

UNIVERZITA PARDUBICE
FAKULTA EKONOMICKO - SPRÁVNÍ

BAKALÁŘSKÁ PRÁCE

2008

Hana Polanská, DiS.

Univerzita Pardubice
Fakulta ekonomicko - správní

**Trendy v přístupech k životnímu cyklu vývoje
informačního systému (SDLC)**

Bakalářská práce

2008

Zadávací List

Poděkování

Ráda bych poděkovala především vedoucímu mé bakalářské práce Ing. Pavlu Jiravovi, Ph.D. za všechny připomínky, rady a především velkou podporu a trpělivost při zpracování bakalářské práce. Chtěla bych poděkovat všem, kteří mi pomáhali radou či připomínkami, případně jinak. Především mé sestře Mgr. Michaele Polanské.

SOUHRN

Bakalářská práce je zaměřena na shrnutí a zhodnocení současných přístupů k životnímu cyklu tvorby informačního systému. Jaký je trend v přístupu k životnímu cyklu informačního systému a v čem se liší. Seznámíme se s konkrétními modely pro tvorbu informačního systému. Klasické modely jsou porovnávány s modely moderními. V závěru práce je podrobně rozebrána problematika informačního systému veřejné správy a nezbytné dokumenty pro jeho tvorbu.

KLÍČOVÁ SLOVA

Informační systém, životní cyklus informačního systému, SDLC = životní cyklus vývoje systému.

TITLE

TRENDS IN ACCESSES TO LIVE CYCLE DEVELOPMENT OF
INFORMATICS SYSTEM

ABSTRACT

The bachelor's work is intent on summarize and evaluate concurrent access of live cycle of making to informatics system. How is trend in access of live cycle informatics system and how are existing different. We acquaint with concrete models for making informatics system. Classical models are compared with modern models. In the end of my work is in detail described problematic of informatics system of public service and necessary documents for make.

KEYWORDS

Informatics system, live cycle system, system development live cycle.

OBSAH

1	Úvod	8
2	Životní cyklus informačního systému	9
2.1	Několik faktů o IS.....	9
2.2	Oblast technického zabezpečení IS .	10
2.3	Samotný životní cyklus IS	10
2.3.1	Slovníček používaných pojmů během sestavování životního cyklu IS	11
3	Ukázky modelů životního cyklu	13
3.1	Vodopádový model.....	13
3.2	Prototypování	15
3.3	Model výzkumník.....	15
3.4	Iterativní model	16
3.5	Spirálový model.....	17
4	Jednotlivé kroky v životním cyklu IS	20
4.1	Zahájení.....	20
4.2	Vývoj	21
4.3	Implementace	21
4.4	Provoz a údržba	21
4.5	Ukončení provozu.....	21
5	Nejdůležitější etapy SDLC	22
5.1	Etapy SDLC	22
5.1.1	Identifikování problému a plánování	23
5.1.2	Analýza potřeb systému	25
5.1.3	Návrh (Design) systému	26
5.1.4	Vývoj softwaru	27
5.1.5	Testování systému	28
5.1.6	Implementace a údržba	29
5.2	Závěr SDLC	30
5.2.1	Oblast omezení	31
5.2.2	Progresivní vzestup.....	31

5.2.3	Předdefinování struktury.....	31
5.2.4	Inkrementální plánování.....	32
6	Trend přístupu k životnímu cyklu IS.....	33
6.1	Vlastní analýza sestává ze 3 relativně samostatných modelů:.....	33
6.1.1	Objektový model.....	33
6.1.2	Dynamický model.....	33
6.1.3	Funkční model.....	34
6.2	Jaké diagramy metodika OMT může obsahovat.....	34
6.2.1	Use case.....	34
6.2.2	Diagram datových toků.....	34
6.2.3	Mapa událostí.....	35
6.2.4	Metodika tvorby systému.....	36
7	Návrh životního cyklu pro IS v oblasti veřejné správy.....	40
7.1	Co je ISVS.....	40
7.2	Standardy ISVS pro náležitosti životního cyklu informačního systému.....	40
7.2.1	Právní předpisy České Republiky se vztahem k ISVS.....	41
7.3	Fáze životního cyklu ISVS.....	44
7.4	Návrh životního cyklu ISVS.....	45
7.4.1	Fáze přípravy IS.....	45
7.4.2	Fáze vývoje, provozu a údržby IS.....	46
7.4.3	Fáze ukončení činnosti IS.....	47
7.5	Grafické vyjádření životního cyklu ISVS.....	48
8	Závěr.....	49
9	Seznam použité literatury.....	51
10	Seznam obrázků.....	54
11	Seznam tabulek.....	54
12	Seznam příloh.....	54

1 ÚVOD

Během mého hledání a přezkoumávání různých podkladů pro bakalářskou práci jsem zjistila, že teorie a praxe životního cyklu je velice odlišná. Životní cyklus informačního systému jako takový není z teoretického hlediska náročný. Byl převzat z klasického životního cyklu produktu a proto určité fáze nám připadají jednodušší a dobře aplikovatelné. Problém nastává v konkrétní situaci a konkrétním případě. Pokaždé potřebujeme vyvíjet informační systém, který bude podle požadavků zákazníka. Tudíž neexistuje striktně dané pravidlo pro vývoj životního cyklu informačního systému. Našla jsem nekonečně mnoho typů životního cyklu, ale každý byl řešen trochu jinak, existuje několik modelů, které se soustřeďují pokaždé na něco jiného.

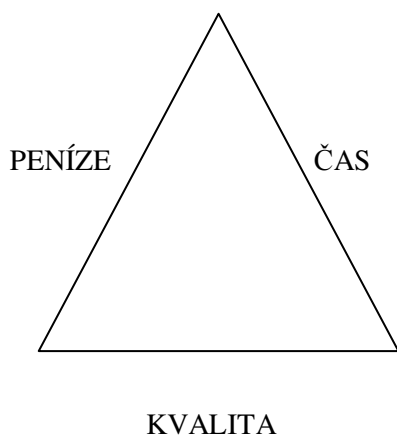
Životní cyklus informačního systému je dynamický, protože se neustále mění požadavky na něj a proto se jedná o jeho neustálou inovaci.

Cílem mé práce bude shrnout a zhodnotit současné přístupy k životnímu cyklu tvorby informačního systému. Popsat jednotlivé modely, které patří mezi klasické přístupy, ale i přístupy současnosti, které jsou tzv. moderní. Snahou é práce bude tyto cykly podrobněji popsat, vysvětlit, znázornit a v neposlední řadě zhodnotit, kdy a jak, který použít. V jedné z mých kapitol se podrobněji zaměřím na informační systém veřejné správy, kde bych ráda vyzdvihla dokumenty, kterými se řídí a samozřejmě vysvětlila tuto problematiku .

2 ŽIVOTNÍ CYKLUS INFORMAČNÍHO SYSTÉMU

Životní cyklus začíná již nejprvnější myšlenkou o programu a končí v ten samý moment, kdy se program přestane používat.

Každý IS by měl být integrovaný (= velké pokrytí), pracovat v reálném čase, minimalizovat datové redundance, být zaveden co nejrychleji a stát co nejméně peněz. Zde je známý trojúhelník peněz, času a kvality.



Obrázek 1: Trojúhelník peněz, času a kvality

2.1 Několik faktů o IS

- Může, ale nemusí být podporován počítačem, přičemž při návrhu IS zkoumáme optimální kombinaci automatizovaných a neautomatizovaných činností.
- Musí disponovat prostředky sběru, kontroly a uchování dat.
- Jsou vyjasněné vztahy mezi informacemi a daty i v rámci jednoho zaměření informačního systému. Informace jsou jen ta data, která dokážeme využít, přiřadit jim význam či smysl. Během návrhu IS je nutné, aby bylo umožněno získávání odlišných informací pro různé zaměstnance – skladník, ředitel atd.

- Informační systém ovlivňují pracovní procesy i organizační struktura podniků. Proto jsou IS produkovány přímo tak říkajíc „na míru“ daného podniku.
- Informační systém je vždy společným dílem dodavatele a zákazníka, proto je velmi důležitá správná a účinná komunikace mezi těmito subjekty.

Prudký rozvoj v informačních technologiích probíhá ve všech jejích komponentech, kterými je myšlen hardware a software.

Výraznými rysy vývoje v jednotlivých oblastech jsou: oblast technologického zabezpečení IS a samotný životní cyklus IS.

2.2 *Oblast technického zabezpečení IS .*

- stálé snižování poměru cena/výkon (cca 20% ročně),
- odklon od velkých sálových počítačů směrem k počítačovým sítím se specializovanými servery a řadou připojených personálních počítačů (pozoruhodný je zejména růst počtu personálních počítačů - odhaduje se, že v roce 1994 bylo na světě 208 mil. kusů PC a tento počet neustále vzrůstá),
- standardizace a z toho vyplývající možnost kombinování hardwarových produktů různých výrobců (ovlivňuje trh IT ve smyslu nezávislosti organizace na určitém dodavateli či výrobcu IT), rozvoj v oblasti komunikací a Internet (nové typy strojů, multimedia, budování informační superdálnice). [15], [10]

2.3 *Samotný životní cyklus IS*

Životní cyklus je obdobím, které začíná prvotní představou o programu a končí okamžikem, kdy se program přestane používat.

2.3.1 Slovníček používaných pojmů během sestavování životního cyklu IS

Následující text vychází ze studia celé řady odborných publikací a textů , především pak „[3], [5], [6], [15] , [16] [21]“...

Trend: základní směr vývoje určitého jevu, jedná se o dlouhodobou tendenci či sklon, který je ovšem určitý. Též můžeme trend definovat jako po delší dobu se projevující proces, který se jako průměr prosazuje mezi individuálními odchylkami od tohoto průměru.

Vývoj: je označení pro soustavný proces, během kterého dochází ke změně aktuálního stavu do stavu nového. Cílem vývoje je na základě zkušenosti, plánu, či náhodné chyby vyvíjet stále lepší verze. Vývoj může být přirozený nebo umělý. Mezi umělé změny patří takové, které jsou přímo ovlivňovány bytostí za účelem zlepšení stávající věci zájmu. Mezi umělé vývoje se může zařadit například vývoj softwaru, kdy na základě objevovaných chyb a nových nápadů dochází k opravám a zlepšováním starší verze.

Informační systém IS: funkční propojení lidí, dat, procesu rozhodování, sítí a technologií, které spolupracují, aby podporovali a zlepšovali každodenní operace („data processing“) v organizaci a zároveň, aby podporovali řešení problémů a proces rozhodování v rámci managementu („information services“). Každý informační systém má sloužit vlastníkově nebo uživateli (ne naopak, jak tomu často bývá).

Informace: Skupina nebo množina informačních jednotek či signálů, která je příjemcem interpretovatelná jako smysluplná. Míjíme sdělení, které odstraňuje nejistotu nebo nevědomost.

Data: zobrazují stavy objektů či probíhající procesy v realitě kolem nás (to, s čím přichází příjemce do styku). V závislosti na způsobu a okolnostech prezentace dat buď představují tato data pro příjemce informaci nebo nikoli.

Znalosti: představují zobecněné poznání určité části reality. Znalosti souvisejí s vymežováním pojmů, s kategorizací, s definováním, s odvozováním závěrů z dostupných faktů na základě abstraktních schémat (hypotéz) a s vymežováním mechanismů (postupů), odvozováním závěrů.

Implementace: znamená uskutečnění, naplnění a doplnění. Odtud sloveso implementovat - uskutečnit či naplnit.

Databázové schéma: představuje seznam implementovaných tabulek a pohledů = definice rozvržení systému. Vzniká z myšlenkového modelu a to převedením do fyzické reprezentace, kterou je již možné implementovat ve zvoleném systému pro správu databází. Databázové schéma je databázový model vyjádřený v pojmech, pomocí kterých jej popisujeme databázovými stroji. Zde se nezabýváme fyzickou implementací.

Databázový model: můžeme rozumět myšlenkový (konceptuální) popis prostoru problému. Nemá však žádnou souvislost s fyzickým rozložením výsledného systému.

3 UKÁZKY MODELŮ ŽIVOTNÍHO CYKLU

Životní cyklus IS se řídí několika základními modely, samozřejmě tyto modely jsou všelijak obměňovány a doplňovány o další kroky. Ukázky modelů slouží jako účinný návod životního cyklu IS, protože modely doplňujeme o jednotlivé kroky dle potřeby, situace a požadavků na systém. [15], [17]

- vodopádový model
- prototypování
- model výzkumník
- iterativní model
- spirálový model

3.1 Vodopádový model

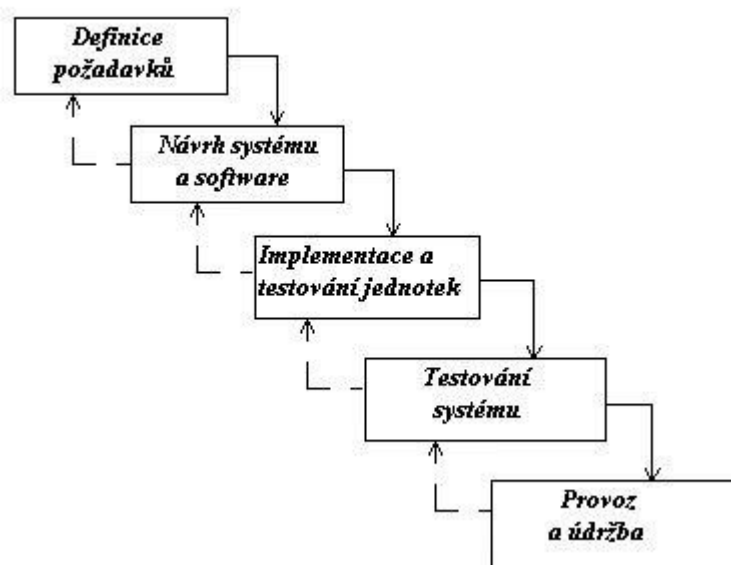
Tento model je jedním z nejstarších modelů, začíná se datovat od roku 1970. Existuje v několika různých verzích, my se zaměříme na jednu z nich. Celý proces začíná definicí požadavků, v rámci kterého je nutné se zaměřit na to, co zákaznická organizace a její uživatelé od systému požadují. Po dokončení analýzy se vytvoří podrobný návrh celého systému. Následuje fáze zavádění a testování jednotlivých jednotek systému, po jejich odladění testováním systému jako celku. Poslední fází projektu je samotný provoz a údržba systému.

