

Univerzita Pardubice

Fakulta ekonomicko-správní

Požadavky na informační systém podporující agendu evidence obyvatel

Kristýna Justová

**Diplomová práce
2014**

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Kristýna Justová**
Osobní číslo: **E12721**
Studijní program: **N6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Požadavky na informační systém podporující agendu evidence obyvatel**
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

Zásady pro vypracování:

Cílem práce bude vytvořit případovou studii jako doplňující studijní materiál pro předmět Informační systémy regionů. Studie se zaměří na problematiku řízení požadavků na IS. V rámci práce budou využity vhodné nástroje a modely pro definování požadavků na agendu evidence obyvatel, se zaměřením na funkční požadavky a způsob formulace požadavků.

Zásady:

- Samospráva a její agendy.
- Řízení požadavků.
- Případová studie na požadavky informačního systému veřejné správy agendy evidence obyvatel.

Rozsah grafických prací:

Rozsah pracovní zprávy: cca 50 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

BUCHALCEVOVÁ, A. Metodiky vývoje a údržby informačních systémů. první vydání. Praha: Grada Publishing, 2005. ISBN 80-247-1075-7.

DOUCEK, P. Řízení projektů informačních systémů. První vydání. Praha: Professional Publishing, 2004. ISBN 80-86419-71-1.

KOLEKTIV AUTORŮ. Úvod do regionálních věd a veřejné správy. druhé rozšířené vydání. Praha: Codex Bohemia, 1998. ISBN 80-85963-69-8.

KOUDELKA Z. Právní předpisy samosprávy. druhé doplněné vydání. Praha: Linde Praha, 2001. ISBN 80-7201-274-6.

SMITH, D. J. Engineering quality software. Amsterdam: Amsterdam Elsevier, 1989. ISBN 1-85166-358-4.


Vedoucí diplomové práce:


doc. Ing. Jitka Komárková, Ph.D.


Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: 1. října 2013

Termín odevzdání diplomové práce: 30. dubna 2014


doc. Ing. Renáta Myšková, Ph.D.
děkanka

L.S.


prof. Ing. Jan Čapek, CSc.
vedoucí ústavu

V Pardubicích dne 1. října 2013

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 4. 2014

Kristýna Justová

PODĚKOVÁNÍ:

Tímto bych ráda poděkovala svému vedoucímu práce doc. Ing. Jitce Komárkové, Ph.D. za vedení mé diplomové práce, její odbornou pomoc a cenné rady, které mi pomohly při zpracování diplomové práce.

Také bych chtěla poděkovat své rodině za podporu, kterou mi během celého studia poskytovali, a které si nesmírně vážím.

ANOTACE

Cílem práce je vytvořit případovou studii jako doplňující studijní materiál pro předmět Informační systémy regionů. Studie je zaměřena na problematiku řízení požadavků na IS. V rámci práce jsou využity vhodné nástroje a modely pro definování požadavků na agendu evidence obyvatel, se zaměřením na funkční požadavky a způsob formulace požadavků.

KLÍČOVÁ SLOVA

Informační systém, řízení požadavků, nástroje.

TITLE

System requirements for IS dealing with the population register

ANNOTATION

The aim of the thesis is to create a case study as a supplementary study material for the subject Information Systems of Regions. The case study is focused on management of requirements on IS. Appropriate tools and models are used for defining requirements on the agenda of population records, with a focus on functional requirements and a suitable way of formulation of requirements.

KEYWORDS

Information system, management requirements, tools.

