitou vlastností CASE systému je možnost kontroly kvality modelu. Systém kontroluje tvořené modely a diagramy podle pravidel tvorby a podle definovaných logických souvislostí. Dále kontroluje izolované a nedefinované jednotky dat, procesy a moduly bez specifikace, uložení dat jako externí zdroje nebo self vazby entit (vazby na sebe samu). K těmto kontrolám je použit slovník, kde jsou uloženy vazby a významy jednotlivých entit. Na základě těchto kontrol jsou generovány zprávy, které uživateli oznamují možnost nebo nemožnost provedení daného kroku. Samozřejmostí by měla být podpora generování seznamů datových položek a jejich atributů.

Ostatní vlastnosti

Jelikož jsou CASE systémy ve velkých firmách používány projekčními týmy v síťovém prostředí, měly by podporovat automatickou kompletnost komplexního modelu z jednotlivých částí (které dělali zvlášť jednotlivý řešitelé týmu), výměnu dokumentů, různé formy komunikace a také použití centrálního slovníku.

Další vlastností CASE nástrojů je možnost připojení na externí databáze (např. pro generování schémat, skriptů nebo struktur databází nebo pro uložení slovníku).

Robustnost a ošetření chyb se vztahuje ke CASE systému v okamžiku nalezení či výskytu chyby. Systém musí snížit námahu uživatele při odstraňování chyby, musí zachovat všechny činnosti. Pokud se vyskytla nějaká ztráta dat, musí být data obnovena. Robustnost systému určuje jeho schopnost nezatuhnout, ustát různé kombinace nestandardních kroků, apod.

Pro běžnou práci je důležitá také odezva systému. Pokud je systém pomalý v odezvě na příkazy, při generování hlášení nebo změn v grafu, je problematické jeho rutinní využití.

Generování kódu nebo programu je také jednou z vlastností, podle které můžeme systém vybrat či hodnotit. Bereme v potaz dostupnost a složitost automatického generování kódu dle specifikace a typu uvedeného jazyka (jazyků), ve kterém je generování umožněno.

8.3 Volba a hodnocení CASE nástrojů

Při volbě moderních vývojových prostředí je třeba brát v úvahu následující kritéria:

- musí být vytvořeno vědomí formalizace metod a řízení projektu,
- předpoklad použití a znalosti blízkých metodik, na kterých je založen CASE,
- CASE by měl zahrnovat prvky procesního pohledu, měl by zahrnovat všechny nástroje střední úrovně a hlavní nástroje horní úrovně,
- výstupy CASE by měli umožňovat spolupráci s vývojovými prostředími a různými DB systémy, určité CASE mají zaměření na různá prostředí, která podporují více (většinou dáno jedním výrobcem vývojového prostředí i CASE),
- CASE by měl podporovat moderní architektury jako C/S, třívrstvé, komponentové a distribuované aplikace, také aplikace založená na WS – webových službách,
- měl by splňovat obecné podmínky otevřenosti, nezávislosti na HW, měl by mít kvalitní podporu ze strany dodavatele.

Pro ohodnocení CASE existuje dokonce norma IEEE, která uvádí několik desítek kritérií. CASE.

Součástí hodnocení CASE je také kvalita jeho podpory určité metodiky. Některé nástroje podporují strukturované i objektově orientované metodiky, většinou však buď strukturované (CASE 4/0 – Yourdan) nebo objektově orientované (Rational Rose - RUP). Pokud je již v organizaci zavedená nějaká metodika, pak námi vybíraný CASE systém musí tuto metodiku podporovat.

Vybraný CASE systém musí umožňovat podporu požadavků na modelování v reálném čase (modelování pomocí grafických nástrojů), správu konfigurací a verzí, správu projektu a pokud možno i správu dokumentů.

Jelikož CASE nástroje sledují ještě stále rychlý vývoj metodik, rychle zastarávají. Investice do CASE nástroje se vyplatí jen tehdy, jsou-li tyto nástroje systematicky a intenzivně využívány. Nástroje starší než 4 roky mohou být v dnešní době již morálně a metodicky zastaralé.

8.4 Zkušenosti s CASE a mylné představy

Použití CASE systému přinese pozitivní výsledky pouze v případě, že metody, na kterých je CASE založen, již tým používá a jsou mu blízké. Pouze samo zavedení CASE nezlepší řízení projektu, pokud do té doby žádné neexistuje.

Dalším úskalím může být odpor k novotám nebo přehnané očekávání. CASE by neměl být napoprvé použit v plné šíři u rozsáhlého softwarového projektu. Je dobré začít od grafických prostředků (ERD, DFD nebo diagramy tříd) a postupně zvládat další prostředky.

CASE nástroj slouží i zákazníkovi, je schopen najít chybu v ERD či DFD (po krátkém zaškolení). Osvědčují se i návrhy obrazovek. Diagramy ulehčují komunikaci v týmu a pomáhají zpřesnit analýzu systému. Za těmito výhodami stojí úspěch středních CASE. Dolní CASE již tolik úspěšné nejsou, zřejmě z již zmíněného důvodu pokrytí jeho funkcí běžnými vývojovými prostředími.

Co CASE systémy ještě dostatečně nepodporují je zavedení, sběr a analýza softwarových metrik a nedostatečné vazby na normy sady ISO 9000.

Jaké jsou některé mylné představy o těchto systémech, které jsme mohli slyšet, si zmíníme nyní. CASE není samo o sobě metodologií, ale používá již zavedené metodologie. CASE není náhradou programovacích jazyků (může generovat části kódu, ale programovací jazyky nenahrazuje). Všechny CASE systémy pracují podobně v tom smyslu, že poskytují stejné výstupy. Užívání CASE zlepší práci vedoucích pracovníků podniku nebo specialistů pro IT – CASE je nástroj, který může zlepšit produktivitu práce, efektivita práce vždy závisí na osobních kvalitách jednotlivých pracovníků.

CASE odstraňuje potřebu disciplíny a přísného vývoje aplikací IT. Praxe ukázala, že systémy CASE často selhávají právě díky nedisciplinovanosti uživatelů. Automatizací chaosu vznikne automatizovaný chaos.

Od CASE se často očekává jako výstup tvorba aplikačního programového vybavení. Hlavní přínos přitom je v úplném poznání fungování podniku a vytváření úplných podkladů právě pro programování aplikací. Produktivita dosažená pomocí CASE není okamžitě zřejmá – na počátku práce je nutné vykonat velmi mnoho práce, která není dlouho vidět.

Užívání CASE nezaručí konzistenci výstupů. Když se dá stejný CASE dvěma systémovým analytikům, dospějí k dvěma naprosto odlišným řešením.

Výčet několika známých CASE nástrojů:

- Power Designer od společnosti Sybase (definování DB struktur, tvorba kostry DB aplikací včetně menu, ...),
- CASE\4\0 německé společnosti microTOOL GmbH. (typický middle CASE),
- Rational Rose od společnosti IBM, dříve Rational,
- ORACLE Designer 2000 od společnosti Oracle (výborný a robustní v kombinaci s databází Oracle),
- System Architect od společnosti Popkin Software and Systems Inc.,
- Select Enterprise od společnosti Select (OO nástroj),
- a spousta dalších.

Kontrolní otázky:

1. Co znamená zkratka CASE?
2. K čemu CASE nástroje slouží?
3. Jaké etapy vývoje SW pokrývá middle CASE?
4. Co je úkolem komponenty CASE, která se jmenuje centrální databáze?

Úkoly k zamyšlení:

V této kapitole jste se dozvěděli, co je to CASE. Zamyslete se, zdali jste se již s takovým nástrojem setkali a kde (popř. v kterém předmětu)?

V předchozí kapitole bylo zmíněno, že pro podporu tvorby UML diagramů se využívá softwarových prostředků. Dají se tyto prostředky nazvat CASE nástroji?

Korespondenční úkol:

Vytvořte stručný přehled CASE nástrojů (název, výrobce, orientační cena, popř. podporovaná metodika a metody) dostupných na našem trhu. Pokuste se najít také nějaké české výrobce, pokud existují.



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili se softwarovými prostředky pro podporu tvorby IS. Dozvěděli jste se, že jejich použití na rozsáhlých projektech je nezbytné a vývojářům usnadňuje práci a komunikaci v týmu. Dále jste se dozvěděli jak tyto nástroje dělíme a z jakých základních komponent se skládají. Nakonec jsme si řekli, co je nezbytné vědět při výběru CASE nástroje.



9 Podnikové informační systémy

V této kapitole se dozvíte:

- Jaká je náplň jednotlivých částí (MIS, EIS, EDI, ...) podnikového IS?
- Z jakých modulů se skládá ERP systém, jaké procesy popisuje a jaké jsou jejich návaznosti?
- Jaké moduly by měly být obsaženy v podnikovém informačním systému?
- Jaké funkce by měly být obsaženy v těchto modulech?

Po jejím prostudování byste měli být schopni:

- Popsat úlohy IS jako je MIS, OIS, CAD apod.
- Vyjmenovat modely podnikového IS a popsat jejich základní funkce.

Klíčová slova této kapitoly:

Úloha, modul, ERP, funkce modulů, finance, obchod, výroba, zdroje, přeprava, servis, stavební výroba.

Doba potřebná ke studiu: 6 hodin

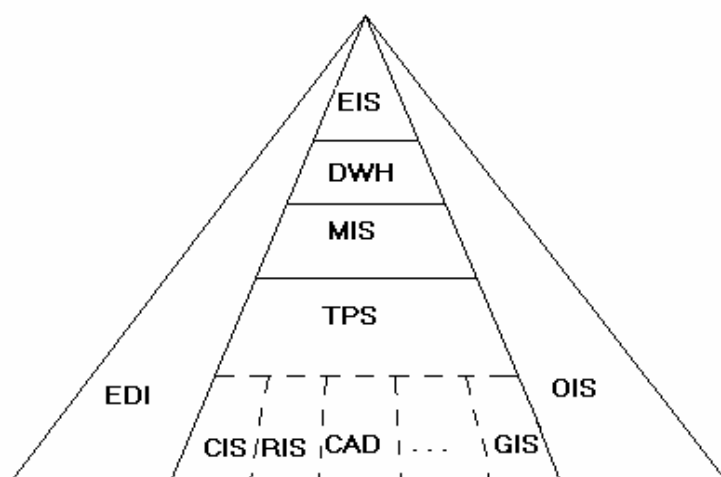
Průvodce studiem

Tato kapitola přiblíží podnikové informační systémy, jejich důležité funkce, které by měly implementovat. Toto rozdělení se mohou částečně měnit podle výrobců jednotlivých podnikových IS.

Na studium této části si vyhradte alespoň 6 hodin.

9.1 Úlohy podnikového informačního systému

V návaznosti na kapitolu 3 si znovu připomeňme architekturu podnikového informačního systému.



Obr. Příklad architektury podnikového systému

Informační systémy 1

Úlohy manažerské - typu EIS

Cíle a vymezení úlohy:

- úlohy poskytují komplexní analýzy aktivit podniku podle nejrůznějších kritérií, z nejrůznějších pohledů, a to s přímým využitím dat získaných v úlohách typu MIS, CIS apod.
- cílem úloh je připravovat podklady pro rozhodování na základě komplexních analýz,

Určení úlohy:

- úlohy jsou určeny především pro pracovníky nejvyšších úrovní řízení, stále častěji se však uplatňují pro operace analytického charakteru i na střední úrovni řízení,
- úlohy jsou orientovány na cca 30 - 40% technicko hospodářských a administrativních pracovníků podniku,

Technologická základna:

- úlohy jsou převážně založeny na tzv. multidimenzionálních databázích,
- pracují v síťovém prostředí, ale velmi často mají individuální charakter (podle potřeb jednotlivých vedoucích pracovníků),
- oproti ostatním typům úloh (MIS, CIS, datové sklady) pracují většinou s menšími rozsahy databází (desítky až stovky Mb),

Úlohy typu datový sklad (DWH)

Cíle a vymezení úlohy:

- cílem úloh je shromažďování vybraných informací z různých databází ostatních úloh do jednotného technologického prostředí a zpracování širokého spektra analýz,
- zajišťují zpracování analýz velkého rozsahu nad uloženými daty, a to i za poměrně dlouhá časová období,

Určení úlohy:

- úlohy jsou určeny většinou vedoucích pracovníků podniku, pracujících s analýzami dat za delší časová období,

Technologická základna:

- založeny na multidimenzionálních databázích, resp. databázích kombinujících multidimenzionální a relační technologie,
- oproti úlohám typu EIS pracují s podstatně větším rozsahem databází, od několika gigabytů až po terabyty dat,

Úlohy pro podporu taktického a operativního řízení - MIS

Cíle a vymezení úlohy:

- úlohy pokrývají všechny oblasti řízení organizace (finance, prodej, nákup,...) na taktické a operativní úrovni s převažující mírou evidenčních a analytických operací (aktualizace datových bází, zpracování základních přehledů, výběrů atd.),
- cílem úloh je zajištění průběžné evidence produkčních procesů a zdrojů podniku, zpracování požadovaných dokumentů daných legislativou i

Informační systémy 1

vnitropodnikovými směrnicemi, zpracování ekonomických a dalších analýz a podkladů pro rozhodování,

Určení úlohy:

- úlohy jsou primárně určeny pracovníkům na střední a nižší úrovni řízení, některé výstupy slouží i pro řízení na nejvyšší úrovni řízení společnosti,
- přímými uživateli úloh je naprostá většina technicko hospodářských a administrativních pracovníků společnosti,

Technologická základna:

- je založena převážně na silných relačních databázích (Oracle, Informix, Sybase, Progress, DB/400...) a jim odpovídajících počítačích,
- ve stále větší míře se uplatňují objektové přístupy k vývoji aplikací a jim odpovídající produkty,

datové báze průměrných podniků představují stovky gigabytů dat,

Úlohy elektronické výměny dat - typu EDI

Cíle a vymezení úlohy:

- cílem úloh je zajistit výměnu dat s obchodními partnery, případně dalšími ekonomickými subjekty v elektronické formě a dosáhnout potřebného zjednodušení a zrychlení obchodních procedur,
- v rámci těchto úloh dochází k výměně pevně strukturovaných dat na základě dohodnutých mezinárodních standardů (objednávek, smluv, faktur, celních deklarácí, ...)

Určení úlohy:

- úlohy jsou zaměřeny na potřeby pracovníků obchodních útvarů, finančních útvarů podniku, kde se realizují přímé kontakty se zákazníky, dodavateli, finančními ústavy, orgány státní správy,

Technologická základna:

- úlohy jsou realizovány pomocí speciálních programových modulů v rámci komplexních aplikačních software nebo specializovaných programových prostředků. V obou případech jde zejména o konverze dat (smluv, faktur,...) ze struktur a formátů odesílající aplikace do dohodnutého mezinárodního standardu a na místě příjmu jde opět o konverzi ze standardu do formátu přijímající aplikace,
- pro přenosy dat jsou většinou využívány veřejné počítačové sítě s přidanou hodnotou (VAN) a s nimi spojené služby, resp. infrastruktura Internetu,

Úlohy pro podporu kancelářských prací - OIS

Vymezení úlohy:

- redukce nároků na běžné administrativní operace (psaní textů, běžné kalkulace, ...) a zvýšení úrovně pořádku v administrativě podniku (správa dokumentů, dokumentace předávaných zpráv, ...)
- zrychlení a zkvalitnění běžné komunikace mezi pracovníky podniku i s pracovníky externích organizací - na bázi elektronické pošty,
- využití dostupných zdrojů nabízených v prostředí Internetu,

Informační systémy 1

- dosažení vyšší formální úrovně kancelářských prací - formální úrovně dopisů, zpráv, katalogů, ...

Určení úlohy:

- úlohy typu OIS jsou určeny nejširšímu okruhu pracovníků ze všech typů úloh, prakticky všem pracovníkům využívajícím počítačovou síť podniku,
- s ohledem na rozsah uživatelské sféry se kancelářské aplikace chápou jako páteř celé architektury informačního systému.

Technologická základna:

- úlohy typu OIS jsou založeny na integrovaných kancelářských balících, zahrnujících textové procesory, tabulkové kalkulátory, elektronickou poštu, grafické editory, pracovní kalendáře a plánovače, plánovače oběhu dokladů v podniku, příp. další produkty.

Úlohy výrobní - typu CAD/CAM

Cíle a vymezení úlohy:

- optimalizace řízení výrobních provozů, snižování provozních ztrát a nákladů,
- podpora zvyšování kvality výroby a služeb od návrhu až po přijímací kontroly,
- zefektivnění výroby automatizací.

Určení úlohy:

- úlohy jsou určeny pro provozní pracovníky na dílnách, stavbách a provozech (vedoucí provozu, mistry, dělníky),
- úlohy jsou ve své konstrukční a projekční části určeny návrhářům, konstruktérům, technologům.

Technologická základna:

- část těchto úloh je většinou realizována jako součást komplexních aplikačních software. Moduly technické přípravy výroby, operativního řízení výroby, dílenského řízení, kapacitního plánování a většinou se realizují na velkých relačních databázových systémech,
- podpora projekčních a konstruktérských prací je realizována specializovanými produkty (CAD - Computer Aided Design) s vysokou převahou počítačové grafiky,
- specializované produkty pak zajišťují podporu řízení výroby, projekční postupy,
- část úlohy orientovaná na řízení a plánování provozů se většinou projektuje na základě metodik uplatňovaných pro úlohy typu MIS a tedy řešení a nasazení komplexních aplikačních software, avšak s velkým důrazem na řešení rozhraní k ostatním produktům, zejména CAD,
- projektování úloh CAD a technologických úloh má zcela specifický charakter daný povahou těchto produktů (uplatnění grafiky, výběr efektivních grafických prostředků, ...).

Úlohy zákaznické - typu CIS

Cíle a vymezení úlohy:

Informační systémy 1

- cílem zákaznických úloh je zajištění základních administrativních operací spojených s evidencí spotřeby a spotřebitelů takových produktů, jako je elektřina, plyn, voda apod.,
- zákaznické úlohy zajišťují i evidenci a monitorování příslušných měřičů spotřeby, plánování jejich údržby, obnovy apod.,
- zákaznické úlohy zabezpečují i příslušnou fakturaci spotřeby, sledování pohledávek, jejich vyhodnocení a zajišťují i potřebné vazby a vstupy do úloh typu MIS (účetnictví, prodej,...).

Určení úlohy:

- charakteristickým rysem zákaznických úloh je jejich určení nejen vlastním provozním pracovníkům, ale i širokému okruhu spotřebitelů elektřiny, plynu, vody, tedy veřejnosti.

Technologická základna:

- zákaznické úlohy jsou většinou založeny na obdobných technologiích jako úlohy MIS, tj. silných serverech, většinou dislokovaných na jednotlivých rozvodných závodech a provozech dodavatele a na silných relačních databázích (Oracle, Sybase, Informix, Progress, DB/2),
- typickým rysem zákaznických úloh je uplatnění rozsáhlých počítačových sítí WAN,
- postupně se prosazují automatická měřidla s přímým vstupem do informačního systému.

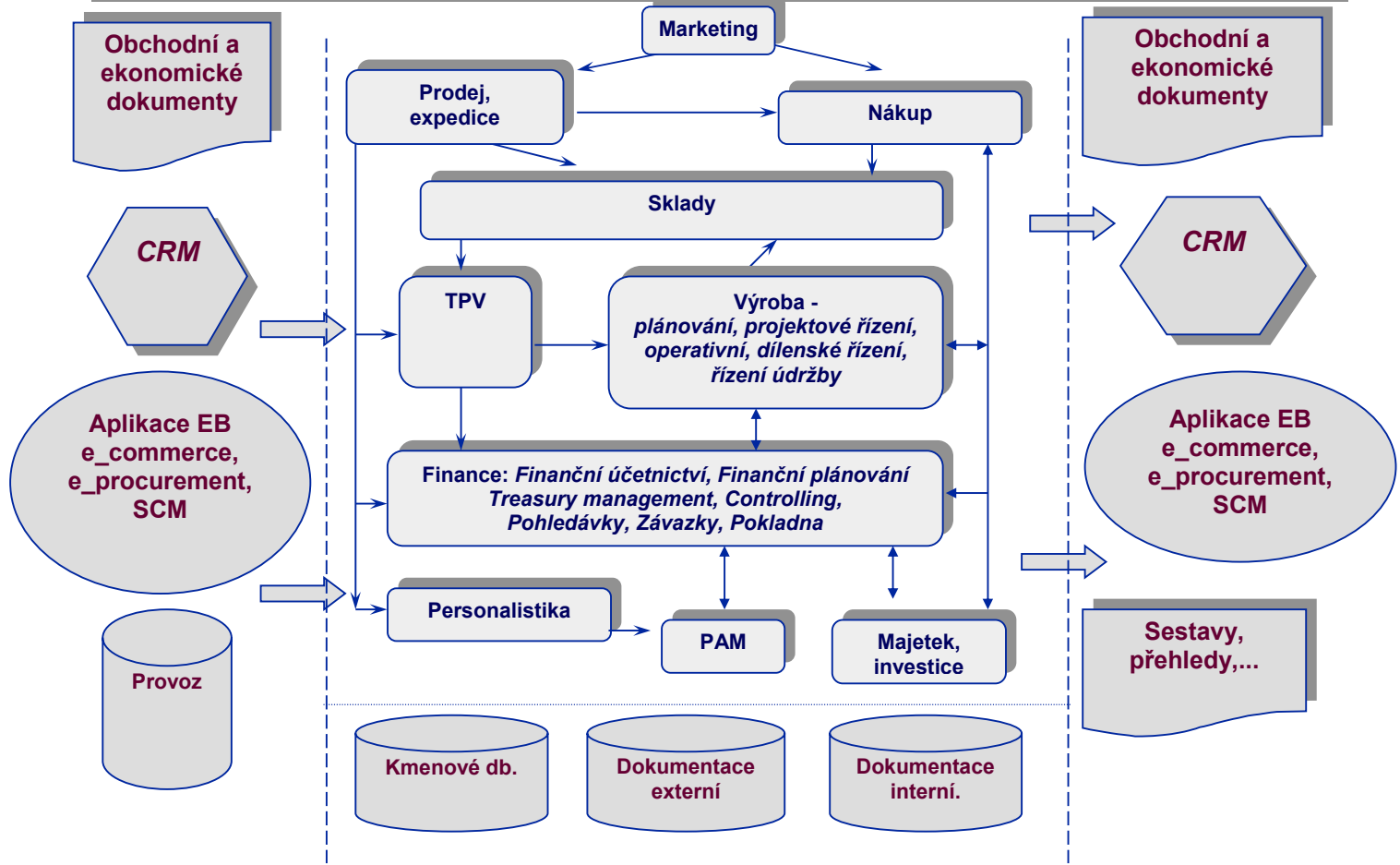
9.2 Jádro informačního systému podniku - ERP

Systémy ERP (Enterprise Resource Planning) vznikly spojením finančních a logistických úloh. ERP tak chápeme jako integraci vnitropodnikových oblastí, zejména výroby, logistiky, financí a lidských zdrojů.

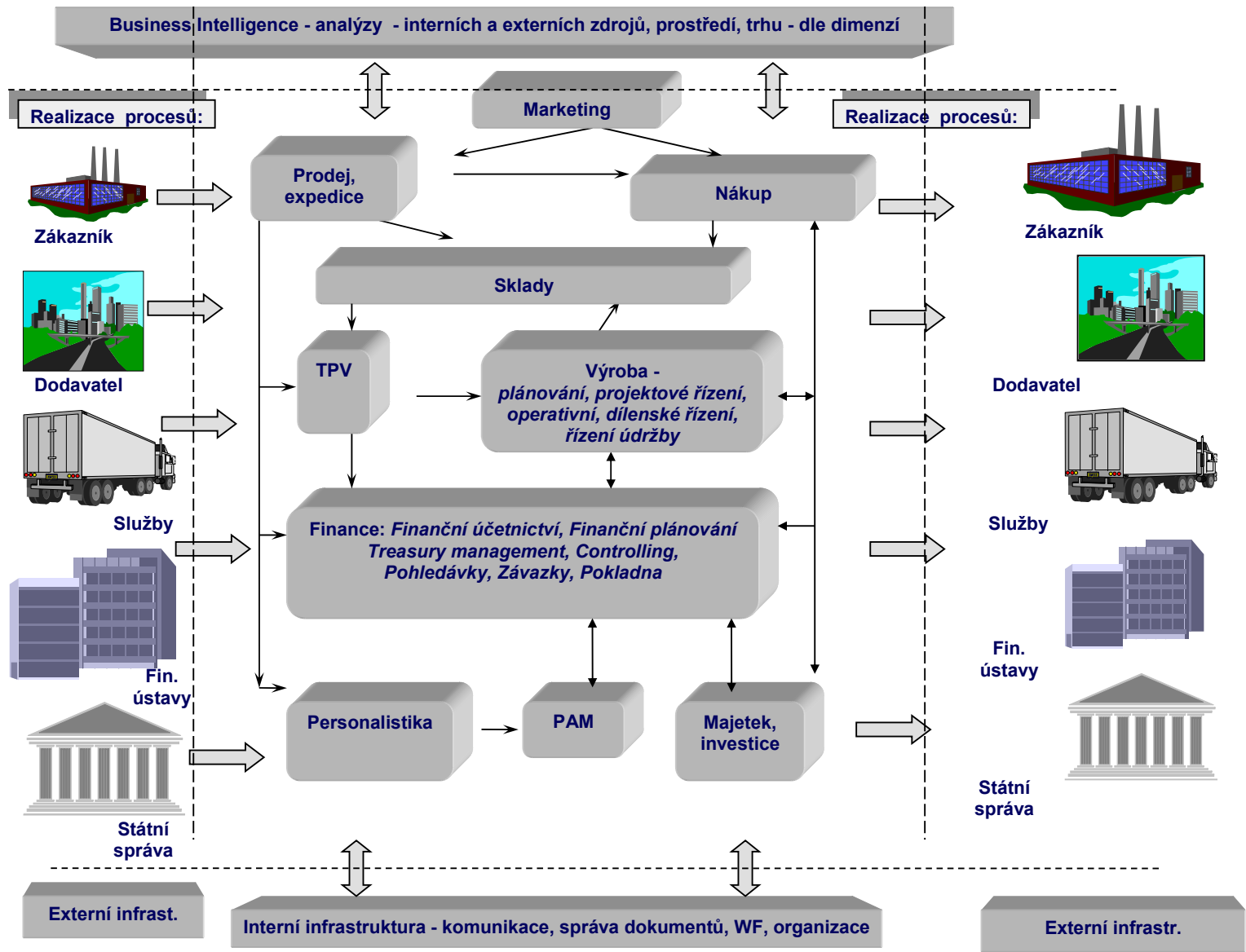
Na následujících obrázcích jsou uvedena základní schémata vazeb (obsahový základ pro analýzu a návrh podnikového IS). Na celkový model navazují popisy vazeb následujících procesů:

- prodej,
- nákup,
- sklady,
- finance,
- finanční účetnictví,
- finanční plánování,
- controlling,
- marketing,
- majetek
- výroba
 - technická příprava výroby (TPV)
 - plánování výroby
 - operativní řízení výroby
 - dílenské řízení výroby.

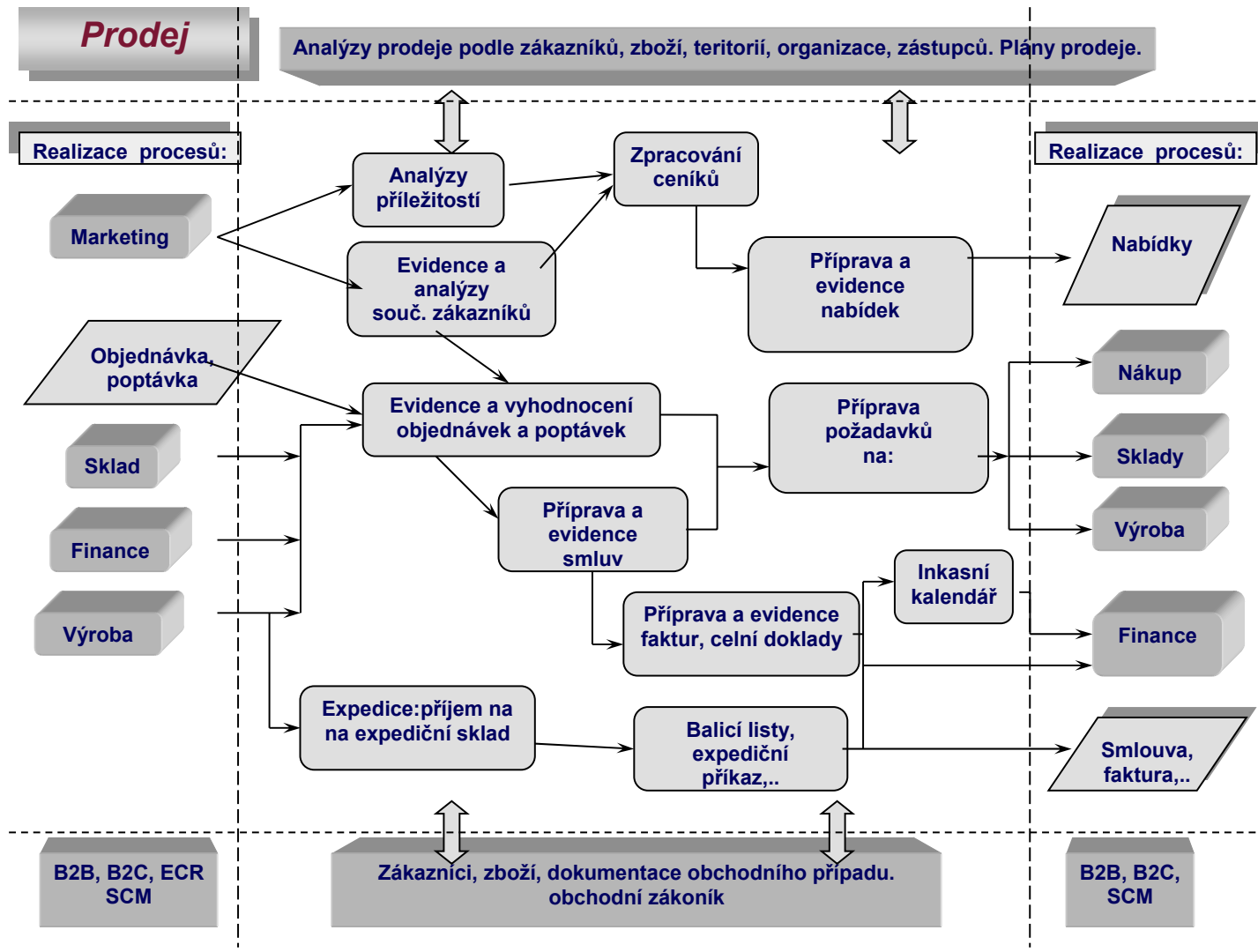
ERP - Enterprise Resource Planning (princip)