Vodopádový model je složen z posloupnosti jednoznačně vymezených etap, které na sebe navazují a vzájemně se prolínají. Naprosto naivní představa pak dokonce počítá s jednoznačným dokončením každé části před spuštěním následující. Pokud by tomu tak bylo, tak by model prováděl rozumnou kontrolu nad rozpočtem, nasazením pracovníků a časem.

Při použití tohoto modelu, je kladen velký důraz na zadavatele, musí být jasně a přesně definované požadavky, neboť vrátit se zpět nebo přeskakovat z jedné etapy do druhé, není možné (z toho paralela s tekoucí vodou, tekoucí z vodopádu). V průběhu realizace nelze měnit požadavky a technologické postupy, proto je v praxi tento model použitelný jen výjimečně.

Pozitiva:

- Model je nedokonalý, ale je lepší než náhodný, nelogický neřízený přístup k řešení projektu.
- Velký teoretický význam- definuje jednotlivé etapy podle logického sledu.



Obrázek 2: Vodopádový model [15],

Negativa:

- Reálné projekty zřídka sledují jednotlivé etapy modelu v předepsaném pořadí.
- Uživatel často nedokáže předem stanovit své požadavky zřetelně a jednoznačně.

- Provozní schopnost verze vidí zákazník až v závěrečných fázích řešení, případně jsou závažné nedostatky odhaleny velmi pozdě. Proto může mít nepříznivý vliv na finanční stránku projektu a jeho celkový neúspěch.

3.2 *Prototypování*

Prototyp je částečně zavedeným produktem, může být v logické nebo fyzické podobě a má všechna vnější rozhraní.

Negativa:

- Náročný na vedení v okamžiku, kdy existuje třeba i několik nezávislých vývojových skupin. Počet pracovníků a způsob vývoje se pak samozřejmě nepříznivě projeví i na nákladech projektu.

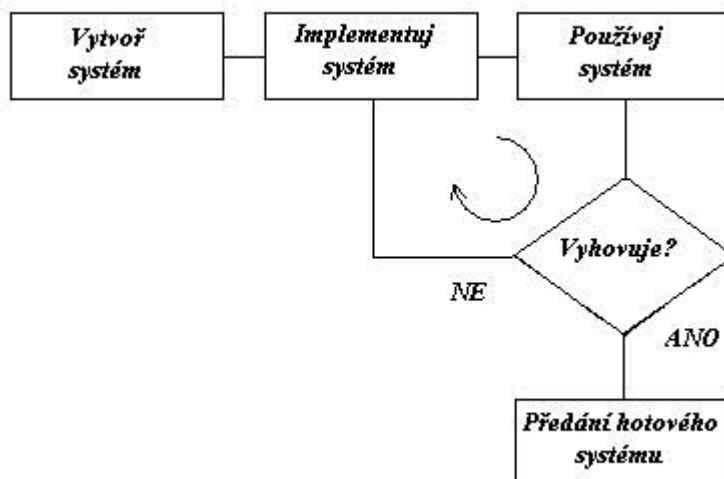
Pozitiva:

- Budoucí uživatelé testují a ověřují prototyp a je upřesňována specifikace požadavků. Zákazníci spolu s řešitelským týmem si průběžně sdělují a vyjasňují požadavky i jejich interpretaci. Některé prototypy jsou tvořeny s vědomím, že budou "vyhozeny", jiné budou použity a dále rozšířeny.

3.3 *Model výzkumník*

Během vývoje systému se řešitelé při získávání poznatků často vracejí k předchozím etapám. Tento model s tím přímo počítá jako se základní charakteristikou vývoje. Ale způsob vývoje s sebou nese především problémy časové a finanční.

Využívá se v případě, kdy je nutné se vrátit k předešlým etapám projektu, z důvodu chyby.



Obrázek 3: Model výzkumník [15]

Negativa:

- Manažersky velmi náročné – etapy lze stěží plánovat časově, finančně i personálně.
- Odklon od původního záměru, z důvodu neustálých úprav (Častý výskyt sporů s realizačním týmem a zadavatelem)
- Dokumentace – pokud nevzniká průběžně, je odrazem hotového díla. Při údržbě není jasné, které požadavky byly stanoveny při zadání, a které jsou nyní nové.

Pozitiva:

- Systém je velice dobře přizpůsobitelný i dodatečným požadavkům zákazníka.
- Není nutné v případě připomínek vracet se k analýze

3.4 Iterativní model

Tento model je vhodný pro řešitele, protože zde hraje velkou roli zadavatel a je spoluodpovědným za další vývoj projektu.

Pilířem modelu je zpracování úvodní studie, kde jsou zjištěny původní požadavky a potřeby zadavatele. Následuje analýza, návrh a implementace. Pokud se zjistí nové požadavky či provádí změna, vrátíme se zpět do fáze analýzy. Proces je ukončen v momentě splnění všech požadavků a neexistují-li další připomínky.

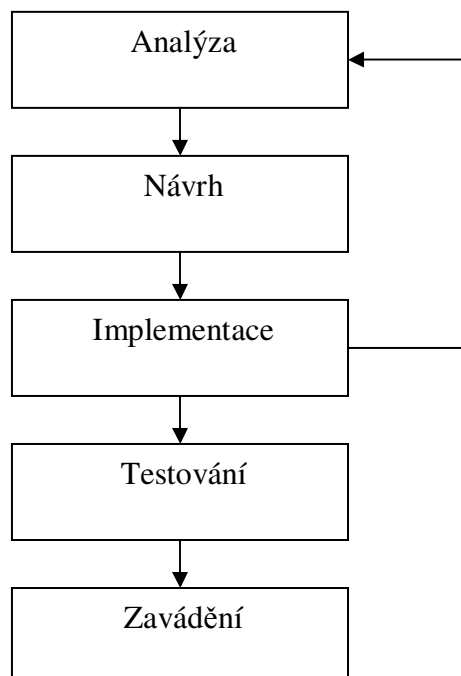
Použití modelu je v případě upřesňování požadavků během vývoje nebo pokud během testování uvedu nové požadavky.

Pozitiva

- Zadavatel vidí postup v realizaci jednotlivých částí návrhu.

Negativa

- Časová a finanční náročnost
- Vracení se stále na počáteční fáze v případě připomínek



Obrázek 4: Iterativní model [17]

3.5 Spirálový model

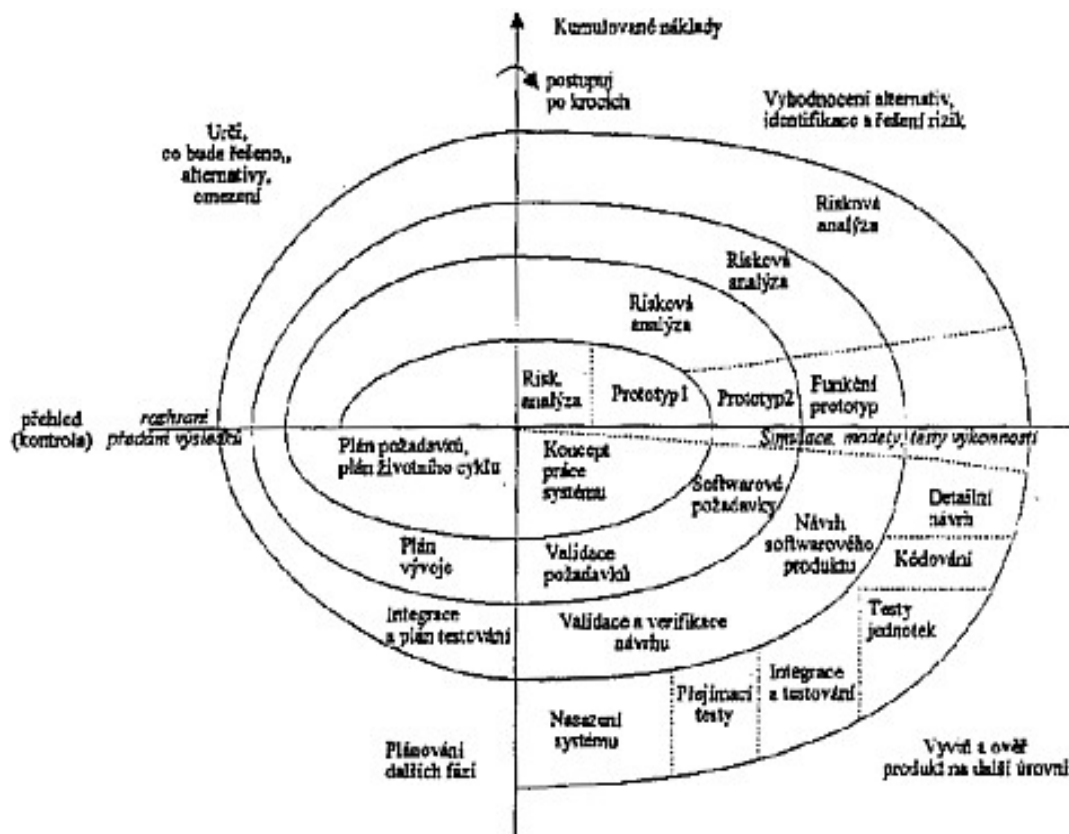
Model je založen na kombinaci prototypování a analýzy rizik. Jednotlivé kroky při vývoji systému se ve spirále opakují, ale pokud možno na vyšším stupni zvládnuté problematiky.

Negativa:

- Výhodné jen pro zkušené řešitelské týmy, zvláště při velkých projektech. Pokud se budete také přesně řídit schématem tohoto modelu, tak celkový záměr projektu můžete vzít do úvahy až velmi pozdě a předchozí výsledky mohou být ztraceny.

Pozitiva:

- Již v časných fázích modelu je pozornost zaměřena na použití SW. Chyby a nevyhovující alternativy jsou vyloučeny co nejdříve (díky "vyhazování" některých prototypů a analyzování rizik).



Obrázek 5: Spirálový model [15]

Pro životaschopný IS, je potřeba věnovat přípravnému procesu nejméně stejné úsilí jako samotnému fyzickému vývoji. Shrňme tedy alespoň základní kroky, které by měly vést po správné cestě.

Při tvorbě nového projektu, je třeba shromáždit velké množství přípravných údajů a zapsat je ve formě, která bude srozumitelná, pokud možno všem zúčastněným. Mnohé údaje, které zdají v počátcích zbytečné, jsou po čase "objeveny" jako zásadní, mnohé údaje opravdu zbytečnými mohou být.

4 JEDNOTLIVÉ KROKY V ŽIVOTNÍM CYKLU IS

- **Zahájení**
- **Vývoj**
- **Implementace**
- **Provoz, podpora, údržba, rozvoj**
- **(ukončení provozu)**

4.1 Zahájení

Na tvorbu nového či dalšího vývoje stávajícího IS zadá podnět zákazník, když má dost síly na prosazení myšlenky a dostatek zdrojů na financování návrhu, vývoje a provozu IS. Prvotním krokem je definování požadavků, které obsahuje: co je požadavkem, co je potřeba řešit (v čem problém spočívá, a do které oblasti spadá), kdo jsou účastníci procesu, proč je potřeba problém řešit (jaké jsou cíle účastníků), kdy je třeba problém vyřešit (časový horizont a limity), jak pomůže software problém vyřešit. [10], [7]

Požadavky se musí specifikovat, např. všechny funkce a výkonnost systému. Toto rozdělit mezi hardware a software, požadavky na bezpečnost a stanovit uživatelské rozhraní.

Požadavky by měly být stanoveny:

- Správně
- Jednoznačně
- Úplně
- Konzistentně – bez rozporů
- Ověřitelně
- Ohodnoceně dle důležitosti
- Modifikovatelně (strukturovaně)
- Vysledovatelně

Pro získání informací existuje několik způsobů a zdrojů. Záleží jen na vyhodnocení toho nejefektivnějšího. Informace můžeme získat např. vzorkováním existující dokumentace,

formulářů a databází, průzkumem na místě, osobní návštěvou, pozorováním pracovního prostředí, různými dotazníky a rozhovory, brainstormingem, ...

4.2 Vývoj

V této druhé fázi je pozornost zaměřena na získání nebo vytváření všech potřebných prostředků (HW, SW, ...), testování, dokumentace, jak by měl IS a produkční systém pracovat. Mělo by být ověřeno, zda systém práce a IS opravdu řeší problémy či obtíže uživatele, které jsou známé z první fáze. Výsledkem této fáze je fungující informační technologie. V tomto okamžiku velice často dochází k selhání ať z pohledu zpoždění nebo překročení daného rozpočtu.

4.3 Implementace

Třetí fáze systému zahrnuje uvedení nového systému do provozu. Naplánování přechodu ze starého IS na nový. To obsahuje školení uživatelů a ověřit zda systém je efektivní. Může nastat zmatek, ale i selhání (není ovšem tak časté jako v předešlé fázi).

4.4 Provoz a údržba

V předposlední fázi je nutné neustálé sledování a kontrolování funkčnosti systému, provádění údržby. Často zde dochází k omylu, že tato fáze je nejjednodušší, tudíž se jí nemusí věnovat pozornost. Opak je pravdou. Tato fáze má důležitý vliv na využívání systému uživateli. Je vhodné drobné změny provést co nejdříve, protože po delším čase je obtížnější provádět údržbu a změny.

4.5 Ukončení provozu

Poslední fází IS je ukončení provozu. IS je zastaralý a nestačí uspokojit požadavky uživatele (náročností, rychlostí, ...).