OBSAH

| | |
|--|-----------|
| ÚVOD | 11 |
| 1 SAMOSPRÁVA | 12 |
| 1.1 SAMOSPRÁVA A VZTAH K VEŘEJNÉ SPRÁVĚ..... | 12 |
| 1.2 ZPŮSOBY VÝKONU VEŘEJNÉ SPRÁVY..... | 12 |
| 1.3 VZTAH STÁTU A SAMOSPRÁVA..... | 13 |
| 1.4 SAMOSPRÁVA..... | 13 |
| 1.4.1 Funkce samosprávy..... | 15 |
| 1.4.2 Agenda evidence obyvatel..... | 15 |
| 1.4.3 Další agendy samosprávy..... | 16 |
| 2 POŽADAVKY NA INFORMAČNÍ SYSTÉM | 18 |
| 2.1 TYPY POŽADAVKŮ..... | 18 |
| 2.2 ŘÍZENÍ POŽADAVKŮ..... | 22 |
| 2.2.1 Definice požadavků..... | 24 |
| 2.2.2 Data..... | 26 |
| 2.2.3 Sběr požadavků..... | 28 |
| 2.2.4 Kontrola požadavků..... | 28 |
| 2.2.5 Funkce a funkcionalita systému..... | 29 |
| 3 NÁSTROJE ANALÝZY INFORMAČNÍHO SYSTÉMU | 31 |
| 3.1 ER-DIAGRAM..... | 31 |
| 3.1.1 Základní prvky používané v ER modelu..... | 31 |
| 3.1.2 Postup tvorby ERD..... | 32 |
| 3.2 DATA FLOW DIAGRAM..... | 33 |
| 3.2.1 Základní prvky používané v DFD..... | 33 |
| 3.2.2 Hierarchie DFD..... | 35 |
| 3.2.3 Pravidla tvorby DFD..... | 35 |
| 3.3 DATOVÝ SLOVNÍK..... | 35 |
| 3.4 STATE TRANSITION DIAGRAM..... | 36 |
| 3.4.1 Základní prvky používané v STD..... | 36 |
| 3.4.2 Hierarchie STD..... | 37 |
| 3.5 PŘÍPAD UŽITÍ..... | 38 |
| 3.5.1 Základní prvky používané v případě užití..... | 38 |
| 3.5.2 Diagram případu užití..... | 39 |
| 3.5.3 Tvorba diagramu případu užití..... | 39 |
| 3.5.4 Scénáře..... | 40 |
| 3.6 METODA IDEF..... | 40 |
| 3.6.1 Základní prvky používané v IDEF0..... | 41 |
| 3.6.2 Hierarchie IDEF0..... | 42 |
| 4 VYBRANÝ INFORMAČNÍ SYSTÉM VEŘEJNÉ SPRÁVY | 43 |
| 5 PŘÍPADOVÁ STUDIE DEFINOVÁNÍ POŽADAVKŮ NA EVIDENCI OBYVATEL INFORMAČNÍHO SYSTÉMU MUNIS | 46 |
| 5.1 POPIS INFORMAČNÍHO SYSTÉMU..... | 47 |
| 5.2 NÁSTROJE DEFINOVÁNÍ POŽADAVKŮ..... | 48 |
| 5.2.1 Hierarchie funkcí..... | 48 |
| 5.2.2 Kontextový diagram datových toků..... | 51 |
| 5.2.3 Diagram datových toků..... | 51 |

| | | |
|-------|--|-----------|
| 5.2.4 | <i>IDEF0</i> | 53 |
| 5.2.5 | <i>Stavový diagram</i> | 56 |
| 5.2.6 | <i>EPC</i> | 57 |
| 5.2.7 | <i>Use Case</i> | 59 |
| 5.3 | FUNKČNÍ POŽADAVKY NA INFORMAČNÍ SYSTÉM | 63 |
| 5.4 | SHRNUTÍ PŘÍPADOVÉ STUDIE..... | 65 |
| | ZÁVĚR | 67 |
| | POUŽITÁ LITERATURA | 68 |
| | SEZNAM PŘÍLOH | 70 |

SEZNAM TABULEK

| | |
|--|--------|
| Tabulka 1: Vlastnosti a popis specifikace požadavků: | 25 |
| Tabulka 2: Popis funkcí | 50 |
| Tabulka 3: Symboly EPC diagramu | 57 |
| Tabulka 4: Symboly eEPC diagramu | 58 |
| Tabulka 5: Hlavní scénář případu užití vydání občanského průkazu | 62 |
| Tabulka 6: Alternativní scénář k případu užití vydání OP | 63 |
| Tabulka 7: Hlavní scénář případu užití Zapsání nového občana | - 84 - |
| Tabulka 8: Alternativní scénář | - 85 - |