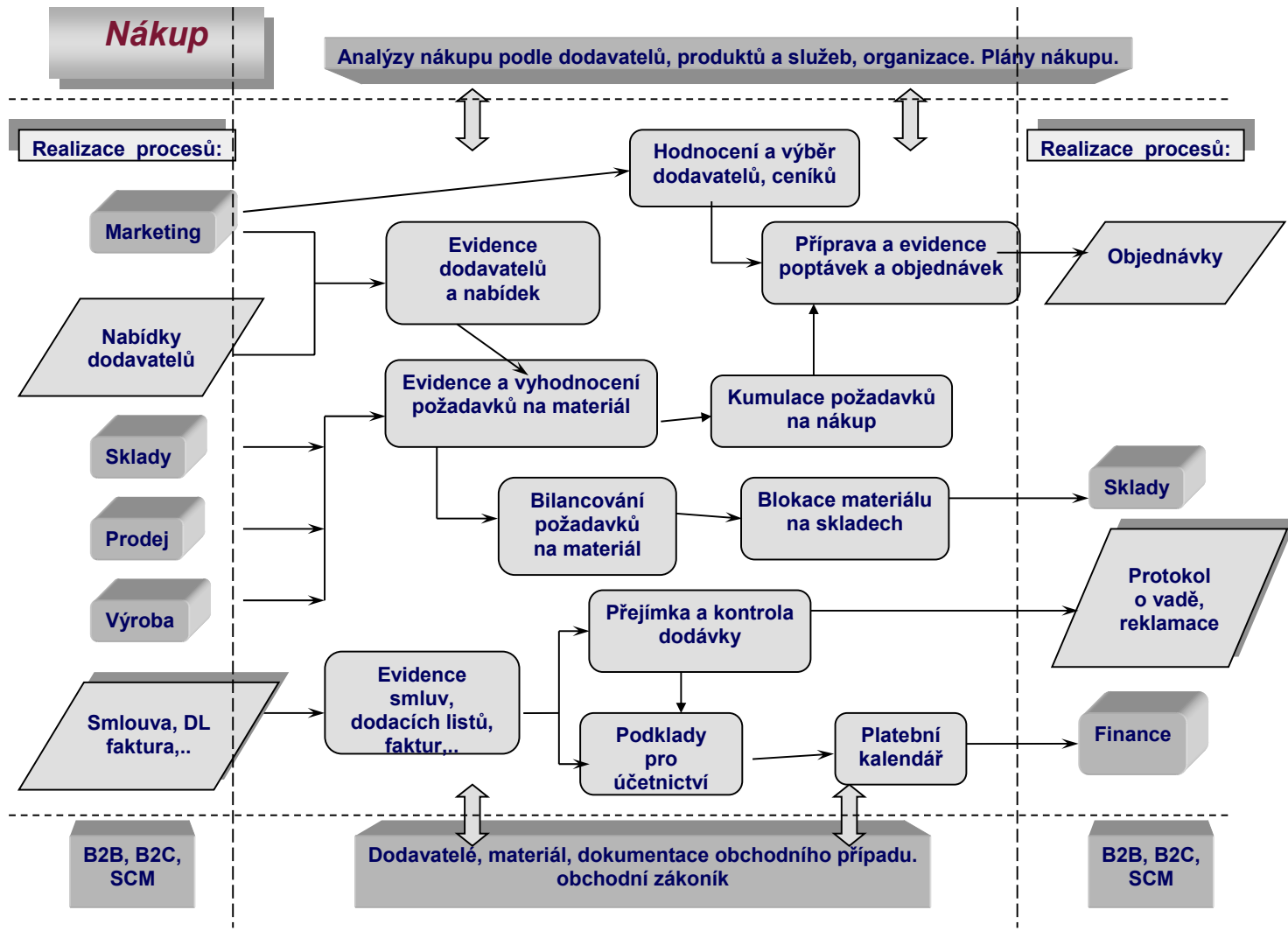
Ekonomické informační systémy

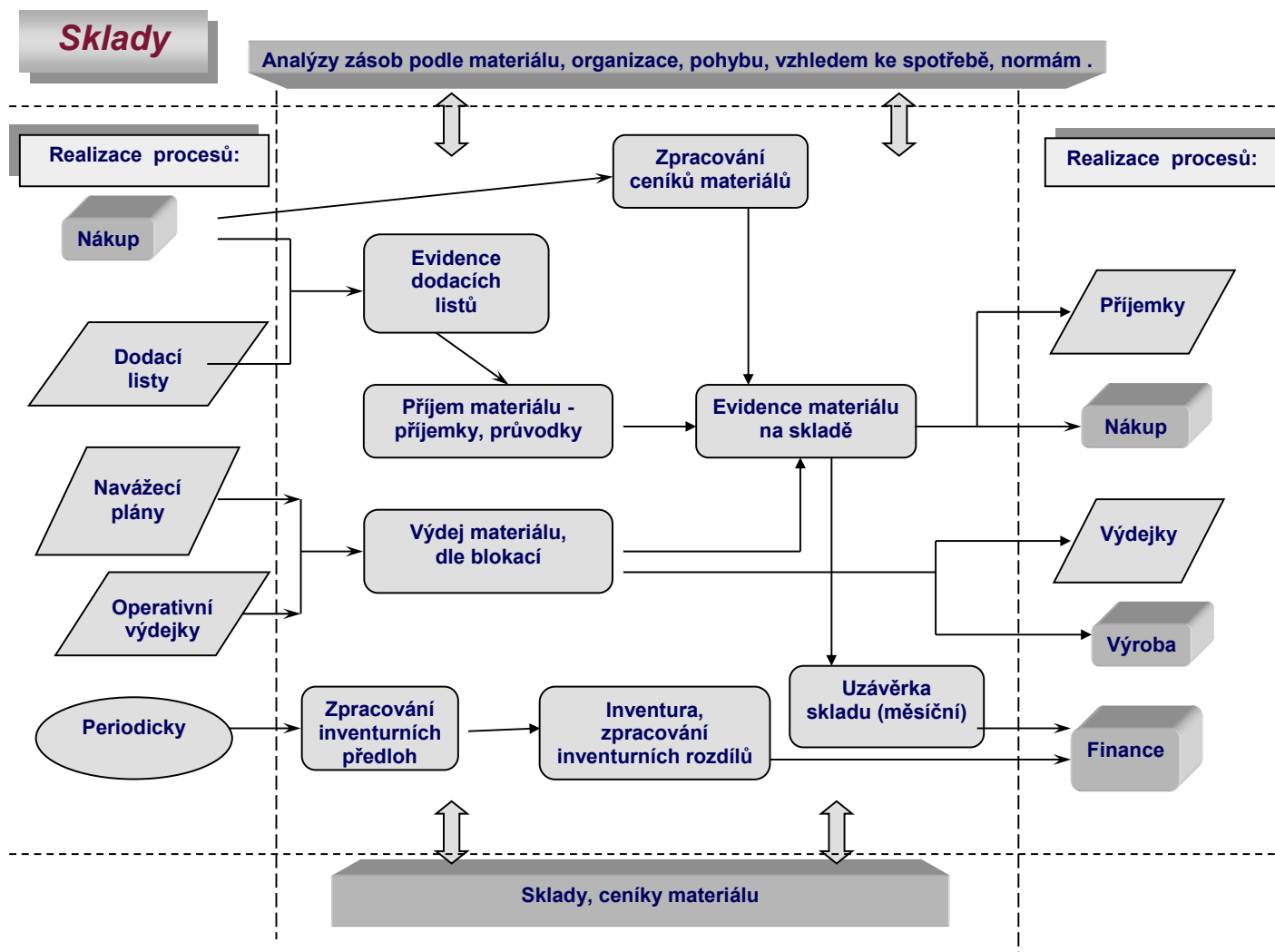


Ekonomické informační systémy

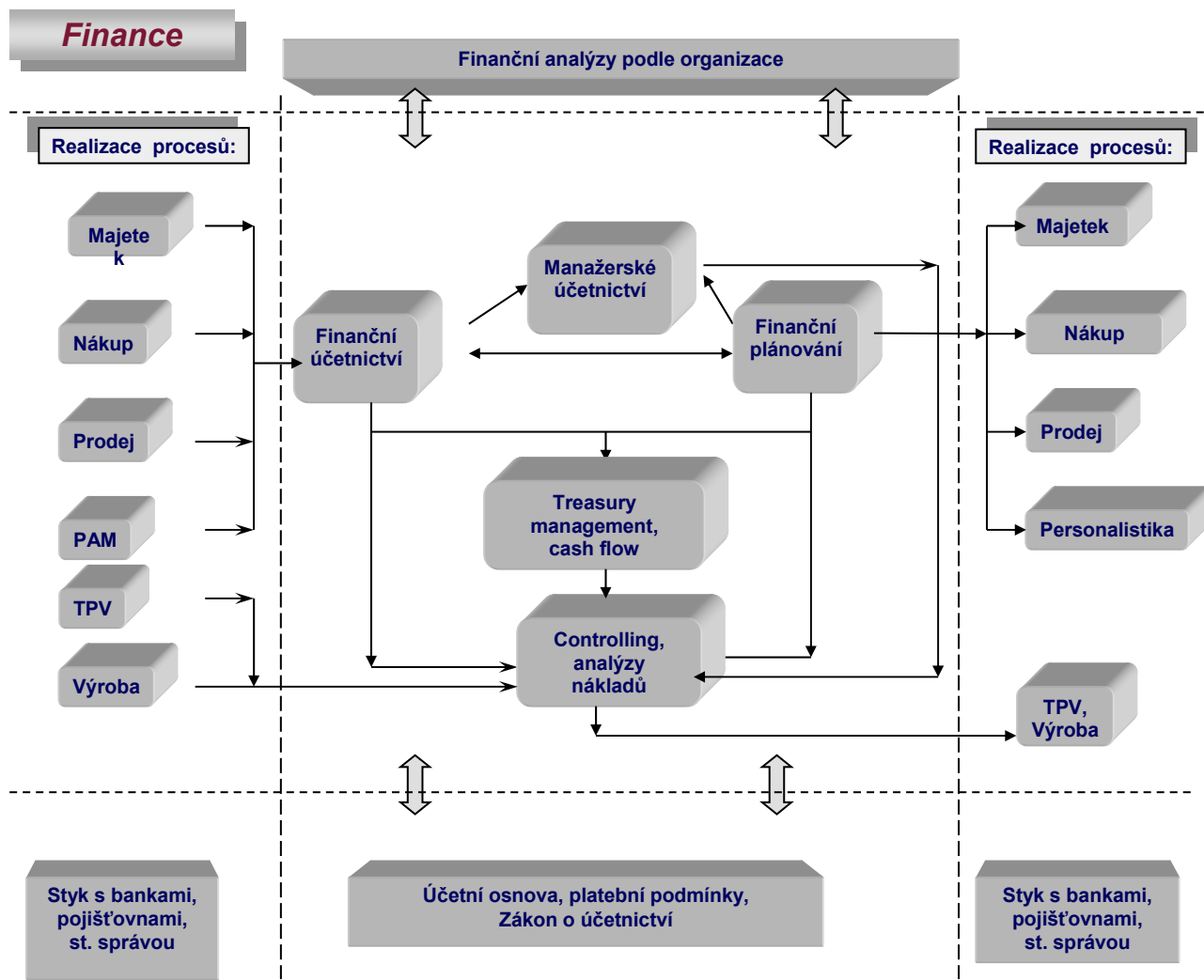


Ekonomické informační systémy

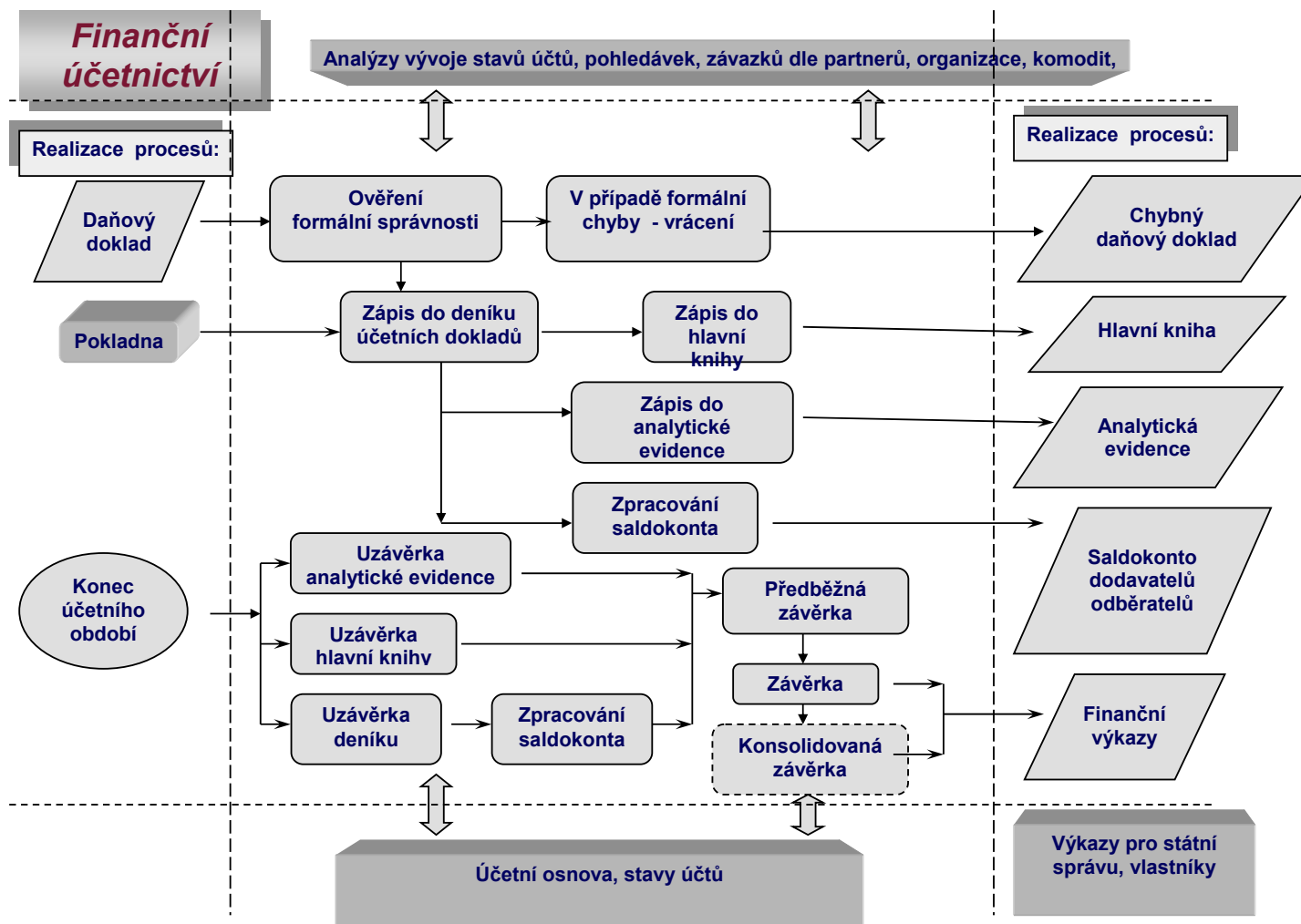




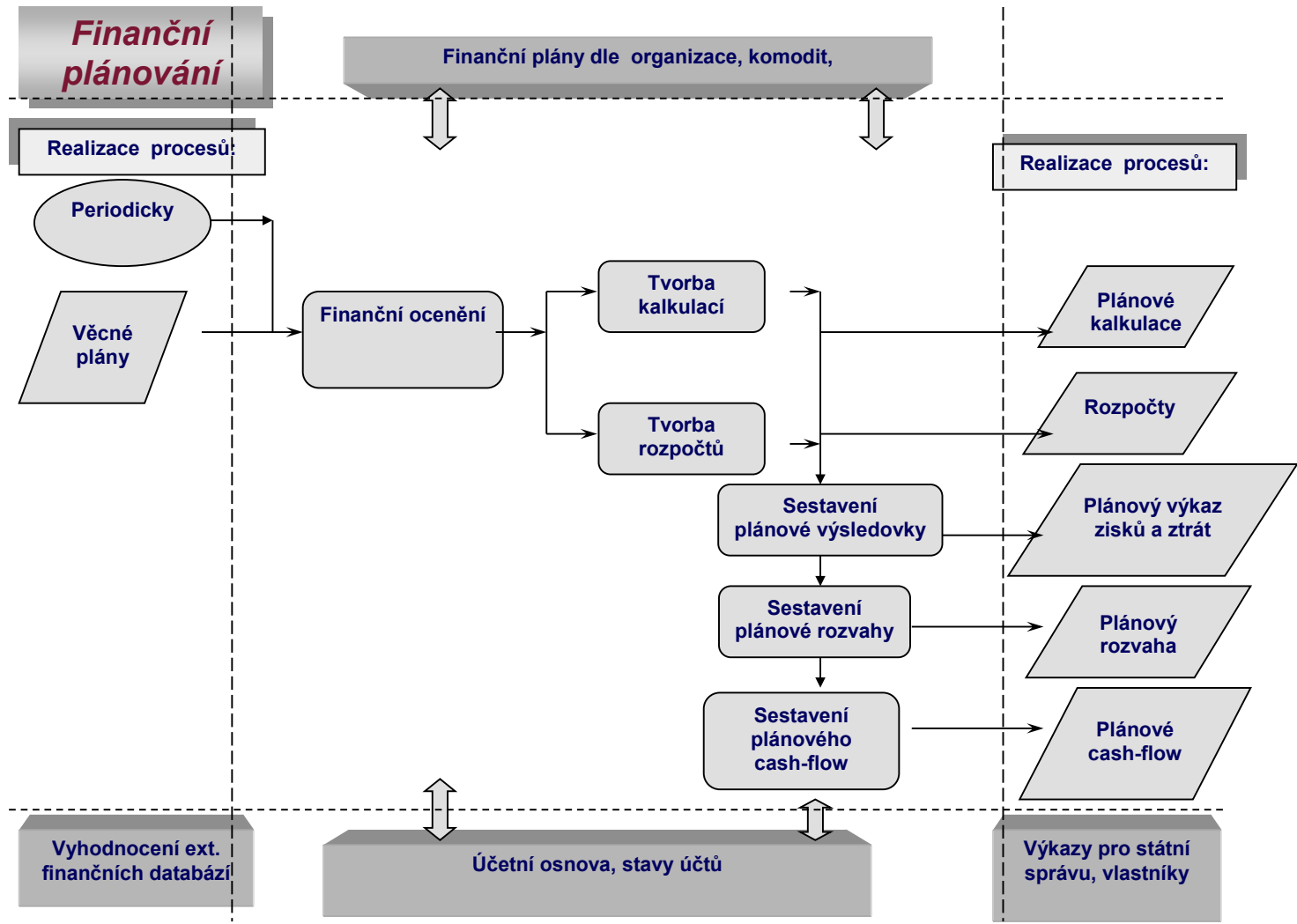
Ekonomické informační systémy



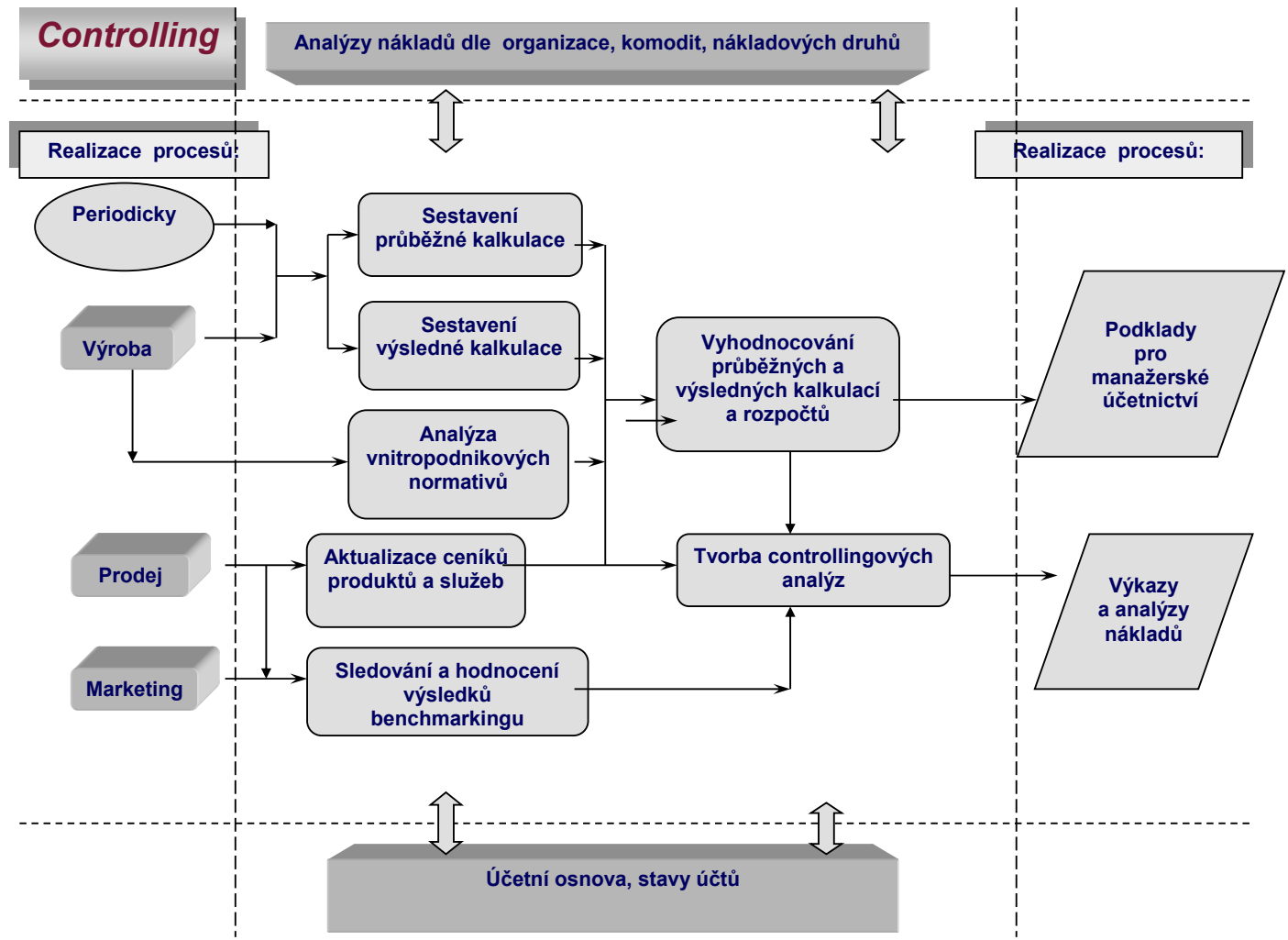
Ekonomické informační systémy



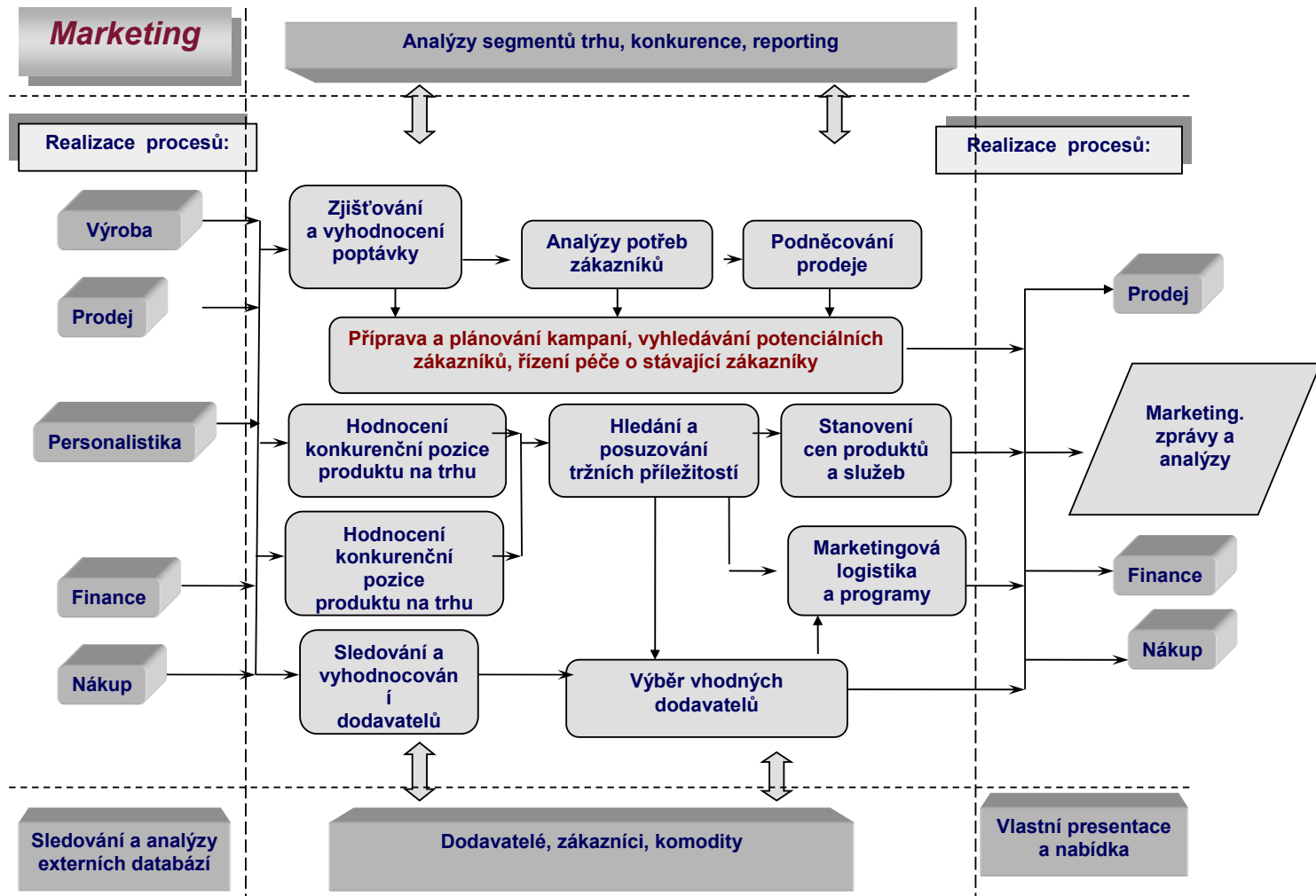
Ekonomické informační systémy



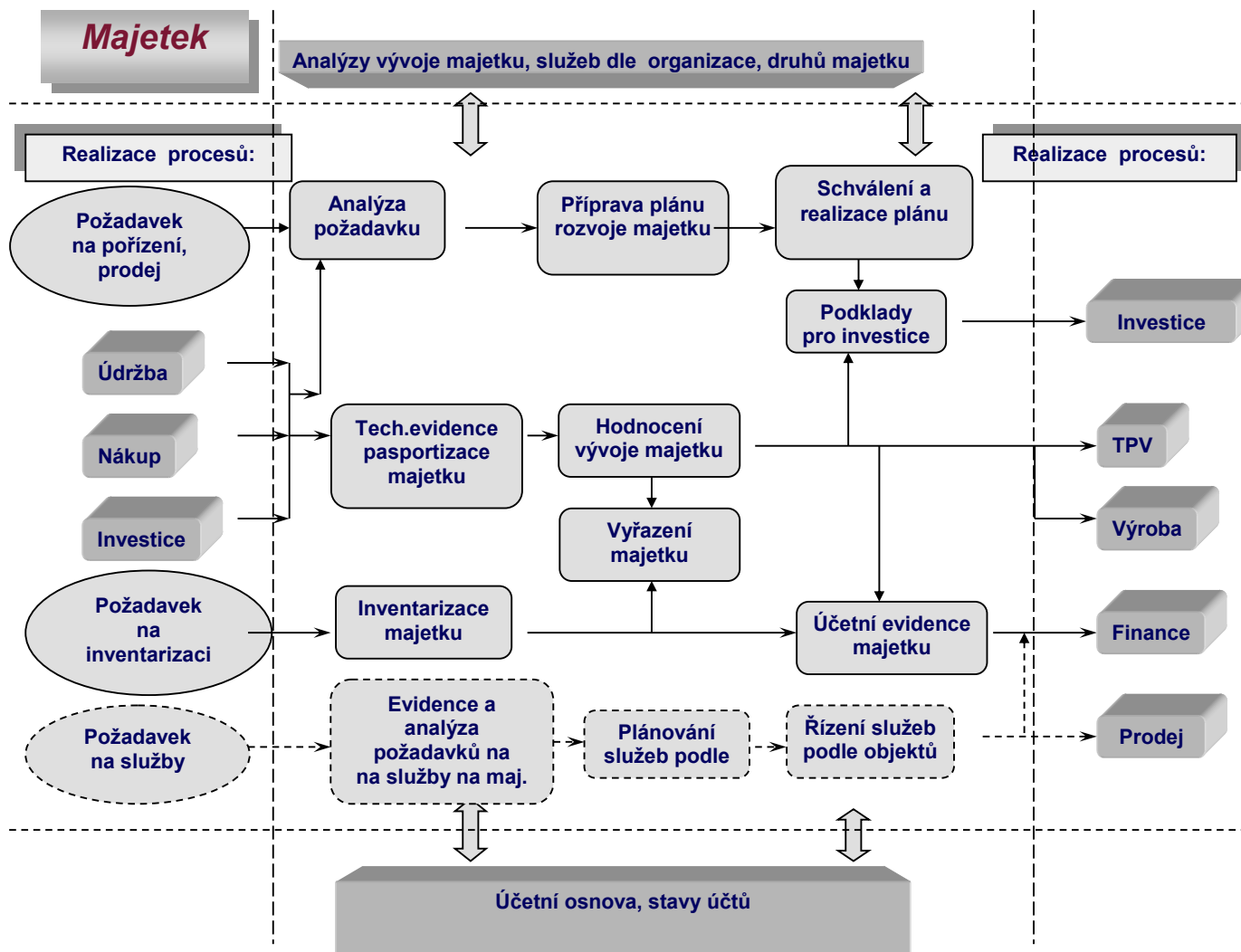
Ekonomické informační systémy



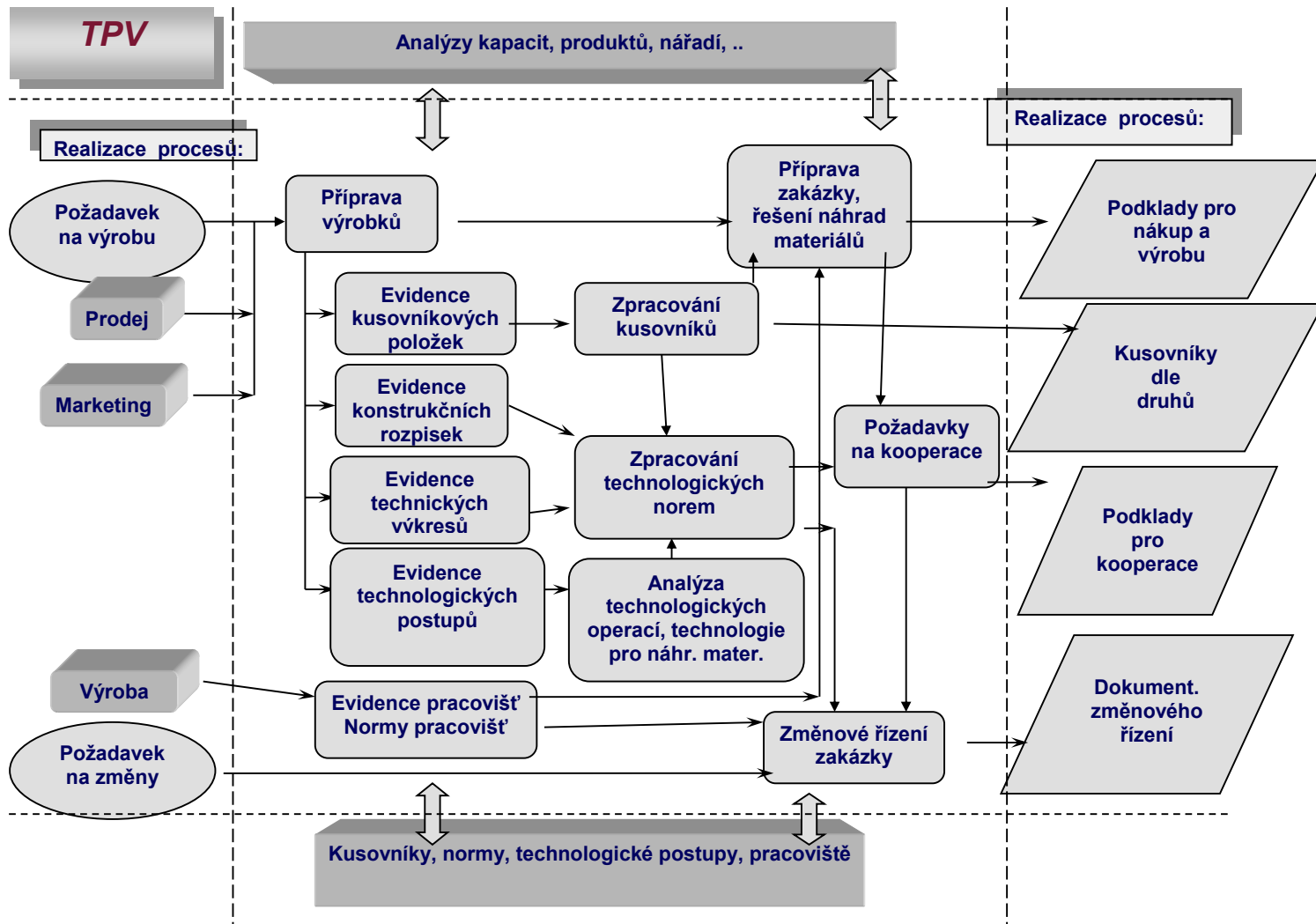
Ekonomické informační systémy



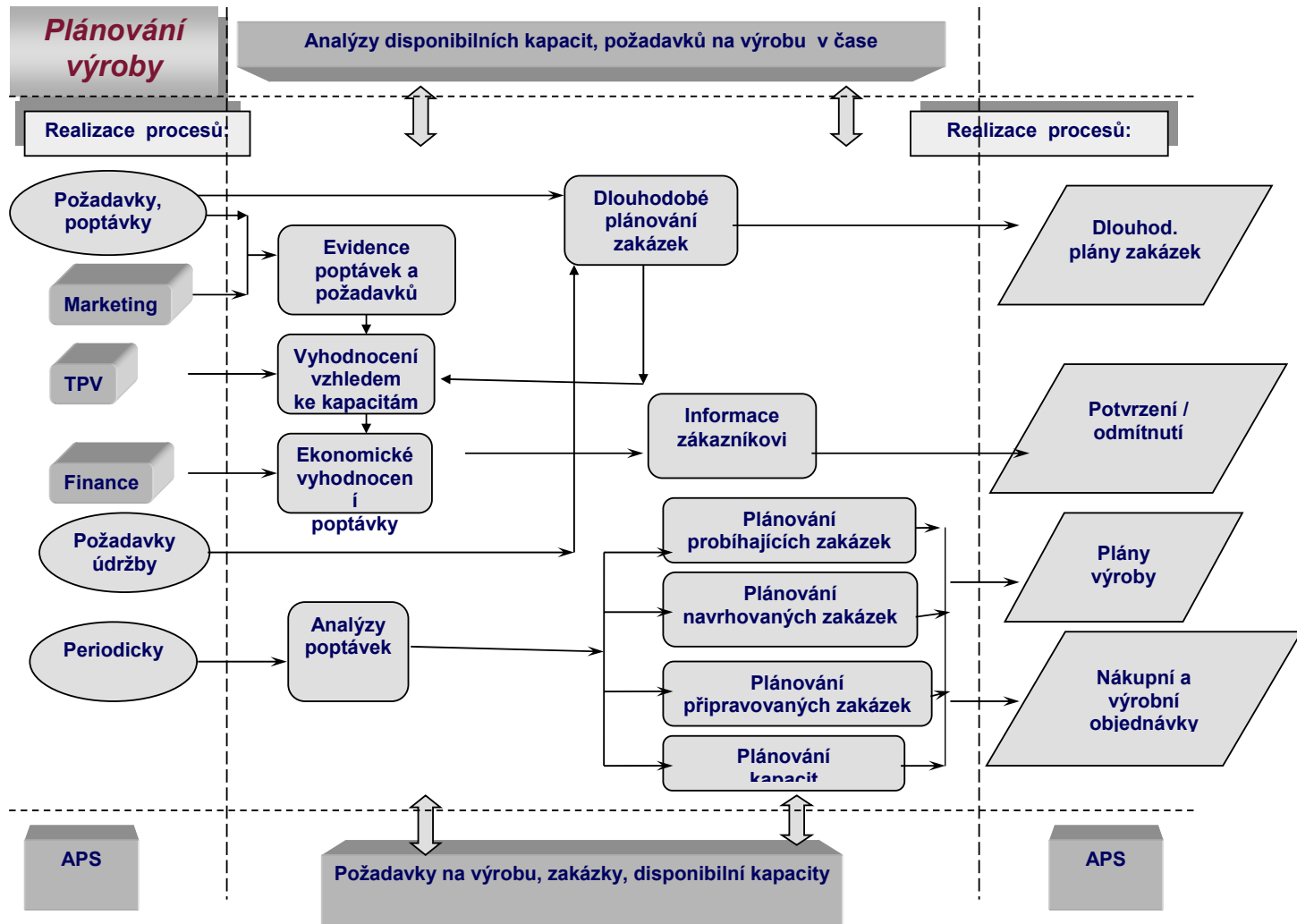
Ekonomické informační systémy



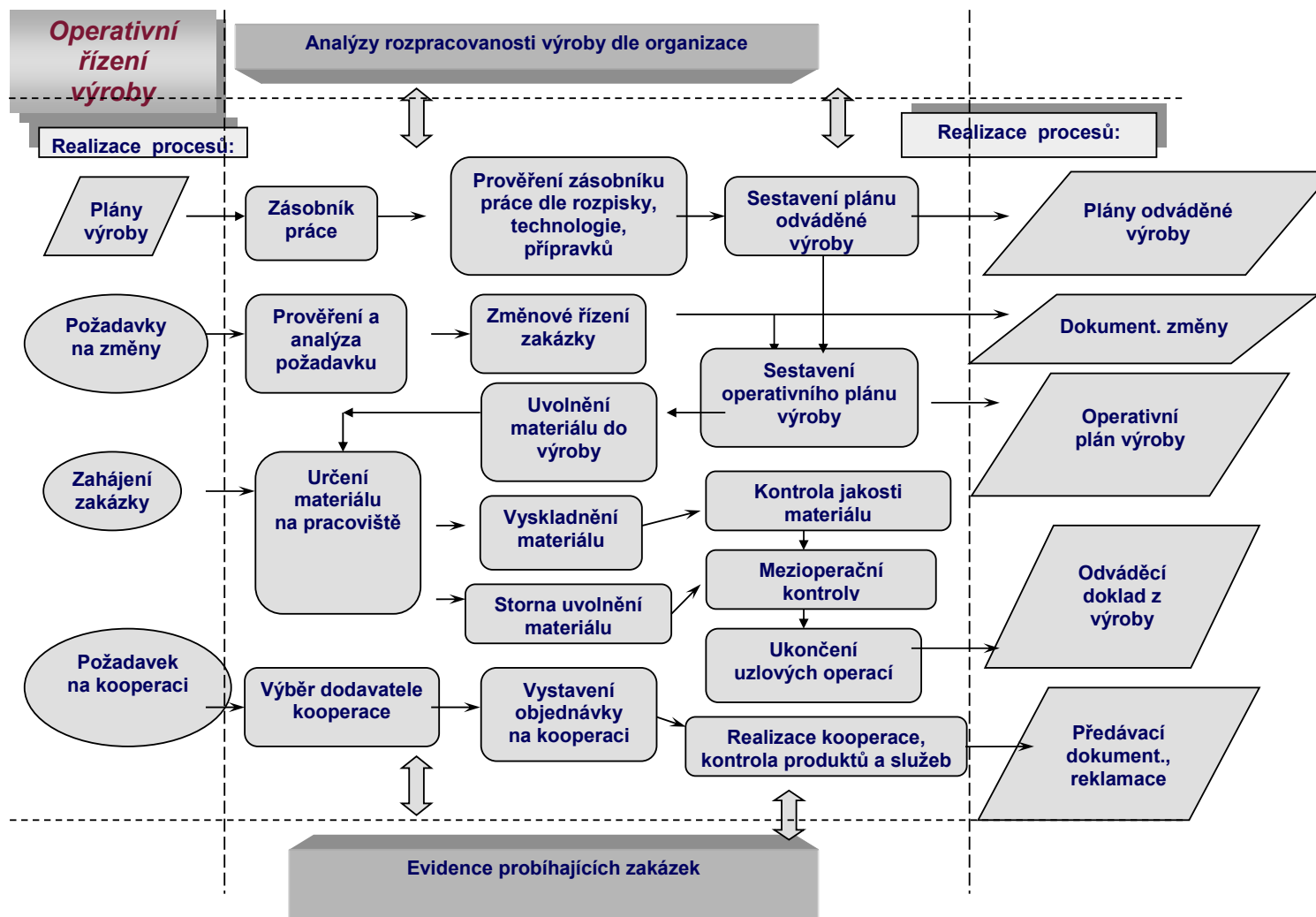
Ekonomické informační systémy

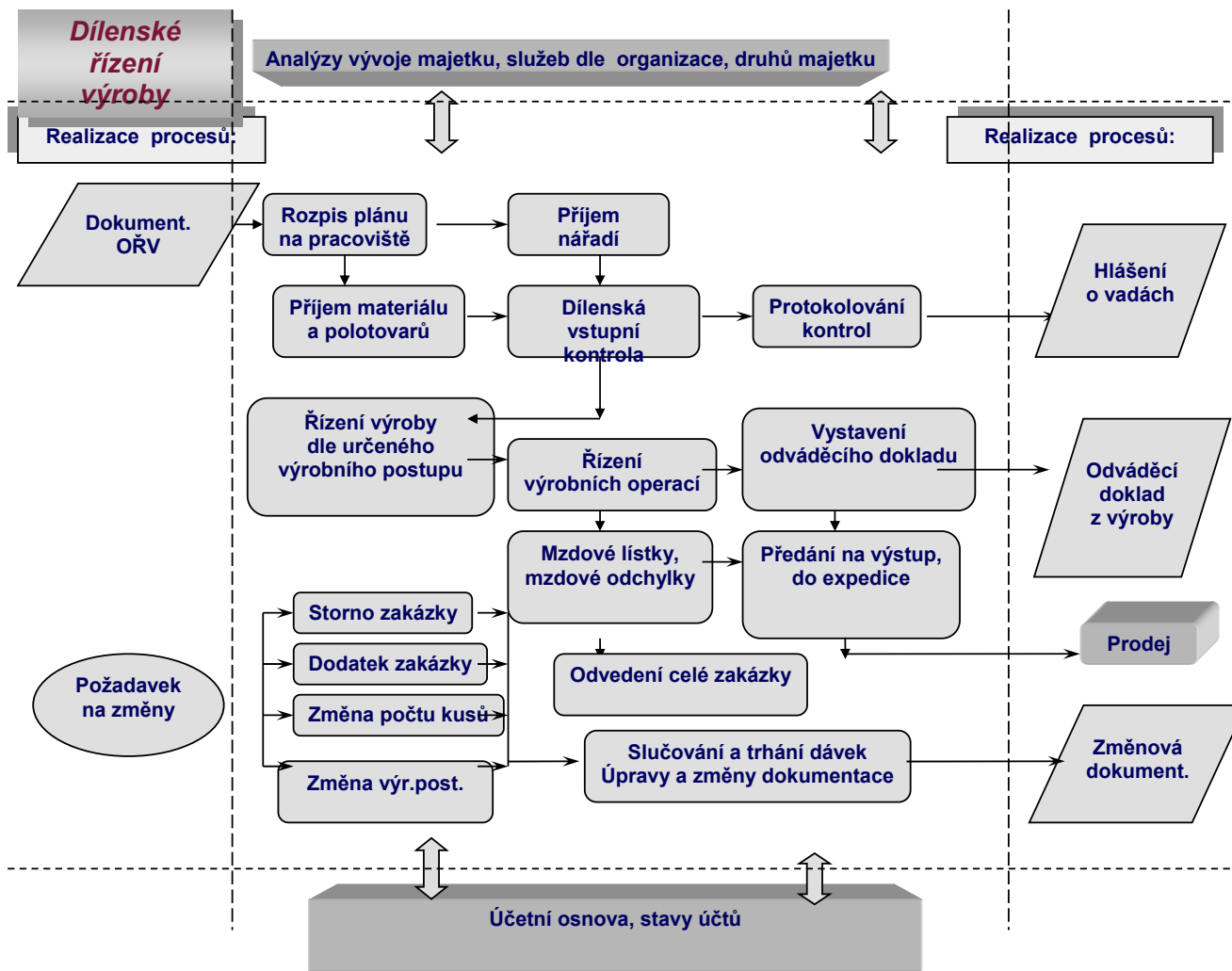


Ekonomické informační systémy



Ekonomické informační systémy





9.3 Moduly informačního systému

Charakteristika jednotlivých modulů navrhovaného informačního systému podniku vychází z analýzy potřeb řízení v nových podmínkách fungování společnosti.

A. Finance :

- hlavní kniha
- závazky
- pohledávky
- pokladna
- controlling
- finanční analýza

B. Obchod :

- prodej
- marketing
- nákup
- sklady
- podpora elektronického obchodu

C. Zdroje :

- personalistika
- PAM
- HNIM

D. Výroba :

- TPV
- Plánování výroby
- Operativní řízení výroby
- Dílenské řízení

E. Přeprava

F. Servis :

- jakost
- správa
- kancelářské úlohy

G. Stavební výroba :

- kalkulace nákladů, nabídkové rozpočty
- rozpočty
- fakturace a čerpání rozpočtu
- vazba na grafické CAD systémy
- datová a normativní základna
- harmonogramy

V kontextu uvedeného schématu a předpokladů systému řízení v podniku se zaměříme na dílčí vymezení jednotlivých modulů, a to v tomto členění:

1. Hlavní vstupy/výstupy modulu,
2. Funkce - obsah modulu, funkční struktura, náplň hlavních funkcí.
3. Realizace - základní požadavky na řešení modulu, projekční, technologické, provozní nároky, problémy realizace.

Informační systémy 1

A. Finance

A.1. Hlavní kniha

Modul Hlavní kniha je základním modulem finančního řízení firmy a často plní centrální úlohu v celém systému řízení. Sleduje stav a pohyb hospodářských prostředků a vnějších nákladů a výnosů za podnik jako celek podle nákladových a výnosových druhů. V rámci Hlavní knihy se vedou jednotlivé účty podle specifikované účetní osnovy. Hlavní kniha poskytuje přehled o výsledku hospodaření za podnik jako celek.

VSTUPY	VÝSTUPY
účetní doklady	účetní deník,
faktury	informace o stavu a pohybech jednotlivých účtů
	přehledy zaúčtovaných dokladů
	souhrnné účetní výkazy, přehledy,
	podklady pro finanční analýzy

Funkce:

Modul Hlavní kniha bude zajišťovat tyto hlavní funkce:

1. Zápis dokladu do deníku a jeho zaúčtování do hlavní knihy, tj.:
 - a) zadání čísla střediska, zakázky, variabilního symbolu,
 - b) aktualizace součtů všech účtovaných účetních případů, ...
 - c) kontrola operace na účtový rozvrh (zda např. účet existuje nebo není zablokován),
2. Zobrazení účetního deníku, tj. zobrazení účetních případů v deníku za zadané období
3. Zobrazení obratu účtu
4. Informace o okamžitém stavu účtů
5. Přehledy DPH
6. Informace o stavu prostředků na bankovních účtech
7. Přehledy zaúčtovaných dokladů - za zadané období, organizační jednotky apod.
8. Závěrka (předběžná, měsíční, roční) - za rozvahové účty
9. Závěrka (předběžná, měsíční, roční) - za výsledkové účty
10. Zobrazení okamžitého hospodářského výsledku, tj. náklady celkem, vnitropodnikové náklady, výnosy celkem, vnitropodnikové výnosy, hospodářský výsledek
11. Obratová soupiska, tj. soupis všech obrátů na účtech

Realizace:

Na modul Hlavní kniha se kladou tyto hlavní požadavky:

- musí být zajištěn soulad s legislativou, tj. řešení musí odpovídat tuzemským, zahraničním účetním a daňovým předpisům (zejména zákon o účetnictví z r. 1990 a další předpisy),
- konsolidace účetnictví za různé organizační jednotky,
- možnost vedení účtů ve více měnách,

Informační systémy 1

- možnost definování účetních transakcí (skupin účetních operací na různých účtech), tzn. že na základě zadání požadované účetní transakce se automaticky provede několik účetních operací najednou,
- možnost definování účetního klíče, resp. účetního řetězce, zahrnujícího nejen účty, účtované částky, ale i další hlediska (hospodářské středisko, zakázka), podle nichž se data v databázi klíčují a lze pak na jejich základě realizovat nejrůznější analýzy.

A.2. Závazky

Modul závazky má sledovat závazky firmy vůči svým dodavatelům, dále saldokonto podle dodavatelů, poskytovat analýzy závazků podle různých kritérií (data splatnosti, dodavatelů apod.), přehledy o platebních kalendářích (harmonogramu jednotlivých plateb), realizovat platební příkazy a analýzy plateb.

VSTUPY	VÝSTUPY
dodavatelské faktury,	saldokonto dodavatelů,
informace o dodavatelích,	analýzy závazků,
celní doklady,	příkazy k úhradě,
zálohové listy,	přehledy a analýzy závazků,
bankovní výpisy	přehledy a analýzy plateb

Funkce:

Modul Závazky zajišťuje tyto hlavní funkce:

1. Zpracování dodavatelských faktur (tuzemských i zahraničních), tj.:
 - a) evidence faktur (zápis do deníku došlých faktur),
 - b) kontrola a ověření faktur,
 - c) kontrola vypočtené DPH,
 - d) opravy a storna faktur,
 - e) rozúčtování závazků dle hospodářských středisek,
 - f) přiřazení závazku dodavateli,
 - g) likvidace faktur,
2. Zpracování celních dokladů
3. Zpracování zálohových listů, tj.:
 - a) finanční sledování zálohových dodávek od dodavatelů, placení jednotlivých záloh,
 - b) vyúčtování zálohové dodávky - po zaevidování konečné faktury - přiřazení jednotlivých záloh ke konečné faktuře,
4. Vytvoření platebního kalendáře
5. Upomínky, penalizace
6. Informace o obrazech na účtech dodavatelů
7. Výkaz DPH za dodavatele - sestavení výkazu DPH za dodavatelské faktury s uvedením daňových položek dodavatelských faktur
8. Saldokonto (tuzemské nebo zahraniční), tj. výběr faktur a plateb spárovaných podle variabilního symbolu, bankovního účtu a kódu banky dodavatele
 - a) přehledy faktur rozdělených podle plateb:

Informační systémy 1

- zaplacené faktury - jednou platbou na celkovou částku faktury nebo vyrovnání jedním účetním dokladem,
 - zaplacené více platbami a ev. účetním dokladem, částečně zaplacené jednou nebo více platbami - vyhodnocení a zobrazení hodnoty přeplatku nebo nedoplatku, vystavení účetního dokladu na vyrovnání salda, (lze využít i pro zaúčtování kursových rozdílů),
 - nezaplacené (určení, zda byl vystaven platební příkaz), vystavení účetního dokladu pro vyrovnání salda,
- b) identifikace nespárovaných plateb - nenašla se k nim faktura,
 - c) zobrazení salda podle dodavatelů, resp. za vybraného dodavatele,
 - d) saldo záloh,
9. Analýzy, přehledy
- a) nezaplacené faktury,
 - b) zlikvidované faktury,
 - c) nezlikvidované faktury,
 - d) přehled příjmek nespárovaných s fakturou, tj. příjemek, jejichž hodnota > částka zlikvidovaná fakturou, ...
 - e) přehledy závazků podle dalších zvolených kritérií,
10. Příkazy k úhradě
- a) tvorba a realizace příkazů k úhradě,
 - b) hromadný příkaz k úhradě,
 - c) přehledy vystavených a neprovedených příkazů k úhradě,
 - d) přehledy provedených příkazů k úhradě,
11. Zpracování plateb
- a) zpracování bankovních výpisů,
 - b) vstup plateb z banky, tj. zaúčtování - dle přiřazených účtů účtového rozvrhu a dle bankovního výpisu,
 - c) opravy plateb,
 - d) přehled plateb podle účtového rozvrhu,
 - e) zúčtování plateb - s vazbou položky na spárovanou fakturu nebo zálohu, opakované platby.

Realizace:

Na modul Závazky se kladou požadavky zejména v souvislosti s různými možnostmi kontrolních operací přijatých dokladů, zpracování analýz závazků podle nejrůznějších definovaných kritérií, jako např. dodavatele, splatnost závazku apod.

A.3. Pohledávky

Modul Pohledávky bude zajišťovat analýzy pohledávek, sledování pohledávek po splatnosti, zpracování inkasních kalendářů, generování upomínek a penalizace dle inkasního kalendáře a zvolených kritérií.