5 NEJDŮLEŽITĚJŠÍ ETAPY SDLC

Systems Development Life Cycle (SDLC = systémový vývoj životního cyklu) je stanovený u United States Department of Justice jako vývoj softwarového procesu, ačkoliv je zřejmé, že se jedná o proces nezávislý na softwaru nebo na jiné informační technologii. [21]

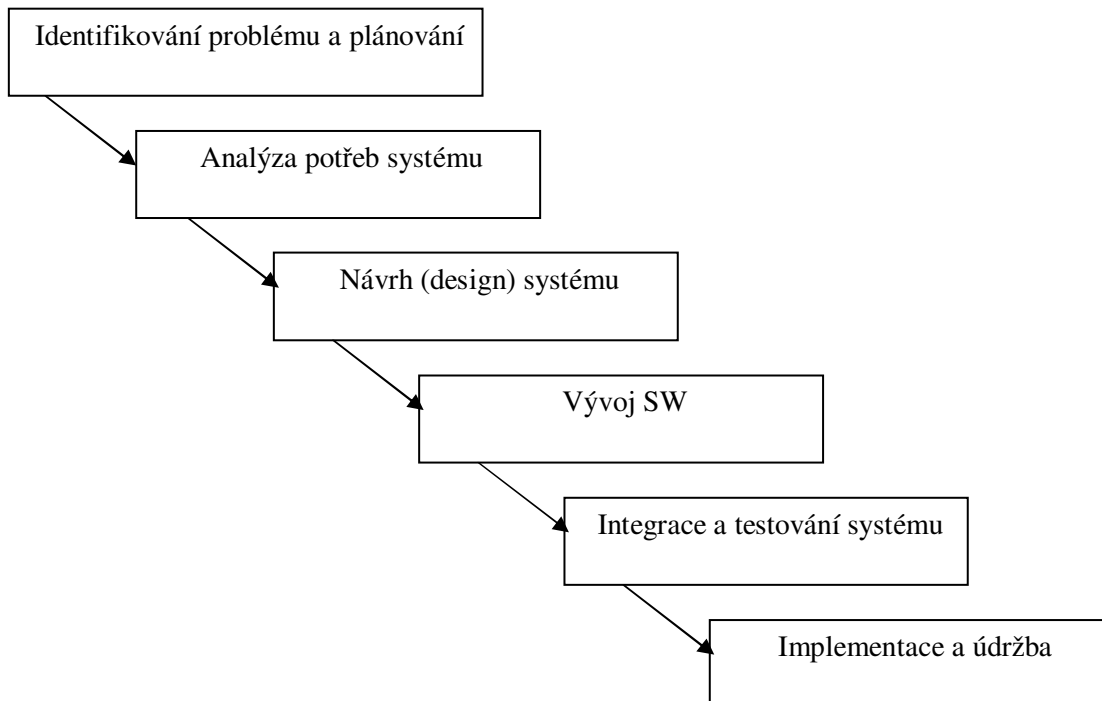
Je používán u systémových analýz vývoje informačního systému, zahrnující požadavky, validaci (ověření), proškolení a používání vlastnických práv přes průzkum, analýzu, design (návrh), implementaci a údržbu. SDLC je také znám jako vývojový, informační a systémový vývoj nebo jako vývoj na požádání.

SDLC může být výsledkem vysoké kvality, který se setkává nebo překonává zákaznicko očekávání v čase, nákladech, pracovní efektivity a je levný na údržbu.

SDLC je systematický přístup k problému řešení a je sestaven z několika fází. Přičemž každá fáze obsahuje několik kroků.

5.1 Etapy SDLC

Prvotní je vždy definování problému a odpověď na otázku, co chci, aby systém uměl. Systém analýzy a návrhu je systematickým přístupem identifikováním problémů, příležitostí a objektivní analýza informačního toku v organizaci a navrhování IS, který bude řešit problémy. Konstrukce pro systematický přístup je nazýván system development live cycle (SDLC). Cyklus může být rozdělen do 3 až 20 částí. Ing. Pavel Jirava, PhD. Rozdělil SDLC do 6 sekvenčních částí, viz obrázek.



Obrázek 6: Etapy SDLC

5.1.1 Identifikování problému a plánování

(Prvotní studie - Feasibility Study)

Jedná se o definování a pojmenování problému a plánování.

Fáze první je velice důležitou pro budoucí vývoj systému ŽC. Zda dojde v této fázi projektu k jakékoli chybě, ale i v pozdějších fázích se vyskytují problémy, které musí být napravené. Proto je tato fáze často označována za kritickou z pohledu projektu. Tato studie by měla odpovědět na otázky managementu ohledně IS.

Jedná se o otázky typu:

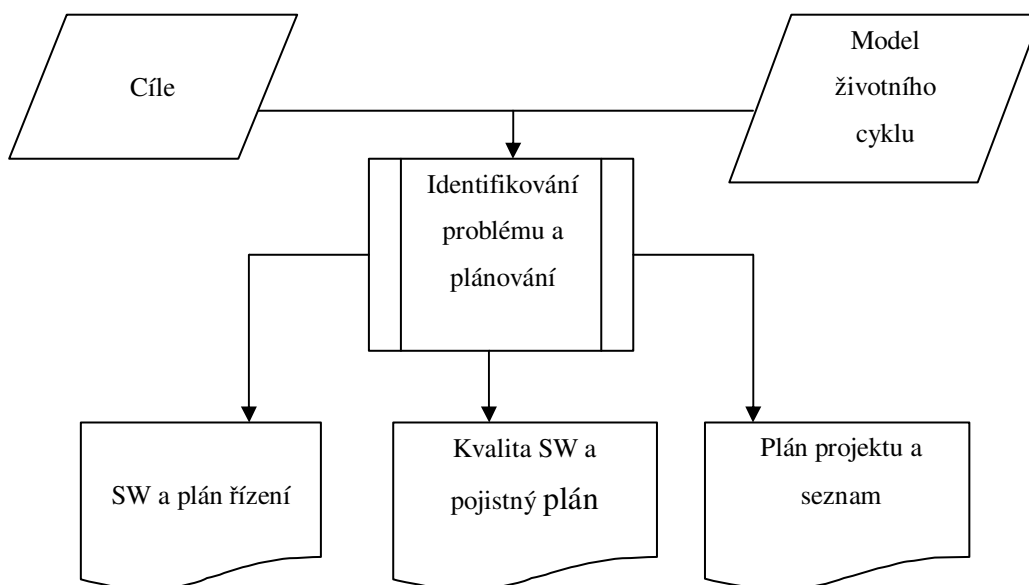
- Co očekáváme od systému.
- Jaký je cíl systému.
- Stanovení podmínek financování.

- Stanovení podmínek z hlediska času.
- Stanovení podmínek z hlediska konečného výsledku.
- Jaká bude originalita systému a jeho rozdílnost.
- Jak často může být existující systém obnovován (update) a inovován.

Na všechny tyto otázky by měla odpovědět prvotní studie. Specialisté zde identifikují požadavky na nový nebo rozšiřující systém. Tyto požadavky jsou uvedeny do projektu. Velmi často potřeby IS vyplývají z požadavků.

Tato fáze obsahuje plán rozvrhu vývoje nového systému. V menších organizacích je určeno, který systém se bude vyvíjet. Zahrnuje základní strukturu modelu životního cyklu, cíle, jaké jsou příležitosti a riziky spojené s projektem a případně popisuje management a technické přístupy.

Výstupem této fáze je uspořádaný plán řízení,



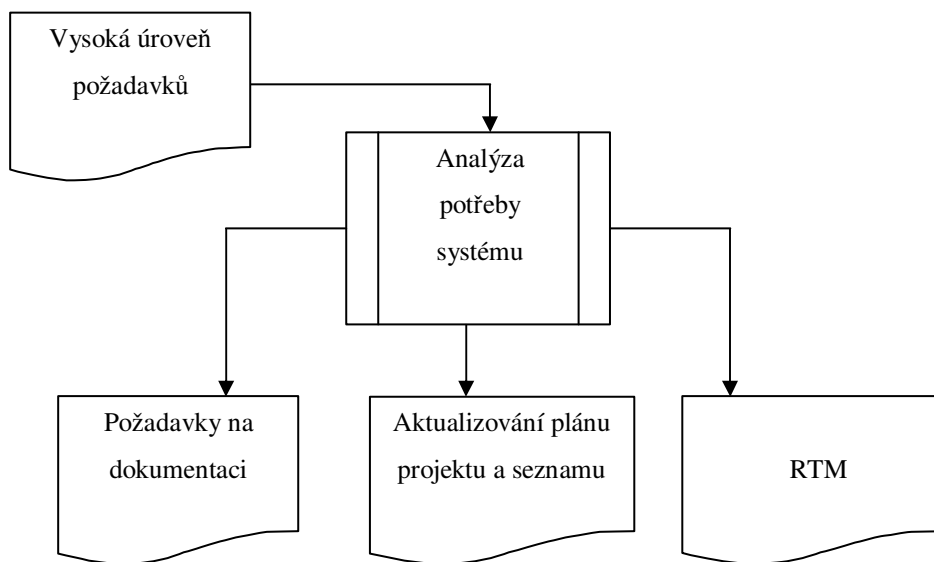
Obrázek 7: Plánovací fáze SDLC [18]

5.1.2 Analýza potřeb systému

Při analýze systému jsou použité znalosti z předcházející fáze jako je informace o prostředí, ve kterém se bude nacházet nový nebo rozšířený IS, dále informace o struktuře systému a jeho elementárních částech.

Dochází k systematizaci požadavků získaných z identifikace problému. Je vytvořen seznam vstupů (= funkčních požadavků), dále seznam událostí a různé varianty reakcí na ně a samozřejmě seznam výstupů. Na základě těchto seznamů se analyzují náklady na systém, to, zda se projekt vyplatí či vrátí. RTM (requirements treasury matrix) je tzv. matice požadavků, tyto požadavky jsou podrobně rozebrány a roztrženy. Každý cíl je uveden zvlášť a věnuje se mu zvláštní pozornost.

Jeden z nástrojů pro získání znalostí budoucího použití systému slouží tzv. Diagram datových toků (DFD data flow diagram viz. příloha).



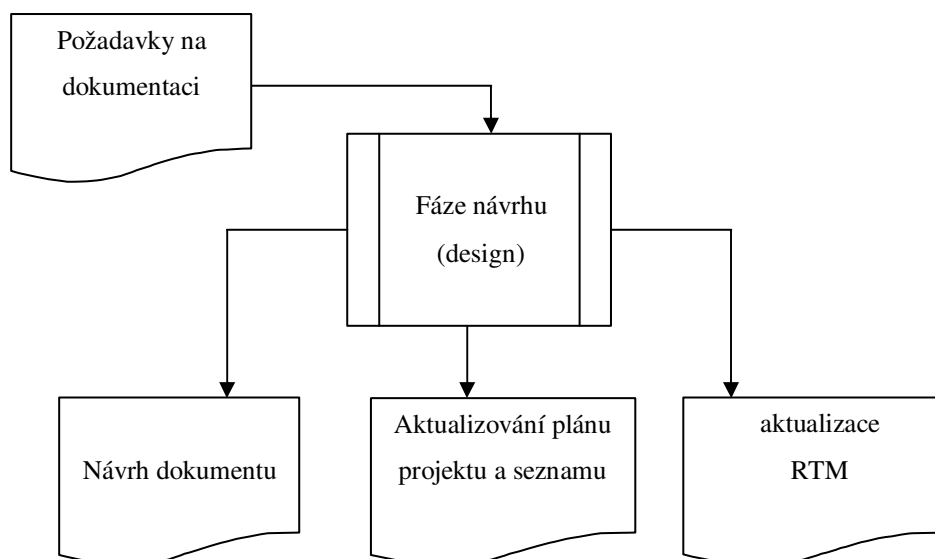
Obrázek 8: Analýza potřeby systému [18]

5.1.3 Návrh (Design) systému

Zatímco se analýza především zabývá tím, co je třeba udělat, design se snaží na základě analýzy určit nejvhodnější způsob, jak to udělat. Některé zdroje dělí design systému na dvě základní části, a to systémový a objektový návrh. Hlavním cílem systémového designu je rozdělit systém na subsystémy, alokovat (rozmístit) subsystémy na HW a SW prvky a vytvořit koncepci detailního návrhu. [7]

V systémovém designu se uvažují mimo modelů systému také možná výpočetní prostředí a možná vývojová prostředí. V tomto návrhu jsou určeny typy architektů, kterými je objektivě řízený systém, systém v reálném čase atd. Systém se člení na podsystémy (vrstvy subsystému), definují se paralelní prvky a činnosti. V systému je nutné definovat chování při mezních stavech. Mezní stav může nastat při uvedení systému do provozu, pádu systému a ukončení běhu systému.

Během třetí fáze SDLC je také vyvíjena efektivní dokumentace softwaru, která musí zahrnovat procesní manuál, on-line pomoc, „často kladené otázky“, čti můj soubor. Dokumentace říká uživatelům, jak používat systém a případně, co dělat, když se vyskytnou nějaké komplikace



Obrázek 9: Fáze návrhu [18]

Vstupem fáze jsou definované požadavky v dokumentaci. Pro každý požadavek je navrženo jeden nebo více elementů, z kterých se vybere výsledný, pomocí interview, workshopy,

Návrh elementů obvykle zahrnuje SW vzhled (detailně), diagramy, tabulky obchodních pravidel, procesní diagram, pseudokód, celkové vztahy diagramů s ucelenými daty.