SEZNAM ILUSTRACÍ

| | |
|--|--------|
| Obrázek 1: Vztahy mezi typy požadavků | 19 |
| Obrázek 2: Jednotlivé disciplíny práce s požadavky | 22 |
| Obrázek 3: Základní prvky IDEF0 | 41 |
| Obrázek 4: Moduly IS Munis | 45 |
| Obrázek 5: Organizační struktura obce | 47 |
| Obrázek 6: Hierarchie funkcí | 49 |
| Obrázek 7: Hierarchie funkce Vyřizování žádostí | 50 |
| Obrázek 8: Kontextový diagram funkce vyřizování žádosti | 51 |
| Obrázek 9: DFD příjmu a výdeje žádosti o občanský průkaz | 53 |
| Obrázek 10: IDEF0 evidence obyvatel | 54 |
| Obrázek 11: Dekompozice IDEF0 vyřizování žádosti | 55 |
| Obrázek 12: Ukázka části stavového diagramu vyřizování žádosti | 56 |
| Obrázek 13: Ukázka části EPC digramu vyřizování žádosti | 58 |
| Obrázek 14: Ukázka diagramu eEPC | 59 |
| Obrázek 15: Případu užití z pohledu občana | 60 |
| Obrázek 16: Případ užití z pohledu úřednice | 61 |
| Obrázek 17: Aplikovaná informatika v ekonomice | - 71 - |
| Obrázek 18: Vodopádový model | - 74 - |
| Obrázek 19: V-model | - 75 - |
| Obrázek 20: Spirálový model | - 76 - |
| Obrázek 21: Stavový diagram vyřizování žádosti | - 81 - |
| Obrázek 22: EPC diagram vyřizování žádosti | - 82 - |
| Obrázek 23: eEPC diagram | - 83 - |

SEZNAM ZKRATEK A ZNAČEK

| | |
|------|--------------------------------------|
| DFD | Data Flow Diagram |
| EPC | Event-driven Process Chain |
| HW | Hardware |
| ICT | Informační a komunikační technologie |
| IDEF | Integration DEFinition |
| IS | Informační systém |
| ISVS | Informační systém veřejné správy |
| SŘBD | System řízení báze dat |
| STD | State Transition Diagram |
| SW | Software |

ÚVOD

Informační systémy jsou v dnešní době nepostradatelnou součástí státní, podnikatelské i soukromé sféry. Příkladem jsou stále častější veřejné zakázky na tyto systémy, které jsou vyhlášeny ve veřejné správě, nebo také růst v odvětví služeb v oblasti ICT. Informační a komunikační technologie je využívána i v domácnostech. Růst těchto technologií je zapříčiněn hlavně větší dostupností informací.

Při zavádění IS je nutné přizpůsobení potřebám organizace a finančním prostředkům dané organizace. Dříve než je IS zaveden, je nutné zjistit, jaké potřeby a procesy jsou potřebné. Jedná se o specifikaci požadavků, které uživatel od IS požaduje. Tato fáze je součástí analýzy, ovšem někdy bývá považována za fázi před analýzou. Dalšími fázemi životního cyklu jsou čerpány funkční požadavky definované ve specifikaci. Specifikace je rozdělena do jednotlivých kroků. K definování požadavků jsou využity nástroje analýzy. Jednotlivé nástroje mají své diagramy, ve kterých je prezentována funkčnost informačního systému.

Práce je orientována na vytvoření případové studie, která je doplňkovým studijním materiálem pro předmět Informační systémy regionů a to jak pro denní, tak pro kombinovanou formu studia. Studie je zaměřena na problematiku řízení požadavků na IS. Při definování požadavků jsou použity nástroje a modely využívané v předmětu. Specifikace požadavků je zaměřena na evidenci obyvatel, se kterou se lidé setkávají nejčastěji. Případová studie je vypracována ve zjednodušené formě.

Definování požadavků bude ukázáno na jednom ze softwarových nástrojů veřejné správy a tím je IS Munis. IS Munis je zaměřen na obce a města. Systém byl vybrán z důvodu jeho zařazení ve výuce. IS Munis disponuje mnoha moduly, ovšem pro tento text je vybrán modul evidence obyvatel, se kterým se občan setkává nejčastěji, aniž by to věděl.

V příloze je uveden rozšiřující materiál pro studenty předmětu ISR a materiál k případové studii.

1 SAMOSPRÁVA

V diplomové práci je nejdříve popsána problematika veřejné správy a samosprávy. Tuto problematiku je nutné pochopit, aby bylo možné lépe definovat požadavky na informační systém. Součástí samosprávy je také agenda obyvatel. V rámci této práce budou popsány pouze agendy, se kterými se občan setkává nejčastěji.