VSTUPY	VÝSTUPY
odběratelské (vydané) faktury,	saldokonto odběratelů,
informace o zákaznících,	výkazy DPH,
platební podmínky,	upomínky, penále,

Informační systémy 1

bankovní výpisy	přehledy a analýzy pohledávek, inkasní kalendáře
-----------------	---

Funkce:

Modul Pohledávky bude obsahovat tyto funkce:

1. Zpracování (tuzemských i zahraničních) odběratelských faktur
 - a) zaúčtování faktury (včetně účetních a daňových položek),
 - b) kontrola při zaúčtování - suma částek účetních a daňových položek se musí rovnat celkové částce faktury,
 - c) opravy a storno faktur,
 - d) zúčtování pohledávek podle hospodářských středisek,
2. Zpracování výkazů DPH za odběratele - přehled všech položek DPH z odběratelských faktur
3. Informace o obratu na účtech odběratelů
4. Analýzy a přehledy pohledávek dle vybraných kritérií
 - a) přehledy nezaplacených faktur,
 - b) přehledy podle zákazníků, lhůty splatnosti, ...
5. Zpracování inkasního kalendáře - přehled termínů plateb
6. Inkaso
7. Upomínky a penalizace
 - a) upomínky k fakturám, které nejsou zaplacené a jejichž datum splatnosti je menší než aktuální datum,
 - b) informace o počtu dnů z prodlení a počtu dnů od minulé upomínky
 - c) výpočet předpokládaného penále,
8. Likvidace pohledávky - párování s platbou
9. Saldokonto - s rozlišením podle vztahu faktur k platbám,
 - a) zaplacené faktury jednou platbou,
 - b) zaplacené faktury více platbami - zahrnuje informace o faktuře a přehled odpovídajících plateb,
 - c) částečně zaplacené faktury - vyhodnocení součtu uhrazených částek, zobrazení nedoplatku a vystavení účetního dokladu na vyrovnání salda,
 - d) nezaplacené faktury (nejsou ani částečně zaplacené) - vystavení účetního dokladu na vyrovnání salda,
 - e) nespárované platby (nenašla se k nim žádná faktura) platby pokrývající více faktur,
10. Saldo na odběratele
 - a) informace o odběrateli a jeho obratech za zvolené účetní období
 - b) přehled faktur pro odběratele a položkové zobrazení plateb,
11. Měsíční závěrka pohledávek

Realizace:

Řešení modulu Pohledávky požaduje integraci vlastních operací s bankovními operacemi, tj. řešení on-line vazeb na příslušnou banku. Kromě toho se požaduje dostatečné spektrum analýz pohledávek podle nejrůznějších kritérií (zákazník, splatnost, objem pohledávky, zástupce,...).

Informační systémy 1

A.4. Pokladna

Modul Pokladna bude poskytovat podporu pokladních operací, zpracování pokladních dokladů, vedení pokladní knihy.

VSTUPY	VÝSTUPY
pokladní doklady příjmové,	Pokladní kniha,
Pokladní doklady výdajové,	Přehledy podle pokladních dokladů,
Cestovní příkazy	Souhrnné pokladní výkazy a analýzy,
	Pokladní závěrka

Funkce:

Modul Pokladna zajišťuje tyto funkce:

1. Realizace příjmových a výdajových operací
2. Zpracování cestovních příkazů
3. Vyúčtování záloh na výdej peněžních prostředků
4. Přehledy nevyúčtovaných záloh
5. Vedení pokladní knihy
6. Účtování pokladních dokladů
7. Pokladní závěrka
8. Přehledy pokladních dokladů podle zvolených kritérií

Realizace:

Modul Pokladna bude umožňovat nejrůznější formy plateb, podporu kontrolních operací vstupních dokladů.

A.5. Controlling

Modul Controlling se orientuje především na nákladové analýzy, analýzy vnitropodnikového hospodaření, předběžné, průběžné a výsledné kalkulace, plánování a účtování nákladových středisek, kalkulace a účtování interních výkonů, analýzy nákladů a výnosů podle činností, výkonů, odpovědnosti, položek kalkulace, kalkulovatelnosti, analýzy činností podle středisek.

VSTUPY	VÝSTUPY
informace z hlavní knihy podle jednotlivých účtů,	Vnitropodnikové zúčtovací doklady,
účetní doklady,	kalkulace,
informace o organizačních jednotkách, nákladových střediscích, ...	nákladové analýzy,
informace o podnikových kapacitách a zdrojích (výrobní zařízení, pracovníci, ...)	analýzy hospodaření středisek

Funkce:

Modul Controlling zahrnuje tyto funkce:

1. Vnitropodnikové zúčtování
2. Zadávání a zúčtování skutečných nákladů
3. Účtování vnitropodnikových výkonů

Informační systémy 1

4. Účetnictví nákladových středisek
5. Vnitropodnikové zakázky a jejich vyhodnocení podle definovaných kritérií
6. Analýzy hospodářského výsledku
7. Kalkulace výrobků
8. Účetnictví jednotlivých středisek

Realizace:

Modul Controlling bude umožňovat variantní sledování vnitropodnikových výnosů a nákladů jak podle středisek, tak podle zakázek.

A.6. Finanční analýzy

Modul Finančních analýz se bude orientovat na celkové finanční analýzy podniku, tj. poskytování hodnoty ukazatelů rentability, aktivity, zadluženosti, likvidity, analýzy cash-flow, bude zahrnovat i podporu tvorby finančních plánů, sledování dalších vybraných ukazatelů.

Vstupy	Výstupy
souhrnné účetní výkazy,	vývoj hospodářských výsledků v čase,
informace o stavu a pohybu účtů,	finanční analýzy - poměrové ukazatele, procentní rozbor,
informace o externích partnerech,	analýzy cash-flow,
informace o podnikových zdrojích,	finanční plány,
	Analýzy úvěrů

Funkce:

Modul Finanční analýzy poskytuje tyto funkce:

1. Ukazatelé finanční analýzy
 - a) ukazatelé likvidity,
 - b) ukazatelé rentability,
 - c) ukazatelé zadluženosti,
 - d) ukazatelé aktivity,
2. Ukazatelé procentního rozboru na základě rozvahy, výsledovky
3. Analýzy finančních toků - Cash Flow
4. Informace o hospodářských výsledcích podle organizačních jednotek
5. Podpora tvorby finančních plánů
 - a) plánový výkaz zisků a ztrát,
 - b) plánová rozvaha,
 - c) plán peněžních toků,
6. Podpora pro řízení vybraných základních položek
 - a) mzdové náklady,
 - b) odpisy HNIM,
 - c) tvorba rezerv,
7. Zpracování podkladů pro finanční řízení investiční činnosti
8. Sledování úvěrového zatížení
 - a) přehled čerpaných úvěrů,
 - b) přehled poskytovaných úvěrů,
 - c) doplňující analýzy úvěrů,

Informační systémy 1

9. Analýzy vývoje kursů a měn
10. Analýzy vývoje úrokových sazeb

Realizace:

Funkce spojené s finanční analýzou jsou stále častěji realizovány v úlohách typu EIS s ohledem na jejich lepší technologické a prezentační možnosti. Často jsou rovněž zahrnuty v ostatních finančních modulech informačního systému a tvoří tak samostatný modul. Záleží tak na dekompozici a způsobu řešení aplikačního balíku.

B. Obchod

B.1. Prodej

Hlavní náplní modulu Prodej je zpracování obchodního případu. Existují např. tyto možné varianty postupů:

- nabídka,
- poptávka - nabídka - objednávka - smlouva - faktura,
- nabídka - smlouva – faktura
- smlouva.

Dalším cílem modulu Prodej je podpora řízení expedičních skladů, analýzy zásob hotových výrobků, sledování a kontrola úvěrových limitů pro zákazníka při uzavírání smluv, prodejní analýzy, podle zákazníků, prodejců, atd.

VSTUPY	VÝSTUPY
informace o zákaznících,	evidence zákazníků,
poptávky, objednávky,	ceníky výrobků a služeb,
dodací a platební podmínky,	nabídky, smlouvy,
informace o zboží,	požadavky na nákup, výrobu,
předpisy pro balení výrobků,	faktury, dobropisy, vrubopisy,
informace o stavu zásob	expediční příkazy, celní doklady,
	analýzy zajištění zakázek,
	prodejní analýzy

Funkce:

Modul Prodej zajišťuje tyto hlavní funkce:

1. Evidence zákazníků - vstup, aktualizace, rušení zákazníků, zpracování základních přehledů zákazníků
2. Zpracování odbytových ceníků,
3. Realizace obchodního případu - zakázky - zahrnuje:
 - a) Evidence poptávek,
 - b) Evidence nabídek,
 - c) Evidence objednávek,
 - d) Zpracování smluv,
 - e) Příprava faktur,
 - f) Dobropis tuzemský / zahraniční,
 - g) Vrubopis tuzemský / zahraniční,
 - h) Celní doklad,
 - i) Zpracování požadavků na výrobu,

Informační systémy 1

- j) Zpracování požadavků na nákup,
 - k) Zpracování požadavků na sklad - vybírají se sklady, na kterých je pro vybraný požadavek volná zásoba, zboží se na skladě zablokuje, definuje se i požadované úložné místo na skladě, pokud je položka na více místech,
 - l) Rušení požadavku ze smlouvy,
 - m) Analýzy zajištění smluv - k jednotlivým položkám se dle situace zobrazuje zajišťující výrobní dávka, odpovídající požadavky na nákup a zpracovávají se přehledy zboží zablokovaných na skladech pro jednotlivé položky smlouvy,
 - n) Požadavky vyplývající z faktury - k faktuře se zobrazí přehled položek a jejich množství zablokované na skladě hotových výrobků,
 - o) Vytvoření expedičního příkazu,
4. Expedice
- a) Tvorba balících listů - vychází se ze standardního předpisu pro balení připraveného v rámci modulu Technické přípravy výroby,
 - b) Plnění balících listů - evidence průběhu balení zboží podle předpisu v balícím listu,
 - c) Kompletace - analyzuje se blokáce zboží v úložných místech skladů podle výrobních čísel položek, provádí se kompletace odesílaného zboží a odblokování ze skladu a realizuje se výdej zboží ze skladu na obchodní případ a odečtení ze zásob,
5. Řízení expedičních skladů
- a) Příjem nakupovaného zboží na sklad - vytváří se příjemka nakupovaného zboží na sklad a automaticky se provádí na úložných místech blokáce podle položky smlouvy,
 - b) Příjem z výroby - příjem zboží z výrobních středisek na sklad,
 - c) Výdej na zakázku - výdej (odblokování) se provádí jen z těch úložných míst, která jsou blokována na konkrétní požadavky a vygeneruje se výdejka zboží,
 - d) Změny úložných míst zboží na skladě - přemístění zboží z jednoho úložného místa na jiné i s odpovídající blokáci,
 - e) Zpracování inventur - tj. vytvoření inventurních předloh, pořízení inventurních stavů a zpracování inventurních rozdílů,
 - f) Obrátová soupiska zásob - tj. seznam všech položek na skladu zahrnující počáteční stav na začátku měsíce, seznam všech příjmů a výdajů za měsíc, konečný stav na konci měsíce, a to v naturálních i hodnotových jednotkách,
6. Prodejní analýzy
- a) Platby zákazníka - přehled všech zákazníků a stav jejich plateb,
 - b) Vývoj výrobních zakázek pro jednotlivé prodejce,
 - c) Analýzy prodejů dle prodejce,
7. Výpočet inkasního kalendáře

Realizace:

Při řešení modulu Prodej se sledují některé z těchto požadavků:

- automatické generování dokladů, např. kopírováním základních údajů z nabídky do smlouvy, faktury apod.
- automatické kontroly - např. nelze provést storno zákazníka, pokud je obsažen na obchodních dokladech,

Informační systémy 1

- sledování a kontrola úvěrových limitů pro zákazníka při uzavírání smluv,
- možnosti zpracování prodejních analýz podle nejrůznějších kritérií (zákazník, zboží, prodejce, zástupce, teritorium, ...)

B.2. Marketing

Modul Marketing zahrnuje podporu hlavních marketingových funkcí, zejména profily zákazníků, analýzy zákazníků, analýzy segmentů trhu a další.

VSTUPY	VÝSTUPY
informace o zákaznících,	analýzy zákazníků,
informace o poskytovaných produktech a službách,	analýzy konkurence,
informace o stavu trhu	analýzy segmentů trhu,
	analýzy cenových, teritoriální a dalších vlivů,
	Materiály pro podporu prodeje

Funkce:

Modul Marketing zajišťuje tyto hlavní funkce:

1. Marketingové analýzy
 - a) Analýzy prodeje dle komodit, organizačních jednotek, teritorií,
 - b) Analýzy cenových vlivů (konkurence, teritoria, sortiment, měna),
 - c) Analýzy nových segmentů trhu (dle komodit, zákazníků, teritorií, ...),
 - d) Analýzy pozice na trhu,
2. Profily zákazníků (platební podmínky, dodací podmínky, komoditní skladba, ...)
 - a) Hodnocení zkušeností styku se zákazníkem,
 - b) Hodnocení aktivit různých organizačních jednotek u téhož zákazníka,
3. Řízení zástupců
 - a) Evidence zástupců,
 - b) Vývoj prodejů u jednotlivých zástupců,
 - c) Informace o zákaznících zástupců,
4. Analýzy konkurence
 - a) Analýzy pozice na trhu vzhledem ke konkurenci,
 - b) Sortimentní, cenová, kvalitativní srovnání,
 - c) Profily rozhodujících konkurentů,
5. Propagace a reklama
 - a) Public relations,
 - b) Veletrhy, výstavy, prezentace,
 - c) Direct mailing,
 - d) Nabídkové katalogy,
6. Informační podpora obchodních aktivit (pro obchodní cestující)
7. Evidence a rešerše cestovních zpráv
8. Souhrnné analýzy výrobních technologií
9. Analýzy potenciálních dodavatelů

Informační systémy 1

Realizace:

V současné době jsou funkce spojené s marketingem často realizovány pomocí specializovaných aplikačních programových balíků (jako např. Infosale, Marketing Manager, Leonardo) a hlavní aplikační balík pouze musí zajistit odpovídající interface na tyto specializované programy. Právě moduly marketingu by měly být schopny integrovat a využívat externí datové zdroje.

B.3. Nákup

Základním cílem modulu Nákup je realizace nákupního obchodního případu. Jde o soubor obchodních transakcí, se sekvencí těchto dokumentů: poptávka – nabídka – objednávka – smlouva – dodací list – faktura. Zahrnuje blokování a odblokování zásob podle potřeb výroby, zajištění materiálu požadovaného na základě zpracování výrobních plánů, zpracování odbytových požadavků na subdodávky a operativních požadavků středisek, např. na režijní materiál.

VSTUPY	VÝSTUPY
informace o dodavatelích,	přehledy, analýzy dodavatelů,
informace o nákupních cenách,	evidence nabídek, smluv, dodacích listů,
požadavky na nákup (materiálu),	potvrzování požadavků,
informace o stavu zásob,	poptávky, objednávky,
nabídky,	analýzy nákupů, zásob,
smlouvy	protokoly, vyhodnocování dodávek,

Funkce:

1. Evidence a analýzy dodavatelů
2. Dodavatelské ceníky
3. Řízení nákupu
 - a) Evidence požadavků na materiál - zajišťuje evidenci požadavků na materiál jednotlivých výrobních a dalších středisek,
 - b) Kumulace požadavků na nákup,
 - c) Provádění storna,
 - d) Blokace materiálu
 - blokování požadovaného množství položky podle termínu vychystání do výroby - globálně za veškeré požadované množství,
 - řeší se podle úložných míst příslušného skladu ve vztahu k jednotlivým materiálovým požadavkům,
 - e) Bilancování požadavků na materiál - analýzy požadovaného a disponibilního materiálu,
 - f) Potvrzování požadavků na materiál - zpětná vazba z modulu Nákup vůči jednotlivým materiálovým nárokům zakázky – informace nárokuje střediskům, ...
 - g) Zpracování poptávek - pro potenciální i stálé dodavatele,
 - h) Evidence nabídek,

Informační systémy 1

- i) Zpracování objednávek,
 - j) Evidence smluv - zahrnuje potvrzení objednávky ze strany dodavatele a evidenci dokladu smlouvy a kontrolu vazby na číslo objednávky, k níž se vztahuje,
 - k) Evidence dodacích listů, součástí je přiřazení čísla objednávky, kterou dodávka plní a vytvoření podkladu pro příjemky materiálu na sklad (v příjmkách čísla objednávek zajistí spárování faktur a příjemek),
 - l) Vyhodnocení dodávky - z hlediska úplnosti, termínů, kvality, zahrnuje pořízení protokolu o vadě, zpracování reklamace, vytvoření ev. upomínky, podkladu pro pokus o smír nebo arbitráž,
4. Analýzy zásob
- a) Přehledy aktuálního stavu zásob, informace o jednotlivých položkách,
 - b) Analýzy zásob vzhledem ke spotřebě,
 - c) Analýzy bezpohybových položek,
 - d) Nadnormativní zásoby,
 - e) Podnormativní zásoby,
 - f) Analýzy nákupu dle zásobovačů.

Realizace:

Základními požadavky je racionalizace zpracování obchodního případu generováním dokladů, resp. kopírováním základních údajů mezi doklady, zajištění základních kontrolních operací a analýz zajištěnosti požadavků na nákup.

B.4. Sklady

Základním cílem modulu Sklady je poskytovat pohotový a přesný přehled o stavu a pohybu materiálu na všech skladech. Přejímku materiálu provádí zásobovač. Vstupním dokladem je dodací list. Vyhotovuje se průvodka materiálu, předloha pro vytvoření příjemky materiálu. Při každém pohybu materiálu se provádí příslušné zúčtování. Součástí modulu Sklady jsou obvykle analýzy zásob podle nejrůznějších hledisek.

VSTUPY	VÝSTUPY
ceníky materiálu,	příjemky / výdejky materiálu,
evidence skladů a skladovacích prostor,	inventurní předlohy a přehledy,
dodací listy	uzávěrky skladu,
	analýzy zásob

Funkce:

1. Ceníky materiálu
2. Skladová evidence materiálu
 - a) Příjem materiálu zahrnuje:
 - příjem dávky dodacích listů,
 - vygenerování průvodek došlého materiálu,
 - možné rozdělení materiálu na několik skladů,
 - b) Vytvoření příjemky,
 - c) Výdej materiálu dle navážecího plánu do výroby zahrnuje:

Informační systémy 1

- výběr z plánovaných výdejů materiálových položek ze zadaného skladu,
 - vygenerování výdejek, určení vydávaného množství a úložných míst, ze kterých se výdaj uskuteční,
 - d) Výdej operativní, tj. jednotlivých materiálů podle požadavků středisek,
 - e) Řešení změn úložných míst materiálu ve skladech,
 - f) Ocenění nově založených materiálů,
 - g) Zpracování inventur:
 - Vytvoření inventurních předloh,
 - Pořízení inventurních stavů,
 - Zpracování inventurních rozdílů,
 - h) Měsíční uzávěrka skladu
 - zpracování všech příjmků a výdejek pro zadaný sklad,
 - aktualizace stavů materiálu k začátku měsíce, ev. k začátku roku,
3. Analýzy zásob na skladě
- a) Výkaz skladu - měsíční pohyby na skladu v agregované podobě podle druhů pohybu,
 - b) Likvidace nepotřebných zásob - evidenční zajištění odepsání nepotřebných zásob ve vazbě na analýzy struktur stavu zásob,
 - c) Analýzy spotřeby materiálu - podle středisek,
 - d) Přehled zůstatků materiálů - seznam aktuálního stavu všech materiálů s uvedením zásoby v měrných jednotkách i v korunách (podle skladů i za podnik),
 - e) Obrátová soupiska zásob - seznam všech položek na skladu (počáteční stav na začátku měsíce, přehled všech příjmů a výdajů za měsíc a stav na konci měsíce),
 - f) Další analýzy zásob podle zvolených hledisek.

Realizace:

Při řešení modulu Sklady je nutné realizovat vazby na skladové technologie, možnosti využití čárových kódů apod. Některé informační systémy umožňují i definování tzv. logických a fyzických skladů, tzn., že na jednom nebo několika fyzických skladech je definován tzv. logický sklad, např. s ohledem na možnosti prodejce prodávat z určité podskupiny všech fyzických skladů.

B.5 Podpora elektronického obchodu

Implementace modulu na podporu elektronického obchodu zahrnuje aplikace s dynamickou tvorbou stránek určené pro nabídku a prodej zboží prostřednictvím Internetu a další podporu elektronického obchodu.

Funkce:

- seznam katalogů nabízeného zboží a služeb,
- katalogy zboží, včetně vyobrazení zboží a upozornění na novinky,
- obsah vybraného katalogu zboží a služeb s možností výběru katalogového listu i dle neúplných indikací v příslušné jazykové mutaci,

Informační systémy 1

- detailní katalogový list o konkrétním zboží a službě, včetně jeho zobrazení (ve formátech JPG, GIF, PNG) s možností prezentace detailních modifikací zboží a s možností poptání zvoleného zboží či služby,
- detailní popis modifikace daného zboží doplněného o technické specifikace,
- poptání a objednání zboží ve zvolené jazykové mutaci, množství, požadované dodací lhůty a doplňkové texty,
- registrace zákazníků (kontakt, adresy, telefony, fax, e-mail atd.),
- použití existujícího zákaznického profilu,
- zobrazování obsahu aktuální poptávky, tisk poptávky,
- závěrečné potvrzení poptávky s celkovou položkovou rekapitulací,
- vícejazyčný návod k obsluze formou funkce HELP, ON-LINE nápověda,
- servisní stránka umožňující autorizovaný přenos denního přírůstku poptávek elektronickou formou pro potřeby interního informačního systému zákazníka,
- odkazy ze statické firemní stránky (Home page).

C.Zdroje

C.1. Hmotný a nehmotný majetek

Modul Hmotný a nehmotný majetek zajišťuje účetní i operativní evidenci hmotného a nehmotného majetku, analýzy stavu a využití majetku, plánování oprav a investic, zpracování účetních i daňových odpisů, analýzy majetku, řízení investic, oprav, údržby majetku.

VSTUPY	VÝSTUPY
technické informace o hmotném investičním majetku,	přehledy a analýzy hmotného a nehmotného majetku,
informace o nehmotném majetku,	podklady pro účtování odpisů,
požadavky na investice,	poklady pro řízení investic,
požadavky na opravy a údržbu	podklady pro plánování a řízení údržby

Funkce:

Modul Hmotný a nehmotný majetek bude zajišťovat tyto funkce:

1. Řízení investic
2. Evidence hmotného a drobného hmotného majetku (HM/DHM) - evidence hmotného investičního majetku (odepisovaného i neodepisovaného), včetně technické evidence,
 - a) Zařazení aktivovaného majetku z investic - zařazení zcela nebo částečně dokončené investiční akce (zakázky) zlikvidované ve finančním účetnictví do evidence HM/DHM - pod jedním nebo více inventárními čísly,
 - b) Technická evidence strojů a zařízení,
 - c) Technická evidence vozidel,
 - d) Technická evidence budov a staveb,
 - e) Technická evidence pozemků,

Informační systémy 1

- f) Převody HM/DHM - hromadný převod mezi nákladovými středisky, evidenčními středisky (podstředisky), umístěním, činnostmi a pracovníky,
 - g) Účetní odsouhlasení přírůstků HM/DHM - potvrzení nebo změna účetních, cenových i daňových údajů a údajů o zařídění nově zařazených předmětů do evidence,
 - h) Individuální odpisy dle odpisového plánu,
 - i) Průběh HM/DHM - zobrazení všech změn a pohybů s konkrétním předmětem, v evidenčním stavu i vyřazeným z evidence,
 - j) Reaktivace HM/DHM,
- 3. Ostatní hmotné předměty - nepočítají se z nich účetní a daňové odpisy,
 - 4. Evidence nehmotného a drobného nehmotného majetku - evidence nehmotného investičního majetku (odepisovaného i neodepisovaného), včetně technické evidence,
 - 5. Ostatní nehmotný majetek - nepočítají se z nich účetní a daňové odpisy,
 - 6. Analýzy hmotného a nehmotného majetku
 - a) Analýzy stavu hmotného majetku podle vybraných kritérií,
 - b) Přehledy majetku dle inventárních čísel,
 - c) Inventurní soupisy majetku,
 - d) Přehledy budov a staveb,
 - e) Analýzy pohybů majetku,
 - f) Přehled majetku dle středisek,
 - g) Přehled majetku dle zodpovědných pracovníků,
 - 7. Opravy a údržba
 - a) Příprava instalací,
 - b) Příprava plánů oprav a obnovy,
 - c) Řízení oprav,
 - d) Vyhodnocování oprav,

Realizace:

Modul by měl poskytovat podporu pro zpracování variantních plánů údržby, generálních oprav. Vytváří důležité podklady pro plánování kapacit.

C.2. Personalistika

Modul Personalistika sleduje plánování, přípravu a optimální využití personálních zdrojů podniku. Zahrnuje evidenci a analýzy personálních zdrojů, přípravu podkladů pro plánování personálních kapacit, pro rekvalifikační programy apod.

VSTUPY	VÝSTUPY
informace o pracovnících,	evidence pracovníků,
informace o potenciálních pracovních zdrojích,	pracovní smlouvy,
požadavky na pracovní zdroje,	personální analýzy,
kvalifikační požadavky podle funkcí	rekvalifikační programy,
	plány personálního rozvoje

Funkce:

Informační systémy 1

1. Osobní evidence pracovníků - personální údaje
 - a) Evidence základních osobních údajů,
 - b) Pracovní poměry pracovníka,
 - c) Detailní popis pracovního místa - informace charakterizující pracovní místo, způsob a kým bylo místo dosud obsazeno, pracovní náplň,
 - d) Průběh kvalifikace pracovníka,
 - e) Osoby ve vztahu k pracovníkovi - údaje o osobách, které mají určitý vztah k pracovníkovi - pro odečitatelné položky ze základu daně,
 - f) Předchozí zaměstnání,
2. Zahájení a ukončení pracovního poměru
3. Analýzy pracovních sil vzhledem k potřebám podniku - analýzy stavu pracovních sil vzhledem k předpokládanému vývoji podniku (z hlediska funkční, kvalifikační, věkové struktury).
4. Příprava rekvalifikačních programů
 - a) specifikace potřebných školení a rekvalifikačních programů ve vazbě k funkčním místům a konkrétním pracovníkům,
 - b) specifikace harmonogramu školení,
 - c) evidence připravovaných a probíhajících vlastních školicích aktivit, evidence o relevantních nabídkách externích školicích možností.

C.3. Práce a mzdy

Modul Práce a mzdy (PAM) zajišťuje evidenci mzdových údajů o zaměstnancích, výpočty mezd, analýzy mzdového vývoje firmy, vstupy do účetnictví. Základním prvkem je mzdová složka - jakákoliv část mzdy, která má odlišný způsob zadávání nebo výpočtu nebo je třeba ji zvlášť zaúčtovat. Mzda pracovníka je potom složena z několika mzdových složek.

VSTUPY	VÝSTUPY
informace o pracovnících,	přehledy mezd, mzdové listy,
pracovní smlouvy,	mzdové uzávěrky,
informace o mzdových složkách	analýzy mzdového vývoje

Funkce:

1. Tabulka mzdových složek - určuje, jak se mzdová složka vyplňuje, jak se vypočte výsledná částka, jak se s výslednou částkou dále nakládá,
2. Odečitatelné položky z daňového základu - údržba číselníku odečitatelných položek,
3. Evidence mzdových údajů pracovníků
 - a) Evidence základních údajů o pracovnících,
 - b) Evidence průběhu pracovních poměrů pracovníka,
 - c) Evidence srážek z mezd,
 - d) Evidence osob se vztahem k pracovníkovi - údaje o osobách, které mají určitý vztah k pracovníkovi pro odečitatelné položky ze základu daně,
 - e) Evidence zdanění - součet příjmů podléhajících dani, sražené pojistné, nezdanitelné částky (vyživované osoby, invalidita, ...),
 - f) Nárok na dovolenou - definuje výši nároku na dovolenou a na dodatkovou dovolenou pro různé počty odpracovaných dnů a výši krácení dovolené pro různé počty zameškaných dnů,

Informační systémy 1

4. Výpočty mezd
 - a) Výpočty záloh,
 - b) Vyúčtování - výpočet mzdového listu pro každého pracovníka, včetně výpočtu daní, vstupem jsou mzdové složky a stálé srážky pracovníka,
5. Zpracování ročních mzdových listů
6. Rekapitulace mzdových složek za středisko - součty jednotlivých mzdových složek za středisko
7. Evidence a analýzy mezd a mzdových složek
 - a) Mzdové složky pro více pracovníků - analýza výskytů mzdové složky ve všech nebo vybraných střediscích,
 - b) Analýzy vyplácených mezd podle zadaných kritérií,
8. Zpracování mzdových uzávěrek
 - a) Měsíční uzávěrka mezd - výstup do účetnictví, generují se opakované mzdové složky, mzdové složky pro dluhy nebo přeplatky na daních, aktualizuje se čerpaná dovolená,
 - b) Roční zúčtování daně - výpočet ročního zúčtování daně pro pracovníky s vygenerováním příslušné mzdové složky,
9. Přehledy a analýzy mzdového vývoje

D.Výroba

D.1. Technická příprava výroby

Modul Technická příprava výroby (TPV) je založen zejména na práci s kusovníky zobrazujícími strukturu výrobku z různých pohledů.

V praxi se rozlišují tyto typy kusovníků:

- strukturální analytický kusovník - je celá struktura výrobku nebo vyráběné součásti a zachycuje všechny přímo i nepřímo vstupující položky. Vstupující součásti jsou zobrazeny tolikrát, kolikrát se vyskytují ve struktuře. Je zde také určeno množství nižší součásti do vyšší podle výrobních stupňů,
- stavebnicový analytický kusovník - zobrazená struktura výrobku nebo vyráběné součásti je pouze na úrovni o 1 stupeň níže než je zadaná úroveň hierarchické struktury,
- souhrnný kusovník - obsahuje všechny přímo i nepřímo vstupující součásti, již v agregované podobě. Každá součást je v přehledu obsažena pouze jednou a množství se sčítá za všechny shodné součásti vyskytující se ve výrobku (včetně ocenění). U nakupovaných položek je použito ocenění ve výši ceny materiálu, u vyráběných ve výši jednicových mzdových nákladů,
- inverzní (stavebnicový přehled použití) na celý sortiment - jsou zobrazeny všechny nižší položky - nakupované i vyráběné, přímo vstupující v jednom výrobním stupni do vyšších vyráběných položek,
- strukturální přehled použití - jsou všechny vyšší položky, do kterých zadaná položka vstupuje přímo i nepřímo.

Modul TPV zahrnuje i podporu konstrukce, projekce, evidenci norem, vývoj nových výrobků, evidenci a zpracování technologických postupů, specifikaci přípravků, zajištění náradí. Středně velký podnik má cca 25.000-30.000

Informační systémy 1

vyráběných a nakupovaných položek, 50.000 - 60.000 vazeb, 100.000 - 120.000 technologických operací.

VSTUPY	VÝSTUPY
technické výkresy, konstrukční rozpisky,	kusovníky dle uvedených typů,
požadavky na výrobu,	evidence kusovníkových položek,
informace o výrobcích,	přehledy a analýzy technologických postupů,
informace o technologiích,	přehledy materiálových a výkonových norem,
informace o výrobních kapacitách, pracovištích,	přehledy a analýzy nářadí a speciálních přípravků,
požadavky na změny	evidence a analýzy výrobních středisek,
	požadavky na kooperace

Funkce:

1. Evidence konstrukčních rozpisek
2. Evidence technických výkresů
3. Základní evidence kusovníkových položek
 - a) Základní údaje o kusovníkové položce - např. číslo položky, číslo provedení, technologického postupu, název položky, druh (vyráběná, nakupovaná, ...), normy rozměrové, jakostní, číslo výkresu, přejímací podmínky, ...
 - b) Údaje pro zásobování a odbyt - např. cena za jednotku, výhledová cena, kalkulační cena a další,
 - c) Údaje konstrukční a technologické - např. číslo změny, příčina změny, tvarové číslo, hmotnost čistá a hrubá, doba atestu, kód technologického postupu,
 - d) Údaje pro výrobu - např. doba kontrol, dávka optimální, minimální, procento zmetků, čas na operaci, kusový v hodinách, čas na operaci přípravný v hodinách,
4. Zpracování kusovníků - zahrnuje tyto kusovníky:
 - a) Stavebnicový analytický kusovník,
 - b) Strukturní analytický kusovník,
 - c) Souhrnný kusovník,
 - d) Inverzní kusovník,
 - e) Strukturní přehled použití,
5. Zpracování přehledů vypočtených norem - materiálových, výkonových, ...
6. Technologické operace
 - a) Operace technologického postupu,
 - b) Nářadí na operaci technologického postupu,
 - c) Určení použitých přípravků v operacích,
7. Popisy výrobních středisek a pracovišť, normy pracovišť
8. Výpočty na kusovníkových položkách
 - a) Dispoziční stupně, tzn. určení nejvyššího konstrukčního stupně položky ze všech kusovníkových struktur, ve kterých se položka vyskytuje,
 - b) Souhrnná norma materiálová,
 - c) Souhrnná norma - dle referentů,

Informační systémy 1

- d) Souhrnná norma výkonová,
- e) Operativní kalkulace,
- 9. Evidence základních technologických postupů
 - a) Zpracování tabulek manipulačních časů,
 - b) Promítnutí norem do pracovních postupů,
 - c) Rozpis pracovního postupu na dílny,
 - d) Vývoj technologických postupů,
- 10. Řízení technologické přípravy zakázky
- 11. Požadavky na kooperace
- 12. Evidence přípravků
 - a) Specifikace a konstrukce přípravku,
 - b) Přiřazení přípravků do pracovních postupů,
- 13. Zpracování mzdových lístků
- 14. Změnové řízení zakázek

Realizace:

Moduly TPV jsou ve většině informačních systémů již řešeny s vazbami na CAD - pro podporu konstrukčních prací a CAM systémy pro automatizovanou podporu řízení výroby.

D.2. Plánování výroby

Modul Plánování výroby zahrnuje především vytvoření a aktualizace výrobních zakázek, zadání výrobních čísel zakázky, přímý rozpis zakázky na výrobní a nákupní objednávky zakázky, kapacitní propočty.

VSTUPY	VÝSTUPY
poptávky, požadavky na výrobu z prodeje,	analýzy poptávek,
informace o výrobcích, postupech, kapacitách,	specifikace výrobní zakázky,
informace o zákaznících	kapacitní analýzy,
	plány finální výroby

Funkce:

1. Plánování zakázek zahrnuje:
 - a) Dlouhodobé plánování zakázek,
 - b) Plánování probíhajících zakázek,
 - c) Plánování navrhovaných zakázek,
 - d) Plánování připravovaných zakázek,
 - e) Plánování kapacit,
2. Evidence a vyhodnocování zákaznických poptávek, tj.:
 - a) Základní evidence poptávek,
 - b) Vyhodnocování poptávky vzhledem ke kapacitám,
 - c) Vyhodnocení ekonomické efektivity poptávky,
 - d) Zpracování a odeslání odpovědi na poptávku,
 - e) Sledování a analýzy poptávek a jejich pokrytí,
3. Zpracování výrobních a nákupních objednávek

Realizace:

Moduly pro Plánování výroby zahrnují i optimalizace tvorby plánů finální výroby na bázi matematických metod. Důležitým požadavkem na jejich řešení je i řešení přímých vazeb na informace poskytované z marketingu.

D.3. Operativní řízení a plánování výroby

Modul Operativní řízení a plánování výroby podporuje hlavně rozpis výrobních objednávek do jednotlivých výrobních operací, zpracování navážecích plánů. Obsahuje i přípravu tzv. průvodek a odběrních lístků na materiál, zpracování kalkulací, vytvoření operativního plánu pro střediska.

VSTUPY	VÝSTUPY
plán výroby,	operativní plány výroby,
informace o výrobcích a postupech,	navážecí plány,
informace o výrobních kapacitách,	plány odváděné výroby,
požadavky na změny	výrobní kalkulace,
	požadavky na kooperace,
	analýzy průběhu a zajištění zakázek,
	doklad změnových řízení

Funkce:

1. Operativní plánování výroby
 - a) Vytvoření zásobníku práce,
 - b) Prověřování zásobníku práce, kontrola zajištění výrobního příkazu - kontrola dle rozpisky, pracovních postupů, přípravků, externích kooperací, dílců z vlastní výroby,
 - c) Sestavení plánu odváděné výroby,
 - d) Změnové řízení zakázek,
 - e) Sestavení operativního plánu výroby,
 - f) Uvolnění materiálu do výroby,
2. Průběh zakázky provozy výroby
 - a) Určení výdeje materiálu na pracoviště,
 - b) Vyskladnění materiálu,
 - c) Kontrola jakosti materiálu, vyhodnocení kvality,
 - d) Storna uvolnění materiálu a řešení změn,
 - e) Vyhodnocování mezioperačních kontrol,
 - f) Evidence o ukončení uzlových operací,
 - g) Vyhodnocování rozpracovanosti výroby,
 - h) Odváděcí doklad po skončení výroby,
3. Zpracování kooperací
 - a) Vystavení požadavku na kooperaci,
 - b) Výběr dodavatele podle zadaných kritérií,
 - c) Vystavení objednávky na provedení práce,
 - d) Zajištění přepravy,
 - e) Dokumentace pracovních postupů,
 - f) Kontrola přijatého výrobku a potvrzení kvality,
 - g) Reklamace,

Informační systémy 1

Realizace:

Velmi důležitou charakteristikou modulu je efektivní podpora změnového řízení výrobních zakázek.

D.4. Dílenské řízení výroby

Modul Dílenské řízení výroby zahrnuje především příjem rozpracované objednávky, příjem náradí, odvedení výrobní objednávky, rozbor výrobních objednávek (nezajištěné materiálem, polotovary, náradím, rozpracované, ...).

VSTUPY	VÝSTUPY
operativní plány výroby,	rozpisy dílenských plánů,
informace o výrobcích, postupech, kapacitách,	zajištění pracovišť materiálem a náradím,
informace o změnách výrobních zakázek,	mzdové výkazy,
	analýzy výrobních objednávek,
	protokol změn. řízení

Funkce:

Rozpis dílenského plánu na pracoviště

1. Zajištění pracovišť materiálem a náradím, tj.:
 - a) Příjem materiálu a polotovarů,
 - b) Objednávky náradí,
 - c) Příjem náradí,
 - d) Operativní nákupy režijního materiálu,
2. Dílenská vstupní kontrola
3. Řízení výroby dle specifikovaného pracovního postupu
 - a) Sledování výroby podle výrobních operací,
 - b) Vystavení odváděcího dokladu včetně dokumentace,
 - c) Předání do expedice - výrobku i dokumentace,
 - d) Předání zákazníkovi,
 - e) Archivování odváděcího dokladu,
 - f) Vykazování mzdových odchylek,
4. Zmetkové hlášení
5. Analýzy stavu výrobních zakázek
 - a) Výrobní zakázky nezajištěné,
 - b) Výrobní zakázky rozpracované,
 - c) Výrobní zakázky v kooperaci,
 - d) Výrobní zakázky k odvedení,
 - e) Výrobní zakázky odvedené k expedici,
 - f) Kontrola a analýza plnění termínů,
 - g) Přehled pozastavených zakázek,
6. Mzdové výkazy pracovníků
7. Výpisy inventurních stavů
8. Změnové řízení
 - a) Storno zakázky,
 - b) Dodatek zakázky,

Informační systémy 1

- c) Změna počtu kusů,
 - d) Změna výrobního postupu,
9. Odvedení celé zakázky

Realizace:

Modul Dílenské řízení je typický značnými rozdíly podle typu výroby. Obdobně jako v modulu Sklady je zde silná vazba na využití čárových kódů. Řízení kvality na úrovni řízení výroby je rovněž součástí tohoto modulu (evidence a vyhodnocování mezioperačních kontrol).

E. Přeprava

Modul Přeprava podporuje zajišťování dopravních a přepravních požadavků firmy, tj. např. stanovení přepravních tras, výběr přepravce apod.

VSTUPY	VÝSTUPY
informace o přepravních,	přehledy a analýzy požadavků na přepravu,
přepravní tarify,	přepravní kalkulace,
informace o přepravních trasách,	analýza přepravců a přepravních možností
požadavky na přepravu	

Funkce:

1. Přehledy a analýzy požadavků na přepravu
2. Kalkulace přepravních nákladů
3. Analýzy přepravců a jejich podmínek
4. Výběr přepravce - analýza dopravních možností podle jednotlivých přepravců (tuzemští nebo zahraniční přepravci), výběr přepravce.
5. Souhrnné analýzy přepravních možností - analýzy vývojových trendů přepravních možností. Analýza aplikovatelnosti a efektivnosti přímého provázání s nejdůležitějšími přepravními partnery.

F. Servis

F.1. Řízení jakosti

Modul Řízení jakosti plní v informačním systému specifickou roli a představuje komplexní systém kontrol.

VSTUPY	VÝSTUPY
metodické materiály v oblasti jakosti,	evidence metodik a norem jakosti,
normy jakosti,	zpracování dokumentace pro jakostní certifikace,
dokumentace ISO 9000,	analýzy kontrol,
požadavky na kontrolní činnosti,	plány kontrolních aktivit
požadavky na kontrolní a testovací zařízení	

Informační systémy 1

Funkce:

1. Metodické podklady pro řízení jakosti
2. Příprava organizačních směrnic
3. Prověřování návrhu smluv
4. Řízení jakosti ve vlastním výrobním a montážním procesu
5. Vytváření a archivace dokumentace kontroly jakosti
6. Kontrolní, měřicí a zkušební zařízení
7. Vyhodnocování kontrol a zkoušek
8. Operativní řízení neshodného výrobku
9. Nápravná opatření
10. Evidence a vyhodnocování záznamů o jakosti výrobků
11. Plánování, evidence, vyhodnocování interních prověrek jakosti

Realizace:

Velmi často je celá oblast řízení jakosti integrována do jednotlivých výrobních, nákupních, distribučních a dalších modulů a tvoří samostatný modul.

F.2: Organizace a správa systému

Modul Organizace a správa systému plní v některých informačních systémech centrální (správcovskou) roli. Zahrnuje správu a údržbu celopodnikových datových zdrojů, standardů vstupujících do připravovaných dokladů, uživatelsky orientované technologické funkce (nastavení zpráv, parametry tisku apod.).

VSTUPY	VÝSTUPY
informace o organizační struktuře, pracovních místech,	správa základních databází IS, centrálních číselníků,
informace o uživatelích informačního systému,	monitorování a analýzy provozu,
informace o zdrojích informačních technologií,	zajištění přístupových práv, evidence uživatelů
požadavky na informační systém	

Funkce:

1. Definice organizační struktury nebo organizačních struktur
2. Evidence a popisy funkčních míst
3. Podpisový řád
4. Podnikové kalendáře
5. Definice účtové osnovy
6. Správu centrálních číselníků
7. Nastavení parametrů provozu IS (parametrů pro tisky, standardní vzhled obrazovek, nastavení jazykového prostředí apod.)
8. Analýzy provozu informačního systému

Informační systémy 1

Realizace:

Moduly správy a organizace mají často charakter metasystémových aplikací. Mohou to být specializované aplikace, které nejsou součástí informačního systému. Rozdíly jsou v jejich komplexnosti a možnostech podpory řízení vývoje a provozu informačního systému.