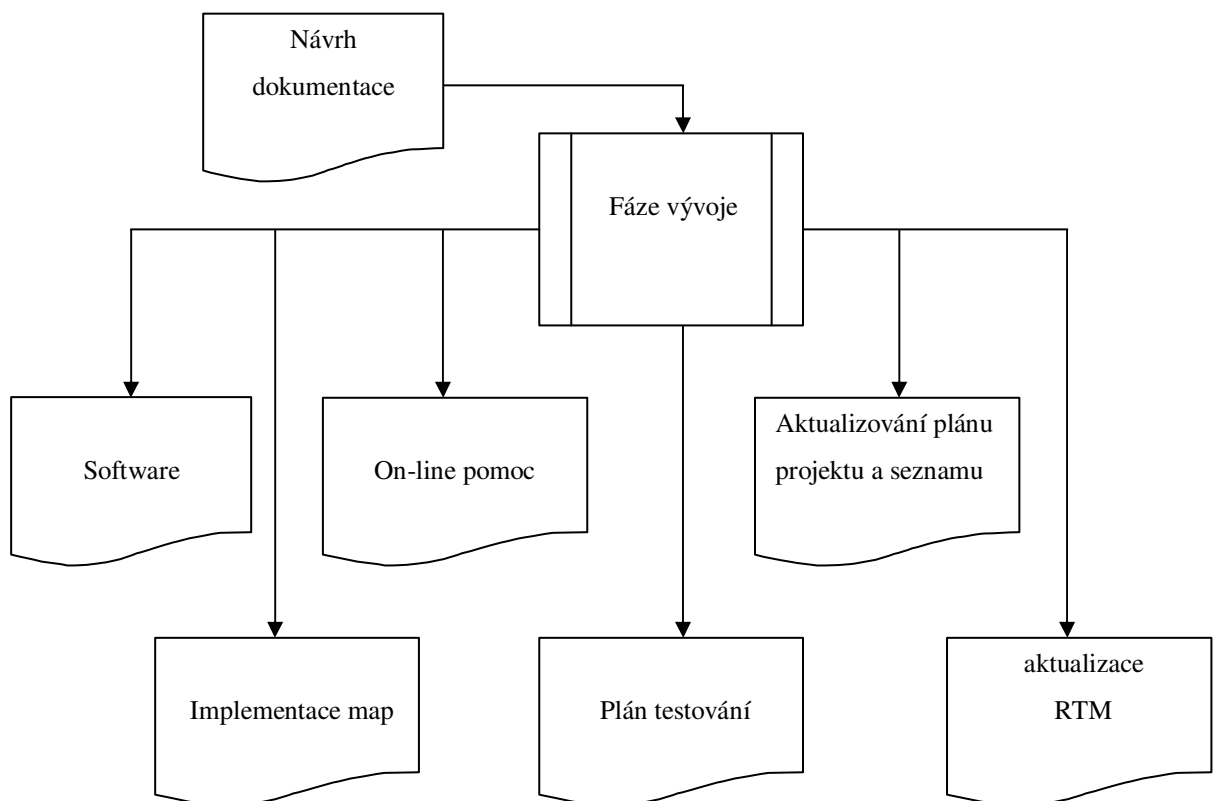
Když návrh dokumentu je ukončen a akceptován provede se aktualizace RTM a každý návrh prvku je porovnán s požadavky.

Výstupem návrhu SW je navržená dokumentace, aktualizovaný plán projektu a RTM.

5.1.4 Vývoj softwaru

V této fázi SDLC je analyzována práce s programátory vývoje softwaru. Software by měl být originální a splňovat potřeby.

Velmi důležitou úlohu mají programátoři, oni navrhují, kódují a odstraňují syntaktické chyby z počítačového programu.



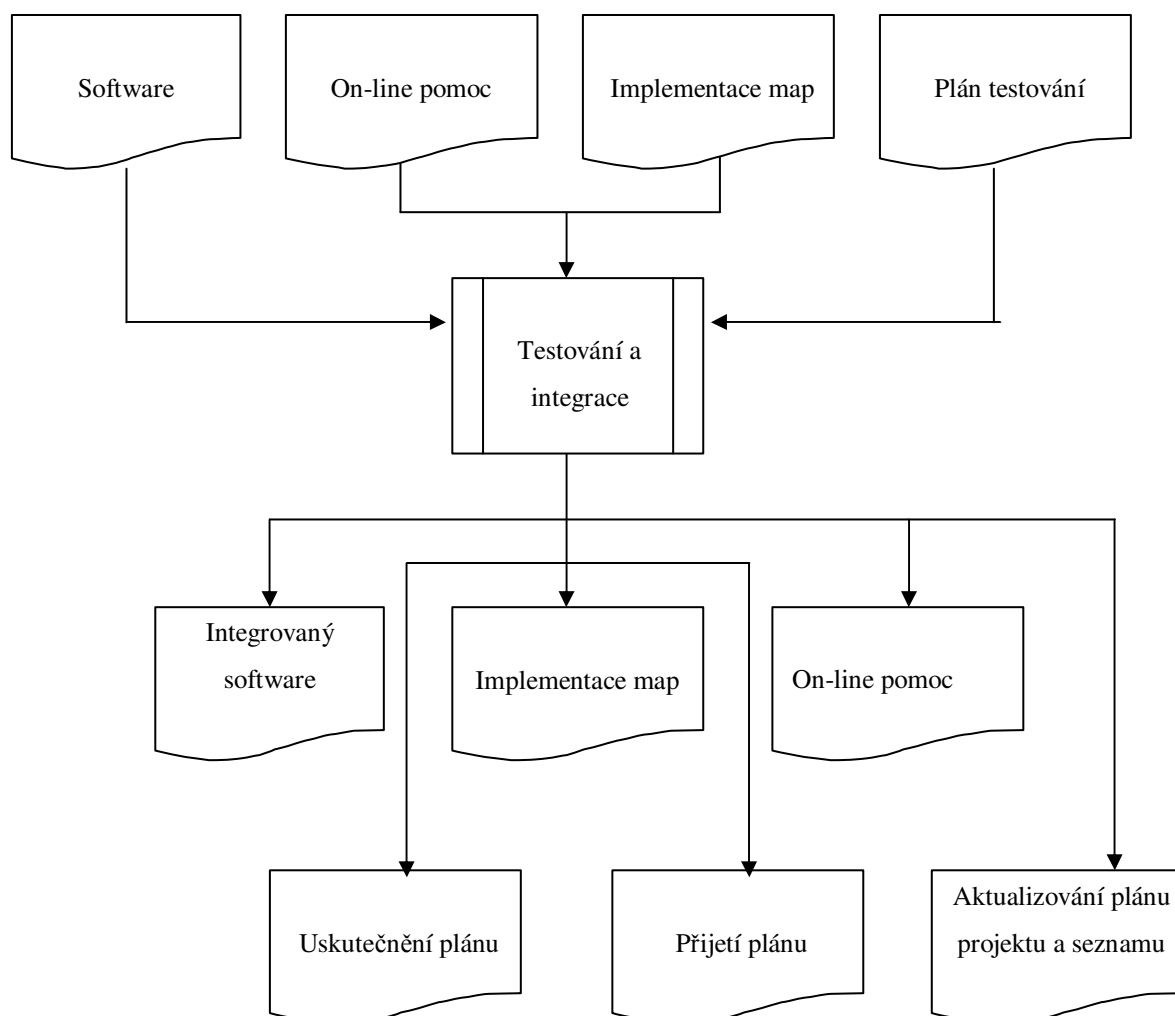
Obrázek 10: Fáze vývoje SDLC [18]

Primárním vstupem fáze vývoje je návrh dokumentace, která je schválená.

Výstupem je plný funkční set softwaru, který splňuje požadavky a shoduje se s návrhem, možná on-line pomoc (popsání jednotlivých operací softwaru), implementační mapy (identifikace primárního kódu pro vstup k funkčnímu systému), plán testování, kde jsou popsány jednotlivé testovací případy, update RTM a plánu projektu.

5.1.5 Testování systému

Po sestavení systému je nutné zjištění, zda opravdu funguje a splňuje dané požadavky. Tuto fázi nazýváme „Testovací“. Testování se provádí před implementací z mnoha důvodů např. rychlejší reakce na chybu a tím i na její nápravu. Pokud je odhalena chyba, program se vrátí zpět do rukou programátorů. Ověření kompletnosti a korektnosti systému.



Obrázek 11: Testování a integrace SDLC [18]

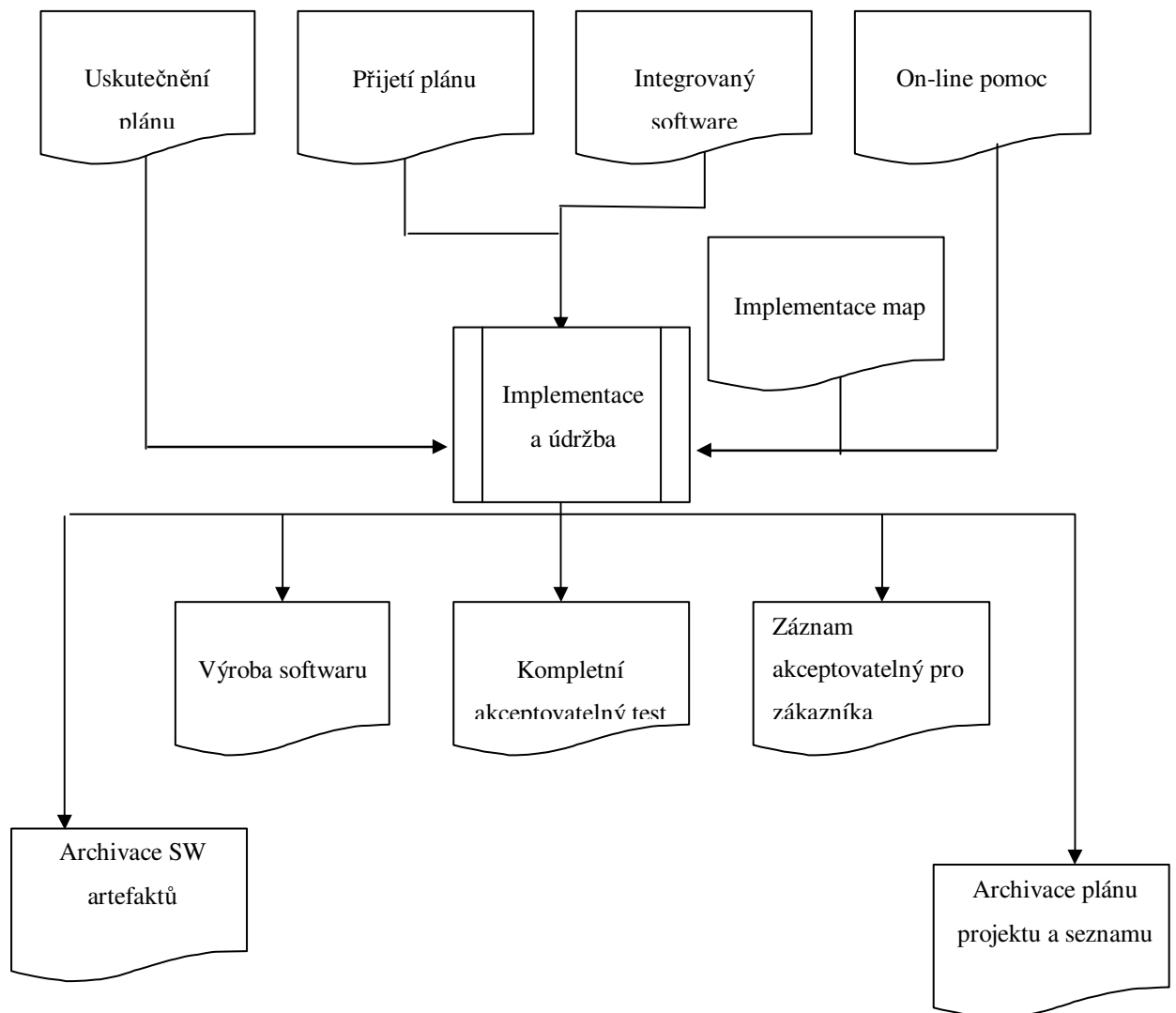
Vstupem je software, on-line pomoc, implementace map a plán testování..

Výstupem předposlední fáze SDLC je integrovaný set SW, on-line pomoc, plány uskutečnění a aktualizace plánu projektu.

5.1.6 Implementace a údržba

Implementace a údržba je poslední fází systému vývoje. Předšlé analýzy pomáhají uvádět do chodu IS. Implementační fáze zahrnuje školení přímých uživatelů. Některá školení jsou prováděna u prodávajících, ale veškerá odpovědnost školení spadá pod fázi analýzy systému. Tato fáze udává, jak převést starý systém na nový co nejefektivněji a při co nejmenší možné ztrátě času. Existuje několik způsobů implementace, např. přímé přepnutí na nový systém (společnost chce rychle vyměnit starý systém za nový, velmi často se implementace provede během víkendu, kdy zaměstnanci jsou mimo kancelář. Tento přístup ale může být velmi riskantní. Po všech testech nemusí být systém funkční, jak předpokládáme), paralelní konverzace (souběžná, společnost instaluje nový systém současně, když starý systém stále funguje. Organizace postupně přesouvá své zaměstnance na nový systém. Možné riziko ze strany zaměstnanců, neochota přejít na systém nový – starý systém znají.), pilotní testování (během počátečního testování společnost instaluje nový systém jen na určitou část organizace nebo na jedno oddělení, pokud se systém osvědčí je instalován všude).

Po implementaci je nutná údržba. 80% celkových nákladů IS jde na údržbu. Údržba zahrnuje opravy chyb, zálohování dat, pomoc uživatelům. Některé z chyb nelze detekovat v testovací fázi, např. když zaměstnanci začínají používat nový systém, tak přicházejí chyby. Oprava těchto chyb je jednou z nejdůležitějších částí údržby. [11], [12]



Obrázek 12: Implementace a údržba [18]

5.2 Závěr SDLC

Předepsaná struktura u SDLC, je specifikovaná návrhem maximální pravděpodobnosti úspěchu vyvíjeného SW. Pro dosažení úspěchu a vložení co největšího úsilí do SDLC existují čtyři základní koncepty.

- Oblast omezení (Scope restrict)
- Progresivní vzestup (Progressive enhancement)
- Před- definovatelná struktura (pre-defined structure)
- Inkrementální (přírůstkové) plánování (Incremental Planning)

Tyto čtyři koncepty pomáhají k předcházení rizik během vývoje softwaru.

5.2.1 Oblast omezení

Projektová oblast je založena na základě požadavků zákazníka, na známosti cílů a jejich začlenění do projektového plánu. Cíle jsou postupně vyříděny do požadavků, dále do návrhu elementů a nakonec do artefaktů softwaru.

Tato hierarchie cílů. Požadavků, elementů a artefaktů jsou zdokumentovány v RTM, který slouží jako kontrola elementů, které omezují projekt původní definované oblasti. Účastníci projektu jsou omezeni adresností těchto požadavků, elementů a artefaktů, které jsou přímo patrné z cílů produktu. Tato prevence značně sníží poruchovost projektu.