1.1 Samospráva a vztah k veřejné správě

Jedna z možných definic veřejné správy je, že veřejnou správou je myšlena správa záležitostí státu, veřejných svazků a jiných subjektů. Je vykonávána příslušnými orgány, které jsou řízeny právním řádem a v oblasti volné správní úvahy politickými kritérii držitele moci.[15]

Jakákoli veřejná moc, která není vykonávána mocí zákonodárnou či soudní, je prováděna jako tzv. veřejná správa. Veřejnou správu můžeme realizovat ve třech podobách, a to jako státní moc, samosprávu a jiné veřejnoprávní korporace.[3]

Členění veřejné správy může být provedeno podle různých kritérií. Jedno z možných kritérií je dělení na státní správu a samosprávu, s ohledem na to, kdo je jejím vykonavatelem, vzhledem k působnosti jednotlivých nositelů.[14]

1.2 Způsoby výkonu veřejné správy

Veřejnou správu je možné zastávat dvěma způsoby výkonu. Jednou z forem je centralizovaná státní správa a druhou formou je decentralizovaná státní správa.

Centralizovaná státní správa

Jediným prvotním normativním právním zdrojem je centrální státní zákonodárná moc. Správa státu je prováděna ústředními orgány a její byrokracií – koncentrovanými a dekoncentrovanými státními orgány plně závislými na centru.

Decentralizovaná státní správa

Tvorba prvotních právních norem je vytvářena centrální zákonodárnou mocí. Výkon státní správy je prováděn i decentralizovanými orgány státní správy či je přenesen na samosprávné veřejnoprávní korporace. Stát tedy není zbaven určité části státní správy, ale je ponechán její výkon veřejnoprávním korporacím pod státním dohledem, řízením a financováním.[14]

1.3 Vztah státu a samospráva

Samospráva není rovnocenným partnerem státu, vždy je vytvořena ze sebeomezení státu a její mocenská rozhodnutí jsou vykonatelná jen za pomoci státního donucení, jelikož nemá k dispozici vlastní mocenské složky, jako například policii, armádu, vězení. Stát může existovat bez samosprávy, samospráva bez státu nikoliv. Stát má dostatečnou možnost zásahu do samosprávy. Zásahy jsou prováděny především přes moc zákonodárnou, kterou jsou určeny meze samosprávy. Tyto meze je možné rozšířit nebo zúžit zákonem.

Další možností, jak stát ovlivňuje samosprávu, je moc soudní. Tento zásah je vykonán přezkumem individuálních právních aktů anebo přezkumem normativních právních aktů samosprávy. Mocí soudní může být zrušen nezákonný či neústavní normativní právní akt a též zrušen individuální akt, který se dotýká práv a povinností fyzických i právnických osob, pokud tato rozhodnutí nejsou v souladu s ústavou a zákony.[15]

1.4 Samospráva

Dle hlavy sedmé Ústavy je Česká republika členěna na obce, které jsou základními územními samosprávnými celky, a kraje, které jsou vyššími územními samosprávnými celky. Územní samosprávné celky jsou územními společenstvími občanů, která mají právo na samosprávu. Obec je vždy součástí vyššího samosprávného celku. Tyto celky jsou korporacemi, které mohou disponovat s vlastním majetkem a hospodařit podle vlastního rozpočtu.[28]

Samospráva je chápána jako svébytná součást veřejné správy vedle státní správy. Má vlastní primární právní novotvorbu, která je vytvářena podle vlastní politické vůle a též má vlastní výkon veřejné správy. Právní a správní možnosti musí být podepřeny vlastním finančním krytím (právo na majetek, vlastní hospodaření, místní daně a poplatky). Pokud takto není učiněno, mohla by být realizace samosprávného rozhodnutí vetována centrální státní mocí odepřením prostředků na jeho finanční krytí.

Je vždy podřízena zákonu a suverénní moc přísluší státu jako celku. Je chápána jako zvláštní druh správy, při níž jsou spravováni spravováni sami. Jedná se o významný prvek moderního demokratického státu, jelikož v rámci teorie dělby moci je omezena centrální státní moc, a to z hlediska vertikálního vedle dělby moci horizontální (zákonodárství, soudnictví, exekutiva).[15]

Územní samospráva má podíl na výkonu tzv. zbývající veřejné moci. Státem je svěřena subjektům nestátního charakteru k samosprávnému hospodaření v úvahu přicházejících

veřejných záležitostí. Subjekty nestátního charakteru je rozuměno veřejnoprávní korporace. Tyto korporace jsou členěny dle kritéria územního a zájmového. Územními korporacemi jsou obce a kraje, zájmovými jsou profesní komory.[3]

Základním prvkem je [3]:

- území nebo činnost, vůči kterým působí,
- osoby podřazené samosprávě,
- existence samosprávy jako právnické osoby veřejného práva se soustavou samosprávných orgánů,
- vlastní hospodaření a rozpočet.