F.3 Kancelářské úlohy

V komplexu implementace aplikačního software organizace je rovněž významné řešení vazeb na kancelářské úlohy. Tyto kancelářské úlohy vytváří aplikace sloužící ke sledování a koordinaci obchodních procesů v organizaci a počítačovou podporu při řešení běžných agend, jak uvnitř organizace, tak i pro styk s obchodními partnery.

Funkce:

- evidence interního oběhu a archivace dokladů a informací, došlých do podniku,
- pořizování, interní oběh a archivace zpráv a dokladů textového charakteru, vznikajících v podniku pro interní i externí potřebu,
- převzetí hotových dokladů datového charakteru do kancelářského systému, jejich interní oběh a archivace,
- identifikace všech dokladů, informací a zpráv v rámci kancelářského systému až na úroveň jednotlivých zakázek,
- využití integrovaného systému elektronické pošty pro řízení událostí pro jednotlivé pracovníky podniku na všech úrovních,
- autorizovaný vzdálený přístup do kancelářského systému,
- napojení na centrální faxový systém,
- napojení na e-mail Internet server,
- transformace Internetových e-mail zpráv do elektronické pošty a naopak.

G. Stavební výroba

G.1. Kalkulace nákladů, rozpočty, výkaz výměr

Modul Kalkulace nákladů, rozpočty, výkaz výměr v sobě integruje moduly pro sestavení výkazu výměr, rozpočtu ve směrných cenách, individuální kalkulace a modul pro výpočet a aktualizaci limitek potřeb přímých nákladů.

VSTUPY	VÝSTUPY
popis objektu,	limitky potřeb přímých nákladů,
položky z normativní základny,	potřeba materiálů, normohodin, mezd, profesí,
výkaz materiálů z CAD systému,	potřeba strojů a OPN,
kalkulační vzorce,	archiv nové akce
archiv starých akcí	

Funkce:

- Uživatelem volně definovatelné členění objektu do kapitol podle tzv. speciálního kódu - např. do oddílů, do technologických etap, do

jednotlivých podlaží, podle subdodavatelů aj. Členění lze navíc kdykoliv během práce účelově měnit. Samozřejmostí je možnost rozdělení akcí na stavby a objekty.

- Pro popis objektu je k dispozici informační karta, umožňující provést předvolbu základních údajů pro kalkulaci ceny i podrobný popis.
- Společné zadání výkazu výměr, rozpočtu a kalkulace. Výměru lze zadat rovnou nebo s využitím aparátu pro její výpočet. Ocenění se automaticky provádí ve dvojích cenách. Jako příklad lze uvést ocenění v individuálně kalkulovaných cenách a ve směrných orientačních cenách. Program bude nabízet velice široké možnosti jak obě ceny dále upravovat.
- Program musí umožňovat využívat archiv akcí při tvorbě nových cenových nabídek. Ze zvolených akcí lze přebírat skupiny položek do nového zadání nebo vytvářet cenovou nabídku ve více variantách.
- Pohodlné vyhledávání položek v normativní základně podle zvoleného klíče (lze vyhledávat i podle názvu). Také je možné si předem navolit určitý interval (obor) položek nebo konkrétní ceník. Při přebírání položek do rozpočtu lze označit více položek a přenést je do zadání najednou.
- Možnost převzít výkaz materiálů a prací z nejpoužívanějších CAD systémů pro projektování (např. CADKON, DDS). Vazba je založena na zpracování výstupů z CAD programů s využitím katalogu agregovaných položek. Realizace této vazby umožňuje nahradit pracné odečítání výměr z projektové dokumentace.
- Podpora plné textace položek až do rozsahu jedné strany. Kteroukoliv položku rozpočtu lze popsat v nabízeném rozsahu jedné strany textu. Texty lze ukládat do normativní základny. Je to krok od počítačově zkratkovitých textů k textům plně popisným a jednoznačným, srozumitelným i neprofesionálům.
- Volně definovatelný kalkulační vzorec akce. K dispozici je přehledný aparát pro tvorbu kalkulačního vzorce jako matematického výrazu, kde i jeho složky (režie, zisk, riziko) lze opět definovat jako výrazy. Možnosti tvorby ceny jsou takřka nevyčerpatelné a připravené na jakýkoliv i velmi neobvyklý způsob kalkulace. Stejně dobře jako ceny prací lze kalkulovat ceny materiálů (výrobky, polotovary) a strojů v provozní sazbě. Správnou funkci vzorce lze odzkoušet kontrolním výpočtem.
- Aparát pro tvorbu a využití agregovaných položek (balíčků). V modulu lze přímo v zadání akce označit skupinu položek a pod zvoleným jménem uložit do katalogu balíčků k dalšímu využití. Balíčky spolu s dalšími položkami lze dále seskupovat do dalších (vyšších) balíčků. Lze tedy vytvářet hierarchicky uspořádané agregované položky. V rozpočtu je možné agregace kdykoliv upravovat a rozpouštět do jednotlivých položek a v případě potřeby opět agregovat.
- Kromě cen prací je možné oceňovat podle zadaného kalkulačního vzorce i materiály (polotovary, výrobky) a stroje.

Informační systémy 1

- Možnost aktualizace normativní základny firmy přímo ze zadání akce. Modul bude dále nabízet modifikaci načtených položek s tím, že úpravy se budou týkat pouze zadávané akce, nebo je možné provedené změny v akci zpětně promítnout do normativní základny.
- Přehledné uspořádání informací na obrazovce umožňující průběžně sledovat ocenění celého objektu i jeho částí (oddílů).
- Pro zadání výměr je k dispozici vestavěná kalkulačka i vysoce výkonný aparát umožňující definovat konstanty, pracovat s figurami, prokládat výpočty komentáři, kopírovat celé výrazy a samozřejmě zaznamenávat kompletní postup výpočtu výměry pro kontrolu a případný tisk výkazu výměr.
- Zpracované akce je nutné ochránit heslem před jejich neoprávněným použitím a přehledně řešenou možnost zálohování zpracovaných akcí.
- Při práci na cenové nabídce musí být k dispozici široké možnosti pro úpravu cen formou indexů zadávaných na úrovni objektu, oddílu nebo položky. Je nutné velice efektivně upravovat mzdy a normohodiny a měnit v případě potřeby procenta režií, zisku a rizika.
- Ze zadané kalkulace lze spočítat limitky potřeb přímých nákladů {materiály, normohodiny, mzdy, profese, stroje, OPN). Položky v limitkách jsou řazeny podle kódového označení a nebo podle jejich podílu na ceně (lze vypíchnout cenově nejvýznamnější materiály, stroje atd.). Limitky je možné dále upravovat (měnit ceny, vypočtená množství, případně nahrazovat položky jinými z normativní základny) a po ukončení aktualizace promítnout provedené úpravy zpět do kalkulovaných cen.
- Tiskové sestavy musí být řešeny velice variabilně, předem lze navolit obsah tiskové sestavy, formu, rozsah stran, lze volit počet výtisků atd. Zvolená sestava je nejprve vytvořena do souboru pro možnost prohlížení před vlastním vytištěním na obrazovce, případně pro editaci textovým procesorem. Vlastní tisk na tiskárnu bude probíhat buď přímo nebo z tiskové fronty na pozadí, tzn. že i v průběhu tisku je možné s počítačem normálně pracovat.

G.2. Komplexní údržba datové základny

Modul Komplexní údržba datové základny musí být pohodlným a komfortním prostředkem pro údržbu datové základny.

VSTUPY	VÝSTUPY
naplnění datové základny musí být součástí dodávky modulu	stavební materiály, ceníky stavebních a montážních prací,
	sazebníky profesí, mzdových tarifů, stavebních strojů, OPN,
	obecné číselníky (měrné jednotky, seznam firem, seznamy VRN,
	kalkulační varianty a plné textace položek stavebních a montážních prací,

	skladby polotovarů, dodavatelské ceny, textace materiálů, skladby sazeb strojů
--	--

Funkce:

Datová (normativní) základna slouží především jako zdroj dat pro tvorbu nabídkových rozpočtů a kalkulací nákladů. Skládá se z jednotlivých katalogů. Každý katalog je popsán jménem (odpovídá jménu fyzického souboru na disku), stručným jednořádkovým popisem, počtem klíčů, maskou hlavního klíče a definicí vazeb na ostatní katalogy. Celá datová základna je tématicky rozdělena do pěti skupin katalogů:

- základní katalogy (stavební materiály, ceníky stavebních a montážních prací),
- sazebníky (profese, mzdové tarify, stavební stroje, OPN),
- obecné číselníky (měrné jednotky, seznam firem, seznamy VRN),
- kalkulační varianty a plné textace položek stavebních a montážních prací,
- ostatní (skladby polotovarů, dodavatelské ceny, textace materiálů, skladby sazeb strojů).

Mezi jednotlivými katalogy existují určité přesně definované vazby, které se při aktualizaci automaticky udržují. Např. katalogy stavebních prací v základní skupině mají přímou vazbu na katalogy kalkulačních variant; variantní katalogy jsou dále napojeny na sazebníky.

Realizace:

- Program Komplexní údržba datové základny umožňuje přehlednou a velmi detailní položkovou aktualizaci. V položkách jednotlivých katalogů lze listovat v řádkovém i formulářovém režimu, je možné vyhledávání a řazení podle několika klíčů (např. u materiálů podle event. čísla, SKP, názvu). Pro rychlejší orientaci v rozsáhlých seznamech položek lze používat zúžené intervaly a filtry.
- Aktualizace zahrnuje základní operace s položkami: opravu, vkládání, vymazání (s možností obnovy), kopírování. Režim rutinní položkové aktualizace lze podle potřeby uživatele zpříjemnit a urychlit řadou konfiguračních nastavení a předvoleb.
- U každé položky datové základny jsou k dispozici minimálně dvě ceny - vlastní (firemní, uživatelská) cena a srovnávací cena.
- K položkám stavebních a montážních prací lze vytvářet neomezený počet technologických variant, cena varianty je kalkulována ze skladby přímých nákladů (norem) pomocí volně definovatelného kalkulačního vzorce. Skladbu nákladů je možné zadávat (tj. kalkulovat) i k položkám materiálů (polotovary - malty, betony) a stavebních strojů.
- Základní jednořádkový popis položek lze rozšířit o tzv. plnou textaci, což je libovolný text v rozsahu až 60 řádek o 80 znacích. Navíc lze plnou textaci rozdělit na dvě části, což lze následně využít v tisku, např. pro hrubé a podrobné popisy nebo dvoujazyčné texty. Pro tvorbu a aktualizaci plné textace je k dispozici pohodlný textový editor.

Informační systémy 1

- Pro profesionální práci s datovou základnou jsou v programu integrovány některé speciální servisní funkce: obnova vymazaných záznamů, rekonstrukce poškozených indexových i datových souborů, export a import dat v různých formátech.
- U každého materiálu lze vedle katalogových cen udržovat také libovolný počet dodavatelských cen, které je možné využít při aktualizaci ceny v limitkách potřeb u zadání konkrétní akce.
- Pokročilí uživatelé mohou provádět údržbu datové základny pomocí tzv. hromadných operací. Hromadným zpracováním je možné v katalogu naráz upravit uživatelem zadané intervaly položek. Výběr položek pro zpracování lze také omezit podle data jejich poslední aktualizace. K provádění hromadných operací musí mít každý uživatel oprávnění, které přiděluje správce systému (ochrana před poškozením dat neodborným zásahem).
- V modulu jsou dostupné tyto hromadné operace:
 - zrušení / obnova položek,
 - kopírování / přesun položek mezi katalogy stejného druhu (volitelně včetně příslušných kalkulačních variant a plných textací),
 - přecenění vybrané ceny (indexové nebo kalkulační),
 - editace norem kalkulačních variant (lze provádět ve třech režimech: záměna, vymazání, vložení norem),
 - import / export položek (formát ASCII s pevnou délkou věty nebo s oddělovači),
 - tisk položek.
- Koncepce modulu je postavena na dvou základních principech:
 - flexibilita - vazby a struktury jednotlivých částí datové základny jsou téměř ve všech úrovních údržby volně definovatelné. Datovou základnu lze takto sestavit ("ušít na míru") přesně podle potřeb firmy,
 - bezpečnost dat - na úrovni operačního systému je pro správu datových souborů normativní základny použita velmi spolehlivá a vyspělá technologie. Data jsou zabezpečena i před neodbornými zásahy uživatelů pomocí přístupových práv, která nastavuje správce systému .

G.3. Vyhodnocení nabídkového řízení

Modul Vyhodnocení nabídkového řízení je určen pro vyhodnocení cenových nabídek dodavatelů (subdodavatelů). Usnadňuje tak výrazně práci spojenou s vyhlášením a vyhodnocením konkursního řízení na dodávku stavby (nebo její části).

Funkce:

Na základě zadání rozpočtu pomocí modulu "Kalkulace nákladů, rozpočty, výkaz výměr" získáme podklady pro výběrové řízení včetně vlastního cenového ohodnocení jednotlivých položek a kapitol. Z tohoto zadání modul "Vyhodnocení nabídkového řízení" umožní vytvořit datový soubor slepého rozpočtu (pouze výměry, nikoliv ceny), který se předává zájemcům. Hlavní

Informační systémy 1

výhodou takového postupu jsou jednotné podmínky pro všechny zájemce o dodávku stavebního díla.

Realizace:

Datový soubor slepého rozpočtu modul uloží na zvolený disk, zároveň s tímto souborem nahraje i další samostatný volně šířitelný program, který slouží pro doplnění cen do slepého rozpočtu. Tyto soubory obdrží každý účastník konkursu a pomocí dodaného programu vyplní k jednotlivým položkám své ceny. V případě, že se mu bude zdát zadání neúplné, může doplnit ty položky prací nebo materiálů, které dle jeho názoru v zadání chybějí. Poté vrátí soubory zpět vyhlášovateli konkursu včetně tiskové sestavy své cenové nabídky.

V následující fázi vyhodnocení nabídek všech zájemců o dodávku díla provedeme pomocí programu cenové porovnání jednotlivých nabídek včetně vlastních cen. Porovnání lze provádět na úrovni celého objektu, kapitol i jednotlivých položek. Na obrazovce počítače je v příslušných sloupcích vždy vyznačena nejnižší cena za objekt, kapitolu či položku. V případě, že byly některým účastníkem konkursu doplněny položky, které původně nebyly předmětem zadání, jsou zobrazeny odlišnou barvou a ceny ostatních dodavatelů vynulovány.

Porovnáním cen se tak získají podklady pro další rozhodování o výběru vhodného dodavatele stavebního díla.

G.4. Fakturace a čerpání rozpočtu

Modul Fakturace a čerpání rozpočtu je určený pro čerpání jednotlivých položek zadaného rozpočtu nebo kalkulace v čase. U každé položky se uchovává historie jejího čerpání po celé období realizace stavby. Program je doplněn o velmi významnou část pro práci s cenami, kromě množství lze při čerpání položky zadávat i novou jednotkovou cenu (např. u materiálů v závislosti na pohybu dodavatelových cen). To umožňuje uchovávat historii vývoje ceny každé položky. Program provádí porovnání nových cen oproti původnímu rozpočtu a automatický výpočet cenových přírůstků za jednotlivé položky, jednotlivé kapitoly a celý objekt.

Funkce:

- Čerpání položek v celém rozsahu kalkulačního vzorce (výrobní faktura). Lze např. získat údaje o plánované výši vyplacených mezd a porovnat se skutečností podle účetnictví.
- Tisková sestava zbytkového rozpočtu.
- Je umožněno procentuální (hromadné) čerpání libovolné kapitoly.
- Možnost úpravy jednotkové ceny formou indexu při hromadném čerpání kapitoly.
- Lze zadat libovolné sumační období a tak vytvářet různé časové pohledy na stav čerpání (sumace tedy není omezena na kalendářní měsíc či rok).
- Existuje přímá vazba na datový soubor rozpočtu nebo kalkulace, tzn. že každá změna v zadání rozpočtu se při sumaci automaticky promítne do stavu čerpání. Při změně zadání rozpočtu je znemožněno vymazání položek, které již mají čerpání.

Informační systémy 1

- U libovolné rozpočtové či kalkulační položky lze provést zákaz čerpání.
- Lze vytvářet několik druhů tiskových sestav pro libovolné časové období (součtové, položkové, přírůstkové, ...).
- Možnosti exportu čerpaných, zbytkových i celkových hodnot.
- Součástí programu je prostředek pro tvorbu, tisk a snadnou evidenci faktur. Všechny údaje, je možné exportovat pro další využití.

G.5. Rozpočtové ukazatele

Modul Rozpočtové ukazatele slouží pro tvorbu rozpočtových ukazatelů stavebních objektů. Na základě těchto ukazatelů je možné rychle stanovit orientační cenu objektu (např. v případě nedostatku podkladů nebo časové tísně). Vybraný objekt lze charakterizovat až pěti libovolnými druhy měrných účelových jednotek (např. m³ obestavěného prostoru, m² užité plochy apod.), pro každou z nich pak získáme jednotkovou cenu na jednotlivé kapitoly (např. oddíly HSV, PSV, ...) a celý objekt.

Ukazatelové ceny se ukládají do katalogu, ve kterém snadno vyhledáme podobný objekt (dle SKP, názvu nebo pořadí), který nás zajímá. Katalog lze velmi lehce doplňovat o nové objekty, na které již byla zpracována kalkulace nákladů (nebo rozpočet).

Požadovaný objekt popíšete měrnými jednotkami, přenesete jednotlivé kapitoly zadané v kalkulaci a popř. upravíme výsledné ceny kapitol. Lze tak získat ukazatele pro další typy objektů, které mohou v budoucnu usnadnit rozhodování při přijímání zakázek.

G.6. Harmonogramy

Modul Harmonogramy je určen pro sestavení harmonogramu postupu výstavby objektů, které byly zpracovány modulem "Kalkulace nákladů, rozpočty, výkaz výměr". Harmonogram sestaví časové ohodnocení jednotlivých kapitol v kalkulaci nákladů.

Funkce:

- Jeden harmonogram může obsahovat libovolný počet objektů (tzn. možnost sestavení harmonogramu celé stavby).
- Po převzetí objektu z kalkulace do harmonogramu je spočtena implicitní délka trvání kapitol a to podle počtu normohodin a délky směny.
- Zadávání harmonogramu dle potřeby v několika časových jednotkách (rok, měsíc, týden, den) s maximální přesností jeden den. Nejmenší jednotkou pro zobrazení je týden.
- Bohaté možnosti v zadávání délky trvání kapitol (posun začátku, konce nebo celé kapitoly), posun označené skupiny kapitol (i z několika objektů současně), posun celého objektu, rozdělení realizace kapitoly až na 14 samostatných částí, posun délky trvání jednotlivých částí podle potřeby.

Informační systémy 1

- Začátky a konce realizace objektů se počítají podle změn začátků a konců jednotlivých kapitol , společně s kapitolami je zobrazena i délka trvání výstavby celého objektu.
- Tři způsoby třídění objektů a kapitol - podle kódu kapitol v rámci jednoho objektu, podle počátků, popř. stejné kapitoly z různých objektů za sebou.
- Výpočet časového průběhu finančních nákladů za celou dobu realizace v měsíčních částkách nebo kumulovaně.
- Výstup harmonogramu na tiskárnu za celou stavbu, objekty a kapitoly.



Kontrolní otázky:

1. Vyjmenujte základní úlohy, které by měly podporovat podnikové IS.
2. Jaké procesy (tzv. ERP) by měly podnikové IS podporovat?
3. Vyjmenujte základní moduly podnikového IS.
4. Jaké jsou základní funkce modulu elektronický obchod?



Úkoly k zamyšlení:

Zamyslete se nad modulem informačního systému sklad, pokuste se popsat jak by se lišily funkčnosti skladu velkého dodavatele např. Mountfield, Ferona, Philips od funkčností malého dodavatele.



Korespondenční úkol:

Zanalyzujte trh ERP systémů a pokuste se vypracovat jednoduchý přehled podnikových informačních systémů, dostupných na našem trhu. Pod pojmem podnikový IS máme na mysli IS, který zahrnuje základní funkčnosti zmíněné v této kapitole. K vyhledání jak domácích, tak zahraničních dodavatelů těchto systémů využijte všech možných kanálů a také srovnání, která jsou dostupná např. v časopisech connect, IT systems, Business World nebo na stránkách české společnosti pro systémovou integraci (www.cssi.cz).



Shrnutí obsahu kapitoly

V této kapitole jste se seznámili s úlohami a základními funkcemi podnikových informačních systémů. Funkce podnikových IS se sdružují do tématických celků, kterým říkáme moduly. Součástí této kapitoly bylo tedy také vymezení funkčností modulů, jejich vstupy a nezbytné požadavky na jejich funkčnost.

10 Tvorba projektů

Cílem následující kapitoly je zopakování některých základních pojmů z problematiky projektového řízení. Předpokladem k vytváření projektů jsou základní znalosti z problematiky projektového řízení.

Znalosti z projektového řízení je možné doplnit prostudováním doporučené literatury.

Klíčová slova:

Projekt, projektové řízení, fáze projektu, rozsah projektu, oblast znalostí, projektové procesy, struktura projektového cyklu, CPM.

*Vysvětlení základních pojmů je důležité pro porozumění mezi účastníky řízení projektů. Ústředním pojmem je pojem **projekt**. Existuje řada definic, které se snaží tento pojem definovat. Uvedeme některé z nich, které poskytnou detailnější pohled na to, co pojem projekt zahrnuje.*



10.1 Projekt

Projekt je výsledek materiální nebo nemateriální povahy založený na strategickém plánu, navržený, organizovaný a realizovaný pod řízením někoho v zájmu vlastníka nebo zadavatele.

Projekt je aktivita omezená v čase, realizovaná pouze jedenkrát bez opakování se značným množstvím charakteristických rysů, ke kterým patří:

- výsledek musí sloužit užívání po celou dobu přesně určenou zadavatelem projektu,
- úspěch projektu při jeho zahájení není zřejmý,
- trvání projektu je časově omezeno,
- projekt je uskutečňován mimo běžnou podnikatelskou rutinu,
- zdroje pro realizaci projektu jsou limitovány,
- projekt má jen jeden výsledek.

Projekt je jednorázový proces:

- směřující k dosažení stanovených cílů,
- během procesu prochází projekt řadou etap a fází,
- s etapami se mění úkoly, organizace a zdroje.

Projekt je prostorově a časově ohraničený soubor technologicky a organizačně souvisejících činností, jehož účelem je dosažení stanoveného cíle při zadaném čase, zdrojích, nákladech a kvalitě.

Činnost je časově ucelená transformace vstupů činnosti (lidské zdroje, finanční zdroje, technologie, zařízení, suroviny, materiál, energie atd.) na výstupy činnosti (výrobky, služby)

Technologické vazby jsou vyvolány technologickou návazností jednotlivých činností na sebe.

Organizační vazby jsou dány časovým a prostorovým uspořádáním omezených zdrojů.

Projektové řízení

Projektové řízení je způsob řízení pomocí projektů. Je to vysoce účinný nástroj řízení změn, komplexní koncepce efektivního dosahování projektových cílů, která umožňuje manažerům dosáhnout odpovídající kvality výstupu s minimálními nároky na čas a ostatní zdroje. Projektové řízení zahrnuje řízení jednotlivých projektů a vytvoření organizační struktury a koordinaci projektů z hlediska termínů a disponibilních zdrojů.

Z rozdílů mezi projektovým řízením a konvenčním řízením vyplývá potřeba specifických nástrojů a technik. Projektové řízení využívá specifických nástrojů, technik, znalostí a dovedností k dosažení stanovených cílů. Nástroje projektového řízení poskytují flexibilitu pro plánování, řízení a sledování projektů. Dávají možnost rychle a efektivně reagovat na nevyhnutelné změny projektů. Sofistikované postupy se používají pro časově limitované a zdrojově omezené rozvrhování i více projektů současně. Mezi základní požadavky projektového řízení patří požadavek splnění dílčích termínů a konečného termínu dokončení projektu. Pro analýzu neurčitosti se používají metody a nástroje rozhodování za rizika a nejistoty. Projektové řízení je řízení týmu, vytvořit práceschopný tým je výrazem umění. Při řešení problémů řešitelským týmem se využívá **synergie** (synchronizace energie) pro dosažení společného cíle. Dochází k **synergickému efektu**, kdy výkon týmu je vyšší než souhrn výkonů jednotlivců, kteří by řešili problém izolovaně.

Projektové řízení ve smyslu řízení projektů lze definovat jako způsob řízení složitých, diskrétních plánovaných úkolů s vysokou mírou neurčitosti a vysokou mírou komplexnosti, který zahrnuje řízení tvorby projektů, řízení realizace projektů a řízení oživování projektů. K pochopení projektového řízení ve smyslu řízení podle projektů mohou přispět následující definice:

Projektové řízení je řízením změn a úlohou ředitele projektu je dbát o to, aby tyto změny byly realizovány správně.

Projektové řízení je řízení všeho, co má začátek a konec. Konec je daleko důležitější než začátek a zahrnuje v sobě efekty z užití letadel, závodů, přehrad, elektráren, výpočetních systémů a ostatních výsledků investic do projektu. Projektové řízení se odlišuje od řízení služeb nebo výroby v tom, že vede vždy přímo ke konci, nikoli k udržování a vylepšování plynulé činnosti, jako je správa hotelu nebo výroba chleba a zákusků.

Projektové řízení představuje mobilizaci multidisciplinárního týmu (ve kterém má každý člen určenou svou odpovědnost) vedoucí k implementaci projektu do cílů zadavatele projektu - cílů termínových, kvalitativních a nákladových.

Projektové řízení umožňuje zadavateli projektu být oproštěn od dennodenních řídicích funkcí nezbytných pro realizaci projektu; přitom nechává v jeho rukou takový stupeň kontroly, který zadavatel považuje za nezbytný pro dosažení jeho požadavků a priorit.

Obsáhlou definici projektového řízení uvádí L.Chrudina: **Projektové řízení** je součást plánování rozvoje sociálních organismů. Je to souhrn činností řídicího subjektu, směřující k dosažení plánovaných cílů v plánovaném čase a nákladech a prováděných v podmínkách charakterizovaných:

- neopakovatelností cílů a způsobu jejich realizace,

Informační systémy 1

- variantností cest realizace,
- značným počtem účastníků (tvůrců i realizátorů projektu se složitými a různorodými vzájemnými vztahy, zájmy a kompetencemi; účastníci i vztahy se v průběhu projektu mění),
- radikálními změnami problémů rozhodování v průběhu života projektu,
- značným výskytem nejasně definovaných problémů, změn, vysokou mírou nejistot a rizik.

Projektové řízení je řízením nespojitých změn v sociálním organismu a vyžaduje vzhledem k uvedeným charakteristikám, aby bylo prováděno na základě projektů.

10.2 Tradiční modely projektového řízení

Program Microsoft Project ve svých modelech projektového řízení řeší softwarově takové metody, jako jsou:

- **Critical Path Method (CPM).** Proces komputerizace projektového řízení začal už v roce 1950. Společnost Dupont Corporation & Remington Rand ve snaze zdokonalit techniku plánování projektů, vyvinula plánovací systém, nazvaný Critical Path Method nebo CPM plánování. CPM je matematický model, který počítá celkové trvání projektu založeného na individuálních úkolech, navzájem na sebe vázaných, s označením, které z činností jsou pro celý projekt kritické. Tento model je fundamentální metodou pro dnešní plánování činností pomocí software, včetně Project4.
- **Program Evaluation Review Technique (PERT).** V roce 1950 vyvíjelo námořnictvo Spojených států Amerických dálkově řízenou střelu POLARIS pro ponorky. Hlavní účastník a řešitel projektu firma Lockheed vyvinula plánovací systém PERT, který používá pravděpodobnostního modelu při stanovení očekávané délky trvání jednotlivých činností projektu. Dnes se PERT diagram (někdy označován jako síťový diagram) používá pro grafické zobrazení vazeb mezi jednotlivými úkoly.
- **Ganttův diagram.** Henry L. Gantt vyvinul grafický způsob zobrazení délky trvání činností a jejich zobrazení na časové ose. Tato grafická reprezentace, nazývaná původně Úsečkový diagram, byla na jeho počest později nazvána jeho jménem.

Otázky

- 1) Co je to projekt? Jak byste jej definovali?
- 2) Co je důležité při řízení projektů?
- 3) S jakými definicemi řízení projektů jste se seznámili?
- 4) Co je to projektový cyklus?

Korespondenční úkol

Definujte zadání projektu, definujte jednotlivé fáze svého projektu, rozhodněte se, jaké zdroje k řešení budete potřebovat a navrhnete zadání pro zpracování v Microsoft Projectu.



11 Ekonomika podniků

V této kapitole se dozvíte:

- Co je to národní hospodářství?
- Jak je definováno podnikání, podnik, podnikatel, obchodní společnost, podnikatelské okolí?
- Jaké zákony se vztahují k této oblasti?
- Co definuje strategie podniku, co je to podnikatelský záměr?

Po jejím prostudování byste měli být schopni:

- Porozumět základním ekonomickým pojmům a definicím.

Klíčová slova této kapitoly:

Národní hospodářství, typologie podniků, podnikání, podnikatel, obchodní společnost, podnikatelské prostředí, strategie podniku, podnikatelský záměr.

Doba potřebná ke studiu: 5 hodin

Průvodce studiem

Tato kapitola Vás seznámí se základními pojmy z oblasti ekonomie. Dočtete se zde o národním hospodářství, podnicích, podnikání a podnikatelském okolí. Většina pojmů pro Vás bude pravděpodobně nová. Doporučujeme projít tuto kapitolu důkladně, jelikož z ní ostatní kapitoly čerpají. Na studium této části si vyhraďte alespoň 5 hodin.

Ekonomické informační systémy (dále IS) jsou nástrojem pohledu na ekonomiku podniku. Z důvodu jasnějšího popisu jednotlivých funkcí ekonomického IS je nutné pochopit pojmy z oblasti ekonomiky podniků.

11.1 Základní pojmy

Co to je ekonomie?

Ekonomii definujeme jako společenskou vědu, která popisuje a analyzuje, jak lidstvo řeší fundamentální problém vzácnosti. *Potřeby každé společnosti přesahují možnosti disponibilních zdrojů (výrobních faktorů), musí proto existovat mechanismus, který tyto zdroje alokuje mezi vzájemně si konkurující užití.*

Národní hospodářství

Uchovávaní a rozvoj života lidí vyvolává nutnost uspokojovat jejich rozmanité a měnící se potřeby, které vytvářejí ve společnosti poptávku po *statcích a službách*. Při vytváření a poskytování statků a služeb se uplatňuje dělba práce, specializace a kooperace. Základní význam mají hmotné statky. Nejprve se vyrábějí, potom se rozdělují, směňují a nakonec se spotřebují buď k další výrobě nebo k činnostem, jejichž výsledkem jsou *nehmotné statky a služby*. Tyto pak dále slouží k uspokojování lidských potřeb. Souhrn uvedených činností tvoří *hospodářský proces*, a oblasti, kde se uskutečňuje, označujeme



Informační systémy 1

jako *ekonomiku – hospodářství*. Hospodářský proces, který probíhá na území státu, označujeme jako *národní hospodářství*. Rozumíme jím tedy určitou soustavu. Národní hospodářství je řízeno na základě zákonů. Věda, která zkoumá jak ovlivňovat procesy v národním hospodářství, aby bylo efektivní, se nazývá makroekonomie. Základními subjekty národního hospodářství jsou stát, spotřebitelské domácnosti a podniky.

Ekonomický systém společnosti

Základní problém každého hospodářství spočívá v tom, jak společnost nakládá s omezenými zdroji, aby uspokojila neomezená lidská přání. To předpokládá určitý ekonomický systém, tj. nalezení způsobu“

Co se bude produkovat?

Jak se to bude produkovat?

Kdo to dostane?

Podle odpovědi na tyto otázky rozeznáváme tři základní ekonomické systémy:

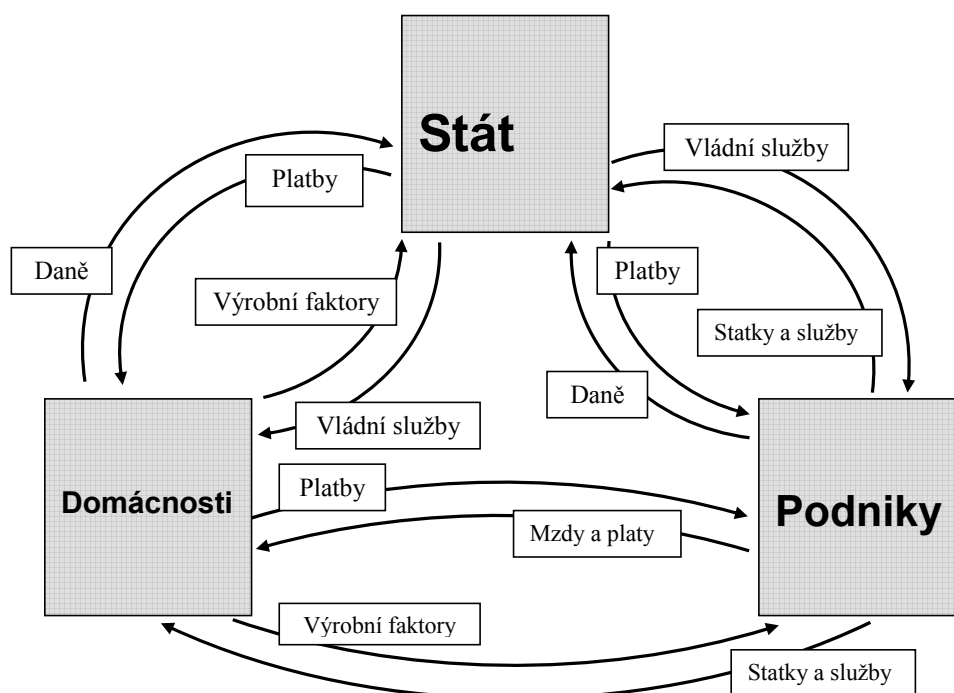
- *tradiční ekonomický systém,*
- *plánovitý ekonomický systém (centrálně plánovanou ekonomiku),*
- *tržní ekonomický systém.*

Struktura národního hospodářství

Strukturu národního hospodářství sledujeme podle druhu vytvářených statků a služeb. Z tohoto hlediska ji členíme do čtyř sektorů - primárního, sekundárního, terciárního a nově se hovoří o kvartérním sektoru. Jednotlivé sektory se pak dále dělí na odvětví.

Koloběh národního hospodářství

Tento koloběh probíhá mezi jednotlivými subjekty národního hospodářství a tak vznikají základní vztahy mezi těmito subjekty.



Obr. 11.1 Základní vztahy mezi subjekty

Informační systémy 1

Základní vztahy mezi těmito subjekty můžeme charakterizovat takto:

- Vztah mezi spotřebitelskou domácností a státem
 - o *Spotřebitelská domácnost (SD)* – obyvatelstvo poskytuje státu nejrůznější statky a služby (pracuje ve státních službách) – za to stát poskytuje SD platby (mzdy státních zaměstnanců).
 - o *Stát* – poskytuje SD také nejrůznější statky a služby (ochrana, zákonodárství, zdravotní a sociální péče atd.) a tyto mu za to platí formou daní.
- Vztah mezi spotřebitelskou domácností a podniky
 - o *Spotřebitelská domácnost (SD)* – obyvatelstvo poskytuje podnikům nejrůznější statky a služby (pracuje v podnicích, případně do nich vkládá majetek) a za to jim podniky platí mzdy, dividendy apod.
 - o *Podniky* dodávají SD statky a služby a tyto jim za ně platí.
- Vztah mezi podniky a státem
 - o *Podniky* zajišťují pro stát nejrůznější statky a služby (státní zakázky) a stát jim za to platí.
 - o *Stát* – zajišťuje prostřednictvím svých statků a služeb (ochrana, zákonodárství, zdravotní a sociální péče atd.) podnikům jejich existenci a tyto mu za to platí formou daní.

V rámci ekonomie se zkoumáním a definováním těchto vztahů zabývá:

- makroekonomie,
- mikroekonomie,
- podniková ekonomika.

Makroekonomie – zkoumá chování ekonomiky (národního hospodářství) jako celku, jeho agregátní veličiny jako např. úroveň produkce, cenovou úroveň, velikost hrubého národního produktu, úroveň zaměstnanosti, míru inflace, měnové kurzy, stabilitu měny apod.

Mikroekonomie – zkoumá chování jednotlivých tržních subjektů, tj. podnikatelů – výrobců, spotřebitelů – odběratelů, firem, domácností, bank, státu, zkoumá individuální trhy.

Podniková ekonomika – zaměřuje se na ekonomické postavení podniku v rámci ekonomického systému, zkoumá jeho efektivnosti, finanční zdraví a postavení podniku ve společnosti.

Základní poznatky o podniku a podnikání a podnikateli.

Národní hospodářství (NH) je velmi složitý celek. Aby ekonomický systém řádně fungoval musí být stanovena pravidla pro vznik a existenci základních organizačních jednotek. Rozhodnutí o uvedených otázkách uskutečňují nejvyšší zákonodárné a výkonné orgány státu, mají povahu zákonů a vyhlášek, které musí být dodržovány – tvoří základní rámec pro podnikání.

Charakteristika podniku

Pro vytváření statků a služeb jsou ve společnosti vytvářeny organizace jako instituce, z nichž největší význam mají podniky.

Informační systémy 1

Podnikem rozumíme právní celek vytvořený předepsaným způsobem z lidí, hmotného a nehmotného majetku za účelem plnění určitých úkolů v rámci NH.

Má způsobilost zakládat, měnit nebo rušit vztahy k jiným organizacím a obyvatelstvu. Při své činnosti má právní subjektivitu tzn., že jedná v rámci zákonných předpisů samostatně, vlastním jménem. O svém hospodaření je povinen vést účetnictví.

Princip hospodaření podniku spočívá v tom, že své náklady (spotřebované prostředky a práci vynaložené na produkty) uhrazuje z výnosů (příjmů za vyrobené produkty). Produktem rozumíme výrobek, službu nebo zboží. Výnosy jsou obvykle větší než náklady. Rozdíl představuje hospodářský výsledek (**zisk**, v opačném případě **ztrátu**), který po odvedených daních státu podnik používá k další produkci, k investicím do podniku a jako odměnu podnikateli za podstoupené riziko. Snahou podniku je obstát v konkurenci, tj. v soutěži s ostatními podniky. Říkáme, že podnik hospodáří na vlastní účet a je tudíž organizací, která je schopna sama sebe obnovovat a rozšiřovat.

Kromě podniků jsou vytvářeny i další skupiny organizací, především státní, ale existují i jiné jako jsou například obecně prospěšné organizace či různá občanská sdružení. Ty však zpravidla působí mimo oblast ekonomiky, tj. zabývají se např. státní správou, zajišťováním obrany a bezpečností, vzděláváním, zdravotní péčí atd.

Typologie podniků

Bližší zkoumání, popis a pochopení podniku vyžaduje jeho podrobnou strukturalizaci. Tuto je možno provést uspořádáním podniků do skupin a podskupin podle jejich společných znaků. Základní třídění podniků se provádí podle nejrůznějších kritérií. Typologie podniků je důležitá pro pochopení odlišností a specifík podniků na straně jedné, ale podobnosti v určitých základních charakteristikách na straně druhé.

Typologie podniků podle různých charakteristik (hledisek):

- národohospodářského,
- vlastnictví,
- zisku,
- právní formy,
- způsobu řízení,
- rozsahu působnosti,
- velikosti,
- podle předmětu činnosti.

Specifická hlediska:

- dle účelu či funkce vůči společnosti (deskriptivní typologie),
- dle prvotního uživatele (analytická typologie).

Dle **národohospodářského** hlediska členíme podniky do sektorů NH:

- **Primární:** zemědělství a lesní hospodářství, rybolov, těžba nerostných surovin
- **Sekundární:** zpracovatelský průmysl, výroba rozvod energie, stavebnictví
- **Terciární:** Služby např. obchod, pohostinství a hotelnictví, doprava, zdravotnictví, školství, bankovníctví atd.

Informační systémy 1

- **Kvartérní:** softwarové a informační technologie

Dle hlediska **vlastnictví** máme podniky:

- soukromé
- státní
- družstevní

Dle hlediska **zisku:**

- ziskové
- neziskové

Dle hlediska **právní formy:**

- podnik jednotlivce
- státní podnik
- obchodní společnosti
- družstvo
- ostatní

Dle hlediska **rozsahu působnosti:**

- obecní
- regionální
- republikové
- nadnárodní

Rozdělení podniků dle hlediska **velikosti:**

(přičemž jako hledisko velikosti můžeme brát: počet zaměstnanců, výši dosahovaného zisku, velikost obrátu, velikost majetku atd..)

- malé
- střední
- velké

Dle **předmětu** činnosti:

- výrobní podniky
- obchodní podniky
- dopravní podniky
- podniky služeb
- atd.

Podnikání

Podnikání je u nás definováno v Obchodním zákoníku (zákon č. 531/1991Sb.) jako:

„soustavná činnost prováděná samostatně podnikatelem vlastním jménem a na vlastní odpovědnost za účelem dosažení zisku“.

Podnikatel vyrábí pro trh nebo prodává na trhu, a to vyžaduje co nejlepší informace o něm. Musí se snažit poznat **poptávku**, pro kterou své **produkty** (výrobky, služby, či jiné výkony) dodává. Na trhu se ale nevyskytuje sám, je tam celá řada dalších podnikatelských subjektů (**konkurentů**), kteří se snaží své produkty uplatnit také. Tyto subjekty tvoří **nabídku**. Znalost situace na trhu, schopnost odhadů vlastních i konkurence, patří mezi hlavní úkoly každého podnikatele.

Podnikáním se spojují výrobní faktory (věcní a lidské činitele) pracovního procesu k vytváření produktu, který se prodá na trhu a realizuje se **zisk**. Snaha dosáhnout zisk vede podnikatele k podnikavosti (tj. vynalézavosti, využívání příležitostí, brát na sebe riziko atd.).

Informační systémy 1

Podnikání je charakterizováno několika podstatnými rysy:

- snahou po dosažení zisku,
- zisk se docílí uspokojením zákazníků,
- potřeby zákazníků uspokojuje podnikatel svými produkty na trhu, kde musí čelit riziku,
- podnikatel vkládá do podniku svůj nebo cizí kapitál a tento se snaží zhodnotit.