5.2.2 Progresivní vzestup

Každá fáze SDLC bere výstup vezme výstupy z předešlé fáze a ty jsou vstupy pro další fáze. Dodatkové informace jsou tímto seskupeny, využití specifických metod v každé fázi. Pak výsledkem jsou výstupy z předešlých fází, u kterých progresivně stoupají dodatečné informace. Tato fáze nám podrobně rozebere každý element. Zakládá se na vzrůstající práci a dodatečné informace můžeme použít i v pozdějších fázích. Nové požadavky jsou formálně mimo vývojový tým pro pozdější srovnání. Výsledkem této fáze je zaměření účastníků projektu na udržení a přesného dodržení původních cílů projektu. Tím se minimalizuje výskyt chyby.

5.2.3 Předdefinování struktury

Každá fáze má předdefinovanou sadu standardních procesů jako je iterační, informální a jiné modely. Účastníci projektu rychle ustálí proces do jednotlivých fází předem určeného modelu, model se vybere podle vývoje, požadavků nebo kultury projektu. Předdefinovaná struktura se utváří podle znalostí z minulosti, současnosti a z toho, co očekáváme od blízké budoucnosti. Toto umožňuje vyšší úroveň komfortu a způsobí lepší spolupráci mezi účastníky. Účastníci poskytují potřebné informace nebo zpětnou vazbu opakovaně. Tím nechybí komunikace mezi jednotlivými články. Toto vynaložené úsilí se navrátí v příštích fázích.

5.2.4 Inkrementální plánování

V přírůstkovém plánování je minimalizované překvapení. Zvyšuje přesnost, poskytuje informace, Poskytuje informace o odchylkách od plánu tak brzy, jak je v SDLC možné. V SDLC se navrhuje každá fáze detailně a omezuje pozdní plánování fází. Předem zná výstup fází, což zvýší celkovou úroveň. Návrh plánu je kompletně aktualizován v každé fázi, průběžně se aktualizují i náklady a důležitá data v kombinaci s odhady dalších aktivit v příštích fázích. Otázky a aktivity dalších fází jsou definovány pouze po skončení probíhající fáze. Výstup každé fáze musí být měřitelný. Toto dovoluje plánování výroby s vysokou přesností v každé fázi. Přímé zkušenosti ukazují, že se jedná o velmi složitý vývoj, více než ukazují odhady předvídatelných struktur a úrovně úsilí pro výstup fází.

Odhady pro výstup z fází zahrnují závěrečné odhady nákladů a seznamů dat. Výsledkem tohoto konceptu je celkový návrh odhadu, který je naprosto přesný. Toto dá zákazníkovi příležitost k porovnání shody projektu a požadavků popřípadě opravy. Zákazník se aktivně účastní projektu.

6 TREND PŘÍSTUPU K ŽIVOTNÍMU CYKLU IS

Objektově orientované metodiky a technologie jsou dle mého názoru trendem přístupu k životnímu cyklu.

Dá se říci, že OMT (model orientovaný na předmět) je jednou z nejoblíbenějších metodik analýzy a návrhu. Proto Metodiku OMT pokládám za trendovou, důkazem toho jsou nové metodiky, které vznikly na základě OMT.

OMT zahrnuje notaci (jednotlivé použitelné diagramy) i popis objektově orientovaného vývoje IS.

6.1 Vlastní analýza sestává ze 3 relativně samostatných modelů:

Každý model popisuje pouze některé aspekty systému. Obsahuje však odkazy a vztahy s ostatními modely. Vztahy modelů jsou velice důležité, protože díky tomu může být metodika zcela kompaktní. [2], [13]

- objektový model (OM - Object Model),
- dynamický model (DM - Dynamic Model),
- funkční model (FM - Functional Model).

6.1.1 Objektový model

Objektový model znázorňuje objektovou strukturu, na které jsou prováděny operace znázorněné v dynamickém a funkčním modelu. Model obsahuje definice tříd a jejich vztahů společně s atributy a metodami. Zachycuje statickou strukturu systému.

6.1.2 Dynamický model

Dynamický model ukazuje akce, které závisí na hodnotách objektů (lépe řečeno jejich atributech). Ty pak způsobují změny hodnot objektů a vyvolávají funkce. Druhý model zachycuje dynamiku objektů a změny jejich stavů. Zabývá se chováním objektů v čase a

tokem zpráv a kontroly mezi objekty. Dynamický model zahrnuje stavové diagramy (STD - State Transition Diagram) pro každou třídu nebo pro důležité části návrhu. Dále pak diagramy interakcí. Součástí DM je také celková mapa událostí (Event Trace Diagram) a diagram událostí (model událostí, v angličtině se nazývá Event Schema nebo i Event-flow Diagram).

6.1.3 Funkční model

Funkční model popisuje funkce vyvolané operacemi (metodami) v objektovém modelu a akcemi v dynamickém modelu. Funkce zpracovávají hodnoty dat definovaných v objektovém modelu. Poslední ze tří modelů popisuje funkční závislosti systému, je podobný diagramu datových toků. Popisuje, co systém dělá (ale nezabývá se tím, jak to dělá).

6.2 Jaké diagramy metodika OMT může obsahovat

Dalšími diagramy metodiky OMT jsou např. use CASE tzv. model jednání a slovní scénáře, Diagram datových toků tzv. Data Flow Diagram nebo např. mapa událostí.

6.2.1 Use case

Model jednání je využíván při definování požadavků, specifikace chování a určení hranic systému. Use case se skládá i ze slovního popisu, který se sestavuje následujícím způsobem. Na počátku sepíšeme seznam událostí probíhajících v dané funkci/use case, které se číslovají. Následně je seřadíme dle pořadí, ve kterém nastávají. Paralelní události a podudálosti se číslovají víceúrovňově. Mohou se sepsat do tabulky, kde se lépe přiřadí jednotlivým událostem či podudálostem vstupy a výstupy. Dále je celý proces popsán slovy, na závěr napsán důsledek plus možná rizika, která mohou nastat.

6.2.2 Diagram datových toků

Diagram datových toků (**Data Flow Diagram**) se skládá z procesů, datových toků, aktérů a data stores. Proces transformuje vstupní data na výstupní. Proces se znázorňuje elipsou. Datový tok zachycuje tok dat mezi procesy. Znázorňuje se šipkou. Pro prvky okolí se původně v diagramech datových toků používalo pojmenování terminátor, v objektové

terminologii OMT jsou aktéři. Znázorňuje se obdélníkem. Data stores pak slouží pro dočasné uložení dat. Znázorňuje se pomocí dvou rovnoběžných čar, mezi kterými je napsán název skladu. Aktoři jsou objekty, data stores mohou být také objekty. (**Aktor** - označení uživatele v diagramech use case. Může to být osoba, jiný systém, hardware, plynutí času.) [8]

Diagram datových toků je modelem hierarchickým. Nejvyšší diagram se nazývá kontextový. V něm jsou uvedeni pouze aktéři, systém jako celek jakožto jediný proces a datové toky mezi aktéry a systémem. Pod kontextovým diagramem se nachází diagram nulté úrovně. Vzniká rozpadem funkce systému jako celku na několik základních procesů (jejich počet se odvíjí od nároku na přehlednost diagramu). Každý proces v nulté úrovni nese před svým jménem číslo. Každý z těchto procesů se pak rozpadá do několika “podprocesů” v diagramech první úrovně. Podprocesy pak nesou číslo svého nadprocesu a za tečku přidávají své vlastní číslo v rámci své úrovně (např. proces 2.3.2 ve druhé úrovni hierarchie).

6.2.3 Mapa událostí

Dalším modelem je mapa událostí. Jejím cílem je zobrazit všechny události, které třídy zpracovávají nebo vysílají. Mapa událostí se podobá objektovému modelu, jsou v ní však pouze třídy, které se vyskytují v diagramech interakcí. Třídy se znázorňují klasicky obdélníky, události šipkami (udávají směr události).

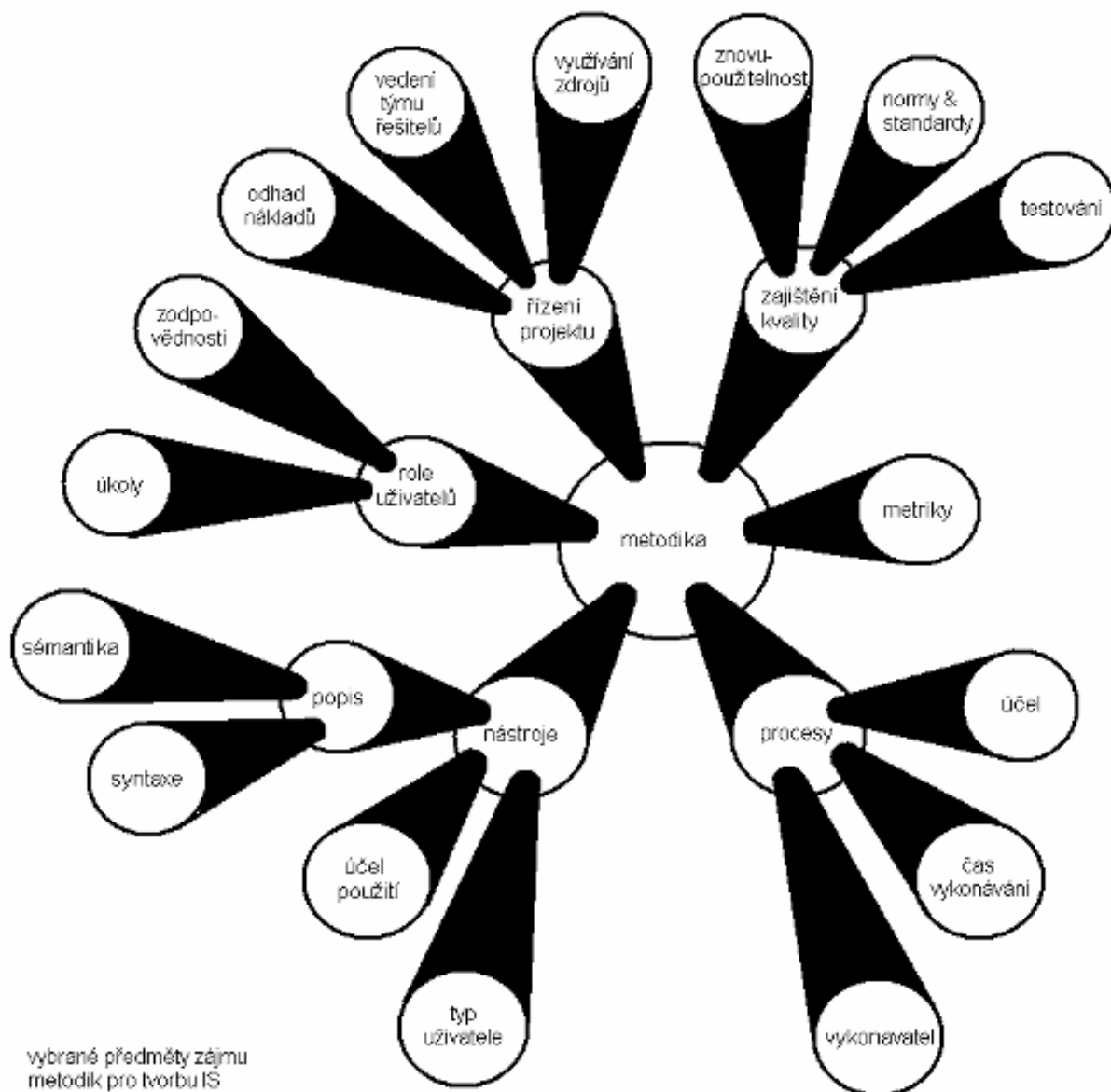
Diagram událostí popisuje všechny činnosti systému jako reakce na nějaké vstupy. Cílem je identifikovat všechny aktivity. Jedná se o síť událostí a aktivit. Aktivity jakožto uzly diagramu se znázorňují obdélníky se zaoblenými rohy, události se znázorňují jako šipky. Celý diagram má jeden vstup a jeden výstup. Mezi základní vlastnosti modelu patří fakt, že popisuje systém bez ohledu na jeho objektovou strukturu. Dále nerespektuje hranice systému (popisuje aktivity systému i okolí, např. uživatele). Události se v diagramu mohou větvit. Větvení událostí se označuje písmenem “S” v kroužku (z anglického “split”), slévání událostí se označuje “M” v kroužku (z anglického “merge”). Větvení a slévání událostí je projevem nějaké činnosti, která se ale v diagramech uvádět nemusí, protože má výhradně technický charakter. Větvením pak vznikají paralelní události (větve), nebo mají události ve větvích k sobě vztah “exclusive or” (tj. dojde pouze k jedné z nich). Závěrem k

diagramu událostí je nutné zmínit, že je možné je vytvářet a uspořádat hierarchicky (obdobně jako DFD). Toho lze využít u složitých aktivit.

6.2.4 Metodika tvorby systému

Tvorba informačního systému s použitím objektového přístupu (ale i bez něj) stojí především na vytváření nejrůznějších modelů. Modelujeme tak například interakci systému s okolím, komunikaci objektů jako nosný prostředek realizace dynamiky aplikace a jako podrobný model můžeme chápat i zdrojový kód. Samotné modely můžeme ze zřejmých důvodů prezentovat a vytvářet ve formě grafických diagramů, jejichž sémantika (význam) a syntaxe (zápis) je zpravidla jasně vymezena. Souhrnně mluvíme o tzv. notaci.

V případě použití objektového přístupu nemusí být, vzhledem k povaze používaných nástrojů, jasná hranice mezi jednotlivými etapami. Obecné schéma posloupnosti analýza/design/implementace/validace však zůstává zachováno. Právě v souvislosti s používáním objektových technologií se nejvíce osvědčil iterativní přírůstkový životní cyklus, jehož filosofie staví na práci s relativně malými částmi systému, které jsou vždy řešeny a zpracovávány přes všechny etapy zkráceného životního cyklu. Celý vývoj IS je tak vlastně neustálým koloběhem klíčových etap při uvažování stále většího množství dílčích celků. Samozřejmě tento model iterativního životního cyklu víceméně padá v případě, kdy není uvažována tvorba kompletního IS, tj. například v již diskutovaném případě analýzy prováděné samostatně. Zatímco při klasickém tzv. vodopádovém modelu byla snaha uzavírat definitivně jednotlivé etapy, při iterativní metodě jsou jednotlivé etapy průchodné oběma směry. [13], [4]



Obrázek 13- Vybrané předměty zájmu metodik pro tvorbu IS [14]

OMT také popisuje fáze objektově orientovaného vývoje informačního systému. Mezi ně patří:

- **Analýza**

Zde je nutné porozumět a namodelovat část reality, kterou bude systém obsahovat.