Obec je základním územním samosprávným celkem a je tvořena územním celkem, který je vymezen hranicí území obce. Obec vystupuje v právních vztazích svým jménem a nese odpovědnost z těchto vztahů vyplývajících. Je pečováno o všestranný rozvoj území a o potřeby občanů. Při plnění úkolů je chráněn též veřejný zájem. Obcí jsou spravovány její záležitosti samostatně. Státní orgány a orgány krajů mohou do samostatné působnosti zasahovat, jen vyžaduje-li to ochrana zákona, a pouze způsobem, který je stanoven zákonem.[29]

Do samostatné působnosti obce jsou zařazeny záležitosti, které jsou v zájmu obce a občanů obce, pokud nejsou zákonem svěřeny krajům. Obcí je pečováno o vytvoření podmínek pro rozvoj sociální péče a pro uspokojování potřeb občanů jako například potřeby bydlení, ochrany a rozvoje zdraví, dopravy a spojů, potřeby informací, výchovy a vzdělávání, celkového kulturního rozvoje a ochrany veřejného pořádku.[29]

Kraj je územním společenstvím občanů, které má právo na samosprávu. Jedná se o veřejnoprávní korporaci, která má vlastní majetek a vlastní příjmy vymezené zákonem a hospodaří za podmínek stanovených zákonem podle vlastního rozpočtu. Kraj vystupuje v právních vztazích svým jménem a nese odpovědnost z těchto vztahů vyplývajících. Krajem je pečováno o všestranný rozvoj jeho území a o potřeby jeho občanů. Kraj může mít znak a vlajku.[30]

Do samostatné působnosti kraje jsou zahrnuty záležitosti, které jsou v zájmu kraje a občanů kraje, pokud nejde o přenesenou působnost kraje. Při výkonu samostatné působnosti kraj spolupracuje s obcemi. Kraj nesmí přitom zasahovat do jejich samostatné působnosti. Kraj je oprávněn vyjadřovat se k návrhům státních orgánů, které se dotýkají působnosti kraje.[30]

1.4.1 Funkce samosprávy

Je vymezeno jakési funkční minimum, jež je společné nejnižší samosprávné úrovni demokratických zemí, bez ohledu na rozdílný historický vývoj, rozdílnou velikostní strukturu obcí, odlišnému nastavení struktur zastupitelských orgánů či rozdílnému postavení komunální úrovně vůči centrálnímu politickému systému.[3]

Funkce lze rozdělit do sedmi oblastí [3]:

- politické funkce – vyhlášení místního referenda, volba a ustavování orgánů obcí, vytváření partnerství s dalšími obcemi, udělování čestného občanství a další,
- ekonomické funkce – sestavování rozpočtu, schvalování závěrečného účtu, hospodaření s majetkem, poskytování a přijímání darů a další,
- sociální funkce – vzdělávání, zdravotnická péče, sport a kulturní oblasti, pečovatelské domy pro staré občany, zřizování hřbitovů a další,
- ekologické funkce – čistota obce, svoz a likvidace odpadů, odvádění a čištění odpadních vod,
- bezpečnostní funkce – pravomoc zřizovat obecní policii,
- infrastrukturní funkce – správa, provozování a udržování zařízení zřízených k uspokojování potřeb občanů, například udržování komunikací, vodovodů, kanalizace,
- prognostické funkce – schvalování programu rozvoje obce, územního plánu, regulačního plánu, přijímání koncepčních dokumentů v oblasti politického, ekonomického, sociálního, právního a organizačního rozvoje v budoucnosti.

Pro plnění těchto funkcí je využito řady nástrojů. Jedná se zejména o usnesení zastupitelstva či rady, uzavírání smluv nebo o vydávání vyhlášek v záležitosti samostatné působnosti a ukládání sankcí.[3]

1.4.2 Agenda evidence obyvatel

V administrativě je agenda chápána jako obor působnosti úřadu, oddělení nebo i jednotlivého referenta, oblast činností, v nichž je nutno rozhodovat.

Dle zákona se v evidenci obyvatel vedou údaje [31]:

- o státních občanech České republiky,
- o osobách, které pozbyly státní občanství České republiky,

- o cizincích, kteří jsou matkou, otcem, popřípadě jiným zákonným zástupcem, manželem, registrovaným partnerem nebo dítětem občana.