Podnikatel, manažer.

Podnikatel je osoba, která podnik zakládá a provozuje jeho činnost, tzn. spojuje práci, kapitál a další výrobní činitele za účelem dosažení prosperity, tedy zisku. Podnikatelem mohou být fyzické osoby (každý jednotlivý občan), právnické osoby (skupina lidí, jiný podnik nebo stát). Zákon rozlišuje podnikatele zapsané v obchodním rejstříku a ostatní.

Právnické osoby podléhají povinnosti zápisu do obchodního rejstříku. Fyzické osoby, které nejsou zapsány v obchodním rejstříku, nabývají postavení podnikatele udělením živnostenského oprávnění. Všeobecné podmínky provozování živnosti přesně upravuje platný Živnostenský zákon.

11.2 Založení podniku

Způsob podnikání je v České republice vymezen zákonem, jedná se o **Živnostenský a Obchodní zákoník**. Provozuje-li někdo podnikání v rozporu se zákonem jedná se neoprávněné podnikání.

Jakou cestu k **prosperitě** si podnikatel zvolí silně ovlivňuje celková hospodářská úroveň společnosti. K prosperitě však vede dlouhá cesta, vyplněná pílí a každodenní aktivitou všech zaměstnanců podniku. **Matematicky lze pro vyjádření výše zisku** použít následující výpočet:

Hrubý zisk = příjmy - výdaje nebo **výnosy - náklady**

Čistý zisk = hrubý zisk - daně a tvorba povinných fondů - **odpočitatelné položky**

Mezi odpočitatelnými položkami najdeme tzv. příspěvky na rozvoj určitých oblastí lidských činností - **sponzorování**. Má-li podnik co nejlépe prosperovat, je důležité, aby jeho činnosti byly **hospodárné a efektivní**. Pod těmito pojmy se skrývá využívání všech hmotných i nehmotných zdrojů podnikatele.

Efektivnost podnikatelské činnosti zjistíme pomocí **ukazatelů rentability** (poměření zisku Z s různými veličinami).

Snahou podnikatelů musí být **dobré jméno podniku (goodwill)**. Od samého začátku musí být uplatňovány **etické principy, sociální a ekologická hlediska**. V současné době jsme svědky často neetického chování různých podnikatelů, ti považují etiku za překážku úsilí o dosahování zisku. V solidním podnikatelském prostředí je takový postup považovaný za sebezničující.

Informační systémy 1

Činnosti související se založením podniku

Mezi klíčová rozhodnutí patří:

- Rozhodnutí o předmětu činnosti.
- Rozhodnutí finančně ekonomické (stanovení velikosti potřeby finančních prostředků a jejich eventuální možnosti získání).
- Rozhodnutí o právní formě podnikání.
- Rozhodnutí o umístění podniku.

Rozhodnutí o předmětu činnosti a posouzení jeho vyhlídek je klíčovým rozhodnutím. Podnikatel vychází z vlastních možností, zkušeností a znalostí. Celou řadu užitečných informací a podnětů lze získat od:

- spotřebitele (zákazníka),
- obchodních partnerů,
- členů distribuční sítě,
- technicky nadaných lidí, kteří často neumí obchodně uplatnit své nápady,
- sledování vývoje oborů,
- například na úrovni místa bydliště.

Jsou to informace o investičních záměrech města nebo obce (do dopravy, do bydlení, do kultury apod.), vymezených podnikatelských zónách, o současné technické a dopravní infrastruktuře, o existující občanské vybavenosti, o počtu obyvatel a jeho struktuře, pracovní síle aj.

Obdobné údaje lze získat na úrovni regionů na krajském úřadě, na pracovním úřadě, v regionálním poradenském a informačním centru, na různých komorách, apod. Ale lze se zaměřit i na úroveň celostátní ba i mezinárodní, neboť existuje celá řada úřadů a institucí, které disponují potřebnými informacemi.

Různé prameny informací pak mohou inspirovat budoucí, ale i již podnikající podnikatele k nejrůznějším **nápadům**. Každý nápad je však třeba zhodnotit z hlediska jeho další životaschopnosti. Jde zejména o zjištění :

- potencionálních zákazníků,
- možnosti rozšíření trhů (export, celní bariéry),
- konkurence a jejím postavení na trhu,
- dodavatelů a jejich možností.

Rozhodnutí o finančním zabezpečení vychází z kvalitně zpracovaného podnikatelského plánu. V rámci tohoto rozhodnutí podnikatel posuzuje finanční sílu zakládaného podniku a jeho schopnost čelit případné ztrátě. Musí být zpracován plán nákladů, výnosů a hospodářského výsledku, jakož i majetku a zdrojů jeho krytí. Důležité je rovněž uvedení způsobu, jak zajistit počáteční finanční prostředky (vloží majitelé, úvěrem, úpisem akcií) na jedné straně a jejich případné splácení na straně druhé.

Rozhodování o právní formě podniku patří k rozhodnutím, které dlouhodobě ovlivňují činnost podniku. Toto rozhodnutí vyvstává nejen při založení podniku, ale vždy při změně osobních, kapitálových, právních či daňových podmínek. Při řešení tohoto problému by měly být vyjasněny následující otázky:

Informační systémy 1

- právní prostor jednotlivých variant,
- prostor pro řízení podniku (možnost vystupování jménem podniku, řízení podniku, spolu rozhodovací pravomoc),
- ručení - spolupodílnictví na zisku a ztrátě,
- možnosti financování vlastním a cizím kapitálem,
- flexibilita vztahů mezi podílíky, přijímání nových a vyloučení stávajících podílníků či společníků,
- daňové zatížení,
- zákonné předpisy pro rozsah, obsah, přezkoušení a zveřejňování roční uzávěrky,
- náklady určité právní formy (zakládací poplatky, registrační poplatky, minimální výše základního jmění atd.).

Všechny tyto faktory je třeba zvážit, přičemž vznikají nemalé potíže, protože ne všechny jsou kvantifikovatelné a některé jsou protikladné, tzn. výhody té které formy musí převýšit její nevýhody.

Rozhodnutí o umístění podniku. Umístění podniku může podstatně ovlivnit úspěšnost podnikání. Dle charakteru činnosti můžeme rozlišit tyto typy podnikání ve vazbě na volbu umístění:

- materiálově orientované podnikání,
- pracovně orientované podnikání,
- energeticky orientované podnikání,
- dopravně orientované podnikání,
- odbytně orientované podnikání,
- poskytování služeb.

Je žádoucí ve vazbě na výše uvedený charakter podnikání určit faktory, které je třeba zohlednit při výběru stanoviště, tyto faktory ocenit z hlediska důležitosti (body dle významu) a ocenit jednotlivá místa.

Příklad faktorů, které může podnikatel použít:

- současný stav a vývoj koupěschopné poptávky v oblasti,
- blízkost místa k surovinovým zdrojům, zásobování energií, vodou,
- blízkost místa k odbytu,
- blízkost místa ke zdroji pracovních sil, průměrná mzda v regionu, kvalifikace lidí,
- reputace místa, cena místa,
- dopravní spojení, možnost parkování, nakládky a vykládky,
- možnost rozšíření podnikání,
- existence konkurence,
- otázky životního prostředí, hygienické normy a předpisy.

Živnostenské podnikání

Podmínky živnostenského podnikání a kontrolu nad jejich dodržováním upravuje **Zákon č. 455/1991 Sb., o živnostenském podnikání** v novelizovaném znění. Tento zákon vymezuje pojem **živnosti** a charakterizuje druhy živností

Druhy živností podle předpokladů vzniku živnostenského oprávnění:

Ohlašovací živnost, která se člení na:

- živnosti řemeslné,
- živnosti vázané,
- volné živnosti.

Informační systémy 1

Koncesovaná živnost, která se provozuje jen na základě státního povolení – koncese.

Druhy živností podle předmětu činnosti:

- živnosti obchodní,
- živnosti výrobní,
- živnosti poskytující služby.

Všeobecné a zvláštní podmínky

Podnikat mohou jen osoby, které splňují všeobecné a zvláštní podmínky pro provozování živnosti. Všeobecné podmínky pro provozování živnosti musí splnit podnikatel u všech druhů živností.

Ke všeobecným podmínkám patří:

- věk; ke dni vzniku živnostenského oprávnění musí fyzická osoba dosáhnout věku 18 let,
- plná způsobilost k právním úkonům,
- bezúhonnost,
- potvrzení finančního úřadu, že občan nemá žádné nedoplatky na daních či jiné pohledávky.

Mezi zvláštní podmínky provozování živnosti patří odborná a jiná způsobilost, pokud ji živnostenský nebo jiný zákon vyžaduje.

Živnostenské oprávnění

Živnostenské oprávnění je právo fyzických i právnických osob provozovat živnost, vzniklé splněním požadavků živnostenského zákona.

Vznik živnostenského oprávnění

U ohlašovacích živností vzniká živnostenské oprávnění dnem ohlášení živnosti nebo dnem pozdějším, pokud je v ohlášení uveden; u koncesovaných živností dnem doručení koncesní listiny. Jde-li však o zahraniční osobu nebo o právnickou osobu, která vzniká až dnem zápisu do obchodního rejstříku. Živnostenské oprávnění může být vykonáváno na celém území ČR; pouze u koncesovaných živností může živnostenský úřad omezit územní rozsah oprávnění.

Prostor, v němž je živnost provozována, se označuje jako provozovna.

Jeden podnikatel může provozovat více živností, pokud má pro každou z nich živnostenské oprávnění.

Zánik živnostenského oprávnění

Živnostenské oprávnění zaniká:

- smrtí podnikatele,
- zánikem právnické osoby,
- uplynutím doby, na kterou byly živnostenský list nebo koncesní listina vydány,
- rozhodnutím živnostenského úřadu.

Obchodní společnosti

Tento způsob podnikání definuje Zákon č. 513/1991 Sb., zvaný Obchodní zákoník. Citovaný zákon vymezuje:

Informační systémy 1

- pojem a obecnou charakteristiku obchodních společností,
- právní formy obchodních společností a družstvo,
- obchodní závazkové vztahy.

Obchodní společnosti se dělí podle rozsahu ručení společníků za závazky společnosti na osobní, smíšené a kapitálové. Osobní společností je v tomto smyslu veřejná obchodní společnost, kde společníci ručí za závazky společnosti celým svým majetkem; smíšenou společností je společnost komanditní, kde někteří společníci ručí osobně, jiní jen svými majetkovými vklady. Kapitálovými společnostmi jsou společnost s ručením omezeným a společnost akciová, kde existuje úplné majetkové oddělení jmění společnosti od jmění jejích kapitálových účastníků (společníků).

Právní formy obchodních společností:

- veřejnou obchodní společnost,
- komanditní společnost,
- společnost s rušením omezeným
- akciovou společnost.

Obchodními společnostmi nejsou družstva a státní podniky.

Základním dokumentem pro vznik společnosti je společenská smlouva nebo zakladatelská listina. Všechny obchodní společnosti se zapisují do obchodního rejstříku a vznikají až dnem, ke kterému jsou tam zapsány (nabývají právní subjektivitu).

Obchodní společnost má vlastnické právo k věcem, které patří do jejího obchodního jmění, jímž nazýváme souhrn hodnot ocenitelných penězi a závazků, které náleží jednomu podnikateli. Zvláštní význam má u obchodních společností tzv. základní kapitál.

Vkladem se rozumí souhrn peněžních prostředků (peněžitý vklad) nebo jiných penězi ocenitelných hodnot (nepeněžitý vklad). Od vkladu se odlišuje podíl, jenž je mírou účasti společníka. Zániku společnosti předchází její zrušení. Společnost zaniká ke dni výmazu z obchodního rejstříku. Společníci nebo orgány společnosti se mohou usnést na zrušení společnosti s likvidací nebo bez likvidace (sloučení, splynutí, rozdělení, přeměna na jinou právní formu obchodní společnosti nebo na družstvo).

Hlediska nutná pro volbu právní formy podnikání:

- počet zakladatelů,
- způsob a rozsah ručení,
- nároky na počáteční kapitál,
- administrativní náročnost při založení podniku a v průběhu podnikání (způsob vedení účetnictví),
- účast na zisku (ztrátě),
- daňové zatížení,
- zveřejňovací povinnost,
- finanční možnosti.

11.3 Podnikatelské prostředí

Vymezení podnikatelského prostředí

Podnik není izolován, ale je obklopen vnějším světem, okolím. Okolím podniku rozumíme vše, co je za pomyslnými hranicemi podniku jako sociálně ekonomického a technického systému a čím je podnik ovlivňován a co případně ovlivnit nemůže. Vliv okolí na podnik je zpravidla velmi silný, zatímco vliv podniku na okolí je spíše malý.

K pochopení podnikatelského prostředí je vhodné se seznámit s pojmem **okolí podniku**. Podnikatelské prostředí definujeme jako souhrn podstatných vlivů působících na podnikatele, podnik a podnikání. Vedle ekonomických a přírodních složek podnikatelského prostředí na významu nabývají složky nehmotného prostředí, jako je kultura (ekonomická i všeobecná), právo, sociální vztahy, vědecké poznatky atd.

Je zřejmé, že podnikatel může být ovlivňován zcela jiným prostředím než podnik, který vlastní. Dokonce rozhodující podnikatelské aktivity, např. investiční akce v zahraničí, se mohou nacházet v dalším odlišném podnikatelském prostředí.

Rozvinuté podnikatelské prostředí se vyznačuje vlastní identitou, vlastní filozofií života, vlastními zájmy, vlastní aktivitou, vývojovými fázemi apod. Vztah kupř. podnik – podnikatelské prostředí si můžeme představit jako dialog, tzn. jako harmonizaci zájmů komunikujících subjektů, konkrétně třeba ředitele podniku a starosty obce. Každý z těchto nositelů reprezentuje určité zájmy, vstupuje do vzájemných vztahů a sehrává určitou roli. Nerespektování vlivu podnikatelského prostředí na podnik, neznalost faktorů, které na podnik působí vede k chybným manažerským rozhodnutím a tím ke krizi podniku a v důsledku toho může dojít i k zániku podniku.

Typologie podnikatelského prostředí

Reálné podnikatelské prostředí je velmi pestré a strukturované. Má svou věcnou, časovou, prostorovou, efektivnostní (účinnostní), účelnostní dimenzi. Můžeme jej modelově popsat pomocí faktorů, které se v podnikatelském prostředí uplatňují. Podle toho, kde se dané faktory nacházejí – zda uvnitř nebo vně daného podnikatelského subjektu, je účelné podnikatelské prostředí rozčlenit na **vnitřní**, specifické pro daný podnikatelský subjekt a **vnější**.

Za nejdůležitější prvky tvořící vnější (externí) podnikatelské prostředí považujeme okolí:

- geografické,
- politické a právní,
- ekonomické,
- ekologické,
- technologické,
- etické,
- kulturně historické.

Informační systémy 1

Tyto prvky pak může charakterizovat pomocí :

- růstu nebo poklesu výkonu ekonomiky, vyjádřený v podobě hrubého domácího produktu, růst nebo pokles inflace, a tedy i vlivy na cenu zboží, služeb, peněz apod.,
- růst nebo pokles zaměstnanosti, a s tím související sociální aspekty,
- udržení nebo narušení vyrovnanosti státního rozpočtu (případně obchodní bilance se zahraničím),
- upevňování nebo narušení politické stability a dalších mimoekonomických parametrů země, skupiny zemí atd. (kupř. vůle a ochota obyvatel ke vstupu do integračního seskupení, mezinárodní pověst apod.),
- přírodně ekologické faktory, které se projevují při interakci podniku, resp. podnikatele s přírodním prostředím (hornatost krajiny, vodní plochy, lesy, přírodní zvláštnosti apod.),
- technické a dopravní infrastruktury, ve které se hodnotí stav technické a dopravní vybavenosti příslušného prostoru s ohledem na podnikatelské aktivity a která má podstatný vliv na rozhodnutí o umístění firmy v dané lokalitě,
- všeobecné a ekonomická kultury - vzdělanost, kulturnost, pracovitost obyvatelstva ve vymezeném území s ohledem na potřeby rozvoje podnikání,
- ekonomická a podnikatelská infrastruktura, což je např. rozvinutost služeb pošty, bank, poradenských firem, hotelů, agentur atd.,
- pilotní (vůdčí) podnikatelské (resp. jiné) subjekty, které za sebou strhávají ostatní subjekty a které jsou schopny se propojovat a vytvářet sítě nebo jinak výrazně ovlivňují podnikatelské prostředí.

O tyto prvky a jejich konkrétní parametry se zajímají jak velké, tak malé a střední firmy. Praktická využitelnost je však odlišná. Velké (globální) podnikatelské subjekty se zpravidla zajímají o parametry velkých regionů, seskupení zemí apod. (kupř. velmi výrazným parametrem pro aktivity velkých firem je politická stabilita země). Naopak malé firmy se soustřeďují pouze na vybrané prvky okolí té země, kde podnikají (např. jde o zaměstnanost, inflaci, vývoj výkonnosti oboru, ve kterém firma podniká ap.).

Vnitřní prostředí

Vnitřní prostředí vymezujeme klimatem, kulturou a dalšími znaky, které vytvářejí pro jednotlivé subjekty v podniku pracovní, provozní, resp. životní prostředí. Za rozhodující faktory tohoto prostředí považujeme:

- umění podnikatele (vlastníka) vlastnit, tzn. optimalizovat, resp. harmonizovat vlastnické podnikatelské portfolio,
- schopnost a umění vést lidi, což znamená rozvíjet vztahy mezi vlastníky a managementem, mezi managementem a podřízenými ap.,
- schopnost uspokojit potřeby, tedy prolnout marketingovou filozofii dovnitř i navenek firmy,
- schopnost komunikovat s vnějším podnikatelským prostředím včetně mimoekonomických aktivit (kupř. různá forma spoluúčasti, dary, soutěže ap.),
- schopnost formulovat podnikatelskou filozofii, zajistit identitu a integritu firmy, tzn. schopnost a umění sjednotit zájmy, představy,

Informační systémy 1

hodnoty a přístupy vlastníků, managementu a ostatních zaměstnanců, popř. subjektů v okolí na principu participace).

Vnitřní prostředí v podstatě vypovídá o vyzrálosti „osobnosti“ (identitě a integritě) podniku (firmy). Velké firmy zpravidla těmto otázkám věnují značnou a stálou pozornost, malé a střední firmy se systematictěji této problematice věnují občas.

Nositelé podnikatelského prostředí

Praktický význam pro formování a rozvoj podnikatelského prostředí má určení jeho klíčových prvků (nositelů, reprezentantů). Jako základní nositelé podnikatelského prostředí vystupují:

- státní centrum reprezentované parlamentem, vládou, ministerstvy apod.,
- centra nadstátních útvarů a globální správy světa jako jsou kupř. OSN, Evropská rada aj.,
- municipality, tzn. obce, vyšší územně samosprávné celky (kraje) ap.,
- obyvatelstvo chápané v roli spotřebitelů, zaměstnanců, voličů, vlastníků ap.,
- reprezentanti sféry podnikatelských či jiných subjektů coby odběratelů, dodavatelů, škol, nemocnic, bank, grantových agentur, rozvojových agentur ap.,
- konkurenční subjekty,
- vědeckovýzkumná centra,
- reziduální sféra coby označení pro ostatní reprezentanty specifického podnikatelského prostředí s vazbou na kulturní, náboženské, politické vlivy, ekologické aspekty ap. (např. etnické organizace, ekologické aktivity, lobby, media apod.).

Každý z těchto nositelů reprezentuje určité zájmy, vstupuje do vzájemných vztahů a sehrává určitou roli. Podstatou formování a rozvoje podnikatelského prostředí je iniciování a rozvíjení dialogu klíčových subjektů.

Význam studia podnikatelského prostředí

V minulosti byly vztahy podniku k podnikatelskému prostředí jednoduché a hlavně velmi stabilní. V současné době hovoříme o turbulentním podnikatelském prostředí. V důsledku vědeckotechnického rozvoje zejména pak telekomunikací, informatiky a globalizace ekonomiky, se okolím podniku stává celý svět. Proměnlivost současného světa výraznou mírou zasahuje do všech oblastí podnikání. S těmito změnami musí podnikatelé a manažeři počítat jak při tvorbě podnikatelských strategií, tak při každém manažerském rozhodování nemají-li být zaskočeni změnami, ale naopak využít všechny příležitosti, které jim prostředí nabízí. Informační systém podniku by měl obsahovat všechny potřebné informace, které manažeři ke svému rozhodování potřebují. Existuje řada zdrojů a technik kde a jak informace získávat.

Strategie podniku

Obecně pojem **strategie** podniku představuje formulaci základních rozvojových procesů. Strategie je zaměření a rozsah činností podniku

Informační systémy 1

v dlouhodobém horizontu, které v ideálním případě vytvářejí soulad mezi podnikovými zdroji a měnícím se vnějším prostředím. Strategie je východiskem pro tvorbu všech podnikových plánů. Strategie zahrnuje to, co podnik hodlá v budoucnu udělat a současně to, co musí udělat, aby zajistil rozvoj svého majetku a byl likvidní.

Strategii podniku můžeme tedy chápat ve třech směrech:

- stanovení základních rozvojových procesů podniku,
- nástroj dlouhodobého vytváření potenciálu podniku,
- základ běžného řízení všech operací.

Cíle formulace strategie lze vymezit následovně:

- dlouhodobé využívání a získávání trhů,
- vytváření a zajišťování likvidity,.

Každá strategie zahrnuje:

- **strategické cíle** (jsou to stavy, kterých má být v průběhu nebo na konci období dosaženo).
- **strategické operace** (jsou to činnosti, jimiž má být dosaženo strategických cílů).

Strategie vychází z:

- **poslání,**
- **vize.**

Poslání je soustředěné na současnost. Vymezuje prostor pro strategické úvahy. Poslání vyjadřuje představy vedení podniku o tom, jaký je současný, ale také budoucí smysl jeho existence. Odpovídá na otázky typu:

- Proč podnik existuje?
- Kdo jsme?
- Co děláme?
- Kam směřujeme?“

Zformulovat účinné poslání znamená splnit tři následující úkoly:

- definovat v jakém podnikání se podnik skutečně nachází,
- rozhodnout, kdy se má poslání změnit a tak upravit strategické zaměření podniku,
- seznámit pracovníky podniku s posláním pomocí srozumitelných a podnětných metod.

Vize naznačuje v základních koncepčních pojmech představu o budoucím směru a postavení podniku. Zaznamenává jevy a trendy, které jsou v současnosti nejasné a nevýrazné. Vize je cílová komplexní představa budoucnosti podniku.

Zpracování strategie a proces strategického řízení

Podnik zpracovává strategii vždy pro určité strategické období, které bývá určeno např. charakterem výrobků, množstvím kapitálu, kvalifikací pracovníků, kvalitou a druhem surovin. Délka strategického období se odvíjí od stability okolí podniku, protože pokud nastávají negativní změny v okolí podniku, délka strategického období se zkracuje a naopak. Pro zpracování strategie je důležité osobní zapojení vrcholového vedení podniku při

Informační systémy 1

zpracování, případně účast kvalifikované poradenské organizace. Základním požadavkem je, aby materiál byl formulován jasně, srozumitelně a mohl poskytnout informace širokému okolí či kolektivu pracovníků. Je v zájmu podniku podpořit zájem pracovníků o rozvoj podniku.

Strategie podniku je základem celého systému řízení podniku. Podílejí se na ní všechny články řízení podniku a všechny úrovně řízení.

Typy strategií s bližší charakteristikou

Progresivní integrace (Forward Integration) - Získávání většího podílu na řízení maloobchodníků a distributorů vlastních výrobků nebo jejich skupování.

Zpětná integrace (Backward Integration) - Získávání většího podílu na řízení firemních dodavatelů nebo jejich skupování.

Horizontální integrace (Horizontal Integration) - Získávání podílu na řízení firem konkurentů, spojování se s nimi nebo jejich skupování.

Proniknutí na trh (Market Penetration) - Zvýšení podílu současných výrobků firmy na jejich současných trzích pomocí zvýšeného marketingového úsilí.

Rozvoj trhu (Market Development) - Představování současných výrobků na geograficky nových trzích.

Vývoj výrobku (Product Development) - Snaha zvýšit prodej zlepšením nebo modifikací současných výrobků či služeb.

Soustředná diverzifikace (Concentric Diversification) - Přidávání nových výrobků a služeb, které se vztahují k dosavadní hlavní činnosti podniku.

Směšená diverzifikace (Conglomerate Diversification) - Přidávání nových výrobků a služeb, které se nevztahují k dosavadní hlavní činnosti podniku.

Horizontální diverzifikace (Horizontal Diversification) - Přidání nových výrobků a služeb, které se k dosavadní činnosti podniku nevztahují, jsou ale zamýšleny pro prodej současným zákazníkům této firmy.

Společné podnikání (Joint venture) - Dvě nebo více firem vytvoří právní celek, firmu pro účely vzájemné spolupráce.

Snížení výdajů (Retrenchment) - Přehodnocování nákladů a redukce majetku za účelem zrušení výroby u produktů s klesajícím prodejem.

Zbavování se majetku (Divestiture) - Prodej divize nebo části organizace.

Likvidace (Liquidation) - Prodej veškerého majetku podniku za jeho skutečnou cenu.

Základním nástrojem řízení je plán. Tvorba tohoto plánu musí vycházet z definované strategie podniku. Tato strategie je formulována pomocí dlouhodobých plánů a jejich konkretizaci představují střednědobé a krátkodobé

Informační systémy 1

plány. Teoretické základy strategického řízení a plánování vychází z předpokladu, že všechny veličiny jednotlivých plánů se vzájemně ovlivňují, a že v praxi působí celá řada nejistých a nepředvídatelných činitelů. Plánování v podniku můžeme chápat jako proces rozvoje podnikových cílů a funkčních oblastí (marketing, odbyt, výroba, ...). Celý proces podnikového plánování je založen na posloupnosti jednotlivých kroků, které za sebou následují v určitém předem stanoveném pořadí:

- formulace cílů a vizí,
- analýza podnikatelského prostředí a určení jeho vývoje,
- výběr činností pro zajištění cílů,
- rozpracování a vypracování plánů,
- kontrola plnění plánů.

Při formulování strategie a jejím následném rozpracování je nesmírně důležité definovat cíle tak, aby podnik udržel finanční rovnováhu.

11.4 Podnikatelský záměr

Podnikatelský záměr je strategický materiál, neboli projekt, který komplexně reprezentuje podnik pomocí čísel a slov. Cílem podnikatelského záměru je:

- definovat základní rozvojové trendy a dynamizující procesy v podniku,
- analyzovat výchozí stav,
- ujasnit základní postup dalšího rozvoje,
- ukázat, že vymezených cílů může být dosaženo,
- dokázat, že výsledky uspokojí předpokládané potřeby společnosti,
- přesvědčit akcionáře, potenciální investory, podílníky či spoluvlastníky aj., jimž je podnikatelský záměr určen, že výsledky dané společnosti budou zajímavé i z jejich hlediska.

Úkolem jeho zpracovatelů je provést analýzu minulosti, současnosti a předpokládané budoucnosti. Podnikatelský záměr je svým způsobem propagační materiál, který seznamuje zájemce s firmou a s jejím vedením. Proto se doporučuje přitažlivé textové zpracování a odpovídající forma celého materiálu. Podnikatelský záměr není jen příležitostný materiál. Měl by být průběžně upravován a aktualizován tak, aby analýza případných odchylek od tohoto záměru umožnila přesnější tvorbu jeho nové varianty.

Důvody pro zpracování podnikatelského záměru:

- Podnikatelský záměr k získání příspěvku či subvence.
- Podnikatelský záměr k získání úvěru.
- Podnikatelský záměr k získání půjčky či kapitálové účasti.
- Podnikatelský záměr pro vnitřní informaci.

Zpracování podnikatelského záměru pro vnitřní potřebu vyplývá ze dvou důvodů:

- z jeho významu pro řízení společnosti, pro mobilizaci jejích vnitřních rezerv,
- z potřeby informovat, zda řízení probíhá efektivně a jak použít finanční prostředky na rozvoj výroby či technologie, výrobku nebo výrobního oboru, na rozvoj perspektivního výzkumu a vývoje.

Informační systémy 1

Struktura podnikatelského záměru

Strukturu podnikatelského záměru můžeme zjednodušeně naznačit např. takto:

- Představení společnosti.
- Stručná analýza okolí společnosti.
- Vymezení specifických předností společnosti.
- Analýza silných a slabých stránek společnosti (lze použít v každé kapitole projektu).
- Charakteristika základního projektu.
- Popis cílového stavu.
- Formulace významných změn při přechodu na konečný cílový stav.
- Návrh dalšího postupu prací v rámci realizace kontroly plnění záměru.

Představení společnosti

Tato část podnikatelského záměru charakterizuje současný stav společnosti. Její vnitřní struktura může být následující:

- stručná historie,
- současný stav,
- nejvýznamnější investice do současnosti,
- teritoriální rozmístění,
- finanční a ekonomické shrnutí,
- vedení a pracovníci.

Stručná analýza okolí společnosti

Smyslem analýzy je pravdivě poukázat na stav společnosti a na její budoucí možnosti. Tato část podnikatelského záměru by měla obsahovat:

- analýzu trhu a postavení společnosti na trhu,
- analýzu vlivu rozvoje vědy a techniky (porovnání podnikové technologie, jakosti výrobků a jejich technické úrovně se světem),
- analýzu teritoriálního rozmístění (obyvatelstvo, energie, doprava, dostupnost zdrojů, ekologické problémy).

Vymezení specifických předností společnosti - specifické přednosti jsou takové vlastnosti firmy, které jí umožňují dlouhodobě dosahovat v určité oblasti nadprůměrných výsledků a zajišťovat její konkurenční schopnosti.

Analýza silných a slabých stránek společnosti

Silnými stránkami společnosti rozumíme její přednosti, tzn. oblasti, v nichž dosahuje bez zvláštního vypětí dobrých nebo nadprůměrných výsledků. Opakem silných stránek jsou slabé stránky.

Charakteristika základního projektu

Je-li podnikatelský záměr zpracován např. pro získání finančních prostředků, měla by společnost v této části projekt stručně popsat a charakterizovat. Tato charakteristika by měla obsahovat:

- popis projektu,
- předpověď uplatnění výrobku na trhu,
- potřebnou technologii a výrobní náklady,
- potřebu v nárůstu kvalifikace pracovníků,
- komplexní finanční náklady,
- ekonomickou rozvahu.

Informační systémy 1

Popis cílového stavu

Tato část podnikatelského záměru charakterizuje stav, kterého chce podnik dosáhnout. Sestavuje se zásadně několik variant cílového stavu. Každá varianta musí obsahovat čtyři základní části:

- Zásadní názor na tempo a kvalitu růstu společnosti.
- Představu o struktuře výrobního programu.
- Názor na to, o kterou specifickou přednost společnosti se varianta opírá.
- Základní představy o systému řízení a organizační struktuře společnosti.

Formulace významných změn při přechodu na konečný cílový stav

Charakteristika přechodové trajektorie musí být zpracována ve variantách tak, aby v případě nenadálých událostí umožňovala rychlou změnu a přechod na jinou přechodovou trajektorii.

Návrh dalšího postupu prací v rámci realizace kontroly plnění podnikatelského záměru

V této části podnikatelského záměru je třeba stručně a jasně formulovat harmonogram postupu prací při jeho rozpracování a realizaci, včetně hodnocení a způsobu kontroly plnění.

Kontrolní otázky:

1. Jaké je členění podniků do sektorů podle národohospodářského hlediska?
2. Jaký je rozdíl mezi ohlašovací a koncesovanou živností?
3. Vyjmenujte všeobecné podmínky podnikání.
4. Jaké jsou nejdůležitější prvky tvořící vnější podnikatelské prostředí?
5. Co by měla zahrnovat každá podnikatelská strategie?
6. Jaké mohou být důvody pro zpracování podnikatelského záměru?

Úkoly k zamyšlení:

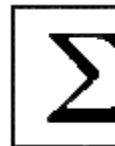
V této kapitole jste se dozvěděli, jak by měla vypadat struktura podnikatelského záměru. Zkuste si představit (zapsat) takový podnikatelský záměr pro firmu ve Vámi zvoleném oboru, kterou byste zakládali.

Korespondenční úkol:

Zjistěte jaké živnostenské oprávnění byste museli (měli) mít, pokud byste chtěli podnikat v IT, konkrétně vývoj software a poskytování konzultací či poradenství v této oblasti.

Shrnutí obsahu kapitoly

V této kapitole jste se seznámili se základními pojmy z oblasti ekonomie a ekonomiky podniků. Dozvěděli jste se, co je to národní hospodářství, jak je definován podnik, podnikání a podnikatelském okolí. V poslední části jste poznali podstatu a důležitost strategie podniku a podnikatelského záměru.



12 Účetnictví podniků z pohledu informačních systémů

V této kapitole se dozvíte:

- Jaký je význam a jaké jsou funkce účetnictví?
- Co je předmětem účetnictví a k čemu slouží účetní doklady?
- Co všechno je majetkem podniku a jak ho rozlišujeme?
- Co jsou to aktiva a pasiva a jaký je mezi nimi vztah?
- Jakými ukazateli zachycujeme hospodářskou činnost podniku?
- Jaké jsou prvky účetnictví?

Po jejím prostudování byste měli být schopni:

- Definovat účel, význam a funkce účetnictví.
- Porozumět rozvaze, účtům, rozlišovat aktiva a pasiva.
- Orientovat se v účetnictví.

Klíčová slova této kapitoly:

Rozvaha, výsledovka, aktiva, pasiva, účet, náklady, výnosy, zisk, účtová osnova a rozvrh, cash flow.

Doba potřebná ke studiu: 6 hodin

Průvodce studiem

Tato kapitola Vás seznámí se základními pojmy z oblasti účetnictví. Dočtete se zde proč účetnictví vedeme, kdo všechno účetní informace vyžaduje, z jakých prvků se účetnictví skládá. Většina pojmů pro Vás bude pravděpodobně opět nová. Doporučujeme dát si po každé podkapitole krátkou pauzu a ověřit si, zda bylo vše pochopeno. Tyto znalosti jsou nezbytné pro pochopení funkčnosti ekonomických IS. Na studium této části si vyhraďte alespoň 6 hodin.

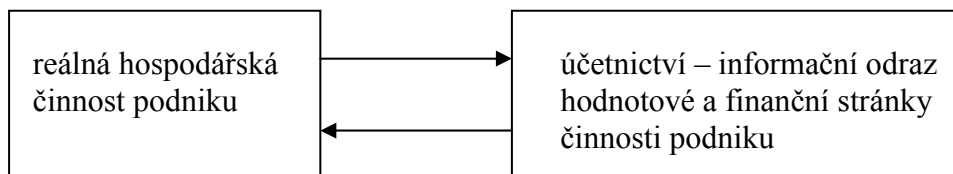
Jádrem ekonomických IS je účetnictví. Pro pochopení funkce účetních programů musíme poznat principy účetnictví (dříve podvojného). Proto si v další části vysvětlíme základní principy účtování – bez detailní znalosti jednotlivých účtů.

12.1 Význam a funkce účetnictví

Obecně můžeme říct, že smyslem účetnictví je utřídit a zpřístupnit ekonomické informace o podniku pro vlastníky, pro zaměstnance, ale také např. pro finanční úřady. Účetnictví poskytuje informační odraz o reálné hospodářské činnosti podniku.



Informační systémy 1



Obr. Vztah účetnictví k činnosti podniku

O různé informace mají pak zájem rozdílné skupiny:

- **Vlastníci** mají největší zájem na prosperitě firmy, přičemž se zajímají o to, jaký je majetek firmy, obrat, zásoby, zisk, jaké jsou pohledávky a závazky, jaký je stav úvěrů. Funkce vlastníků záleží na formě podniku:
 - pokud je vlastník jediným pracovníkem, všechno řídí sám,
 - vlastník je i pracovníkem (malé firmy),
 - řídicí funkce vlastníka,
 - pouze vlastník - tichý společník, akcionář: toho pak zajímá zejména základní jmění a výše jeho vlastního podílu.
- **Zaměstnanci** představují vzhledem k firmě poněkud odlišné zájmy: dá se říct, že jsou zčásti antagonistické výše platů. Zajímají je zejména výše platu (odměny), pojištění, perspektiva zaměstnání.
- **Odběratelé** jsou to jednotlivci nebo firmy, kterým podnik dodává zboží nebo služby za úplatu. To je základní vztah k odběratelům (zákazníkům).

Základní požadavky přitom jsou, aby účetní systém mohl podat:

- informace o ekonomickém stavu podniku,
- informace o hospodářském výsledku podniku,
- prognózy budoucího vývoje.

Přitom účetnictví plní řadu funkcí:

- **Registrační funkce.** Účetnictví je písemnou formou vedení soustavných záznamů o ekonomických jevech. Její výjimečnost a nezastupitelnost spočívá ve schopnosti systémového, nepřetržitého, úplného a kvantitativního vyjádření průběhu hodnotové a finanční stránky hospodářské činnosti vyčleněné hospodářské jednotky.
- **Informační funkce.** Při podnikání či hospodářské činnosti je nutné mít přehled o výši a skladbě prostředků vložených do podnikání a o dosažených výsledcích prováděné činnosti, o míře zhodnocení či znehodnocení vložených prostředků za určité období a o tom, zda se vyplatí dále provozovat podnikatelskou činnost či ne. Z toho plyne, že účetnictví má poskytovat **spolehlivé informace** o podnikatelské zdatnosti podniku, o jeho **výdělečných schopnostech**.
- **Důkazní prostředek.** Vzhledem k tomu že účetnictví poskytuje informace ohledně ekonomické situace podniku, může sloužit jako důkazní prostředek **při vedení sporů**, zejména při uznávání a ochraně práv věřitele před dlužníkem a dlužníka před věřitelem. Stejně tak je účetní evidence důkazem při likvidaci **pojistných událostí**. Proto mimo jiné účetnictví podniku, jež splňuje zákonem určené parametry, musí být schváleno auditorem.

- **Podklad pro vyměření daňové povinnosti.** Účetnictví také slouží ke zjištění informací pro účely daňové. O to, zda všichni podnikatelé vedou řádně účetní agendu, se zajímají **finanční úřady**. Na základě zákona o správě daní a poplatků jsou oprávněny činit opatření potřebná ke správnému stanovení a plnění daňových povinností.
- **Podklad pro rozhodovací procesy.** Z účetnictví čerpá majitel nebo vedení podniku informace důležité pro rozhodování a řízení. Na základě údajů o minulosti nebo současnosti firmy se může rozhodovat o budoucích investicích, obchodech, rozšíření nebo naopak likvidaci podniku.
- **Základ pro hodnocení managementu.** Zejména tam, kde je vedení odděleno od vlastníků (vedení je hodnoceno na základě dosazených hospodářských výsledků).
- **Prostředek kontroly.** S růstem velikosti podniků (ve smyslu velikosti obrátu či vlastního kapitálu, množství zaměstnanců apod.) dochází k oddělování výkonného řízení firmy od vlastnictví – organizace jsou řízeny profesionálním týmem odborníků. Pro majitele či společníky se pak účetnictví stává důležitým prostředkem kontroly a ochrany majetku svěřeného do správy profesionálnímu vedení, tzn. společníci kontrolují vedení podniku, jak udržuje a rozšiřuje jejich majetek.

12.2 Využití účetních informací

Z výše uvedených funkcí účetní agendy je zřejmé, že uživatelem zpracovaných informací jsou různé subjekty zainteresované určitým způsobem na činnosti podniku. O informace nacházející se v účetnictví nemá zájem pouze management a jeho orgány, ale celá řada externích subjektů.

Interními uživateli výstupů z účetnictví jsou všechny úrovně podnikového řízení, tedy:

- strategický management,
- taktický management,
- operativní management.

Důvodů, proč je účetnictví nástrojem podnikového řízení (plánování, organizování, rozhodování, vedení a kontroly) je několik:

- **Dynamická, zpětnovazební kontrola** směřující v zásadě buď k realizaci opatření k nápravě (tzv. korektivní opatření) ve skutečném průběhu činnosti, nebo k revizi plánu či záměru (žádoucího stavu).
- **Poskytování rozsáhlého informačního materiálu pro tvorbu plánu, rozpočtů, formulací rozhodnutí** atd.
- **Zapojení kvalifikace, zkušeností a schopností** zejména vedoucích účetních pracovníků do podnikového plánovacího a rozhodovacího procesu.

Spektrum vnějších subjektů je velmi pestré a jejich požadavky na informace jsou značně diferencované.

Informační systémy 1



Obr. Uživatelé účetnictví

O účetní informace mohou jevit zájem zejména:

- **Vlastníci.** Mají největší zájem na zhodnocení vložených peněz, protože z něj mají přímý ekonomický prospěch.
- **Dodavatelé.** Mají zájem zejména na perspektivních dodávkách zboží, a také na splacení existujících pohledávek, sledují tedy zejména platební schopnost podniku.
- **Ostatní věřitelé.** Stejně tak např. banky mají zájem na tom, aby podnik byl schopen dostát svým závazkům; protože jejich pohledávky jsou zpravidla dlouhodobějšího rázu (úvěry), zajímá je spíše prognóza vývoje.
- **Zaměstnanci.** Základní otázkou je, zda podnik je schopen plnit mzdové požadavky, resp. možné zlepšení (prognóza).
- **Potencionální investoři.** Kapitálový vstup do podniku je ovlivněn jeho situací.
- **Státní orgány.** Požadují po podniku placení daní a zdravotního a sociálního pojištění, kontrolují firmu (daňové úniky!), též sbírají statistické údaje pro makroekonomické ukazatele.
- **Zákazníci.** Zejména větší zákazníci mají zájem na dobré situaci svého dodavatele.
- **Veřejnost.** Veřejné mínění se projevuje hlavně u velkých podniků s mnoha zaměstnanci (sociální otázky).