- Systémový design – určuje celkovou architekturu systému (plus podsystémy, paralelní úkoly, uložení dat, ...).
- Objektový design optimalizuje modely analýzy z hlediska konceptu implementace.
- Implementace obsahuje programování vlastního počítačového systému.
- Testování probíhá průběžně ve fázích inkrementálního vývoje

OMT dále detailněji popisuje kroky, které by měly být provedeny v jednotlivých fázích objektově orientovaného vývoje IS. Ve fázi analýzy je třeba vytvořit objektový, dynamický a funkční model.

Tvorba objektového modelu by měla obsahovat následující kroky:

- Vytvořit slovní popisy modelovaného problému.
- Určit třídy objektů.
- Zrušit nepotřebné a chybné třídy.
- Připravit knihovnu znalostí (data dictionary)
- Určit asociace mezi třídami.
- Zrušit nepotřebné a chybné asociace.
- Určit atributy tříd.
- Zrušit nepotřebné a chybné atributy.
- Určit vazby dědičnosti.
- Projít vše znova a určit nedostatky.

Tvorba dynamického modelu by měla obsahovat tyto kroky:

- Určit use cases a připravit scénáře typických interakčních sekvencí.
- Určit události mezi objekty a připravit mapu událostí pro každý scénář.
- Vytvořit diagram událostí systému.
- Vytvořit stavové diagramy pro třídy s významným dynamickým chováním.
- Zkontrolovat konzistenci a úplnost událostí sdílených mezi stavovými diagramy.

Tvorba funkčního modelu zahrnuje následující kroky:

- Určit vstupní a výstupní hodnoty.
- Vytvořit diagramy datových toků pro vyjádření funkčních závislostí.
- Popsat každou funkci, co dělá.
- Určit omezení.
- Specifikovat optimalizační kritéria.

7 NÁVRH ŽIVOTNÍHO CYKLU PRO IS V OBLASTI VEŘEJNÉ SPRÁVY

7.1 Co je ISVS

Informační systémy veřejné správy (dále jen „ISVS“) představují významný nástroj výkonu veřejné správy na všech úrovních. S ohledem na potřebu zajistit transparentnost, vzájemnou kompatibilitu, komunikaci a určitý stupeň jednotnosti všech ISVS, je kladen velký důraz na řízení této oblasti.

Informační systém veřejné správy (dále jen „ISVS“) má životní cyklus podobný jako klasický IS např. pro podnik. Přesto se, ale liší v jednotlivých fázích a zpracování, je zde kladen větší důraz na dokumentaci a vše, co s ní souvisí např. aktualizace.

7.2 Standardy ISVS pro náležitosti životního cyklu informačního systému

Při vývoji životního cyklu se používají standardy a metodiky ISVS. Jedná se o soubory pravidel pro výkon odborných činností spojených s vytvářením, rozvojem a využíváním informačních systémů veřejné správy. Jedním z nejvýznamnějších je "Standard ISVS pro náležitosti životního cyklu informačního systému". Jeho cílem je:

- Zajistit řízení a zejména kvalitní organizaci a kontrolu projektů rozvoje IS v rámci organizace správce ISVS.
- Zajistit kvalitní řízení vývoje, provozu a údržby IS jako celku.
- Připravit věcný podklad pro atestace IS nebo projektů jejich rozvoje.
- Poskytnout správcům IS veřejné správy možnost efektivněji kontrolovat dodavatele komplexních řešení (zejména externí subjekty) rozšířením počtu a přesnou definicí kontrolních bodů, ve kterých lze zpracováváný projekt ovlivnit nebo v krajním případě zastavit tak, aby nedocházelo k dalším zbytečným finančním a časovým ztrátám.

Standard předepisuje základní kroky, které musí být provedeny v průběhu životního cyklu informačního systému a zejména základní strukturu dokumentů, které musí být v průběhu životního cyklu informačního systému vypracovány a pravidelně aktualizovány.

7.2.1 Právní předpisy České Republiky se vztahem k ISVS

Právní předpisy platné pro oblast řízení ISVS jsou uvedeny v tabulce č. 1. Základním právním předpisem je zák. č. 365/2000 Sb., o informačních systémech veřejné správy, ve znění pozdějších předpisů (dále též jen "zákon"), od něhož se odvíjejí všechny právní předpisy nižší právní síly a další nástroje používané Ministerstvem vnitra. [19]

Předpis (poslední novela)	Zkrácený název	Název
<u>zák. č. 365/2000 Sb.</u> (81/2006) ¹	o informačních systémech veřejné správy	O informačních systémech veřejné správy a o změně některých dalších zákonů
<u>vyhl. č. 529/2006 Sb.</u> (529/2006)	O dlouhodobém řízení ISVS	o požadavcích na strukturu a obsah informační koncepce a provozní dokumentace a o požadavcích na řízení bezpečnosti a kvality informačních systémů veřejné správy (vyhláška o dlouhodobém řízení informačních systémů veřejné správy)

Předpis (poslední novela)	Zkrácený název	Název
vyhl. č. 530/2006 Sb. (530/2006)	o dlouhodobém řízení ISVS	O požadavcích na strukturu a obsah informační koncepce a provozní dokumentace a o požadavcích na řízení bezpečnosti a kvality informačních systémů veřejné správy (vyhláška o dlouhodobém řízení informačních systémů veřejné správy)
vyhl. č. 528/2006 Sb. (528/2006)	o postupech atestačních středisek při posuzování dlouhodobého řízení ISVS	o postupech atestačních středisek při posuzování dlouhodobého řízení informačních systémů veřejné správy
vyhl. č. 528/2006 Sb. (528/2006)	o informačním systému o informačních systémech veřejné správy	o formě a technických náležitostech předávání údajů do informačního systému, který obsahuje základní informace o dostupnosti a obsahu zpřístupněných informačních systémů veřejné správy
vyhl. č. 469/2006 Sb. (469/2006)	o informačním systému o datových prvcích	o formě a technických náležitostech předávání údajů do informačního systému o datových prvcích a o postupech Ministerstva vnitra a jiných orgánů veřejné správy při vedení, zápisu a vyhlašování datových prvků v informačním systému o datových prvcích

Předpis (poslední novela)	Zkrácený název	Název
Vyhl. Č. 52/2007 Sb. (52/2007)	o postupech atestačních středisek při posuzování vazeb ISVS	o postupech atestačních středisek při posuzování způsobilosti k realizaci vazeb informačních systémů veřejné správy prostřednictvím referenčního rozhraní
Vyhl. č. 53/2007 SB. (53/2007)	o referenčním rozhraní	o technických a funkčních náležitostech uskutečňování vazeb mezi informačními systémy veřejné správy prostřednictvím referenčního rozhraní
<u>zák. č. 137/2006 Sb.</u> (137/2006)	o veřejných zakázkách	o veřejných zakázkách
<u>zák. č. 81/2006 Sb.</u> (81/2006)	Kterým se mění zákon č. 365/2000 Sb., o ISVS	kterým se mění zákon č. 365/2000 Sb., o informačních systémech veřejné správy a o změně některých dalších zákonů, ve znění pozdějších předpisů, a další související zákony
zák. č. 110/2007 Sb. (110/2007)	o zrušení Ministerstva informatiky	zákon o některých opatřeních v soustavě ústředních orgánů státní správy, souvisejících se zrušením Ministerstva informatiky a o změně některých zákonů

Tabulka 1 přehled předpisů souvisejících s ISVS [20]

Poslední významnější novela zák. č. 365/2000 Sb., která byla přijata jako zák. č. 81/2006 Sb., přináší v oblasti řízení ISVS významné změny, které spočívají především v **přenesení důrazu z jednotlivých ISVS na dlouhodobé řízení celé oblasti**

ISVS jako komplexní dlouhodobý proces probíhající v rámci každého orgánu veřejné správy. U samotných informačních systémů je pak kladen důraz především na kvalitu jejich komunikačního rozhraní, což platí též o provozních informačních systémech, které si s ISVS vyměňují data.

Zákon zmocňuje MV k vydávání prováděcích předpisů v několika oblastech. Z pohledu informační koncepce je nejdůležitější vyhl. č. 529/2006 Sb., vyhláška o dlouhodobém řízení ISVS a vyhl. č. 530/2006 Sb., o postupech atestačních středisek při posuzování dlouhodobého řízení ISVS. Účinnost vyhlášek je stanovena od 1.1.2007. Zejména první vyhláška je stěžejním podkladem a poměrně přesným návodem pro přípravu informační koncepce.

Další změnu pravidel v této oblasti přinesl zák. č. 110/2007 Sb., zákon o zrušení Ministerstva informatiky, kterým byly kompetence v oblasti dlouhodobého řízení ISVS, původně svěřené Ministerstvu informatiky, kompletně převedeny na Ministerstvo vnitra, a to ke dni 1. června 2007. [20]

7.3 Fáze životního cyklu ISVS

Životní cyklus ISVS se skládá ze tří částí. Vztah fází životního cyklu informačního systému, strategických procesů a realizačních projektů IS nebo jeho vztah fází životního cyklu IS, strategických procesů a realizačních projektů IS je uveden v Tabulce 2.

FÁZE ŽIVOTNÍHO CYKLU IS	STRATEGICKÉ PROCESY IS	REALIZAČNÍ PROJEKTY IS
Příprava IS	<ul style="list-style-type: none"> • Definice potřeby IS • Příprava na zpracování nebo aktualizaci informační strategie IS • Příprava nebo aktualizace nástrojů strategického řízení IS 	
Vývoj, provoz a údržba IS	<ul style="list-style-type: none"> • Tvorba a údržba informační strategie • Řízení bezpečnosti • Plánování a koordinace 	<ul style="list-style-type: none"> • Projekty akvizice • Projekty základního postupu vývoje • Projekty provozu a

	projektů <ul style="list-style-type: none"> • Plánování a řízení jakosti • Řízení požadavků a jejich monitorování 	údržby <ul style="list-style-type: none"> • Kombinované projekty
Ukončení činnosti IS	<ul style="list-style-type: none"> • Vyřazení IS 	

Tabulka 2 fáze životního cyklu ISVS [23]

7.4 Návrh životního cyklu ISVS

Celý návrh životního cyklu IS se musí řídit standardem ISVS, tudíž i jednotlivé kroky, které zde popisují budou rozděleny do tří hlavních skupin viz . tabulka. V této kapitole se pokusím jednotlivé fáze přesněji specifikovat a vysvětlit.

7.4.1 Fáze přípravy IS

Před zahájením jakékoli činnosti musíme určit tzv. „správce ISVS“, osobu, která určuje účel a prostředky zpracování informací a za informační systém odpovídá. Fáze přípravy informačního systému obsahuje činnosti a úlohy správce. Definujeme potřeby vytvoření informačního systému a dále pokračujeme přípravou a zpracováním záměru rozvoje a informační strategie. Pokud v organizaci je provozováno více IS musí tato fáze zahrnovat kroky spojené s aktualizací informační strategie a nástrojů strategického řízení. Z důvodu shody požadavků s nově připravovaným systémem.

7.4.1.1 Strategické procesy

Strategické procesy obsahují minimální požadavky na dokumentaci, v případě, že organizace nemá vnitřním předpisem definovanou metodiku pro vývoj, provoz a údržbu IS, vést IS v souladu s pravidly, postupy a činnostmi uvedenými ve standardu ISVS a zpracovávat dokumenty, které tento standard požaduje. Pokud mají vnitřním předpisem definovanou metodiku pro vývoj, provoz a údržbu IS, musí ji upravit tak, aby zahrnovala i pravidla, postupy, činnosti a dokumenty uvedené ve standardu ISVS a řídit IS v souladu s takto upraveným vnitřním předpisem.

- Definice potřeby IS

V této fázi je nutno, aby správce popsal koncepci a zdůvodnil potřebu IS. Koncepce musí být v souladu s principy uvedenými v aktuální verzi dokumentu Státní informační politika a Informační politika resortu (resortu správce) v organizacích. Tento požadavek platí pouze pro organizace správců, kteří jsou povinni se těmito dokumenty řídit.

- Příprava na zpracování nebo aktualizaci informační strategie IS

Správce definuje a analyzuje systémové požadavky IS, do kterých patří definování a analýza zdroje a východiska IS, definování a analyzování výchozího stavu pro rozvoj IS, definování cílového stavu IS, definování prvotního návrhu transformace ze současného stavu IS do cílového.

- Příprava nebo aktualizace nástrojů strategického řízení IS

Správce připraví základní nástroje strategického řízení IS, bude definovat jejich základní principy a připraví úvodní verze dokumentů, kde bude uvedeno: definování základní procedury řízení, bezpečnosti ICT, definování principu plánování a řízení projektu, definování principu řízení jakosti IS, definování principu monitorování a aktualizace požadavky na IS.

7.4.2 Fáze vývoje, provozu a údržby IS

Tato fáze obsahuje činnosti a úlohy správce a provozovatele v různých rolích. V této fázi budou probíhat strategické procesy IS periodicky a realizační projekty IS probíhají jednorázově.

7.4.2.1 Strategické procesy

obsahují tvorbu a údržbu informační strategie, kterou správce musí aktualizovat pravidelně (nejméně jednou za dva roky) a např. pokud dojde ke změně legislativy.

- Řízení bezpečnosti

Správce řídí bezpečnost IS podle postupů bezpečnostní politiky IS. Opět ji musí aktualizovat po 2 letech.

- Plánování a koordinace projektů

Tuto fázi opět spravuje správce e vede o ní požadovanou evidenci. Dále správce vede plánování a řízení jakosti, řízení požadavků a jejich monitorování. Správce musí evidovat požadavky na IS v dokumentu Systémové požadavky.