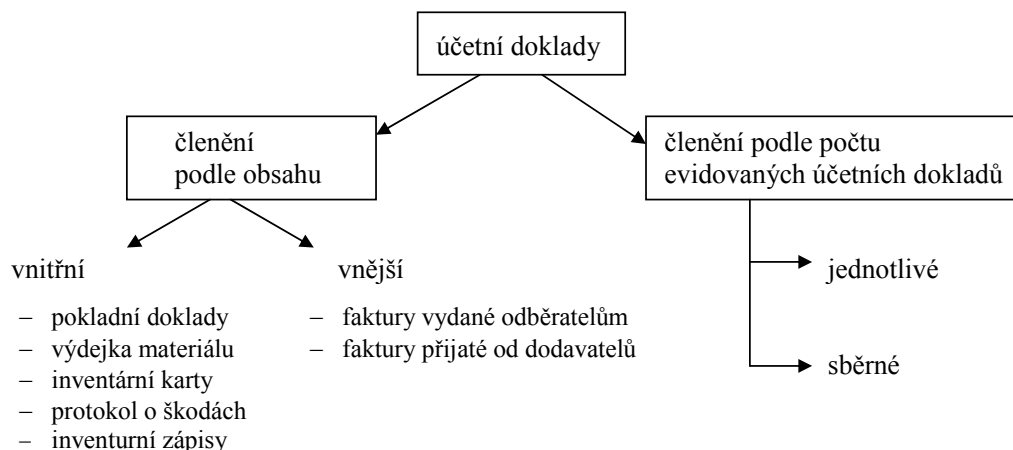
12.3 Účetní doklady, účetní zápisy a účetní knihy

Skutečnosti, které jsou předmětem účetnictví (**účetní případy**), se dokládají **účetními doklady**. Za účetní doklady se považují průkazné účetní záznamy, které musí obsahovat podle zákona o účetnictví:

- a) označení účetního dokladu (název a pořadové číslo),
- b) obsah účetního případu a jeho účastníky,
- c) peněžní částku nebo informaci o ceně za měrnou jednotku a vyjádření množství,
- d) okamžik vyhotovení účetního dokladu, jenž musí nastat bez zbytečného odkladu po zjištění skutečností, které se účetním dokladem zachycují,

Informační systémy 1

- e) okamžik uskutečnění účetního případu, není-li shodný s okamžikem podle písmene d),
- f) podpisový záznam (vlastnoruční podpis nebo obdobný průkazný účetní záznam v technické formě) osoby odpovědné za účetní případ a podpisový záznam osoby odpovědné za jeho zaúčtování.



Obr. Členění účetních dokladů

Každý druh účetního dokladu vyjadřuje konkrétní typ účetních případů. Sběrné doklady vznikají shrnutím několika původních účetních dokladů téhož typu.

Účetní zápisy představují záznam účetních dokladů do příslušných účetních knih. Účetní zápisy se prokazují buď účetními doklady nebo, odvozuje-li se tyto zápisy programem zpracování dat, způsobem uvedeným v projekčně programové dokumentaci software. Účetní jednotky provádějí účetní zápisy v souladu s předepsanými charakteristikami účetní agendy a způsobem zaručujícím jejich trvanlivost (viz. úschova účetních záznamů).

Účetní zápisy je nezbytné provádět **průběžně** v účetním období **po vyhotovení účetního dokladu** a uspořádat způsobem, který umožní **ověřit zaúčtování** všech účetních případů v účetním období (chronologicky), a tak, aby se **zabránilo neoprávněným změnám** a úpravám těchto zápisů. Z tohoto důvodu se například proškrtávají volné řádky nebo prázdné plochy mezi jednotlivými účetními zápisy.

Účetní knihy představují chronologický soupis účetních případů (účetních dokladů) za určité účetní období. V podstatě lze účetní knihy rozdělit do dvou skupin:

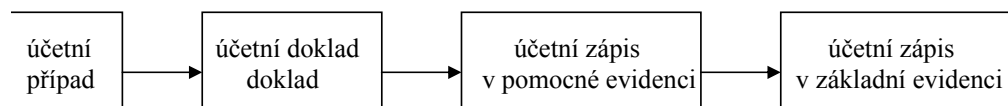
- a) **základní účetní knihy** – jsou specifické pro každou soustavu vedení účetnictví,
- b) **pomocné účetní knihy** – společné pro obě účetní soustavy. Patří sem ku příkladu kniha pohledávek (kniha vystavených faktur), kniha závazků (kniha přijatých faktur), pokladní kniha, kniha cenin, kniha jízd, kniha odeslané pošty, skladní karty, soupis drobného majetku, mzdové listy apod.

Informační systémy 1

Každá pomocná účetní kniha slouží k zapisování jen určitého konkrétního typu účetních dokladů. V pomocných knihách je možné mimo vyjádření v peněžních jednotkách používat také záznamy v měrných jednotkách s vyjádřením množství.

Účetní jednotky mají povinnost vést seznamy účetních knih a seznamy symbolů či zkratk používaných při účetních zápisech s uvedením jejich významu.

Postup provádění účetních záznamů lze zobrazit následujícím řetězcem:



Obr. Sled provádění účetních záznamů

Faktura

Faktura je doklad, vystavený v zásadě při dodání zboží, který zavazuje odběratele ve stanovené lhůtě splatnosti uhradit částku za poskytnuté zboží nebo služby. Rozlišujeme tyto druhy:

- standardní faktura,
- zálohová faktura, po které následuje vyúčtování, je to faktura na celou částku; záloha je zaplacená předem; má to smysl pro plátce DPH, protože záloha se platí bez daně,
- proforma faktury - je vystavena před dodáním zboží platba předem, není to účetní doklad - po dodání se vystaví normální faktura,
- penalizační faktura - až po dni neplacení nebo nezaplacení.

Mezi náležitosti faktury patří:

- číslo faktury - pokud možno v souvislé číselné řadě,
- IČO, DIČ, adresa a bankovní spojení dodavatele,
- datum odeslání, datum splatnosti, datum uskutečnění zdanitelného plnění pokud je faktura i daňovým dokladem,
- číslo dodacího listu, číslo objednávky – popis, za co je vystavena
- cenové údaje: měrná jednotka, cena za jednotku, celková cena - bez DPH, DPH, s DPH, částky se zaokrouhlují např. na koruny.

12.4 Majetek podniku

Majetkem podniku rozumíme všechny prostředky, které podnik využívá při své činnosti. Jsou to jednak finanční prostředky hotovosti, peníze na účtech, dále hmotné prostředky - stroje, zařízení, budovy, ale také nehmotné prostředky - software, patenty; také např. zásoby.

Na majetek přitom pohlížíme dvojitým způsobem:

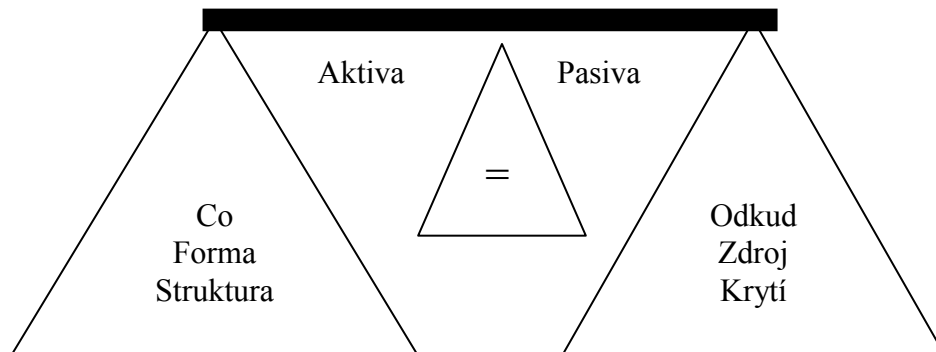
- forma, ve které je majetek vázán,
- původ majetku - z vkladů vlastníka, z vlastní produkce atd.

Informační systémy 1

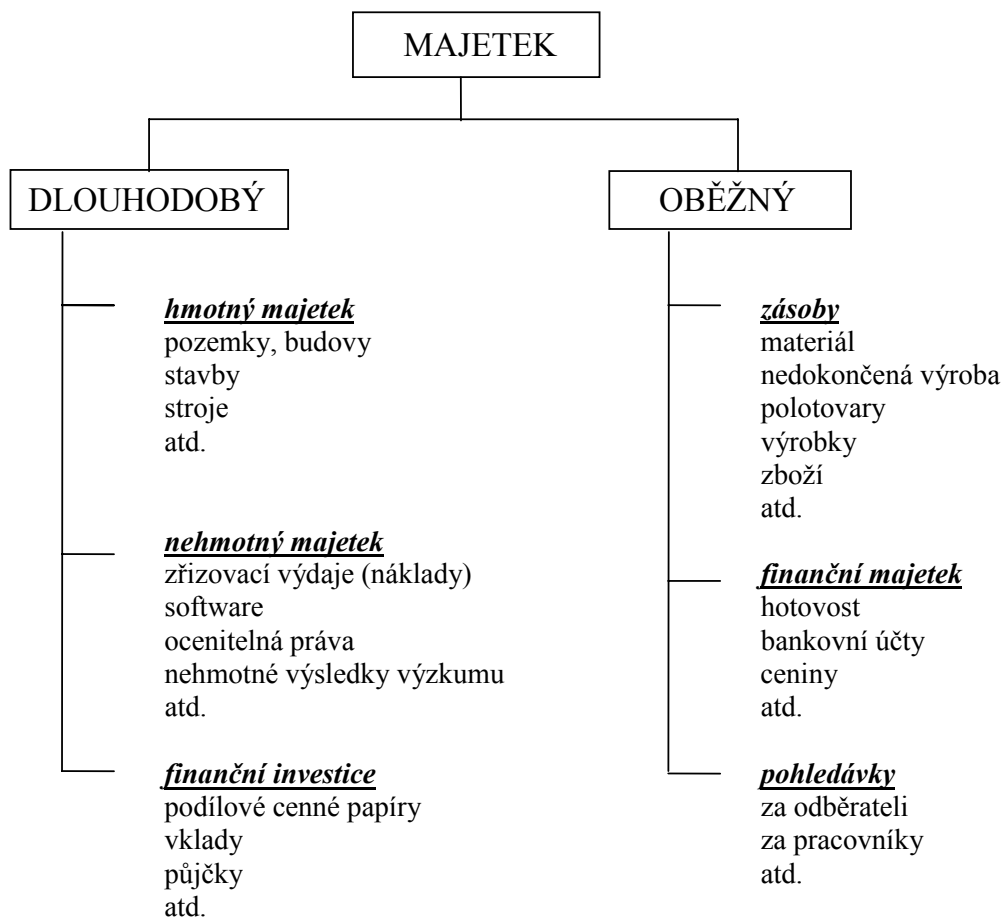
Proto rozlišujeme:

- aktiva - formy majetku - co podnik vlastní,
- pasiva - zdroje majetku - co se muselo použít, aby existoval,

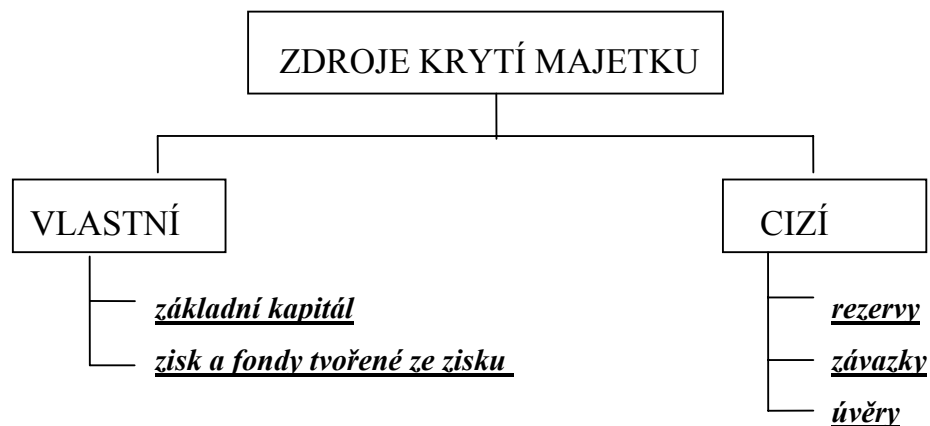
přičemž vždy platí tzv. bilanční princip:



Obr. Vyrovnanost aktiv a pasiv



Obr. Majetková struktura účetní jednotky



Obr. Kapitálová struktura účetní jednotky

Rozvaha

Je účetní výkaz, který podává přehled aktiv a pasiv daného podniku.

ROZVAHA	
AKTIVA	PASIVA
I STALÁ (FIXNÍ, NEOBĚŽNÁ) AKTIVA	I VLASTNÍ ZDROJE (VLASTNÍ KAPITÁL, VLASTNÍ JMĚNÍ)
II BĚŽNÁ (OBĚŽNÁ) AKTIVITA	II CIZÍ ZDROJE (CIZÍ KAPITÁL) pasiva v užším pojetí
III PŘECHODNÁ AKTIVITA	III PŘECHODNÁ PASIVITA

$$A = P$$

Obr. Obecná struktura rozvahy

Nejprve si ukážeme tzv. zahajovací rozvahu, kdy máme pouze počáteční majetek podniku.

Informační systémy 1

Příklad:

Nechť např. byl složen vlastní kapitál 200 000,- Kč, z toho 30 000,- Kč v hotovosti, 170 000,- Kč budova. Zahajovací rozvaha pak vypadá takto:

Aktiva		Pasiva	
Peněžní prostředky	30 000	Vlastní kapitál	200 000
Budovy	170 000		

Zde jsme tedy uvedli zahajovací rozvahu; dále se sestavuje počáteční rozvaha (na počátku určitého účetního období), konečná rozvaha (na konci účetního období, popř. při prodeji nebo likvidaci firmy).

Účel rozvahy je jednak statistický opět pro státní orgány - sledování makroekonomických ukazatelů, jednak je rozvaha podkladem pro určení hospodářského výsledku, tedy jestli podnik hospodařil se ziskem nebo se ztrátou. Pro účely daňové stačí vytvářet rozvahu jednou ročně, pro řízení firmy častěji - např. čtvrtletně nebo dokonce měsíčně.

Klasifikace aktiv

Řekli jsme si, že aktiva představují majetek podniku. Nyní rozlišíme dlouhodobý majetek (tzv. stálá aktiva), a majetek ke krátkodobé spotřebě (spotřebuje se do jednoho roku - oběžná aktiva).

Stálá aktiva.

Jedná se zejména o:

- Hmotný investiční majetek (HIM) dlouhodobé povahy - budovy, stroje, dopravní prostředky, inventář dlouhodobější povahy (např. zabudovaný nábytek do stavby).
- Nehmotný investiční majetek (NIM) - licence, koncese, patenty, autorská práva, software (zejména větší), „goodwill (dobré jméno podniku). Zpravidla se nezařazuje do obvyklých rozvah, ale jen při prodeji nebo ukončení podniku, tedy vždy, kdy se musí ocenit majetek podniku.
- Finanční prostředky dlouhodobého charakteru - cenné papíry a vklady - dluhopisy, vkladové listy, akcie; též finanční účast v jiných podnicích.

Oběžná aktiva.

Sem patří:

- Zásoby - tedy základní prostředky, které jsou vstupy do činnosti podniku (materiál), dále nedokončená výroba, neprodané výrobky, zboží zakoupené za účelem prodeje.
- Pohledávky - jednak z obchodní činnosti (pohledávky za odběrateli), též pohledávky vůči ostatním fyzickým a právnickým osobám (např. plnění škod od zaměstnanců, přebytek daně k zaplacení finančním úřadem).
- Finanční prostředky krátkodobé povahy - rychle obchodovatelné cenné papíry.
- Peněžní prostředky - peníze v hotovosti (na pokladně), vklady na účtech u finančních institucí, šeky, poukázky apod.

Informační systémy 1

Klasifikace pasiv

Řekli jsme si, že pasiva jsou zdroje, kterými jsou aktiva (majetek) financována. Zdroje mohou být vlastní nebo cizí, rozlišujeme tedy:

Vlastní zdroje.

Tvoří je:

- vlastní kapitál,
- rezervní a jiné fondy,
- zisk (v zásadě je dobré část zisku ponechat jako zdroj, ne jej celý rozdělit).

Cizí zdroje.

Jsou to:

- závazky podniku, a to
 - krátkodobé: závazky dodavatelům, krátkodobé úvěry,
 - dlouhodobé: dlouhodobé úvěry, směnky, vydané obligace (se splatností přes jeden rok),
- rezervy (jiné než z rezervních fondů - např. rezervy na překlenutí kursových rozdílů).

Příklad:

Rozvaha, ve které jsou aktiva a pasiva takto rozčleněna, může vypadat takto:

Rozvaha

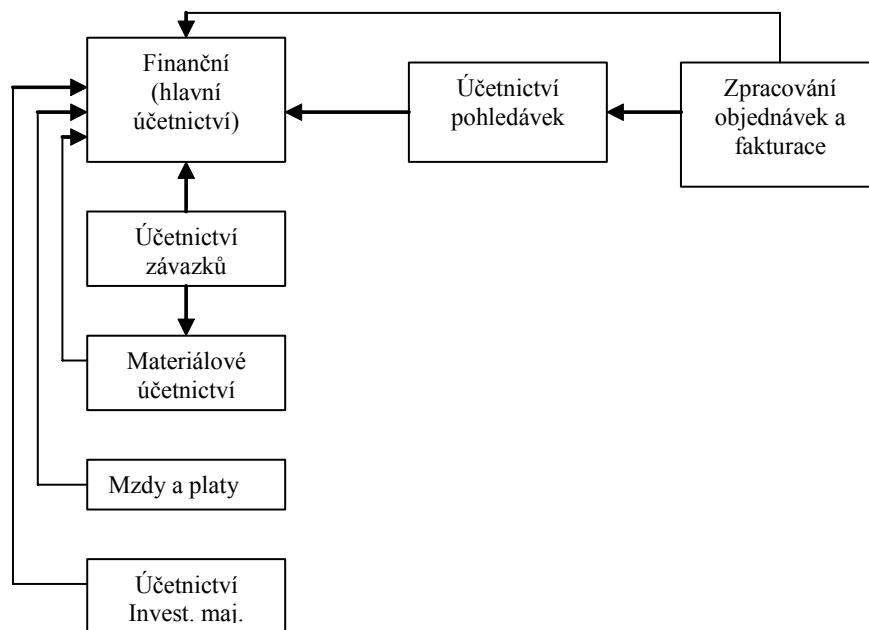
<i>Aktiva</i>	<i>Pasiva</i>
1. Stálá aktiva	1. Vlastní zdroje
Hmotný inv. majetek	Základní kapitál
Nehmotný inv. majetek	Rezervní apod. fondy
Dlouhodobý finanční majetek	Zisk
2. Oběžná aktiva	2. Cizí zdroje
Zásoby	Krátkodobé závazky
Pohledávky	Dlouhodobé závazky
Krátkodobý finanční majetek	3. Rezervy
Finanční prostředky	4. Ostatní pasiva
3. Ostatní aktiva	
(aktiva v průběhu úč. období)	

$$\sum \text{aktiv} = \sum \text{pasiv}$$

Hospodářské operace

Podnik vyvíjí určitou činnost, a proto dochází na obou stranách rozvahy k pohybům; vždy však při zachování bilančního principu. Podnik tak nakupuje materiál, investiční majetek, prodává výrobky nebo služby atd.

Informační systémy 1



Obr. Systémy účtů a toky dat pro finanční účetnictví

Přítom jednotlivé hospodářské operace mohou mít:

1. Vliv pouze na aktiva. Např. hotové peníze se vyberou z účtu na pokladnu.
2. Vliv pouze na pasiva. Např. splacení závazku zvýšením úvěru, převedení zisku do rezerv.
3. Přírůstek aktiv i pasiv. Např. si vezmeme úvěr a nakoupíme zboží.
4. Úbytek aktiv i pasiv. Např. úhrada dluhu penězi z účtu.

Uvedeme si na závěr ještě jeden příklad.

Příklad:

Mějme počáteční rozvahu podniku

Aktiva		Pasiva	
Stálá aktiva	200 000 Kč	Základní jmění	280 000 Kč
Materiál	100 000 Kč	Závazky k dodavatelům	70 000 Kč
Peníze	50 000 Kč		
Σ aktiv	350 000 Kč	Σ pasiv	350 000 Kč

Nyní podnik vyvíjí určitou činnost, v rámci které dojde (např. během měsíce) k těmto pohybům:

- opotřebení strojů v hodnotě 10 000 Kč
- spotřeba materiálu za 50 000 Kč
- výplata mezd ve výši 40 000 Kč
- prodej výrobků za 120 000 Kč

Informační systémy 1

Nová rozvaha pak vypadá takto:

Aktiva		Pasiva	
Stálá aktiva	190 000 Kč	Základní jmění (vč. zisku)	300 000 Kč
Materiál	50 000 Kč	Závazky k dodavatelům	70 000 Kč
Peníze	130 000 Kč		
Σ aktiv	370 000 Kč	Σ pasiv	370 000 Kč

12.5 Náklady, výnosy, zisk

Jedná se o zachycení hospodářské činnosti podniku:

- podnik z něčeho vyrábí, a proto má náklady,
- prodejem výrobků získá výnosy (aby měl čím pokrýt náklady),
- jako rozdíl výnosů a nákladů vzniká zisk (resp. ztráta).

Mezi náklady tak patří opotřebení strojů, mzdy, materiál, energie apod.; výnosy tvoří zejména tržby.

Změna základního jmění

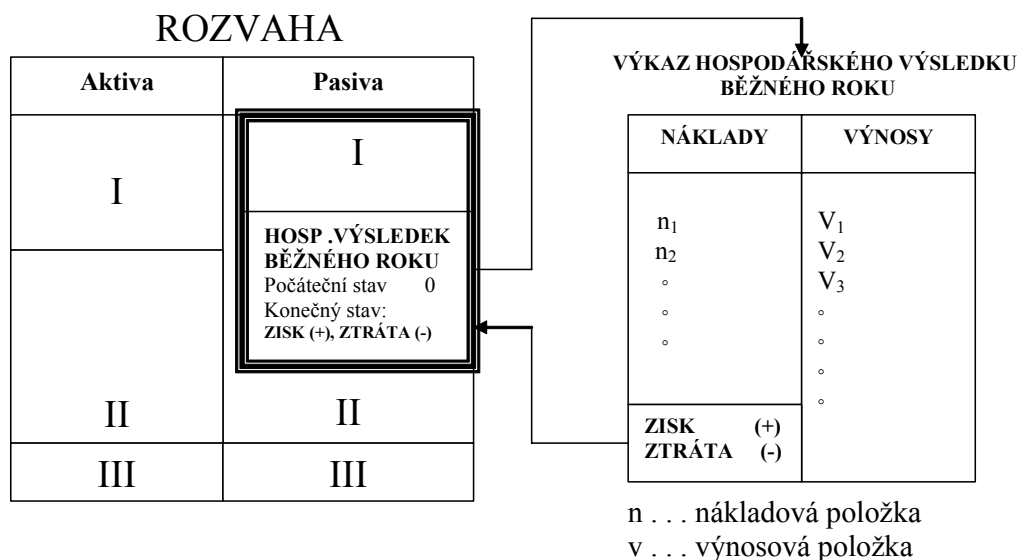
Nyní nemáme na mysli základní jmění jako součet vkladů společníků, ale celé základní jmění. Musí se zde proto promítnout všechny náklady a výnosy.

Vrátíme se k předchozímu příkladu, kde podnik provedl určité hospodářské operace. Základní jmění se směnilo na $280 - 10 - 50 - 40 + 120 = 300$ tisíc Kč; porovnáním konečného a počátečního základního jmění dostaneme zisk ve výši 20 000 Kč.

Vidíme, že při větším počtu účetních operací by byl celý systém značně nepřehledný; dostaneme se proto k dalším prostředkům, a konečně i k účtům.

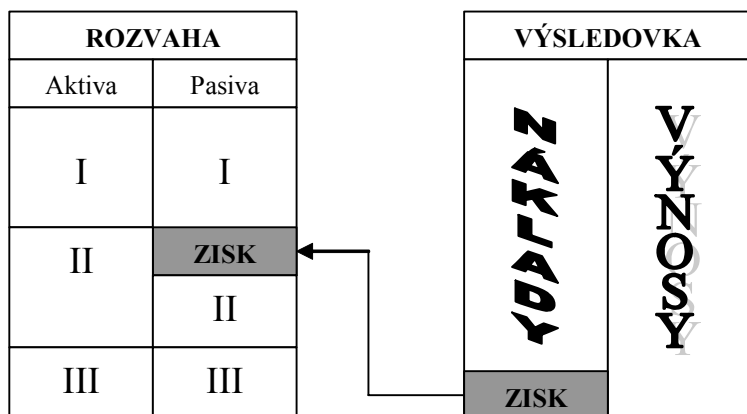
Osamostatnění nákladů a výnosů

Nyní se budeme zabývat tzv. *výsledovkou*, jejíž součástí je zisk nebo ztráta (proto se taky někdy nazývá výkaz zisků a ztrát - profit and loss account). Obsahuje tedy veškeré náklady, výnosy a dosažený zisk (resp. ztrátu), a to tak, aby opět platil bilanční princip. S rozvahou je výsledovka spojená právě přes zisk.



Obr. Zrození výkazu o složkách hospodářského výsledku běžného roku

HOSPODÁŘSKÝ VÝSLEDEK = ZISK



Obr. Vazba výsledovky a rozvahy

Příklad:

Opět využijeme operaci z předchozího příkladu a sestavíme výsledovku:

<i>Náklady</i>		<i>Výnosy</i>	
Odpisy	10 000	Tržby	120 000
Materiál	50 000		
Mzdy	40 000		
Zisk	20 000		

Σ 120 000 Kč

Σ 120 000 Kč

Zisk se tedy dopočítá tak, aby platil bilanční princip.

Informační systémy 1

Kalkulační vzorec

Jednou z funkcí účetnictví je vyhodnocování všech položek. Kalkulační vzorec vyhodnocuje náklady, režii apod.; váže se k zisku podniku. K jeho položkám tedy patří:

- přímé náklady - materiál, mzdy, opotřebení strojů,
- režie - zpravidla se stanovuje procentem přímých nákladů; případné další strukturování záleží na podniku - jestliže chceme mít podrobnější přehled, rozlišíme např. správní režii, podnikovou režii, odbytovou režii, ostatní,
- inflační vývoj,
- energie, vytápění atd.

Konkrétní podoba kalkulačního vzorce je závislá na konkrétním podniku.

12.6 Prvky účetnictví

V průběhu účetního období s rozvahou a výsledovkou nevystačíme; tak se každému řádku rozvahy, resp. výsledovky, přiřadí jeden nebo více účtů, a tak máme zajištěný „on line“ přístup k informacím.

Účetnictví tak sleduje hospodářské operace, které by se teoreticky daly sledovat v rozvaze a výsledovce.

Účty

Účet je výkaz, ke kterému se vztahují účetní operace, a ty jej buď obohacují, nebo z něj odčerpávají. Nejjednodušší způsob uvedení účtů je ten, že každému řádku rozvahy odpovídá jeden účet.

Při ručním zpracování se používala tzv. T-forma, která zachycuje strany nazývané Má dáti a Dal. Zápis účtu v T-formě vypadá následovně:

MD	D
poč. stav z1	
z3	z2
kon. stav	

Přítom přírůstky aktivních účtů se objevují na straně Má dáti, přírůstky účtů pasivních na straně Dal. Vzpomeneme-li si tedy na typy hospodářských operací z odstavce 2.2 podle vlivu na aktiva a pasiva, můžeme si rozmyslet, kde se objeví změna příslušného aktivního, resp. pasivního účtu. Zjistíme tak, že se vždy objeví jedna položka na straně Má dáti, a jedna na straně Dal. Platí tedy bilanční princip

$$\sum MD = \sum D.$$

Informační systémy 1

počáteční stav	
přírůstky	úbytky
obraty	obraty
konečný stav	

	počáteční stav
úbytky	přírůstky
obraty	obraty
	konečný stav

přírůstky	úbytky
obraty	obraty
výsledný obrat	

úbytky	přírůstky
obraty	obraty
	výsledný obrat

Obr. Schéma zápisů na jednotlivých druzích účtů

Proč se objevují přírůstky u aktivních a pasivních účtů na opačných stranách, to plyne z bilančního principu, navíc: pokud přibude něco např. v penězích, je to pro podnik opravdu „plus“, zatímco pokud přibudou závazky, je to naopak. Znamená to tedy, že na straně Dal se vždy objeví částky, které podnik platí.

Náležitosti účtů

Účty rozvahové. Zobrazují nám část rozvahy; rozlišujeme ještě dále:

účty aktivní - mapují určitou část aktiv rozvahy. Jak jsme si řekli, přírůstky (zde: xi) se zachycují na straně Má dáti, úbytky (zde: yj) na straně Dal:

MD	D
poč. stav	
x1	y1
xi	yj
ΣMD	$-\Sigma D$

kon. stav

Konečný stav přitom vznikne jednoduše tak, že k počátečnímu stavu přičteme součet obrátů strany Má dáti a odečteme součet obrátů Dal.

účty pasivní - podobně mapují určitou část pasiv rozvahy; přírůstky (zde: xi) se zachycují na straně Dal, úbytky (zde: yj) na straně Má dáti:

Informační systémy 1

MD	D
	poč. stav
x1	y1
xi	yj
- Σ MD	Σ D
	kon. stav

Konečný stav zde analogicky vznikne z počátečního stavu přičtením obratu Dal a odečtením obratu Má dáti.

Účty výsledkové.

Stejně jako rozvaha, dělí se i výsledovka na účty. Poněvadž sledujeme náklady a výnosy, máme účty nákladové a výnosové.

Pozor: výnosové účty se chovají jako pasivní, zatímco nákladové účty jako aktivní; totiž např. materiál je zcela typicky aktivní účet, a do nákladů je zahrnuta **spotřeba** materiálu; pokud snížíme množství materiálu, zvýšíme tak jeho spotřebu; aby byl zachován bilanční princip, musí být spotřeba účtem aktivním.

Účty syntetické a analytické. Některé účty se nám mohou dále rozpadat: tak např. již zmíněný materiál můžeme evidovat podle jeho druhů, popř. podle různých středisek a skladů. Dostáváme tak

$$S_i = \Sigma A_{ij}$$

tedy jeden **syntetický** účet jako sumu více účtů **analytických**. Přitom analytické účty mají stejnou povahu (aktivní, pasivní) jako příslušný účet syntetický.

Předepsané vedení účtů. Zákon o účetnictví ukládá nejhrubší mocné členění účtů; další rozčlenění na analytické účty je na naši libovůli.

Dále je uvedena účtová osnova syntetických účtů pro podnikatele:

Účtová třída 0 - Dlouhodobý majetek

01 - Dlouhodobý nehmotný majetek

010 - Dlouhodobý nehmotný majetek

011 - Zřizovací výdaje

012 - Nehmotné výsledky výzkumu a vývoje

013 - Software

014 - Ocenitelná práva

019 - Ostatní dlouhodobý nehmotný majetek

02 - Dlouhodobý hmotný majetek odpisovaný

021 - Stavby

022 - Samostatné movité věci a soubory movitých věcí

025 - Pěstitelské celky trvalých porostů

026 - Základní stádo a tažná zvířata

029 - Ostatní dlouhodobý hmotný majetek

Informační systémy 1

03 - Dlouhodobý hmotný majetek neodpisovaný

031 - Pozemky

032 - Umělecká díla a sbírky

04 - Pořízení dlouhodobého majetku

040 - Pořízení dlouhodobého majetku

041 - Pořízení dlouhodobého nehmotného majetku

042 - Pořízení dlouhodobého hmotného majetku

043 - Pořízení dlouhodobého finančního majetku

05 - Poskytnuté zálohy na dlouhodobý majetek

050 - Poskytnuté zálohy na dlouhodobý majetek

051 - Poskytnuté zálohy na dlouhodobý nehmotný majetek

052 - Poskytnuté zálohy na dlouhodobý hmotný majetek

053 - Poskytnuté zálohy na dlouhodobý finanční majetek

06 - Dlouhodobý finanční majetek

061 - Podílové cenné papíry a podíly v podnicích s rozhodujícím vlivem

062 - Podílové cenné papíry a podíly v podnicích s podstatným vlivem

063 - Realizovatelné cenné papíry a podíly

065 - Dlužné cenné papíry držené do splatnosti

066 - Půjčky podnikům ve skupině

067 - Ostatní půjčky

069 - Ostatní dlouhodobý finanční majetek

07 - Oprávky k dlouhodobému nehmotnému majetku

070 - Oprávky k dlouhodobému nehmotnému majetku

071 - Oprávky ke zřizovacím výdajům

072 - Oprávky k nehmotným výsledkům výzkumu a vývoje

073 - Oprávky k softwaru

074 - Oprávky k ocenitelným právům

079 - Oprávky k ostatnímu dlouhodobému nehmotnému majetku

08 - Oprávky k dlouhodobému hmotnému majetku

081 - Oprávky ke stavbám

082 - Oprávky k samostatným movitým věcem a souborům movitých věcí

085 - Oprávky k pěstitelským celkům trvalých porostů

086 - Oprávky k základnímu stádu a tažným zvířatům

089 - Oprávky k ostatnímu dlouhodobému hmotnému majetku

09 - Opravné položky k dlouhodobému majetku

091 - Opravná položka k dlouhodobému nehmotnému majetku

092 - Opravná položka k dlouhodobému hmotnému majetku

093 - Opravná položka k dlouhodobému nedokončenému nehmotnému majetku

094 - Opravná položka k dlouhodobému nedokončenému hmotnému majetku

095 - Opravná položka k poskytnutým zálohám

096 - Opravná položka k dlouhodobému finančnímu majetku

097 - Opravná položka k nabytému majetku

098 - Oprávky k opravné položce k nabytému majetku

Informační systémy 1

Účtová třída 1 - Zásoby

11 - Materiál

111 - Pořízení materiálu

112 - Materiál na skladě

119 - Materiál na cestě

12 - Zásoby vlastní výroby

121 - Nedokončená výroba

122 - Polotovary vlastní výroby

123 - Výrobky

124 - Zvířata

13 - Zboží

131 - Pořízení zboží

132 - Zboží na skladě a v prodejnách

139 - Zboží na cestě

19 - Opravné položky k zásobám

191 - Opravná položka k materiálu

192 - Opravná položka k nedokončené výrobě

193 - Opravná položka k polotovarům vlastní výroby

194 - Opravná položka k výrobkům

195 - Opravná položka ke zvířatům

196 - Opravná položka ke zboží

Účtová třída 2 - Finanční účty

21 - Peníze

210 - Peníze

211 - Pokladna

213 - Ceniny

22 - Účty v bankách

221 - Bankovní účty

23 - Běžné bankovní úvěry

231 - Krátkodobé bankovní úvěry

232 - Eskontní úvěry

24 - Jiné krátkodobé finanční výpomoci

241 - Emitované krátkodobé dluhopisy .

249 - Ostatní krátkodobé finanční výpomoci

25 - Krátkodobý finanční majetek

251 - Majetkové cenné papíry k obchodování

252 - Vlastní akcie a vlastní obchodní podíly

253 - Dlužné cenné papíry k obchodování

255 - Vlastní dluhopisy

256 - Dlužné cenné papíry se splatností do jednoho roku držené do splatnosti

Informační systémy 1

257 - Ostatní realizovatelné cenné papíry
259 - Pořizování krátkodobého finančního majetku

26 - Převody mezi finančními účty
261 - Peníze na cestě

29 - Opravné položky ke krátkodobému finančnímu majetku
291 - Opravná položka ke krátkodobému finančnímu majetku

Účtová třída 3 - Zúčtovací vztahy

31 - Pohledávky
311 - Odběratelé
312 - Směnky k inkasu
313 - Pohledávky za eskontované cenné papíry
314 - Poskytnuté provozní zálohy
315 - Ostatní pohledávky

32 - Závazky
321 - Dodavatelé
322 - Směnky k úhradě
324 - Přijaté zálohy
325 - Ostatní závazky

33 - Zúčtování se zaměstnanci a institucemi
331 - Zaměstnanci
333 - Ostatní závazky vůči zaměstnancům
335 - Pohledávky za zaměstnanci
336 - Zúčtování s institucemi sociálního zabezpečení a zdravotního pojištění

34 - Zúčtování daní a dotací
341 - Daň z příjmů
342 - Ostatní přímé daně
343 - Daň z přidané hodnoty
345 - Ostatní daně a poplatky
346 - Dotace ze státního rozpočtu
347 - Ostatní dotace
35 - Pohledávky ke společníkům a sdružení
351 - Pohledávky k podnikům ve skupině
353 - Pohledávky za upsaný vlastní kapitál
354 - Pohledávky za společníky při úhradě ztráty
355 - Ostatní pohledávky za společníky
358 - Pohledávky k účastníkům sdružení

36 - Závazky ke společníkům a sdružení
361 - Závazky k podnikům ve skupině
364 - Závazky ke společníkům při rozdělování zisku
365 - Ostatní závazky ke společníkům
366 - Závazky ke společníkům a členům družstva ze závislé činnosti
367 - Závazky z upsaných nesplacených cenných papírů a vkladů

Informační systémy 1

368 - Závazky k účastníkům sdružení

37 - Jiné pohledávky a závazky

371 - Pohledávky z prodeje podniku

372 - Závazky z koupě podniku

373 - Pohledávky a závazky z pevných termínových operací

374 - Pohledávky z pronájmu

375 - Pohledávky z emitovaných dluhopisů

376 - Nakoupené opce

377 - Prodané opce

378 - Jiné pohledávky

379 - Jiné závazky

38 - Přejícné úcty aktiv a pasiv

381 - Náklady příštích období

382 - Komplexní náklady příštích období

383 - Výdaje příštích období

384 - Výnosy příštích období

385 - Příjmy příštích období

386 - Kurzové rozdíly aktivní*

387 - Kurzové rozdíly pasivní*

388 - Dohadné úcty aktivní

389 - Dohadné úcty pasivní

39 - Opravná položka k zúčtovacím vztahům a vnitřní zúčtování

391 - Opravná položka k pohledávkám

395 - Vnitřní zúčtování

398 - Spojovací účet při sdružení

Účtová třída 4 - Kapitálové úcty a dlouhodobé závazky

41 - Základní kapitál a kapitálové fondy

411 - Základní kapitál

412 - Emisní ážio

413 - Ostatní kapitálové fondy

414 - Oceňovací rozdíly z přecenění majetku a závazků

418 - Oceňovací rozdíly z přecenění při přeměnách

419 - Změny základního kapitálu

42 - Fondy ze zisku a převedené výsledky hospodaření

421 - Zákonný rezervní fond

422 - Nedělitelný fond

423 - Statutární fondy

427 - Ostatní fondy

428 - Nerozdělený zisk minulých let

429 - Neuhrazená ztráta minulých let

43 - Výsledek hospodaření

431 - Výsledek hospodaření ve schvalovacím řízení

45 - Rezervy

451 - Rezervy zákonné

Informační systémy 1

459 - Ostatní rezervy

46 - Bankovní úvěry

461 - Bankovní úvěry

47 - Dlouhodobé závazky

471 - Dlouhodobé závazky k podnikům ve skupině

473 - Emitované dluhopisy

474 - Závazky z pronájmu

475 - Dlouhodobé přijaté zálohy

478 - Dlouhodobé směnky k úhradě

479 - Ostatní dlouhodobé závazky

48 - Odložený daňový závazek a pohledávka

481 - Odložený daňový závazek a pohledávka

49 - Individuální podnikatel

491 - Účet individuálního podnikatele

Účtová třída 5 - Náklady

50 - Spotřebované nákupy

500 - Spotřebované nákupy

501 - Spotřeba materiálu

502 - Spotřeba energie

503 - Spotřeba ostatních neskladovatelných dodávek

504 - Prodané zboží

51 - Služby

510 - Služby

511 - Opravy a udržování

512 - Cestovné

513 - Náklady na reprezentaci

518 - Ostatní služby

52 - Osobní náklady

520 - Osobní náklady

521 - Mzdové náklady

522 - Příjmy společníků a členů družstva ze závislé činnosti

523 - Odměny členům orgánů společnosti a družstva

524 - Zákonné sociální pojištění

525 - Ostatní sociální pojištění

526 - Sociální náklady individuálního podnikatele

527 - Zákonné sociální náklady

528 - Ostatní sociální náklady

53 - Daně a poplatky

530 - Daně a poplatky

531 - Daň silniční

532 - Daň z nemovitostí

538 - Ostatní daně a poplatky

Informační systémy 1

- 54 - Jiné provozní náklady
- 540 - Jiné provozní náklady
- 541 - Zůstatková cena prodaného dlouhodobého nehmotného a hmotného majetku
- 542 - Prodaný materiál
- 543 - Dary
- 544 - Smluvní pokuty a úroky z prodlení
- 545 - Ostatní pokuty a penále
- 546 - Odpis pohledávky
- 548 - Ostatní provozní náklady
- 549 - Manka a škody

- 55 - Odpisy, rezervy a opravné položky provozních nákladů
- 551 - Odpisy dlouhodobého nehmotného a hmotného majetku
- 552 - Tvorba zákonných rezerv
- 554 - Tvorba ostatních rezerv
- 555 - Zúčtování komplexních nákladů příštích období
- 557 - Zúčtování oprávek k opravné položce k nabytému majetku
- 558 - Tvorba zákonných opravných položek
- 559 - Tvorba opravných položek

- 56 - Finanční náklady
- 560 - Finanční náklady
- 561 - Prodané cenné papíry a podíly
- 562 - Úroky
- 563 - Kurzové ztráty
- 564 - Náklady z přecenění majetkových cenných papírů
- 566 - Náklady z finančního majetku
- 567 - Náklady z derivátových operací
- 568 - Ostatní finanční náklady
- 569 - Manka a škody na finančním majetku

- 57 - Rezervy a opravné položky finančních nákladů
- 574 - Tvorba rezerv
- 579 - Tvorba opravných položek

- 58 - Mimořádné náklady
- 580 - Mimořádné náklady
- 581 - Náklady na změnu metody
- 582 - Škody
- 584 - Tvorba rezerv
- 588 - Ostatní mimořádné náklady
- 589 - Tvorba opravných položek
- 59 - Daně z příjmů a převodové účty
- 591 - Daň z příjmů z běžné činnosti - splatná
- 592 - Daň z příjmů z běžné činnosti - odložená
- 593 - Daň z příjmů z mimořádné činnosti - splatná
- 594 - Daň z příjmů z mimořádné činnosti - odložená
- 595 - Dodatečné odvody daně z příjmů

Informační systémy 1

- 596 - Převod podílu na výsledku hospodaření společníkům
- 597 - Převod provozních nákladů
- 598 - Převod finančních nákladů

Účtová třída 6 - Výnosy

- 60 - Tržby za vlastní výkony a zboží
- 600 - Tržby za vlastní výkony a zboží
- 601 - Tržby za vlastní výrobky
- 602 - Tržby z prodeje služeb
- 604 - Tržby za zboží

- 61 - Změny stavu vnitropodnikových zásob
- 610 - Změny stavu vnitropodnikových zásob
- 611 - Změna stavu nedokončené výroby
- 612 - Změna stavu polotovarů
- 613 - Změna stavu výrobků
- 614 - Změna stavu zvířat

- 62 - Aktivace
- 620 - Aktivace
- 621 - Aktivace materiálu a zboží
- 622 - Aktivace vnitropodnikových služeb
- 623 - Aktivace dlouhodobého nehmotného majetku
- 624 - Aktivace dlouhodobého hmotného majetku

- 64 - Jiné provozní výnosy
- 640 - Jiné provozní výnosy
- 641 - Tržby z prodeje dlouhodobého nehmotného a hmotného majetku
- 642 - Tržby z prodeje materiálu
- 644 - Smluvní pokuty a úroky z prodlení
- 646 - Výnosy z odepsaných pohledávek
- 648 - Ostatní provozní výnosy

- 65 - Zúčtování rezerv a opravných položek provozních výnosů
- 652 - Zúčtování zákonných rezerv
- 654 - Zúčtování ostatních rezerv
- 655 - Zúčtování komplexních nákladů příštích období
- 657 - Zúčtování oprávek k opravné položce k nabytému majetku
- 658 - Zúčtování zákonných opravných položek
- 659 - Zúčtování opravných položek

- 66 - Finanční výnosy
- 660 - Finanční výnosy
- 661 - Tržby z prodeje cenných papírů a podílů
- 662 - Úroky
- 663 - Kurzové zisky
- 664 - Výnosy z přecenění majetkových cenných papírů
- 665 - Výnosy z dlouhodobého finančního majetku
- 666 - Výnosy z krátkodobého finančního majetku

Informační systémy 1

667 - Výnosy z derivátových operací
668 - Ostatní finanční výnosy
67 - Zúčtování rezerv a opravných položek finančních výnosů
674 - Zúčtování rezerv 679 - Zúčtování opravných položek
68 - Mimořádné výnosy
680 - Mimořádné výnosy
681 - Výnosy ze změny metody
684 - Zúčtování rezerv
688 - Ostatní mimořádné výnosy
689 - Zúčtování opravných položek

69 - Převodové účty
697 - Převod provozních výnosů
698 - Převod finančních výnosů

Účtová třída 7 - Závěrkové a podrozvahové účty

70 - Účty rozvažné
701 - Počáteční účet rozvažný
702 - Konečný účet rozvažný .

71 - Účet zisků a ztrát
710 - Účet zisků a ztrát

75 až 79 - Podrozvahové účty

Účtové třídy 8 a 9 - Vnitropodnikové účetnictví

Měsíčně se musí v hlavní knize vykázat obraty Má dáti a obraty Dal spolu s počátečními a konečnými zůstatky všech účtů; někdy ještě obraty Má dáti a Dal od začátku roku.

Zpravidla se měsíčně tiskne výsledovka a rozvaha (není povinné ze zákona).

K poslednímu dni účetního roku se dělá roční účetní uzávěrka a dále tzv. přemostění účtů, což je převod do nového roku. K tomu slouží následující dva prostředky:

- Konečný účet rozvažný. Bývá to zpravidla skupina účtů, na které se převedou zůstatky všech rozvahových účtů (tím se účty vyprázdní); na začátku roku se převedou zpět jako počáteční zůstatky rozvahových účtů. Takto je možné na přelomu roku změnit účetní osnovu.

- Účet zisků a ztrát. Zde se vypočte zisk, který se posléze rozdělí, popř. ztráta, která se uhradí.

Účetní rozvrh.

Je to příklad (rozpis) používaných účtů. Měl by obsahovat řádek rozvahy nebo výsledovky, do kterého má konečný stav toho kterého účtu vstupovat.

Informační systémy 1

Princip podvojnosti a souvztažnosti účtů

Již jsme se zmínili o tom, že každá účetní operace vyvolává změnu v některém účtu na straně Má dáti, a jinou, stejně velkou změnu na straně Dal. Platí tedy bilanční princip $\Sigma MD = \Sigma D$. Podle bilančního principu se tak dá zkontrolovat správnost účtování.

Systém účetnictví je tak určitým způsobem uzavřený a každá účetní (hospodářská) operace přelévá prostředky z jednoho účtu na jiný.

Přitom je třeba zachovat souvztažnost účtů, tedy při každé účetní operaci provést změnu na stranách Má dáti a Dal u správných účtů. Totiž bilanční princip zůstane zachován, i když operaci zaevidujeme na nesprávné účty.

Příklad:

Nyní si ukážeme praktické promítnutí hospodářských operací do účtů. Všechny hodnoty necht' jsou např. v tisících Kč nebo chcete-li, v miliónech.

Náš podnik má následující počáteční rozvahu:

<i>Aktiva</i>		<i>Pasiva</i>	
Stálá aktiva	2000	Základní jmění	2000
Materiál	1000	Rezervní fond	185
Účet	150	Závazky k dodavatelům	<u>1000</u>
Peníze	<u>35</u>		
	3185		3185

Nyní provedeme následující operace:

1. nákup materiálu,
2. úvěr a z něj úhrada dodavatelům,
3. převod peněz z účtu na pokladnu,
4. splátka úvěru.

Na účty, které nám rozčleněním rozvahy vzniknou, budou mít operace takovýto vliv:

Stálá aktiva		Základní jmění	
PS 2000		PS 2000	
KS 2000		KS 2000	
Materiál		Rezervní fond	
PS 1000		PS 185	
(1) 250		KS 185	
KS 1250		PS 35	
(2) 200		(3) 15	
KS 1050		KS 50	
(4) 30		KS 170	
Dodavatelé		Účet	
PS 1000		PS 150	
(1) 250		(3) 15	
(2) 200		(4) 30	
KS 1050		Σ 45	
(4) 30		Úvěr	
KS 1050		PS 0	
KS 1050		(2) 200	
KS 1050		(4) 30	
KS 1050		KS 170	

Informační systémy 1

Všimněme si přitom, do kterých účtů a na které jejich strany se promítají jednotlivé účetní operace; mějme na paměti bilanční princip a uvědomme si, které účty jsou aktivní a které pasivní.

Můžeme ještě na závěr vytvořit konečnou rozvahu

	<i>Aktiva</i>		<i>Pasiva</i>	
Stálá aktiva	2000	Základní jmění		2000
Materiál	1250	Rezervní fond		185
Účet	105	Závazky k dodavatelům		1050
Peníze	50	Úvěr		170
Σ	3405	Σ		3405

Účetní zápisy

Každý účetní zápis musí obsahovat datum, číslo dokladu, částku, účet MD, účet D a textový popis. Oprava zápisu není smazání dokladu, ale opravná účetní operace s opačným znaménkem.

U rozvahových účtů si musíme držet počáteční stavy (u výsledkových to není třeba, protože počáteční stav je vždy nulový); účetní systémy si zpravidla drží počáteční stav každého měsíce. Existují rovněž účty pomocné, které slouží v průběhu roku, a na konci roku musí být nulové.

Účetní zápisy se evidují v deníku, který každá organizace musí vést ze zákona o účetnictví. V počítači to tak může být jeden soubor, do kterého se sbíhají záznamy o všech účetních operacích. Nemusí se jednat o fyzický soubor, ale je možné jej podle potřeby generovat z jiných záznamů. Měl by vypadat asi takto:

Číslo položky

Datum

Číslo dokladu

Popis dokladu

Datum účetní operace

Účet MD

Účet D

Částka

Čísla dokladů by přitom měla tvořit souvislou řadu (kvůli věrohodnosti).

Chyby a jejich opravy

V účetních záznamech se v zásadě nesmí škrtnat a přepisovat, proto se používají tzv. opravné účetní zápisy, které napraví účinek chybné operace. Může jít přitom buď o přímou opravu na správný stav, anebo tzv. storno, což znamená, že chybný zápis se anuluje, a poté se provede správná účetní operace.

Problém nastává, pokud dojde k chybě po uzávěrce. V takovém případě by bylo nutné uzávěrku nebo všechny uzávěrky za období od chyby přepočítat, a

Informační systémy 1

to je pracné (zejména provedl-li se mezitím např. odvod DPH). Proto i v tomto případě se použijí opravné účetní doklady, nejlépe již zmíněné storno.

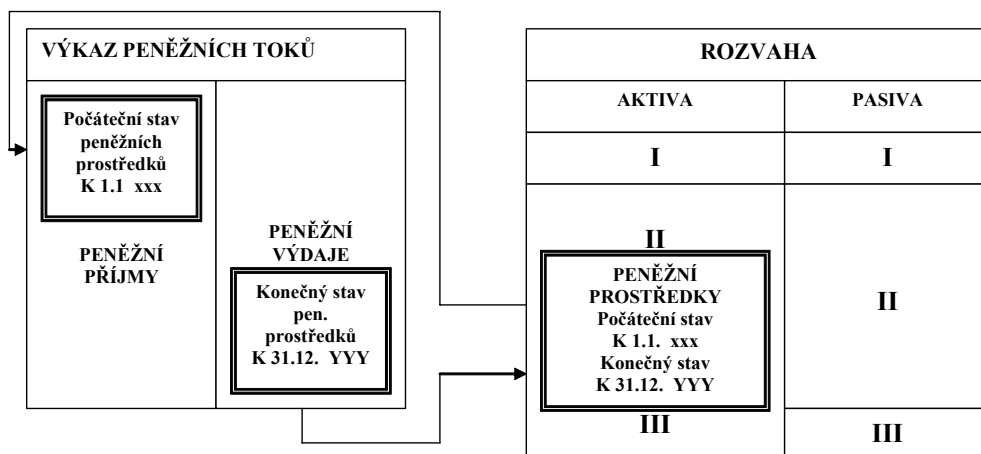
12.7 Cash Flow

Účetně může být výsledkem zisk, i když peníze na účtech nejsou: do výnosů se totiž započítávají i dosud nezaplacené pohledávky, a naopak do nákladů dosud neuhrazené závazky podniku. Zisk je dále „zkreslen“ odpisy, které zatěžují podnik (jako náklady) delší dobu, přestože majetek již uhrazen byl.

Proto se vykazuje zisk z provozní činnosti, kam se tyto částky nezahrnují. (Daň se nicméně platí stále ze zisku, který se počítá obvyklým způsobem.)

Zavádí se dále cash flow neboli tok peněz podnikem. Tento model je znám z jednoduchého účetnictví (nyní daňové evidence). Cash flow vlastně tvoří rozdíl ve stavu peněžních prostředků (na účtech a na hotovosti) na začátku a na konci účetního období. Podnik je potom „plusový“, pokud má kladný cash flow. Protože se zpravidla nepodaří podchytit některé položky, jako úroky, poplatky za platby, apod., je cash flow pokládán za správně sestavený, pokud vyjde +-1%.

U nás zatím cash flow není oficiálním účetním výkazem, ale je jen otázkou času, kdy se situace změní: normy ES jej totiž vyžadují, a proto se s ním operuje např. u již umíněných podniků se zahraniční majetkovou účastí.



Obr. Zrození výkazu peněžních toků a jeho sestavení na základě příjmů a výdajů peněz

Motivaci k cash flow jsme si již uvedli; mezi nejdůležitější důvody (resp. výhody) dále patří:

- analýza a hodnocení hospodaření (lépe vystihuje situaci v podniku než rozvaha a výsledovka),
- rozpočtování,
- struktura příjmů a výdajů je dobře zmapována,
- je snazší výběr variant rozvoje,
- provádí se pomocí něj ohodnocení tržní ceny podniku.

Informační systémy 1

Výpočet cash flow

Nejjednodušší výpočet cash flow vychází ze zisku (ztráty), ke kterému připočteme s kladným, resp. záporným znaménkem tyto položky:

- + odpisy
- výdaje na nákup investičního majetku (náhradou za odpisy)
- nezapočítané výdaje (tvorba rezerv, zálohy na nákup materiálu)
- + čerpání rezerv, materiál zaplacený zálohově
- pohledávky
- + závazky

a takto zkorigovaný účetní zisk nám přechází v cash flow.

Přitom je možné uvést podrobnější členění:

1. Provozní oblast.

Zisk, resp. ztráta z provozní činnosti, je analogický výsledku hospodaření středisek: dali jsme materiál do výroby a vyprodukovali jsme výrobky, odkud nám vyšel zisk nebo ztráta. Dále započteme:

- + odpisy
- + změna stavu krátkodobých pohledávek (Zde - a stejně i u ostatních položek, představujících změnu - znamená „+“ , že zvýšení se započítá s kladným znaménkem a snížení se záporným, zatímco symbol „-“ naopak.
- + změna stavu násob
- + změna zůstatku nákladů příštích období
- + - změna krátkodobých závazků
- + - změna výdajů rezerv
- + přijaté úroky a dividendy
- zaplacené úroky
- zaplacená daň z příjmu

2. Finanční oblast.

Sem patří:

- + příjmy z prodeje akcií
- + příjmy z prodeje dluhopisů
- + čerpání úvěru
- výplata dividend
- výdaje na nákup vlastních akcií (zejména při upisování)
- splátky úvěrů
- umoření vlastních dluhopisů

Informační systémy 1

3. Investiční oblast.

- + úhrady za cizí dluhopisy (po uplynutí jejich splatnosti)
- + příjem z prodeje cizích akcií
- + příjem z prodeje cizích dluhopisů
- + příjem z prodeje základních prostředků
- výdaje za nákup cizích akcií
- výdaje za nákup cizích dluhopisů
- Výdaje za nákup základních prostředků
- výdaje za nákup ostatních aktiv

Celkový cash flow pak dostaneme sečtením těchto tří kategorií. Pokud je však podnik menší, nečerpá úvěry a neobchoduje na burze, zcela mu postačí jednoduchý cash flow bez podrobného členění.

Příklad

Na závěr si uvedeme ještě poměrně obsáhlý příklad na výpočet cash flow.

1. Provozní oblast.

zisk z provozní činnosti	6400
odpisy	900
výnosy	<u>-1000</u>
	6300

změna (zvýšení) pohledávek	-2600
změna (snížení) zásob	1900
změna (snížení) závazků	<u>-500</u>
	5100

přijaté úroky	400
přijaté dividendy	400
vyplacené úroky	-540
zaplacená daň	<u>-1800</u>
	3560

zaplacené pokuty	-100
přijaté pokuty	<u>460</u>
	3920

V posledním řádku nám tak vyšel čistý cash flow z provozní činnosti.

2. Finanční oblast.

vydání nových akcií	500
dlouhodobé půjčky	500
splátky půjček	-180
vyplacené dividendy	<u>-2400</u>
	-1580

Informační systémy 1

3. Investiční oblast.

nákup akcií	-1100
nákup drobného majetku	-700
příjmy z prodeje aktiv	<u>40</u>
	-1760

Nyní všechny položky sečteme, přidáme ztrátu ze změn kursů zahraničních měn.

provozní činnost	3920
finanční činnost	-1580
investiční činnost	-1760
kursové ztráty	<u>-80</u>
	+500

Tento výsledek představuje čisté zvýšení finančních prostředků.

Vazba na ostatní účetní výkazy

Návaznosti cash flow na rozvahu a výsledovku ukážeme opět na příkladu.

Příklad

Mějme počáteční rozvahu

<i>Aktiva</i>		<i>Pasiva</i>	
Základní prostředky	370 000	Kmenové akcie	100 000
Odpisy	-122 000	Přednostní akcie	150 000
Fin. účasti v podnicích	90 000	Emisní ážio	50 000
Peníze	40 000	Rezervní fond	100 000
Pohledávky	105 000	Nezapl. daň z příjmu	40 000
Zásoby	160 000	Dluhopisy	150 000
Náklady příštích období	<u>7 000</u>	Závazky	<u>60 000</u>
Σ aktiv	650 000	Σ pasiv	650 000

a konečnou rozvahu

<i>Aktiva</i>		<i>Pasiva</i>	
Základní prostředky	530 000	Kmenové akcie	107 000
Odpisy	-150 000	Přednostní akcie	150 000
Fin. účasti v podnicích	80 000	Emisní ážio	93 000
Peníze	50 000	Rezervní fond	115 000
Pohledávky	107 000	Nezapl. daň z příjmu	90 000
Zásoby	180 000	Dluhopisy	160 000
Náklady příštích období	<u>6 000</u>	Závazky	<u>85 000</u>
Σ aktiv	800 000	Σ pasiv	800 000

Výsledovka nám vyjde takto:

tržby	1 900 000
pořizovací náklady	<u>-1 386 000</u>

Informační systémy 1

zisk z prodeje	514 000
odpisy	-51 000
ostatní náklady	<u>-259 700</u>
zisk před zdaněním	203 300
úroky	-9 300
daň	-90 000
ztráta z prodeje ZP	-15 000
zisk z prodeje podílů	<u>11 000</u>
čistý zisk	100 000

Odtud můžeme spočítat cash flow:

provozní zisk	203 300
odpisy	51 000
změna pohledávek	-2 000
změna zásob	-20 000
změna nákladů příš.obd.	1 000
změna závazků	<u>25 000</u>
cash flow	258 300

úroky	-9 300
zaplacená daň	-40 000
nákup ZP	-160 000
prodej ZP	15 000
prodej podílů	11 000
dluhopisy	10 000
výplata dividend	<u>-85 000</u>
cash flow celkem	0

Kontrolní otázky:

1. Co zachycuje účetnictví, jaké jsou jeho funkce?
2. Kdo všechno může vyžadovat účetní informace a za jakým účelem?
3. Jaké údaje by měl obsahovat účetní doklad faktura?
4. Jaký je vztah mezi aktivy a pasivy?
5. Jaký je vztah mezi syntetickými a analytickými účty?
6. Jaké účetní knihy v účetnictví používáme?
7. Co je to cash flow?

Úkoly k zamyšlení:

Zamyslete se nad tím, jaké znalosti, kromě finančního účetnictví, by měl mít vedoucí pracovník podniku a vedoucí softwarového projektu. Vaše návrhy zdůvodněte a podložte příklady z praxe, kde tyto dovednosti mohou (či musí) uplatnit.

Korespondenční úkol:

Účetní jednotka měla k 1.1.2004 následující stav majetku a zdrojů jeho krytí:

Základní kapitál	8 570 000	Stroje	6 400 000
Odběratelé	340 000	Software	100 000
Pokladna	50 000	Zásoby	1 110 000
Bankovní účty	1 500 000	Budovy	4 720 000

Informační systémy 1

Dodavatelé	650 000	Zaměstnanci	800 000
Dlouhodobý fin.majetek	500 000	Dlužné daně	400 000
Dlouhodobý bank.úvěr	3 100 000	Rezervní fond	200 000
Krátkodobý bank. úvěr	1 000 000		

Sestavte rozvahu k datu 1.1.2004

Shrnutí obsahu kapitoly

V této kapitole jste se seznámili se základními pojmy z oblasti účetnictví. Dozvěděli jste se proč účetnictví vedeme, kdo všechno účetní informace vyžaduje, z jakých prvků se účetnictví skládá. Poznali jste jeho principy a nástroje a měli byste být schopní se v oblasti finančního účetnictví lépe orientovat. Získané znalosti Vám umožní pochopit funkčnosti ekonomických IS.

13 Informační zdroje

- [1] KRÁL, J. *Informační systémy (specifikace, realizace, provoz)*. 1. vyd. Veletiny: SCIENCE. 1998. ISBN 80-86083-00-4.
- [2] DOHNAL, POUR. *Architektury informačních systémů v průmyslových a obchodních podnicích*. 1. vyd. Praha: EKOPRESS. 1997. ISBN 80-86119-02-5.
- [3] TVRDÍKOVÁ, M. *Zavádění a inovace informačních systémů ve firmách*. 1. vyd. Praha: GRADA. 2000. 110 str. ISBN 80-7169-703-6.
- [4] MOLNÁR, Z. *Efektivnost informačních systémů*. 2. rozš. vyd. Praha: GRADA. 2002. ISBN 80-247-0087-5.
- [5] ŘEPA, V. *Analýza a návrh informačních systémů*. 1. vyd. Praha: EKOPRESS. 1999. ISBN 80-86119-13-0.
- [6] VOŘÍŠEK, J. *Strategické řízení informačních systémů a systémová integrace*. 1. vyd. Praha: Management Press. 1999. ISBN 80-85943-40-9.
- [7] ŠEŠERA, MIČOVSKÝ, ČERVENĚ. *Datové modelování v příkladech*. 1. vyd. Praha: GRADA. 2001. ISBN 80-247-0049-2.
- [8] ARLOW, J., NEUSTADT, I. *UML a unifikovaný proces vývoje aplikací*. 1. vyd. Brno: Computer Press. 2003. ISBN 80-7226-947-X.
- [9] BOOCH, G., JACOBSON, I., RUMBAUGH, J. *The unified modeling language User guide*. Addison-Wesley. 1998. ISBN 0201571684.
- [10] BOOCH, G., JACOBSON, I., RUMBAUGH, J. *The unified modeling language Reference guide*. Addison-Wesley. 1998. ISBN 020130998X.
- [11] Coad, P., Yourdan, E.: *Modern structured analysis*. Prentice-Hall. Englewood Cliffs. New Jersey. 1989. ISBN 0135986249.
- [12] Yourdon, E.: *Structured Walkthroughs*, Prentice-Hall, 1989. ISBN 0138552894.
- [13] Coad, P., Nicola, J.: *Object oriented programming*. Prentice Hall. 1993. ISBN 013032616X.
- [14] Wolf, P., Kaluža, J., Maňasová, Š., Ostrožná, J.: *Projektování informačních systémů*, učební texty VŠB-TU Ostrava, 1994, 1. vydání, Ostrava, ISBN 80-7078-232-3.
- [15] Gane, Ch.: *Computer-Aided Software Engineering*, Prentice-Hall International, Englewood Cliffs, New Jersey, 1990, ISBN 0-13-172776-1.
- [16] KLIMEŠ, C. *Projektování informačních systémů I*. [CD-Rom]. Ostrava: Ostravská univerzita. Listopad 2003 - [cit. 2003-11-5].
KLIMEŠ, C., MELZER, J. *První elastický informační systém QI*. [CD-Rom]. Ostrava. Říjen 2002 - [cit. 2003-12-01].
- [17] VOŘÍŠEK, J. *MDIS – principy a konceptuální modely*. [online]. Praha: VŠE Praha. Listopad 2001. Dostupné na Internetu: <http://nb.vse.cz/~vorisek/>.
- [18] DC CONCEPT. *Vývojářské CD QI Builder*, verze 38.0 – 47.2 [CD-Rom]. Brno: DC Concept. Červen 2002 – leden 2004.
- [19] Kraval, I.: *Objektové modelování a UML v praxi 2000*. [e-kniha]. 2000. Dostupné na Internetu: <http://www.objects.cz/>.
- [20] OMG's UML homepage. [on-line]. Dostupné na Internetu: <http://www.omg.org/uml/>

- [Carda et al. 2003] Carda A., Merunka V., Polák J.: *Umění systémového návrhu*, Grada 2003, ISBN 80-247-0424-2
- [Hall et al. 2004] Hall J. et al.: *Accounting Information Systems 3rd edition*, South-Western Publishing, 2004, ISBN 0538877960. (spoluautoři kapitoly 4. – System Development Activities)
- [Knott et al. 2000] Knott, Roger P.; Merunka, Vojtěch; Polák, Jiří: *Process Modeling for Object Oriented Analysis using BORM Object Behavioral Analysis*, in Proceedings of Fourth International Conference on Requirements Engineering ICRE 2000, Chicago 2000. IEEE Computer Society Press ISBN 0-7695-0565-1
- [Knott et al. 2003] Knott, Roger P.; Merunka, Vojtěch; Polák, Jiří: "*The BORM methodology: a third-generation fully object-oriented methodology*", Knowledge-Based Systems 3(10) 2003, Elsevier Science Publishing, New York.
- [Liu et al. 2005] Liu L., Roussev B. et al: *Management of the Object-Oriented Development Process*, Virgin Island 2005, ISBN 1-59140-605-6 (spoluautoři kapitoly 15. – The BORM Methodology)
- [Merunka et al. 2000] Merunka, V.; Polák, J.; Kofránek J.: *Úvod do metody BORM*, ve sborníku konference Objekty 2000, ISBN 80-213-0682-3
- [Merunka 2001] Merunka V.: *Zkušenosti s objektovou analýzou a návrhem*, ve sborníku konference Objekty 2001, ISBN 80-213-0829-X
- [Merunka 2002A] Merunka V.: *Objekty v databázových systémech*, skriptum ČZU Praha, Praha 2002. ISBN 80-213-0318-2
- [Merunka 2002B] Merunka V.: *Řídící a podpůrné procesy v objektově orientované tvorbě softwaru*, ve sborníku konference Objekty, listopad 2002 Praha, ISBN 80-213-0947-4
- [Merunka 2003] Merunka V.: *Objektový databázový systém Gemstone*, sborník konference OBJEKTY 2003. Ostrava 2003. ISBN 80-248-0274-0
- [Merunka et al. 2004] Merunka V., Pergl R., Pícka M., *Objektově orientovaná tvorba software*, ČZU Praha, 2004, ISBN 80-213-1159-2
- [Abadi 1996] Abadi M., Cardelli L.: *A Theory of Objects*, Springer 1996, ISBN 0-387-94775-2
- [Agha et al. 1994] Agha Gul, Hewitt Carl: *Actors - A Conceptual Foundation for Cocurrent Object-Oriented Programming*, pp 49-74, Research in OOP 1994, MIT Press ISBN 0-262-19264-0
- [Ambler 1997] Ambler Scott: *Building Object Applications That Work, Your Step-By-Step Handbook for Developing Robust Systems Using Object Technology*, Cambridge University Press/SIGS Books, 1997, ISBN 0521-64826-2
- [Ambler 1998] Ambler, Scott W.: *Process Patterns - Building Large-Scale Systems Using OO Technology*, Cambridge University Press - Managing Object Technology Series 1998, ISBN 0-521-64568-9
- [Ambler 1999] Ambler, Scott W.: *More Process Patterns - Delivering Large-Scale Systems Using OO Technology*, Cambridge University Press - Managing Object Technology Series 1999, ISBN 0-521-65262-6
- [Ambler et al. 2004] Ambler Scott, Beck Kent: *Object Orientation – Bringing data professionals and application developers together*, <http://www.agiledata.org/essays/objectOrientation101.html>, prosinec 2004
- [Ambler 2005] Scott W. Ambler, John Nalbone, and Michael Vizdos, *The Enterprise Unified Process*, Prentice Hall PTR, 2005, ISBN 0-13-191451-0
- [AP 2001] *AP manifesto*, <http://www.agilealliance.org>, December 2004
- [Arlow 2003] Jim Arlow and Ila Neustadt: *UML and the Unified Process: Practical Object-Oriented Analysis and Design*. Addison Wesley. 2003
- [Barry 1998] Barry D.: *The Object Database Handbook: How to Select, Implement, and Use Object-Oriented Databases*, ISBN 0471147184

Informační systémy 1

- [Bartoška et al. 2004] Bartoška Jan, Adámková Milena, Hnátková Běla, *Tvorba softwaru nad biometrickou strukturou dat*, ve sborníku konference Objekty 2004
- [Beck 1989] Kent Beck and Ward Cunningham: A Laboratory for Teaching Object-Oriented Thinking. In *Proceeding of OOPSLA 89*. SIGPLAN Notices, Vol.24, No. 10, pp 1-6
- [Beck 1997] Beck K.: *Smalltalk Best Practice Patterns*, Prentice Hall 1997, ISBN 0-13-476904-X
- [Beck 2002] Beck K.: *Extrémní programování* (český překlad) Grada 2002, ISBN 80-247-0300-9
- [Beck 2003] Beck K.: *Agile Database Techniques - Effective Strategies for the Agile Software Developer*, John Wiley & Sons; 2003, ISBN 0471202835
- [Bellin et al. 1997] Bellin, David; Simone, Susan Suchman: *The CRC Card Book*, Addison-Wesley 1997 ISBN: 0201895358
- [Bertino et al. 1995] Bertino E., Martino L.: *Object-Oriented Database Systems - Concepts and Architectures*, Addison Wesley 1995, ISBN 0-201-62439-7
- [Biermann 1990] Biermann Alan W.: *Great Ideas in Computer Science*, MIT Press 1990, ISBN 0-262-02302-4
- [Blaha et al. 1995] Blaha M., Premerlani M.: *Object-Oriented Modeling and Design for Database Applications*, Prentice Hall 1995, ISBN 0-13-123829-9
- [Booch 1994] Booch Grady: *Object-Oriented Analysis and Design with Applications*, Second Edition, Benjamin Cummings 1994, ISBN 0-8053-5340-2
- [Booch 1996] Booch, Grady: *Object Solutions: Managing the Object-Oriented Project*, Addison-Wesley, 1996, ISBN 0805305947
- [Buchalcevová 2005] Buchalcevová A.: *Metodiky vývoje a údržby informačních systémů*, Grada 2005, ISBN 80-247-1075-7
- [Burlison 1999] Burlison D.: *Inside the Object Database Model*, CRC Press 1999, ISBN 0-8493-1807-6
- [Catell 1998] Catell R. G.: *The Object Data Standard: ODMG 3.0*, ODMG 1998, ISBN 1558606475
- [Choppy 2004] Choppy C.: *Improving Use-Case Based Requirements*, in proceedings of FASE – Fundamental Approaches to Software Engineering 2002, pp. 244-261, Springer ISSN 0302-9743
- [Coad 1990] Coad, P., Yourdon E., *Object Oriented Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1990.
- [Coad 1993] Peter Coad and Jill Nicola: *Object-Oriented Programming*, Yourdon Press, 1993
- [Coad 1995] Peter Coad , David North and Mark Mayfield: *Object Models: Strategies, Patterns and Applications*, Yourdon Press, 1995
- [Coleman et al. 1994] Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H., Hayes, F. Jeremaes, P., *Object-Oriented Development - The Fusion Method*, Englewood Cliffs, N.J.: Prentice-Hall, 1994.
- [Darnton 1997] Darnton, Geoffrey, Darnton, Moksha: *Business Process Analysis*, International Thomson Publishing 1997 ISBN: 1861520395
- [Davis 1993] Davis, A.: *Software Requirements - Objects, Functions and States* , Prentice Hall 1993
- [Derr 1995] Derr K.W.: *Applying OMT - A Practical Guide to Using the Object Modelling Technique*, Sigs Books 1995, ISBN 1-884842-10-0, Prentice Hall 1995 ISBN 0-13-231390-1

Informační systémy 1

- [Doskočil 2004] Doskočil Tomáš, Merunka Vojtěch, *Zkušenosti z výuky s objektovým databázovým systémem Gemstone/S v předmětu DBII na KII PEF ČZU*, ve sborníku konference Informatika o výuce informatiky, Zlín 2004, ISBN 80-7302-066-1
- [Drbal 1996] Drbal Pavel, *Proudy v objektově orientovaných metodikách*, ve sborníku konference Tvorba softwaru 1996, Ostrava.
- [Drbal 2002] Drbal P.: *Extrémní programování a metodický přístup*, ve sborníku konference Tvorba softwaru 2002, ISBN 80-85988-74-7
- [Ewusi 2003] Ewusi K.: *Software Development Failures*, MIT Press 1993, ISBN 0-262-05072-2
- [Fowler 1999] Fowler M., Kendall S., *UML Distilled (2nd Edition)*, 1999, Addison-Wesley ISBN: 0-201-65783-X
- [Gamma et al. 1995] Gamma, E.; Helm, R.; Johnson, E.R.; Vlissides, J.: *Design Patterns - Elements of Reusable Object Oriented Software*, Addison-Wesley, New York, USA, 1995. ISBN 0201633612
- [Gemstone 2004] *Gemstone Object Server - documentation & non-commercial version download*, <http://www.gemstone.com>, <http://smalltalk.gemstone.com>, prosinec 2004
- [Goldberg 1995] Goldberg Adele, Kenneth Rubin S.: *Succeeding with Objects - Decision Frameworks for Project Management*, Addison Wesley 1995, ISBN 0-201-62878-3
- [Graham et al. 2002] Graham, Ian; Simons, Anthony J H: *30 Things that Go Wrong in Object Modeling with UML 1.3*, University of Sheffield & IGA Ltd. <http://www.iga.co.uk>, květen 1992
- [Hammer et al. 1994] Hammer, M. – Stanton, A.: *The Reengineering Revolution*, Collins 1994
- [Henderson-Sellers 1994] Henderson-Sellers B, Edwards JM.: *MOSES - A Second Generation Object-Oriented Methodology*, pp 68-73, Object Magazine 4(3) 1994
- [Henderson-Sellers 1998] Henderson-Sellers, B.; Bulthuis, A.: *Object-Oriented Metamethods*, Springer-Verlag New York, 1998, ISBN 0-387-98257-4
- [Hopkins 1992] Hopkins T.: *Animating an Actor Programming Model*, Computer Science Dept. University of Manchester 1992
- [HOPL 1988] Wexelblat Richard L.: *History of Programming Languages*, Academic Press Pennsylvania 1988, ACM Monographic Series, from the ACM SIGPLAN History of Programming Languages Conference, June 1-3, 1988, ISBN 0-12-745040-8
- [Hruška 1995] Hruška T.: *Objektově orientované databázové technologie*, sborník konference Tvorba softwaru 1995, ISBN 80-901751-3-9
- [Jacobson et al. 1992] Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G., *Object-Oriented Software Engineering - A Use Case Driven Approach*, 1992, Addison Wesley ISBN 0-201-54435-0
- [Jacobson 1985] I. Jacobson: *Concepts for Modelling Large Real-Time Systems*, Doctoral Dissertation, Department of Computer Systems, The Royal Institute of Technology, August 1985.
- [Jacobson et al. 1999] Jacobson, G. Booch, J. Rumbaugh: *The Unified Software Development Process*, Addison Wesley, 1999, ISBN 0-2015-7169-2
- [Kelly 1997] Kelly, S.: *GOPRR Metamodel, appendix of doctor thesis „Towards a Comprehensive MetaCASE and CAME Environment: Conceptual, Architectural, Functional and Usability Advances in Metaedit+“*, Ph.D. Thesis, Jyväskylä University, 1997
- [Khodorkovsky 2002] Khodorkovsky V. V.: *On Normalization of Relations in Databases*, Programming and Computer Software 28 (1), 41-52, January 2002, Nauka Interperiodica.
- [Kotonya 1999] Kotonya G. – Sommerville I.: *Requirements Engineering - Processes and Techniques*, 1999, J. Wiley and Sons

Informační systémy 1

- [Kroha 1995] Kroha P.: *Objects and Databases*, McGraw Hill, London 1995, ISBN 0-07-707790-3.
- [Kruchten 2000] Kruchten Philippe: *The Rational Unified Process – An Introduction*, Addison-Wesley, 2000, ISBN 0201707101
- [Lacko 1996] Lacko B.: *Komparativní analýza strukturovaného a objektově orientovaného přístupu*, sborník konference OBJEKTY 1996. ČZU Praha 1996 str.69-76
- [Larsson 2002] Larsson M., Crncovic I.: *Building Reliable Component-Based Software Systems*, Artech House 2002, ISBN 1-58053-327-2
- [Loomis 1994] Loomis M.: *ODBMS - Hitting the Relational Wall*, pp 56-60, JOOP January 1994
- [Loomis et al. 1994] Loomis M., Chaundri A.: *Object Databases in Practice*, Prentice Hall 1994, ISBN 013899725X
- [Martin et al. 1995] Martin James, Odell James J.: *Object-Oriented Methods - A Foundation*, Prentice Hall 1995, ISBN 0-13-63856-2
- [Meyer 1997] Bertrand Meyer: *Object Oriented Software Construction*, Prentice Hall, 1997, ISBN 0136291554
- [Molhanec 2004] Molhanec M.: *Několik poznámek k porozumění objektového paradigmatu*, ve sborníku konference Objekty 2004
- [Nierstrasz et al. 1990] Nierstrasz Oscar, Papathomas Michael: *Viewing Objects as Paterns of Communicating Agents*, pp 255-266 , in Object Management 1990, Centre Universitaire d' Informatique - Universite de Geneve, also at at <ftp://cui.unige.ch/OO-articles/viewingObjectsAsPatterns.ps.Z>
- [Nootenboom 2004] Nootenboom Henk Jan: *Nut's - a online column about software design*. <http://www.sum-it.nl/en200239.html>, prosinec 2004
- [OMG 2002a] Object Management Group (OMG). *Meta Object Facility (MOF) specification – version 1.4*. 2002. <http://www.omg.org>
- [OMG 2002b] Object Management Group (OMG). *Model Driven Architecture (MDA) specification – Version 1.4*. 2002. <http://www.omg.org>
- [OMG 2003] Object Management Group (OMG). *OMG Unified Modeling Language Specification - Version 1.5*. 2003. <http://www.omg.org>
- [Partridge 1996] Partridge C.: *Business Objects - Reengineering for Reuse*, Butterworth-Heinemann 1996, ISBN 07506-2082X
- [Rashid 2004] Rashid A.: *Aspect Oriented Database Model*, Springer 2004, ISBN 3-540-00948-5
- [Riecken 1994] Riecken D.: *Software Agents*, pp 18-147, Communications of ACM, 37 (1994)
- [Rubin et al. 1992] Rubin K.S., Goldberg A.: *Object Behavioural Analysis*, pp 48-62 Communications of the ACM 35(9) 1992
- [Rubin et al. 1993] Rubin K.S, Goldberg A.: *Object Behavior Analysis*, pp 48-57 Communications of the ACM, vol 35 no 9. 1993
- [Rubin et al. 1994] Rubin Kenneth S, McClaghry Patrick, Pellergini David.: *Modelling Rules using Object Behavior Analysis and Design*, pp 63-68, Object Magazine 4(3) 1994
- [Rumbaugh et al. 1991] Rumbaugh James, Blaha Michael, Premerlani William, Eddy Frederic, Lorensen William: *Object-Oriented Modelling and Design*, Prentice Hall 1991, ISBN 0-13-630054-5
- [Rumbaugh et al. 1998] Rumbaugh, I. Jacobson, G. Booch: *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1998, ISBN 0-2013-0998-X
- [Rumbaugh 1996] James Rumbaugh: *OMT Insights*. SIGS Book. 1996.
- [Shiver et al. 1987] Shriver Bruce, Wegner Peter.: *Research Directions in OOP*, MIT Press 1987, ISBN 0-262-19264-0

Informační systémy 1

- [Shlaer 1992] Mellor Stephen, Shlaer Sally: *Object Lifecycles: Modeling the World in States*, ISBN 0136299407
- [Shlaer 1989] Sally Shlaer and Stephen. P. Mellor: *Object-Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press, 1989
- [Shoham 1993] Shoham Y.: *Agent Oriented Programming*, pp 51-92, Artificial Inteligence 60 (1993)
- [Silbershatz 2004] Silbershatz A et al.: *Database Systems Concepts – 4th Edition*, McGraw Hill 2004, ISBN 0-07-228363-7
- [Taylor 1995] Taylor, David A.: *Business Engineering with Object Technology*, John Wiley 1995 ISBN: 0471045217
- [UML 2004] *The Unified Modeling Language Documents*, <http://www.uml.org> , prosinec 2004
- [Vaniček 2004] Vaniček J.: *Měření a hodnocení jakosti informačních systémů*. Praha: ČZU PEF, 2004, ISBN 80-213-1206-8
- [Wai 1992] Wai Y. Mok, Yiu-Kai Ng and David W. Embley, *An Improved Nested Normal Form for Use in Object-Oriented Software Systems*. Proceedings of the 2nd International Computer Science Conference: Data and Knowledge Engineering: Theory and Applications, pp. 446-452, Hong Kong, December 1992.
- [Warmer 1998] Jos Warmer and Anneke Kleppe: *The Object Constraint Language: Precise modeling with UML*. Addison-Wesley, 1998.
- [Wilkie 1993] Wilkie G.: *Object-Oriented Software Engineering*, Addison Wesley 1993, ISBN 0-201-6276-1
- [Wirfs-Brock 1990] Rebeca Wirfs-Brock, Brian Wilkerson and Lauren Wiener: *Designing Object-Oriented Software*, Prentice Hall, 1990
- [Yonghui et al. 2001] Yonghui Wu, Zhou Aoying: *Research on Normalization Design for Complex Object Schemes*, Info-Tech and Info-Net, vol 5. 101-106, Proceedings of ICII 2001, Beijing.
- [Yourdon 1994] Yourdon E.: *Object-Oriented System Design - An Integrated Approach*, Prentice Hall 1994, ISBN 0-13-176892-1
- [Yourdon 1995] Yourdon E.: *Mainstream Objects - An Analysis and Design Approach for Business*, Prentice Hall 1995 ISBN 0-13-209156-9
- [Yourdon 1989] Yourdon E.: *Modern Structured Analysis*. Yourdon Press/Prentice-Hall. New York 1989.
- [Yourdon 1991a] Peter Coad and Ed Yourdon: *Object-Oriented Analysis*, Yourdon Press, 1991
- [Yourdon 1991b] Peter Coad and Ed Yourdon: *Object-Oriented Design*, Yourdon Press, 1991
- [Zahir et al. 1997] Tari Zahir, Stokes John, Spaccapietra Stefano: *Object Normal Forms and Dependency Constraints for Object-Oriented Schemata*, ACM Transactions on Database Systems 513-569, Vol 22 Issue 4, December 1997.