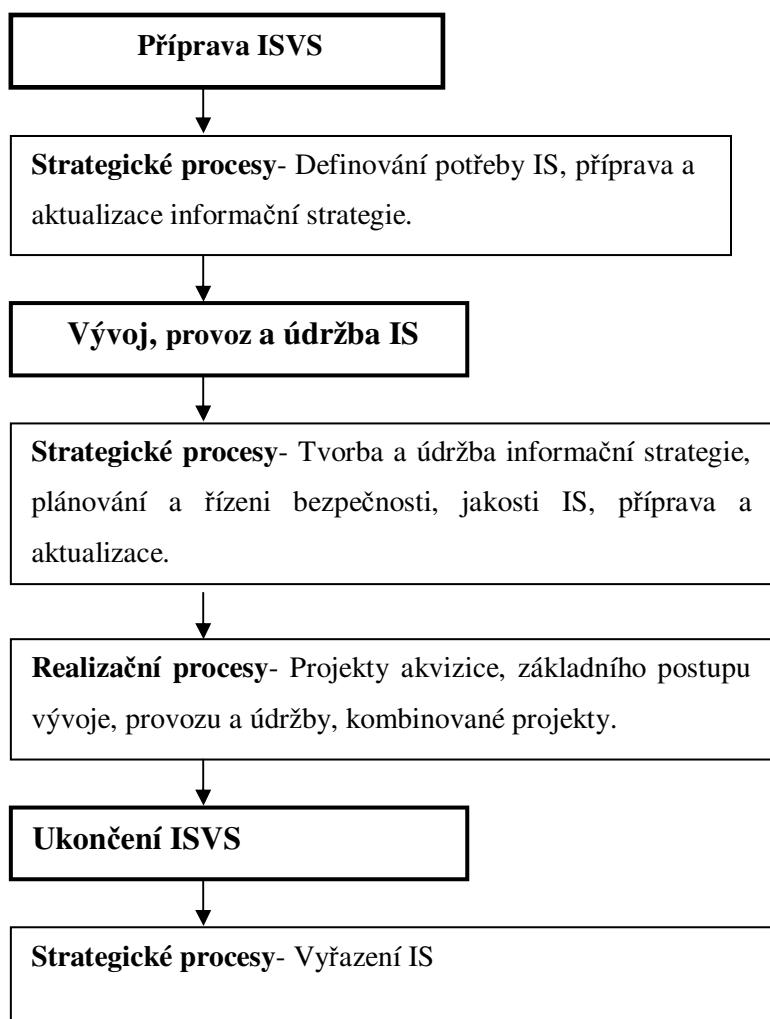
7.4.2.2 Realizační projekty

Standardem ISVS se nepředepisuje použití specifické metody vedení realizačních projektů . Je nutné definovat specifickou metodiku k vedení realizačních projektů IS. Standard ISVS definuje pět druhů projektů: Projekty akvizice (získání IS od externího dodavatele), Projekty vývoje (vývoj IS např. jeho část, softwarový produkt nebo služba. Může se zde i IS modifikovat nad rámec běžné údržby), Projekty základního postupu vývoje, (vypracovává se, pokud nejsou splněny podmínky pro použití postupu redukovaného projektu), Projekty redukovaného postupu vývoje (jde o projekty, ve kterých se vyvíjí IS, jeho část, softwarový produkt nebo služba. Jedná se o projekty, u kterých výsledek nevyžaduje změnu právního předpisu nebo výsledek projektu má jednoduchou strukturu a omezené vazby na jiné IS popř. je zcela izolovaný), Projekty provozu a údržby (v tomto projektu se provozuje IS nebo jeho část nebo se poskytuje provozní podpora uživatelům, můžeme sem zahrnout také drobné opravy systému, které zásadně nemění jeho funkčnost nebo datové rozhraní. Dále tento projekt zahrnuje činnosti spojené s migrací systému nebo jeho části na novou verzi, které zásadně nemění jeho funkčnost nebo datové rozhraní a které jsou řešeny v rámci provozních činností IS) a Kombinovaný projekt (jedná se o kombinaci dvou či více z výše uvedených projektů, do této kategorie patří např. projekty, ve kterých je proces akvizice kombinován např. se souběžně probíhajícím procesem vývoje realizovaným vlastními silami správce).

7.4.3 Fáze ukončení činnosti IS

Strategické procesy obsahují pouze jednu fázi a to vyřazení IS. Vyřazení může být z důvodu uplynutí určeného období nebo IS je nahrazován novou verzí (migrace), což je součástí procesu provozu. O ukončení činnosti IS musí být zpracován plán. Do plánovacích činností se musí zapojit všichni uživatelé.

7.5 Grafické vyjádření životního cyklu ISVS



Obrázek 14: Životní cyklus ISVS [vlastní]

8 ZÁVĚR

Po dokončení mého důsledného sběru a prostudování všech materiálů potřebné k sepsání bakalářské práce na téma trendy v přístupech k životnímu cyklu vývoje IS jsem zjistila, že pokud chci sestavit opravdu funkční a kvalitní IS musím znát několik oborů, nestáčí znalost pouze z informatiky, ale musím znát i ekonomiku, protože jsou spolu úzce provázané.

V mé práci jsem vyjádřila tři různé způsoby pohledů na životní cyklus první byl klasický informační systém, který lze vyjádřit několika modely, druhý byl americký model životního cyklu, který je nazýván SDLC a jako poslední jsem rozepsala životní cyklus informačního systému veřejné správy. Na první pohled jsou odlišné, ale pokud se podíváme pořádně zjistíme, že základ každého životního cyklu je podobný, vždy začíná podrobným rozpoznáním problému a definováním problému či potřeby a končí ukončením provozu nebo tzv. migrací, což je inovací. Střed životního cyklu má většinou rozdílné fáze, ale vždy obsahuje vývoj, provoz a údržbu. Nedokážeme v tuto chvíli říci, který model je nejlepší, protože s každým novým projektem se otvírá šance na nový cyklus vývoje informačního systému tudíž i vzniká nový trend k přístupu vývoje IS.

Jak jsem již zmínila na začátku, téma je natolik rozsáhlé, že jsem použila ekonomické, počítačové výrazy a samozřejmě pokud je řeč o ISVS tak i normy řady ISO, především obecnou normu ISO 9001 a k tomu další související normy řady ČSN EN ISO , kterými se řídí veškerá dokumentace pro ISVS. Životní cyklus pro vývoj ISVS je odlišný v mnoha věcech, např. je zde důležité dodržet dané kroky, které předepisuje ministerstvo.

Závěrem bych ráda vyzdvihla několik faktů a poznatků o životním cyklu IS. Neexistuje žádný model životního cyklu IS, který nám zaručí naprosto plynulý a nezávadný chod IS, ale v každém modelu se počítá s určitými riziky, které mohou v životním cyklu nastat. Dalším a podle mého názoru nejdůležitějším faktem je fáze vývoje. Ve vývoji dochází k rozsáhlé analýze všech faktorů, získávání nebo vytváření všech potřebných prostředků jako je hardware a software, testování, dokumentace IS jak by měl vypadat a produkční systém pracovat. Tato fáze je nejdůležitější, protože je i nejvíce nebezpečná. Pokud v této fázi dojde k selhání může to vést ke zpoždění IS a překročení rozpočtu. Z tohoto důvodu je velice důležité věnovat fázi vývoje velkou pozornost.

Domnívám se, že cíle mé práce formulované v úvodu byly naplněny. Myslím, že se mi podařilo shrnout a zhodnotit cykly vývoje informačního systému. Popsat jednotlivé fáze každého z cyklů, poukázat na problematické fáze nebo na rizika, které mohou během vývoje nastat. Dále jsem zřehlednila a ucelila jednotlivé pohledy na životní cyklus vývoje IS a snažila jsem se je vždy vyjádřit graficky v diagramech, pro jejich lepší poznání.

9 SEZNAM POUŽITÉ LITERATURY

Klasicá literatura

- [1] Schmuller, Joseph: *Myslíme v jazyku UML*, Grada Publishing, Praha 2001, první vydání, ISBN 80-247-0029-8
- [2] Drbal, Pavel a spolupracovníci: *Objektově orientované metodiky a technologie - 1. a 2. díl*, VŠE, Praha 1997, první vydání, ISBN 80-7079-740-1
- [3] Molnár, Z: *Moderní metody řízení informačních systémů*, Praha:Grada, 1992, ISBN 80-85623-07-2
- [4] Jilková Helena, Ryant Ivan: *Tvorba aplikací v objektovém prostředí*, Grada, Praha 1994, ISBN 80-85623-82-X
- [5] Hoffer, J. : *Modern systemanalysis and design*, fourth edition, Pearson Prentice Hall, New Jersey, 2002
- [6] Kendall, K.E., Kendall, J.E., *System Analysis and design*. New Jersey:Pearson Education, 2004. ISBN 0-13-145-455-2.
- [7] Polák, J., Merunka, V., Carda, A. *Umění systémového návrhu*. Grada, Praha 2003, ISBN 80-247-0424-02
- [8] Kaluža, J., *Tvorba datového modelu v prostředí strategických informačních systémů*, Grafie, Ostrava 1996
- [9] Molnár, Z. *Efektivnost informačních systémů*. Grada Publishing, Praha 2000, ISBN 80-247-0087-5
- [10] Řepa, V. *Analýza a návrh informačních systémů*. Ekopress, Praha 1999, ISBN 80-86119-13-0
- [11] Voříšek, J. *Strategické řízení informačního systému a systémová integrace*. Management Press, Praha 1999, ISBN 80-85943-40-9
- [12] Tvrđíková, M. *Zavádění a inovace informačních systémů ve firmách*. Grada Publishing, Praha 2000, ISBN 80-7169-703-6

Virtuální literatura

- [13] Beneš, M., *Objekty: Přehled OO notací a metodik – OMT*, [online]. [cit. 2008-03-03]. Dostupný z WWW: <http://objekty.vse.cz/Objekty/MethodikyANotace-OMT#Methodika>
- [14] Smil, P., *KOMIX, systémový integrátor, integrace, internet, Úvod do tvorby IS s použitím objektového přístupu*, [online]. [cit. 2008-04-16]. Dostupný z WWW: http://www.komix.cz/Tisk/Clanky/Historie/Uvod_IS.aspx
- [15] Jiří Lukáš, *Vyvíjíme databázový a informační systém II*, [online]. [cit. 2008-04-16]. Dostupný z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2004051201>
- [16] Jiří Lukáš, *Vyvíjíme databázový a informační systém III*, [online]. [cit. 2008-04-16]. Dostupný z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2004051201>
- [17] Dudka, Kamil, *Softwarové inženýrství- studijní materiály*, [online]. [cit. 2008-05-20]. Dostupný z WWW: <http://dudka.cz/studyIUS>
- [18] Eluci data, *The software development life cycle (SDLC)*, [online]. [cit. 2008-06-18]. Dostupný z WWW: www.elucidata.com/refs/sdlc.pdf
- [19] Ing. Novotný, O., Ota Novotný, Publikace, *Standard ISVS pro náležitosti IS*, [online]. [cit. 2008-04-19]. Dostupný z WWW: www.otanovotny.cz/publikace.php
- [20] ÚVIS, *Standard ISVS pro národní prostředí*, [online]. [cit. 2008-04-19]. Dostupný z WWW: http://crypto-world.info/pravo/isvs/s_004_02_06.pdf
- [21] Wikipedie, [online]. [cit. 2008-02-19]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Speci%C3%A1ln%C3%AD:Search?search=&go=J%C3%ADt+na>
- [22] Avenue, *DFD 0: Internetový obchodní systém*, [online]. [cit. 2008-07-19]. Dostupný z WWW: <http://avenue-de.net/school/dfd.html>

- [23] ÚVIS, *Standard ISVS 005/02, 01 pro náležitosti životního cyklu informačního systému*, [online]. [cit. 2008-04-19]. Dostupný z WWW: <http://helpdesk.e-praha.info/pages/akce/servefile.php?filename=1>

10 SEZNAM OBRÁZKŮ

Obrázek 1: Trojúhelník peněz, času a kvality	9
Obrázek 2: Vodopádový model.....	14
Obrázek 3: <i>Model výzkumník</i>	16
Obrázek 4: Iterativní model.....	17
Obrázek 5: <i>Spirálový model</i>	18
Obrázek 6: Etapy SDLC	23
Obrázek 7: Plánovací fáze SDLC	24
Obrázek 8: Analýza potřeby systému	25
Obrázek 9: Fáze návrhu	26
Obrázek 10: Fáze vývoje SDLC	28
Obrázek 11: Testování a integrace SDLC.....	28
Obrázek 12: Implementace a údržba	30
Obrázek 13- Vybrané předměty zájmu metodik pro tvorbu IS	37
Obrázek 14: Životní cyklus ISVS.....	48