13.1 Slovníček pojmů

<i>Pojem</i>	<i>anglický překlad</i>	<i>vysvětlení</i>	<i>oblast použití</i>
agilní přístup	agile approach	Varianta →evolučního životního cyklu tvorby softwaru. Pro agilní přístupy je typická orientace na vlastní tvorbu programu na úkor analytických a dokumentačních aktivit.	všechny fáze BORMu
agregace objektů	object aggregation	Zvláštní případ →skládání objektů, kdy je existence a →identita objektu závislá na jiném objektu, který je v něm skládán. (Změna skládaného objektu by v systému způsobila i změnu skládajícího objektu.)	CM, SM
aktivita	activity	Aktivita představují jednotlivé části chování →business objektů tak, jak byly rozpoznány technikou →OBA. V →procesních diagramech se pomocí vybraných aktivit provádějí →přechody mezi →stavy objektů. Aktivita různých objektů mezi sebou komunikují (→komunikace). Z aktivit se odvozují →metody.	BM
aktor	actor	Aktor je specifický název objekt, který je schopen →delegovat zprávy. (→čistý objektový programovací jazyk)	CM, SM
aktorový model výpočtu	actor computational model	Varianta objektově orientovaného chování systému, kde dochází k →delegování →zpráv mezi objekty. (→aktor, čistý objektový programovací jazyk)	CM, SM
analýza	analysis	Souhrnný název pro fáze tvorby systému, při kterých se zjišťuje zadání systému a modeluje se jeho požadovaná funkčnost. Po analýze následuje →návrh. V případě →spirálního způsobu tvorby softwaru se analýza kryje s →expansí systému. V →BORMu se analýza skládá z fáze →strategické analýzy, →úvodní analýzy a →podrobné analýzy.	BM, CM
architektura systému	system architecture	Komplexní model systému, který se skládá z několika vrstev, které jsou samy o sobě modelem zaměřeným na jednu stránku systému. Je to například vrstva →procesů, logický model (popis dat, funkcí a pravidel) a model komponent (např. softwarové aplikace nebo organizační struktury). Prvky z různých vrstev architektury jsou mezi sebou vzájemně provázány, a proto je vhodné při každé změně provést rozbor dopadů na prvky jiných úrovní. (→konvergenční inženýrství, BPR)	všechny fáze BORMu
AS-IS asociace	AS-IS association	První fáze →BPR. AS-IS znamená „tak, jak to je“ . Vztah mezi objekty, který vyjadřuje, že jeden objekt v modelu potřebuje nebo používá nebo k němu jinak přísluší druhý objekt. Tento vztah se nesmí vykládat jako vazba mezi záznamy v tabulkách →relačních databázích a na rozdíl od jiných metod se v BORMu nepoužívá pro →softwarové modelování. Z asociací lze odvodit vazbu →skládání objektů.	BM, BPR BM, CM
asynchronní zpráva, paralelní zpráva	asynchronous message, parallel message, fork message	→Zpráva, na jejíž výsledek →objekt, který zprávu poslal, nečeká. Objekt tedy poslanou zprávou spustil vykonávání nějakých aktivit a současně dál pokračuje ve svých aktivitách. Tento typ zpráv je v →čistých objektových jazycích základem paralelního programování a také umožňuje →delegování a →závislost mezi objekty. (→synchronní zpráva)	CM, SM
atribut, vlastnost	attribute	Vlastnosti →objektu, kterými se objekt prezentuje svému okolí v systému. (Například „barva“, „váha“, „cena“ atd.) V →BORMu by atributy neměly být chápány jen jako data uvnitř objektů, protože atributy lze realizovat i jako →metody, které dané hodnoty poskytují. (→zapouzdření, protokol)	všechny fáze BORMu
automat	automaton (mn.č. „automata“), finite state machine	Zařízení, které přijímá vstupní informace a jako odpovědi na ně vydává informace výstupní, které závisí jednoznačně na vnitřním stavu tohoto zařízení a na vstupní informaci. Během své činnosti automat vykonává →přechody čímž mění svoje →stavy. Teorie automatů je v →BORMu základem pro modelování chování →objektů v →procesech. (→ORD).	všechny fáze BORMu

Informační systémy 1

BM	BM	Zkratka pro →business modelování.	BM
BO	BO	Zkratka pro →business objekt.	BO
BORM	BORM	Zkratka pro „Business Object Relation Modeling“. BORM je metodika vyvinutá v rámci výzkumného projektu VAPPIENS Britské rady (British Council) ve spolupráci s Deloitte&Touche. Základní filosofií BORMu je plná podpora →objektového přístupu, využívání →procesního modelování pro →získávání požadavků a myšlenky →konvergenčního inženýrství.	všechny fáze BORMu
BPR – reinženýring business procesů	BPR – Business Process Reengineering	Přehodnocení a rekonstrukce →procesů za účelem jejich zdokonalení. Tato činnost je někdy důležitá pro nalezení správného zadání informačního systému. (→ <i>získávání zadání</i>) Provádí se ve dvou fázích: V první etapě →AS-IS se podrobně modeluje stávající stav procesů. Po vyhodnocení výsledků první etapy se přistupuje k druhé etapě →TO-BE. Zde se navrhuje nové procesy a provádí se návrh nové organizační struktury. Pro podklady na BPR se používá technika →OBA.	BM, BPR
business inženýrství	business engineering	Inženýrský obor, který se zabývá analýzou, modelováním a návrhem organizačních a řídicích struktur. (→ <i>BPR</i>) Některými autory je považován za součást →informačního inženýrství. Podle BORMu jsou tyto aktivity potřebné pro získání správných požadavků na informační systém. (→ <i>získávání požadavků</i>)	BM, BPR
business modelování	business modeling	První etapa BORMu, kdy se vytváří model →procesů systému tak, aby došlo k rozpoznání zadání problému a případnému přepracování stávajících procesů a nebo organizační struktury za účelem lepší implementace softwaru. (→ <i>BPR</i>) Zahnuje fáze →strategické analýzy a →úvodní analýzy.	BM, BPR
business objekt	business object	Objekt reálného světa, který slouží k analýze a popisu zadání systému. (→ <i>participant</i>)	BM
business proces	business process	Business procesy jsou východiskem pro funkční popis podniku nebo organizace. (→ <i>proces, BPR</i>)	BM, BPR
C# („cis“)	C# („c-sharp“)	Jazyk z dílny Microsoftu. Velmi podobný jazyku →Java.	SM
C++ („cé plus plus“)	C++	→Smíšený programovací jazyk. Patří mezi složité programovací jazyky, ale je v něm možné dosáhnout efektivního kódu. Jeho první verze pocházejí z 80. let.	SM
CASE – nástroje pro podporu softwarového inženýrství	CASE – Computer Aided Software Engineering	Programy, které pomáhají analytikům a vývojářům postupovat podle nějaké metodiky. Správný CASE by měl mít podporu týmového vývoje, nástroje pro práci s →diagramy, možnost tvorby reportů a analýz nad projektovými daty a generovat kód. CASE nemusejí být pouze pro metody →softwarového inženýrství, ale i pro →business inženýrství. Pokud podporuje →metamodelování (např. Proforma Workbench, Metaedit), tak lze CASE uživatelsky „nastavovat“ a „vylepšovat“ jím podporovanou metodiku. (→ <i>metamodelování</i>)	všechny fáze BORMu
cíl	goal	V kontextu →BPR to je součást detailního popisu →aktivity. Cíle vymezují, kam má podnik směřovat. (Například ziskat dominantní postavení na trhu nebo ve vybraném tržním segmentu.) Jejich znalost je důležitá pro rozhodování nad souvislostmi zadání informačního systému, →podnikových procesů s dalšími atributy organizace. (→ <i>získávání zadání</i>)	BM, BPR
CM	CM	Zkratka pro →konceptuální modelování.	CM
CO	CO	Zkratka pro →konceptuální objekt.	CO
CORBA	CORBA	Standard pro práci s objekty v distribuovaných softwarových systémech. Je doporučován sdružením →OMG a je podporován v →objektových databázích. (→ <i>objektově relační databáze, Gemstone</i>)	SM
CSF	CSF	Zkratka pro „Critical Success Factor“ (→ <i>faktor</i>).	BM, BPR
činnost, pracovní činnost	real activity	V kontextu →BPR to je součást detailního popisu →aktivity. Je to konkrétní popis týkající se plnění jednoho pracovního úkolu. (Například „zajištění pracoviště“ nebo „oprava vozidla“). Jedna aktivita typicky obsahuje více činností. (→ <i>BPR</i>)	BM, BPR

Informační systémy 1

čistý objektový programovací jazyk	Environment pure object-oriented programming language (EPOL)	Programovací jazyk, který důsledně podporuje vlastnosti objektového přístupu. Takový jazyk není úpravou nějakého neobjektového jazyka. Ve srovnání se →smíšenými jazyky je v nich objektové programování snadnější – například dovolují programovat aplikaci i za jejího chodu. Odhaduje se, že čistý jazyk potřebuje asi 2× až 5× méně příkazů na jeden →funkční bod. Mezi čisté jazyky patří například →Smalltalk.	SM
databáze, SŘBD – systém řízení báze dat	database, DBMS – database management system	Software, který dovoluje uchovávat a vydávat data pro potřeby uživatelů a nebo jiných systémů k němu připojených. Databáze je důležitou součástí většiny →informačních systémů. Podle principu činnosti se člení na síťové, →relační, →objektově relační a →objektové.	SM
datový tok	data flow	Data, která si objekty vyměňují při →komunikacích nebo posílání →zpráv. Rozlišují se →parametry zprávy a →návratové hodnoty.	všechny fáze BORMu
dědičnost	inheritance	Vazba mezi →softwarovými objekty, pomocí které lze programovat nové objekty (potomky) s využitím již existujících objektů (předků). Nové objekty jsou potom v programu schopny využívat →metody svých předků. Ve →smíšených programovacích jazycích je dědičnost jediná možnost, jak zajistit →polymorfismus mezi objekty. (→ <i>hierarchie typů</i>)	SM
delegování	delegation	Zpracování →zprávy objektem tím způsobem, že objekt, který přijal zprávu, ji nebude zpracovávat a vyvolá přenesení této zprávy na jiný objekt, který ji již dokáže zpracovat. Výpočet se pak dál odehrává jen u delegovaného objektu a objekt, který zprávu od sebe delegoval se ho již neúčastní (ani mu nemusí být předán výsledek). V případě, že implementační prostředí tuto vazbu nepodporuje, musí se ve fázi →softwarového modelování nahrazovat. (→ <i>smíšený programovací jazyk</i>)	CM, SM
deliverable design diagram	deliverable design diagram	Synonymum pro →výrobek. Synonymum pro →návrh.	SM, BM CM, SM
diagram	diagram	Schéma, které popisuje pomocí značek a dalších dle standardu dohodnutých způsobů kreslení nějaký model. V →BORMu se diagramy označují jako →ORD.	všechny fáze BORMu
esenciální objekt	essential object	Synonymum pro →business objekt.	BM
evoluční způsob tvorby softwaru, evoluční model	evolutionary development lifecycle	Varianta životního cyklu tvorby softwaru. Je založen na myšlence postupného vývoje po malých částech, kdy může docházet k doplňování požadavků od uživatele. Tento způsob je vhodný k využití →objektového přístupu. (→ <i>sekvenční model, →agilní přístup</i>)	všechny fáze BORMu
expanse	expansion	První část cyklu →spirálního modelu. Zde dochází ke hromadění informací, potřebných pro pochopení zadání a vytvoření aplikace. Expanze končí dokončením →konceptuálního modelu, který na logické úrovni reprezentuje požadované zadání a formálně popisuje řešený problém. (→ <i>konsolidace, analýza</i>)	BM, CM
faktor, kritický faktor pro úspěch	CSF – critical success factor	V kontextu →BPR to je součást detailního popisu →aktivity. Faktor je činitel, který má zásadní vliv na podnikatelský úspěch. (Je to například „existence podnikové informační strategie“, „hodnocení na základě měření“ nebo „dodržování standardů“.) Jejich znalost je důležitá pro rozhodování nad souvislostmi zadání informačního systému, podnikových →procesů s dalšími atributy organizace. (→ <i>získávání zadání</i>)	BM, BPR
funkce systému	system function	Požadovaná funkce je nejjednodušším slovním popisem požadovaných →procesů v systému podle →OBA. Pro pozdější bezproblémovou komunikaci analytiků a zadavatelů je vhodné do tohoto seznamu zahrnout a zvlášť vyznačit i funkce, které popisují, co se modelovat nebude. Ze seznamu funkcí se odvozuje seznam →scénářů.	BM, OBA

Informační systémy 1

funkční body	function points, feature points	→Měření, které slouží k odhadování pracnosti vyvíjeného softwaru metodou výpočtu funkčních bodů, které se vypočítají z rozsahu požadavků na systém (např. počet datových struktur, uživatelských vstupů a výstupů, ...). Z výsledného počtu funkčních bodů lze odhadnout velikost budoucího programu, protože pro různé →programovací jazyky je na základě statistických měření známo, kolik příkazů je třeba na pokrytí jednoho funkčního bodu. Metoda „feature points“ je variantou této metody pro případ objektové tvorby softwaru.	BM, CM
Gemstone	Gemstone	Představitel →objektového databázového systému. Databázový systém Gemstone používá programovací jazyk →Smalltalk nebo →Java. Podporuje standard →CORBA.	SM
generalizace-specializace	generalization-specialization	Synonymum pro →hierarchii typů.	CM
generický model	generic model	Generický model je takový model, který v sobě kombinuje vlastnosti modelu s vlastnostmi →metamodulu, což může usnadnit návrh systému. (→ <i>metamodelování, znovupoužitelnost, návrhové vzory</i>)	všechny fáze BORMu
hierarchie typů	type hierarchy	Tato vazba vyjadřuje vztahy mezi →protokoly objektů. To znamená, že objekty nadtypu jsou →polymorfni s objekty podtypu. K upřesnění vazby lze uvést seznam →zpráv, které se polymorfismu týkají. Tato hierarchie objektů vzniká z hierarchie →JE-JAKO a později je z ní odvozována hierarchie →dědičnosti mezi objekty.	CM
identita objektů	object identity	Spolu s →polymorfismem a →zapouzdřením základní vlastnost →objektů, díky které jsou různé objekty v jednom systému nezaměnitelné ani tehdy, mají-li shodná data a nebo →metody. V některých konkrétních softwarových prostředích (→ <i>smíšené programovací jazyky</i>) je ale třeba identitu zvlášť zajišťovat. Například v →relačních databázích musí mít objekty pro vzájemné rozlišení navíc zvláštní údaj nazývaný primární klíč (neboli index). V →objektových databázích to není potřeba.	všechny fáze BORMu
implementace	implementation	V této fázi se vytváří (programuje, sestavuje či generuje z CASE nástroje), testuje a předává uživateli požadovaný software. Při →spirálním způsobu tvorby softwaru je součástí →konsolidace systému.	SM
informační inženýrství	information engineering	Inženýrský obor, který podle některých autorů zahrnuje →softwarové inženýrství a →business inženýrství. Jedná se o disciplínu, která nahlíží komplexně na organizační a řídicí struktury a →informační systémy. (→ <i>sociotechnický systém, konvergenční inženýrství</i>)	všechny fáze BORMu
informační systém	information system	Systém sloužící k poskytování informací a sestavený podle nějakého zadání složený z hardwarových, softwarových a organizačních prvků.	všechny fáze BORMu
instance	instance	Objekt, který má svoje vlastnosti (= strukturu dat a metody) popsané v nějaké →třídě. Všechny instance téže třídy tedy mají stejné metody a stejnou strukturu dat a liší se mezi sebou jen konkrétním obsahem těchto dat. Instance se odvozují z →business objektů.	CM, SM
iterativní způsob tvorby softwaru, iterativní model	iterative development lifecycle	Způsob vývoje softwaru, kdy se v případě potřeby navrácí do počátečních fází vývoje. Pro upřesnění zadání se mohou vytvářet →prototypy. Jeho variantou je →spirální model. (→ <i>vodopádový model</i>)	všechny fáze BORMu
Java („džáva“)	Java	→Smíšený programovací jazyk. Jeho první verze pocházejí z 80. let, ale rozšířil se až ve druhé polovině 90. let s nástupem WWW. Je podobný →C++, ale je jednodušší a podporuje více objektových vlastností. Podle některých autorů je považován za →čistý objektový programovací jazyk.	SM

Informační systémy 1

Je-Jako	IS-A	Vztah hierarchie objektů. Objekty na nižší úrovni této hierarchie jsou prvky domény, která je podmnožinou domény objektů vyšší úrovně. Tento vztah se v →BORMu nesmí vykládat jako →dědičnost mezi →softwarovými objekty, protože je na dědičnost postupně transformován přes →hierarchii typů.	BM
klient	client	Objekt, na kterém jsou →závislé jiné objekty (→ <i>server</i>).	CM, SM
klient	client	Softwarová aplikace, která využívá data nebo služby jiné aplikace na jiném počítači v síti. (→ <i>server</i>)	SM
kolekce, sada	collection	Synonymum pro →množinu objektů.	CM, SM
komunikace	communication	Rízení →aktivit →business objektů. Komunikace je abstrakce →zpráv mezi objekty. V pozdějších fázích modelování se zprávy z komunikací odvozují.	BM
konceptuální modelování	conceptual modeling	Druhá etapa →BORMu, která zahrnuje tvorbu modelu z →konceptuálních objektů. Takový model stojí napůl cesty mezi zadáním a řešením, kdy je problém považován za rozpoznáný a nalézá se v něm část k softwarovému řešení. V konceptuálních modelech se na systém nazírá jako na ideální svět počítačových objektů. Mezi modelované vazby patří například →asociace, →skládání, →závislost, →delegování a →hierarchie typů. Zahrnuje fáze →podrobné analýzy a →úvodního návrhu.	CM, UML
konceptuální objekt	conceptual object	Objekt, který se používá v →konceptuálních modelech k popisu první podoby modelu softwaru. Jsou to →instance, →třídy nebo →množiny. Na rozdíl od →softwarových objektů konceptuální objekty nejsou zatíženy detaily konkrétního implementačního prostředí. Konceptuální objekty se odvozují z →business objektů a samy slouží k popisu →softwarových objektů.	CM, UML
konsolidace	consolidation	Druhá část cyklu →spirálního způsobu tvorby softwaru. Model se zde postupně po předchozí myšlenkové "expanzi" stává fungujícím programem. V tomto stadiu může dojít k tomu, že některé z návrhů bude nutno vypustit vzhledem k časovým, kapacitním, implementačním nebo i intelektuálním omezením. (→ <i>expanse, návrh, implementace</i>)	CM, SM
kovergenční inženýrství	convergent engineering	Inženýrský obor, který neklade hranici mezi podnikovým systémem a informačním systémem. (→ <i>sociotechnický systém</i>). Je to směr →informačního inženýrství, který techniky a postupy používané v →softwarovém inženýrství aplikuje do →business inženýrství. Opírá se o →objektový přístup. Je jednou ze základních filosofí →BORMu.	BM, CM
logický objekt	logical object	Synonymum pro →konceptuální objekt.	CM
metamodel	metamodel	Metamodel je model systému, který sám obsahuje prvky k modelování něčeho jiného. Znalost metamodelu metody může pomoci při pochopení metody a také usnadnit návrh konkrétního řešení. (→ <i>metamodelování, generický model</i>)	všechny fáze BORMu
meta-modelování	metamodeling	Metamodelování znamená tvorbu modelu na vyšší úrovni abstrakce tak, že předmětem modelování je něco, co je samo o sobě modelem něčeho jiného. (→ <i>metamodel</i>). Dovoluje popsat jednotným způsobem odlišné datové struktury více systémů, což poskytuje možnost skládat je dohromady ve vyšší systém. Zajišťuje standardizaci metodologií a interoperabilitu →CASE nástrojů.	všechny fáze BORMu
metoda	method	Část programového kódu, kterou objekt spouští, pokud přijme příslušnou →zprávu. Metody mohou například měnit data uvnitř objektů (→ <i>skládání, zapouzdření</i>), posílat další zprávy dalším objektům nebo provádět →přechody mezi →stavy objektu. Metody se odvozují z →aktivit.	CM, SM
měření (v praxi používaný pojem je „metriky“)	measurement (metrics)	Nástroj pro podporu formálního měření kvality softwarových produktů nebo modelů. Aparát pro softwarové měření je numerická matematika a statistický aparát. Jejich sběr a vyhodnocování je důležitou součástí řízení projektů →BPR nebo tvorby softwaru. (→ <i>funkční body</i>).	všechny fáze BORMu

Informační systémy 1

množina objektů, sada, kolekce	collection	Skupina →objektů, které mají dohromady nějaký zvláštní význam pro model, kde se s nimi pracuje nejen po prvcích, ale i jako s celkem. Množiny do systému zavádíme když není vhodné skupinu objektů realizovat jako →třidu. Prvky množiny mohou být objekty různých tříd, pokud jsou mezi sebou →polymorfni. Množiny objektů se odvozují z →business objektů.	CM, SM
modelovací karta, modelová karta	modeling card	Modelovací karta je tabulka, která podrobněji popisuje →objekt. Je součástí techniky →OBA. Může obsahovat →aktivity objektu, s objektem spolupracující objekty a nebo také vlastnosti objektu, které v modelu potřebujeme. (→proces)	BM, OBA
nadtyp-podtyp	supertype-subtype	Synonymum pro →hierarchii typů.	CM
návratová hodnota	return value	Data přenášená jako odpověď na posláni →zprávy mezi objekty. Lze je považovat za →datový tok ve směru opačném k směru posláni zprávy.	CM, SM
návrh, design	design	Souhrnný název pro fáze tvorby systému, při kterých se na základě již provedené →analýzy model přetváří do podoby, kterou je již možné implementovat. (→implementace). V případě →spirálního způsobu tvorby softwaru je návrh součástí →konsolidace systému. V →BORMu se návrh skládá z fáze →úvodního návrhu a →podrobného návrhu.	CM, SM
návrhový vzor	design pattern	Návrhové vzory jsou znalosti z předchozích úspěšných projektů. Přestavují mechanismus k uchování znalostí o tvorbě modelů, které se týkají →znovupoužitelných řešení. Je to významný prostředek k předávání znalostí mezi tvůrci systému při →softwarovém modelování. Někteří autoři také poukazují na možnost využívání návrhových vzorů při →BPR.	SM, BPR
OBA – analýza objektů podle chování	OBA – Object Behavioral Analysis	Technika, která slouží k získávání strukturovaných podkladů ze zadání pro potřeby konstrukce prvotního objektového modelu. Má pět kroků. Výstupy z OBA je například seznam →funkcí systému, →scénářů systému, →modelovacích karet a →procesní diagramy.	BM, OBA
objekt	object	Základní prvek modelování objektově orientovaným způsobem. Objekt tvoří jeho data a operace (aktivity nebo metody), které vymezují jeho chování. (→zapouzdření, protokol, polymorfismus) Objekty mají v systému →identitu a jejich vlastnosti se mohou v čase měnit. (→automat) V →BORMu se nazírání na objekt mění podle fáze projektování. Proto se rozlišují →business objekty, →konceptuální objekty a →softwarové objekty.	všechny fáze BORMu
objektová databáze	object-oriented database, objectbase	→Databázový systém, který dovoluje uchovávat a také zpracovávat →softwarové objekty přímo v databázi. Takové systémy nepoužívají relační tabulky a pracují na jiném principu. Mezi objektové databáze patří například →Gemstone. (→objektově-relační databáze, relační databáze)	SM
objektově relační databáze	object-relational database, hybrid object database	→Databázový systém, který uchovává →softwarové objekty v relační databázi a podporuje některé vlastnosti →objektového přístupu. Principem jeho činnosti zůstává relační datový model. Mezi tyto systémy například patří →Oracle od verze 8.0 (→objektové databáze)	SM
objektový datový model (ODM)	object-oriented data model	Datový model, na jehož principu pracují → objektové databáze.	SM
objektový programovací jazyk	object-oriented programming language	Programovací jazyk, který dovoluje využívat ve svých příkazech →objektový přístup. Rozlišují se →smíšené programovací jazyky a →čisté objektové programovací jazyky.	SM

Informační systémy 1

objektový přístup, objektově orientovaný přístup, paradigma	object-oriented paradigm, object-oriented approach, OOP	Souhrn myšlenek o způsobu nazírání na svět, chápání zadání a hledání způsobu jejich řešení. Podle objektového přístupu se systém modeluje jako soustava vzájemně komunikujících →objektů. (→ <i>konvergenční inženýrství, proces</i>) V užším slova smyslu to je jeden ze způsobů tvorby softwaru. (→ <i>objektový programovací jazyk, objektová databáze</i>)	všechny fáze BORMu
omezení podle chování	behavioral constraints	Sada pravidel, typicky zobrazovaná ve formě rozhodovacích tabulek, která popisují za jakých podmínek lze konkrétní vazby mezi →business objekty transformovat na vazby mezi →konceptuálními objekty.	BM, CM
OMG	OMG	Object Management Group. (http://www.omg.org) Je to odborné sdružení firem, univerzit a dalších institucí, které podporuje objektovou technologii. Vydává také publikace a standardy (→ <i>OQL, CORBA</i>).	SM
OOP	OOP	Zkratka pro Object-Oriented Paradigm. (→ <i>objektový přístup</i>)	všechny fáze BORMu
operace	operation	→Metoda nebo →aktivita objektu.	CM, SM
OQL	OQL	Návrh rozšíření jazyka →SQL pro práci s →objekty. Je doporučován sdružením →OMG a je součástí standardu →CORBA.	SM
Oracle	Oracle	Představitel →relačního databázového systému. Nové verze Oraclu jsou →objektově relační. Používá jazyk →SQL.	SM
ORD – diagram vztahů mezi objekty	ORD – Object Relation Diagram	Diagram, který zobrazuje současně datové, funkční i časové vztahy a souvislosti mezi objekty. OR diagramů je v →BORMu víc druhů podle toho, jaké objekty zobrazují. Například pro fázi →business modelování to je →procesní diagram (nazývaný také procesní mapa) a pro pozdější fáze se používají upravené diagramy UML.	všechny fáze BORMu, UML
parametry zprávy participant	message parameters participant	Data přenášená jako součást →zpráv mezi objekty. Parametry lze považovat za →datový tok ve směru zprávy.	CM, SM
podnikový proces	business process	→Objekt, který se účastní nějakého →procesu a je popsán v nějakém →scénáři a nebo →procesním diagramu.	BM
podrobná analýza,	advanced analysis	Synonymum pro →business proces.	BM, BPR
podrobný návrh, podrobný design	advanced design	Vymezení softwarové domény uvnitř modelovaného problému a rozpracování analýzy do detailů jednotlivých druhů a vazeb →konceptuálních objektů. Při →spirálním způsobu tvorby softwaru je součástí →expanse systému. (→ <i>analýza</i>)	CM
polymorfismus	polymorphism	V této fázi dochází k přeměně prvků již existujícího →konceptuálního modelu do podoby, která je podřízena cílovému implementačním prostředím. (→ <i>softwarový model, softwarový objekt</i>) Při →spirálním způsobu tvorby softwaru je součástí →konsolidace systému. (→ <i>návrh</i>)	SM
pozdní vazba	late binding	Schopnost objektů reagovat různou →metodou na stejnou →zprávu. Rozlišujeme polymorfismus 1) mezi různými objekty, 2) mezi různými →stavy téhož objektu a 3) v závislosti na objektu, který zprávu vyslal. Polymorfismus významně ulehčuje modelování a způsobuje →znovupoužitelnost objektů. Díky polymorfismu mohou v jednom systému pracovat a zastupovat se objekty s různou strukturou. (→ <i>protokol</i>) V některých →objektových programovacích jazycích je však implementován jen částečně. (→ <i>dědičnost</i>)	všechny fáze BORMu
		Druh vazby mezi →zprávou a →metodou, kdy se výběr metody provádí až po poslání zprávy během chodu objektového programu, což je v souladu s filosofií →polymorfismu. Tento typ vazby je na rozdíl od →včasné vazby typický pro →čisté objektové programovací jazyky.	SM

Informační systémy 1

pracovní pozice	job position	V kontextu →BPR to je součást detailního popisu →participantu. Je to konkrétní popis pracovního místa daného pracovníka s případným počtem konkrétních pracovníků na této pozici. Jeden participant typicky obsahuje více pracovních pozic. (→BPR)	BM, BPR
problém, issue	issue	Problém je něco, co stojí v cestě při plnění nějakého cíle nebo úkolu a je třeba to vyřešit. Je to například „velké emise do životního prostředí“ nebo „státní regulace“ apod. Jejich znalost je důležitá pro rozhodování nad souvislostmi zadání informačního systému, podnikových →procesů (→business proces) s dalšími atributy organizace. (→získávání zadání, BPR)	BM, BPR
proces	process	Sled →aktivit objektů, které dohromady realizují nějakou funkci systému. Mezi procesy může být hierarchie. Procesy popisujeme →scénáři a →procesními diagramy. (→BPR, role objektu v procesu, business proces)	BM, BPR
procesní diagram, procesní mapa	process diagram, process map	Tento →diagram představuje mapu všech možných průběhů →procesu pomocí současného zobrazení dvou dimenzí tohoto problému. První dimenzí jsou →role (= průběh aktivit jednoho objektu v procesu) participujících objektů jako →automaty se →stavy a →přechody. Druhou dimenzí je sled →komunikací mezi objekty, který představuje řídicí a datové toky mezi objekty v procesu.	BM, OBA
protokol objektu	object protocol	Množina →zpráv nebo →komunikací, které lze objektu poslat. Pomocí protokolu se u objektů popisuje, k čemu v modelu slouží a jaké operace mohou provádět. Má-li více objektů neprázdný průnik protokolů, jsou mezi sebou →polymorfni (→nadtyp-podtyp). Znalost protokolu stačí k tomu, aby se s objektem mohlo v systému pracovat, protože pokud dokáže příslušné zprávy přijímat, tak již nezáleží na jeho vnitřní struktuře. (→zapouzdření, znovupoužitelnost)	všechny fáze BORMu
prototyp	prototype	Program, který napodobuje funkčnost vytvářeného softwaru, ale byl sestaven s výrazně menšími náklady a nižšími nároky na robustnost. Slouží k upřesnění a ověření zadání. V →BORMu se používají i jako prostředky k demonstraci nově navržených →procesů. (→BPR)	BM
přechod	transition	Jeden úkon →automatu, kterým automat změni svůj stávající →stav na nový stav na základě nějaké přijaté informace. V →BORMu se na →objekty v →procesech nahlíží jako na automaty, kde jejich →přechody jsou realizovány prováděním →aktivit. (→role objektu v procesu)	všechny fáze BORMu
refaktoring	refactoring	Technika, při které se přeuspořádává vnitřní struktura programu za účelem větší srozumitelnosti, lepší údržby a pro snadnější potenciální znovupoužití v jiných aplikacích. Na rozdíl od ladění zde nejde o opravy a vylepšení funkčnosti programu. Po refaktoringu totiž program funguje z vnějšího pohledu stejně jako před refaktoringem.	SM
relační databáze	relational database, table-based database	→Databázový systém, který organizuje data do vzájemně propojených tabulek. Pokud dovoluje přímou práci s objekty a ne pouze s jednoduchými hodnotami např. typu text, číslo nebo datum, tak se hovoří o →objektově relačním systému.	SM
relační datový model (RDM)	relational data model	Datový model, na jehož principu pracují →relační databáze a →objektově relační databáze.	SM
relační vztah	relation	Znázorňuje spojení mezi záznamy v relačních tabulkách →relačních databází. Pro →objektově databáze se nepoužívá, protože tyto databáze pracují s objekty přímo propojenými.	SM

Informační systémy 1

role objektu v procesu	object role in a process	Sled →stavů a →přechodů objektu, který se účastní nějakého procesu. Je to jedna ze dvou dimenzí →procesního diagramu. Na tento sled lze nazírat jako na dílčí →diagram, který popisuje →proces ze zorného úhlu diskutovaného objektu, a který lze výhodně použít ke kontrole správnosti a realizovatelnosti celkového modelovaného procesu. (→ <i>automat</i>)	všechny fáze BORMu
sada objektů	collection	Synonymum pro →množinu objektů.	CM, SM
scénář	scenario	Scénář systému je podrobnější popis →procesu v technice →OBA. Odvozují se z →funkcí systému. U scénáře zvlášť popisujeme 1) začátek procesu, 2) vlastní akce procesu, 3) seznam participantů a 4) konec procesu. Mezi scénáři může být hierarchie skládání procesů, hierarchie nadtypů a podtypů a časová návaznost procesů na sebe, pro které se používají stejné značky jako pro objektové diagramy (→ <i>UML, ORD</i>). Typické zobrazení scénářů jsou tabulky s uvedenými čtyřmi políčky.	BM, OBA
sekvenční způsob tvorby softwaru, vodopádový model	sequential development lifecycle, waterfall development lifecycle	Způsob vývoje softwaru, kdy se ke každé další fázi projektu přistupuje až po úplném dokončení předchozí fáze. Vracení se zpět není přípustné. Tento styl se dobře plánuje a řídí, ale není vhodný v případech, kde na začátku projektu není ještě přesně rozpoznané zadání. Jiné způsoby vývoje jsou →evoluční a →iterativní.	všechny fáze BORMu
server	server	Objekt, který je →závislý na jiném objektu (→ <i>klient</i>).	CM, SM
server	server	Softwarová aplikace, která poskytuje data nebo služby jiným aplikacím na jiných počítačích v síti. (→ <i>klient</i>) Může se jednat o →databázový systém.	SM
skládání objektů, kompozice objektů	object composition	Vazba mezi objekty, kdy jeden nebo více objektů představují datové složky jiného objektu. Skládání objektů se v →BORMu odvozuje z →asociací.	CM, SM
SM	SM	Zkratka pro →softwarové modelování.	SM
Smalltalk	Smalltalk	→Čistý objektový programovací jazyk. Jeho první verze byly vyvíjeny již v 70. letech. Smalltalk může být také použit pro práci s daty v →objektových databázích. (→ <i>Gemstone, OOL</i>)	SM
smíšený programovací jazyk, hybridní jazyk	hybrid object-oriented programming language	→Programovací jazyk, který má základ v neobjektovém programování a dovoluje využívat některé vlastnosti objektového přístupu, ale některé, jako např. →závislost nebo →delegování nepodporuje. (→ <i>čistý objektový jazyk</i>). Mezi smíšené jazyky patří například →C++, →Java a →C#.	SM
SO	SO	Zkratka pro →softwarový objekt.	SO
sociotechnický systém	socio-technical system	→Informační systém včetně jeho podnikového prostředí tvořeného procesy uživatelů, organizační a řídicí strukturou popsaný a modelovaný jako jeden související celek. (→ <i>informační inženýrství, konvergenční přístup</i>)	všechny fáze BORMu
software	software	V kontextu →BPR to je součást detailního popisu →aktivity. Jsou to komponenty nebo moduly zamýšlených nebo již existujících →informačních systémů, které jsou potřeba pro vykonání dané aktivity v →procesu. Je to například „osobní evidence“ nebo „mzdy“ nebo „skladové hospodářství“ apod. (→ <i>architektura systému</i>)	BM, BPR
softwarové inženýrství	software engineering	Inženýrský obor, který se zabývá analýzou, modelováním, návrhem a provozem softwarových systémů. Některými autory je považován za součást →informačního inženýrství. Podle →BORMu při tvorbě systému by měly aktivitám softwarového inženýrství předcházet aktivity →business inženýrství.	CM, SM
softwarové modelování	software modeling	Třetí a poslední etapa →BORMu, která zahrnuje tvorbu modelu →softwarových objektů. Jeho prvky a vazby vycházejí z modelu →konceptuálních objektů, ale na rozdíl od něj musejí zohlednit problémy implementačního prostředí. (→ <i>smíšený programovací jazyk, zděděný systém, objektově relační databáze</i>) Skládá se z fáze →podrobného návrhu a →implementace.	SM, UML

Informační systémy 1

softwarový objekt	software object	→Objekty odvozené z →konceptuálních objektů během →softwarového modelování. Na rozdíl od nich zohledňují omezení implementačního prostředí.	SM, UML
spirální způsob tvorby softwaru, spirální model	spiral development lifecycle	Varianta →iterativního způsobu tvorby softwaru. Fáze projektu se provádějí několikrát za sebou tak, aby při každém opakování došlo ke zpřesnění zadání na základě předchozí implementace. První část jednoho vývojového cyklu se nazývá → <i>expandse</i> a druhá → <i>konsolidace</i> . Tento způsob je vhodný k využití →objektového přístupu. (→ <i>sekvenční model</i>)	všechny fáze BORMu
SQL	SQL (“sequel”)	Structured Query Language. Programovací jazyk který je standardem pro práci s daty v →relačních databázích. Jeho první verze pochází z počátku 70. let. V současné době probíhá jeho rozšiřování tak, aby mohl pracovat i s objekty. (→ <i>OQL</i> , <i>Smalltalk</i>)	SM
stav	state	Konkrétní konstelace →automatu v čase. Pokud automat přijme nějakou informaci, může to vyvolat přechod z jednoho jeho →stavu do druhého. V →BORMu se nahlíží na objekty v procesech jako na →automaty, které v různých stavech mohou provádět různé →aktivity. (→ <i>role objektu v procesu</i>)	všechny fáze BORMu
strategická analýza	strategic analysis	První fáze →BORMu, kde dochází k vymezení samotného problému, je stanoveno jeho rozhraní, jsou rozpoznány základní procesy, které se v systému mají odehrávat. Při →spirálním způsobu tvorby softwaru je součástí → <i>expandse</i> systému. (→ <i>získávání zadání, analýza</i>)	BM
synchronní zpráva	synchronous message	→Zpráva, na jejíž výsledek musí →objekt, který zprávu poslal čekat. Objekt tedy pokračuje ve svých aktivitách až když jsou dokončeny všechny aktivity, které byly zprávou vyvolány. (→ <i>asynchronní zpráva</i>)	CM, SM
TO-BE třída	TO-BE class	Druhá fáze →BPR. TO-BE znamená „tak, jak by to mělo být“. V čistých →objektových programovacích jazycích to je zvláštní objekt, který jako svoje data obsahuje vlastnosti (= strukturu dat a metody) pro svoje instance. Ve →smíšených jazycích to je pouze pojem ze syntaxe programovacího jazyka, pomocí kterého se programují vlastnosti objektů. Třídy se v →BORMu odvozují z → <i>business</i> objektů.	BM, BPR CM, SM
událost	event	Podnět z okolí k vykonání →operace nějakého →objektu. V →BORMu jsou zdroje událostí modelovány také jako objekty a jejich události jako →zprávy nebo →komunikace. (→ <i>závislost</i>)	BM, CM
úkol	target	V kontextu →BPR to je součást detailního popisu →aktivity. Úkoly popisují, čeho je třeba dosáhnout. Je to například „snížit surovinovou náročnost“ nebo „zkrátit průběh vyřízení objednávky“. Jejich znalost je důležitá pro rozhodování nad souvislostmi zadání informačního systému, →podnikových procesů s dalšími atributy organizace. (→ <i>získávání zadání</i>)	BM, BPR
UML – unifikovaný modelovací jazyk	UML – Unified Modeling Language	UML poprvé publikovali G. Booch, J. Rumbaugh a I. Jacobson v roce 1996. Je doporučovaným standardem pro notaci (způsob kreslení) objektových →diagramů a představuje sjednocení myšlenek původních metod svých autorů. Stále se ještě vyvíjí. Pro potřeby →BORMu je třeba UML doplnit. (→ <i>ORD</i>)	CM, SM
úvodní analýza	initial analysis	Fáze rozpracování samotného problému, jsou mapovány požadované →procesy v systému a vlastnosti základních →objektů, které se na diskutovaných procesech podílejí. Může při ní dojít k →BPR. Při →spirálním způsobu tvorby softwaru je součástí fáze → <i>expandse</i> . (→ <i>analýza</i>)	BM, BPR

Informační systémy 1

úvodní návrh, úvodní design	initial design	Je to fáze →BORMu, ve které se →konceptuální model upravuje tak, aby byl schopen softwarové implementace. Z pohledu zadání by mělo již vše být hotovo a rozpoznáno. Úvodní návrh používá shodné nebo velmi podobné nástroje jako předchozí fáze →podrobné analýzy, ale liší se způsobem práce s nimi. Při →spirálním způsobu tvorby softwaru je součástí fáze →konsolidace. (<i>→návrh</i>)	CM
včasná vazba	early binding	Druh vazby mezi →zprávou a →metodou, kdy již v době překladač kódu programu je ke zprávě překladačem jednoznačně přiřazena metoda, která se má provést, což může vést k omezení míry →polymorfismu mezi objekty. Tento typ vazby je na rozdíl od →pozdní vazby typický pro →smíšené objektové programovací jazyky.	SM
vrstva	layer	Jedna součást →architektury modelovaného systému.	všechny fáze BORMu, BPR
výrobek, deliverable	deliverable	Programový výstup z jednoho cyklu →spirálního vývoje softwaru. Označuje se takto nejen →prototyp, ale i produkt „posledního“ cyklu, protože i ten může posloužit jako nové zadání pro další vývojový cyklus. Může také sloužit pro tvorbu nové verze produktu - a to nejen skrze zkušenosti s ním, ale přímo i svým kódem jako výchozí model nové →expanse.	SM, BM
zapouzdření	encapsulation	Spolu s →polymorfismem a →identitou základní vlastnost →objektů. Díky zapouzdření může objekt obsahovat data nebo →metody, se kterými pracuje jen objekt sám, nejsou součástí jeho →protokolu, a neposkytuje je ostatním objektům v systému. Z vnějšího pohledu se potom objekt tváří, jako by tyto vlastnosti neměl.	všechny fáze BORMu
zařízení	device	Součást detailního popisu →aktivity v →BPR. Zařízení je konkrétní pracovní pomůcka nebo stroj (například „služební auto“ nebo „osobní počítač“ nebo „ochranný oděv“), který je potřeba k vykonání dané aktivity.	BM, BPR
závislost	dependency	Závislost vyjadřuje, že provedení nějaké →metody řídicího objektu (→klienta) nepřímo bez posláni →zprávy vyvolává provedení nějaké metody řízeného objektu (→serveru). V případě, že implementační prostředí tuto vazbu nepodporuje, musí se ve fázi →softwarového modelování nahrazovat. (<i>→smíšený programovací jazyk</i>)	CM, SM
zděděný systém	legacy system	Existující programy, data nebo organizační struktura či procesy, které nelze měnit nebo ignorovat, což komplikuje tvorbu nového systému ve fázi →softwarového modelování, protože se jim musí struktura nového systému přizpůsobit.	SM
získávání zadání, requirement engineering	requirement engineering	Inženýrský obor, který se zabývá rozpoznáváním, analýzou a verifikací zadání pro →informační systémy. (<i>→informační inženýrství, business inženýrství, softwarové inženýrství</i>)	BM, BPR
znovu-použitelnost	reusability	Schopnost →objektu sloužit v jiném systému jiným způsobem, než pro jaký byl objekt vytvořen. Objekty jsou znovupoužitelné díky →polymorfismu, protože jsou snáze mezi sebou zaměnitelné. (<i>→protokol</i>) Znovupoužitelnost je základem techniky →návrhových vzorů.	všechny fáze BORMu
zpráva	message	Žádost o provedení →metody →objektem, kterému je zpráva poslána. Na rozdíl od volání funkce v klasickém programování je v objektovém programování od sebe oddělena žádost o operaci a její vlastní provedení. (<i>→polymorfismus, včasná vazba, pozdní vazba</i>) Zprávy mohou přenášet data. Data ve směru posláni zprávy se nazývají →parametry zprávy, data posílaná opačným směrem jsou →návratové hodnoty. Zprávy jsou →synchronní nebo →asynchronní.	CM, SM