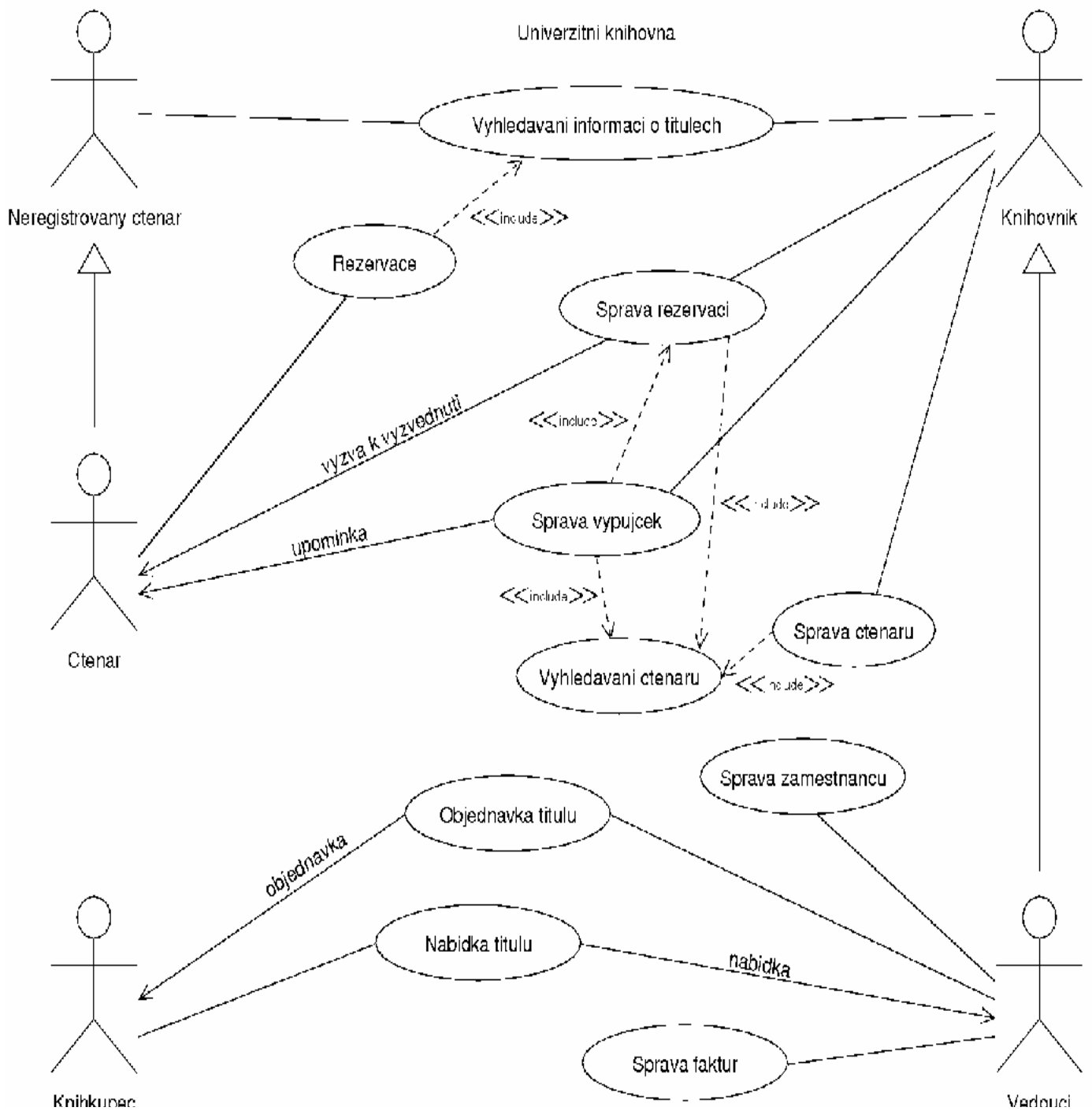
11 SEZNAM TABULEK

Tabulka 1 přehled předpisů souvisejících s ISVS	43
Tabulka 2 životní cyklus ISVS.....	45

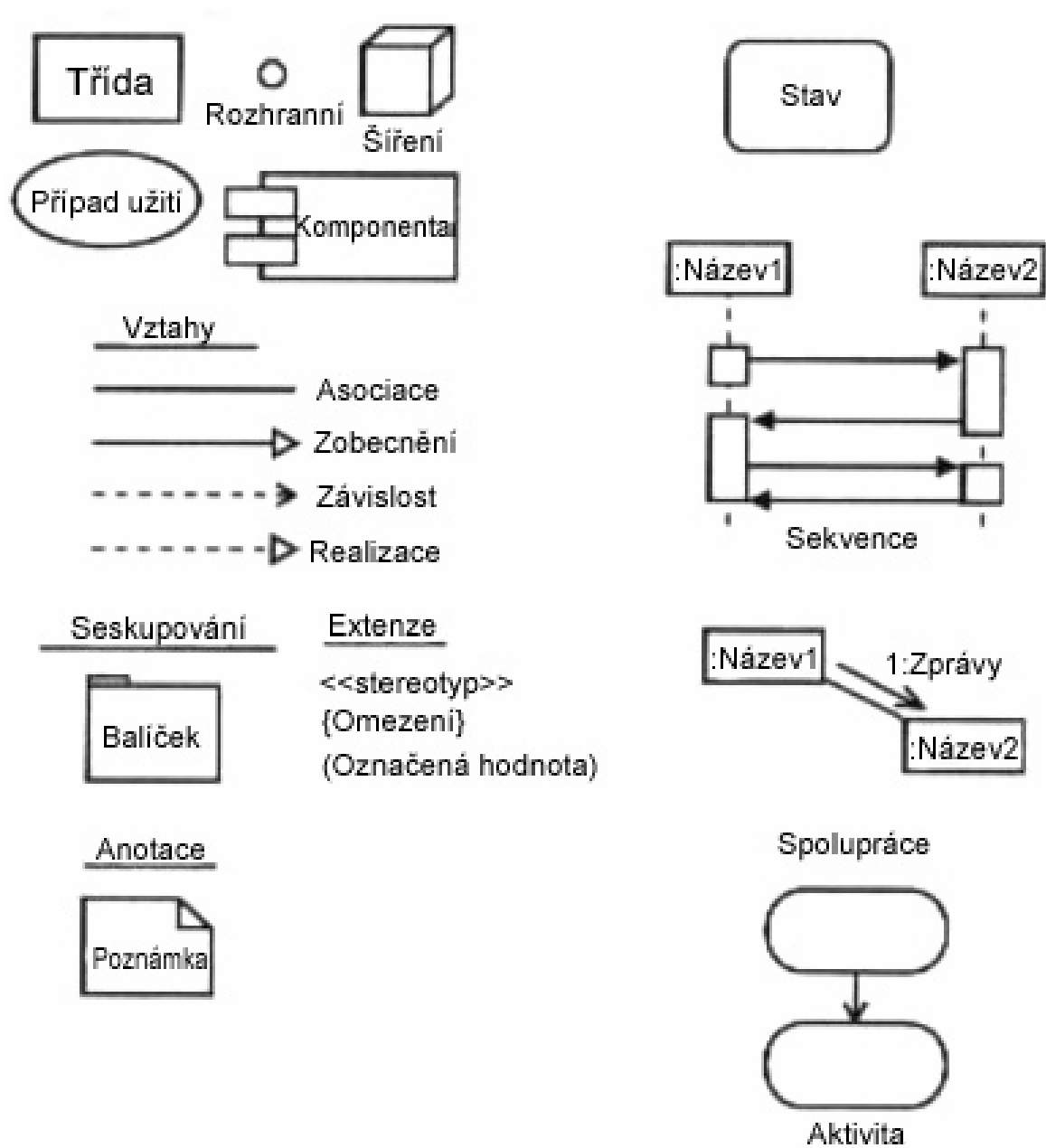
12 SEZNAM PŘÍLOH

Příloha 1- Use case	55
Příloha 2: Základní prvky notace UML	56
Příloha 3: DTD- Data flow diagram = diagram datových toků	57
Příloha 4: Schéma řízení ISVS	58
Příloha 5 Dlouhodobé řízení ISVS	559

Příloha 1- Use case [22]

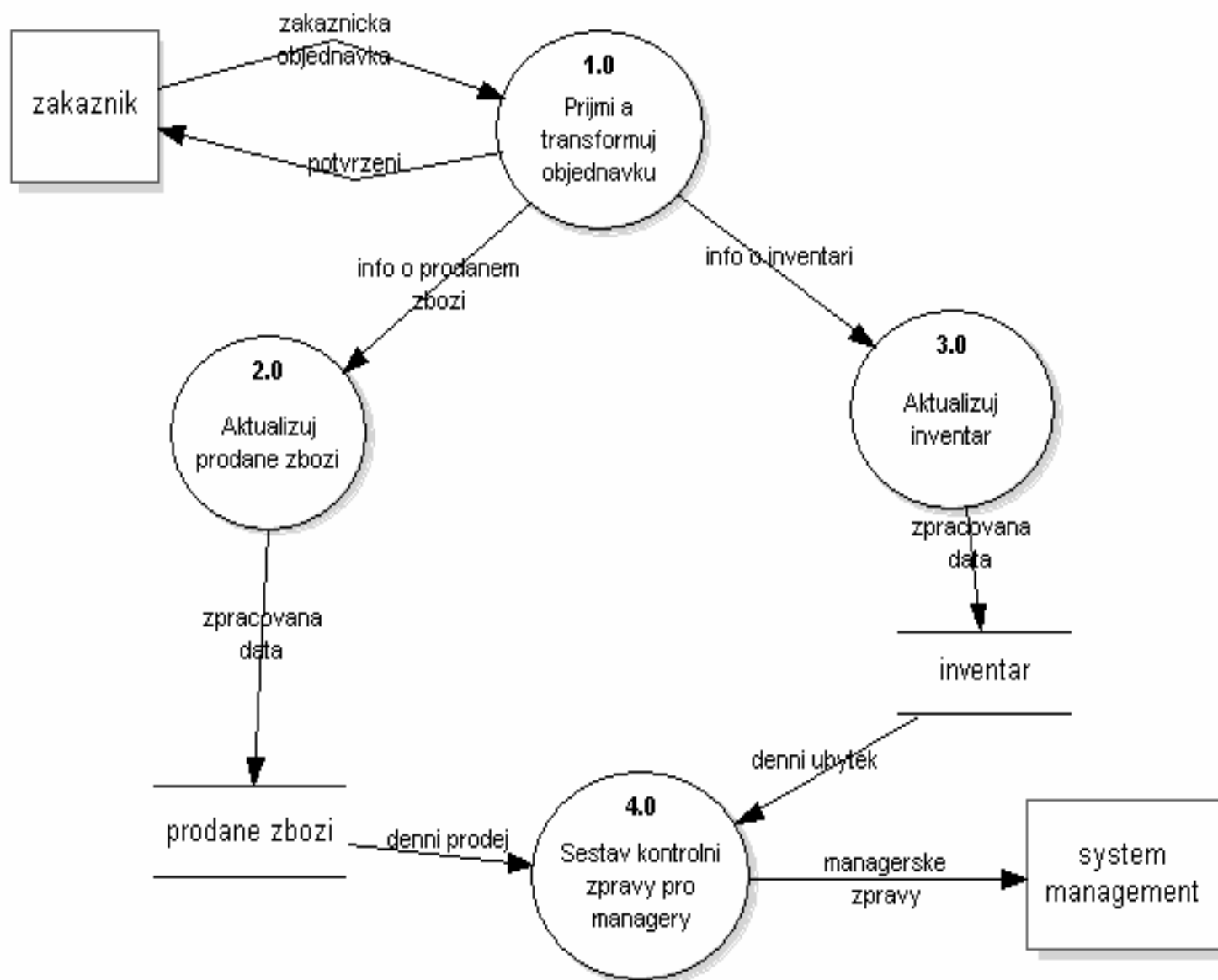


Příloha 2: Základní prvky notace UML [1]

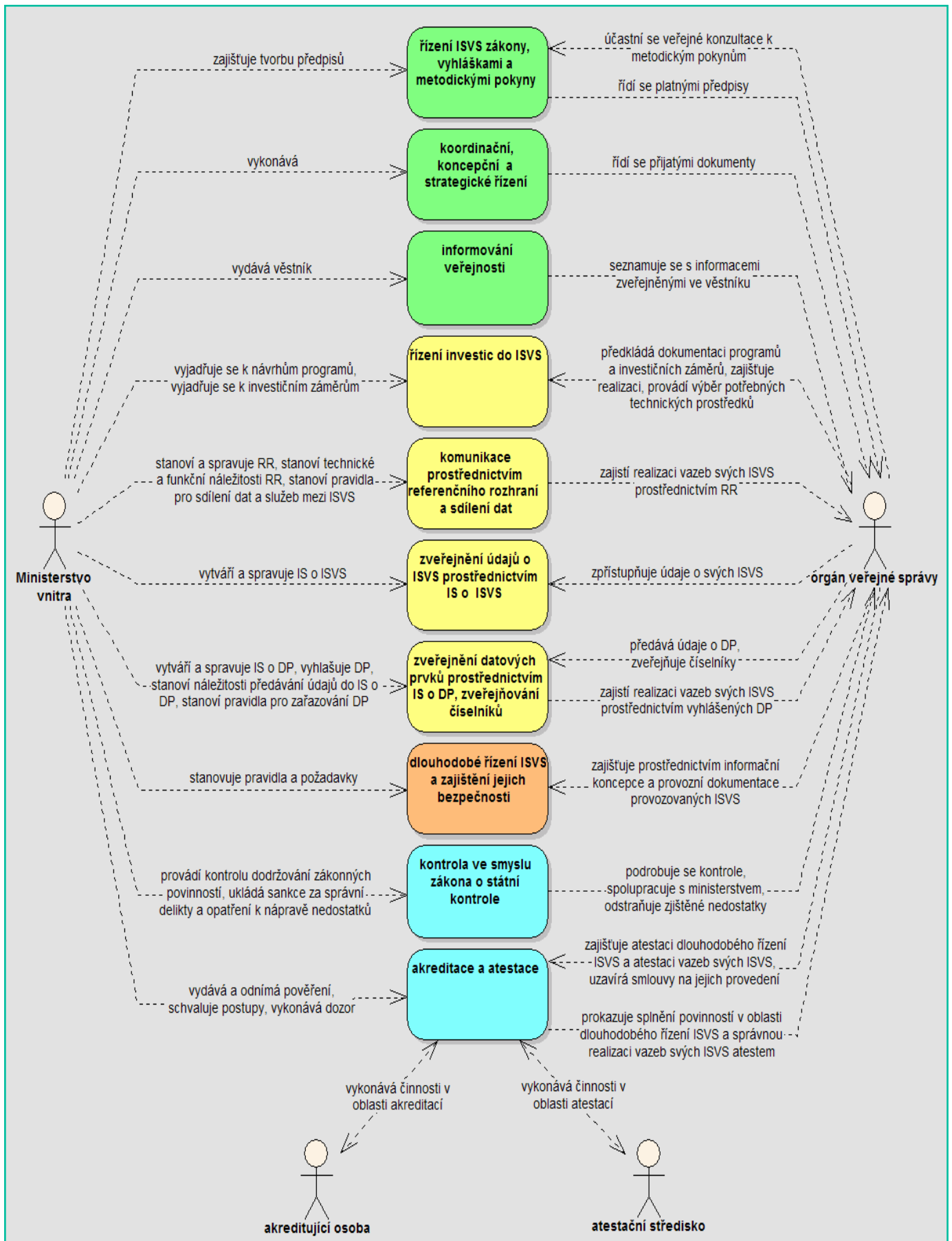


Příloha 3: DTD- Data flow diagram = diagram datových toků [22]

Ukázka DFD Internetový obchodní systém



Příloha 4: Schéma řízení ISVS [20]



Příloha 5 Dlouhodobé řízení ISVS [20]