Státní správu podle tohoto zákona vykonávají [31]:

- Ministerstvo vnitra,
- krajské úřady,
- obecní úřady obcí s rozšířenou působností, v hlavním městě Praze městské části určené Statutem hlavního města Prahy a v územně členěných statutárních městech magistráty těchto měst,
- obecní úřady, v hlavním městě Praze a v územně členěných statutárních městech úřady městských částí nebo městských obvodů, pokud tak stanoví statuty těchto měst, a území vojenských újezdů, újezdní úřady.

Osobní základ obce je reprezentován občany, tj. fyzickými osobami, které jsou státními občany České republiky a jsou spojeni s územím trvalým pobytem, podle zákona o evidenci obyvatel.[14]

Občané starší 18 let mají právo [14]:

- aktivní a pasivní volební právo do zastupitelstva obce,
- hlasovat v místním referendu,
- za podmínky, že vlastní na území obce nemovitost, mají právo:
 - svého stanoviska k projednávaným věcem na zasedání zastupitelstva,
 - vyjádření k rozpočtu a závěrečnému účtu,
 - nahlížení do rozpočtu obce a závěrečného účtu,
 - usnesení a zápisů z jednání zastupitelstva, rady, výborů zastupitelstva a komisí rady.

Stejná oprávnění jako občanům obce jsou udělena také cizím státním příslušníkům starším 18-ti let. Tito lidé musí být přihlášení v obci k trvalému pobytu.[14]

1.4.3 Další agendy samosprávy

Hřbitovní řády

Samostatná působnost obce je dána tím, že hřbitovy jsou v naprosté většině provozovány obcemi a jsou v jejich vlastnictví. Řada norem ve hřbitovních řádech se dotýká udržování

čistoty a veřejného pořádku na hřbitovech. Ve hřbitovních řádech je zpravidla upravena otevírací doba hřbitova, úkoly správce pohřebiště, pravidla pro propůjčování místa na pohřebišti a podobně.[15]

Při vydávání hřbitovních řádů je odstoupeno od pojmu místo na pohřebišti a je zaváděn pojem hrobový pozemek. Tento pozemek však není chápán jako pozemek ve smyslu občanského práva věcného, nejedná se o nemovitou věc.[15]

Místní poplatky

V obcích je možné vybírat místní poplatky podle zákona o místních poplatcích. Ústavní úpravou je umožněna existence povinnosti platit jakékoliv veřejnoprávní dávky (daně, poplatky) jen na podkladě zákona. Samostatné rozhodnutí obce pro výběr místních poplatků i určení jejich výše při respektování zákonné horní hranice je v samostatné působnosti obce.[15]

Obcí jsou vybírány místní poplatky [15]:

- za užívání veřejného prostranství,
- ze vstupného,
- za provozovaný výherní hrací přístroj,
- za provoz systému shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálních odpadů.

Komunální odpad

Do působnosti obcí je zahrnováno zabezpečování svozu, třídění a likvidace komunálního odpadu. Je to dáno tím, že obec je považována zákonem za původce komunálního odpadu. Podle zákonné dikce je odpad nepotřebná movitá věc, které se chce nepodnikající fyzická osoba zbavit. Dle právních předpisů není možné regulovat nakládání s jiným druhem odpadu než s komunálním a stavebním, neboť nakládání s jiným odpadem je svěřen zákonem o odpadech státní správě.[15]

Obec může obecně závaznou vyhláškou v samostatné působnosti stanovit systém shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálního odpadu, včetně míst určených k odkládání odpadu. Tomuto systému jsou podřízeny všechny fyzické osoby, pokud není prokázáno, že komunální odpad je likvidován jinak. Důkazní břemeno je tak na osobě, která se chce tomuto systému vyhnout. Způsob je stanoven obcí samotnou v příslušné obecně závazné vyhlášce.[15]

2 POŽADAVKY NA INFORMAČNÍ SYSTÉM

Většina práce při definování a specifikaci požadavků probíhá ve fázích začátek a rozpracování. Tyto fáze jsou k nalezení hned na počátku projektu. Dříve než je započato s analýzou či návrhem, je nutné mít alespoň rámcový přehled o tom, čeho je možné dosáhnout a jaký je smysl požadavků a jejich specifikace.

2.1 Typy požadavků

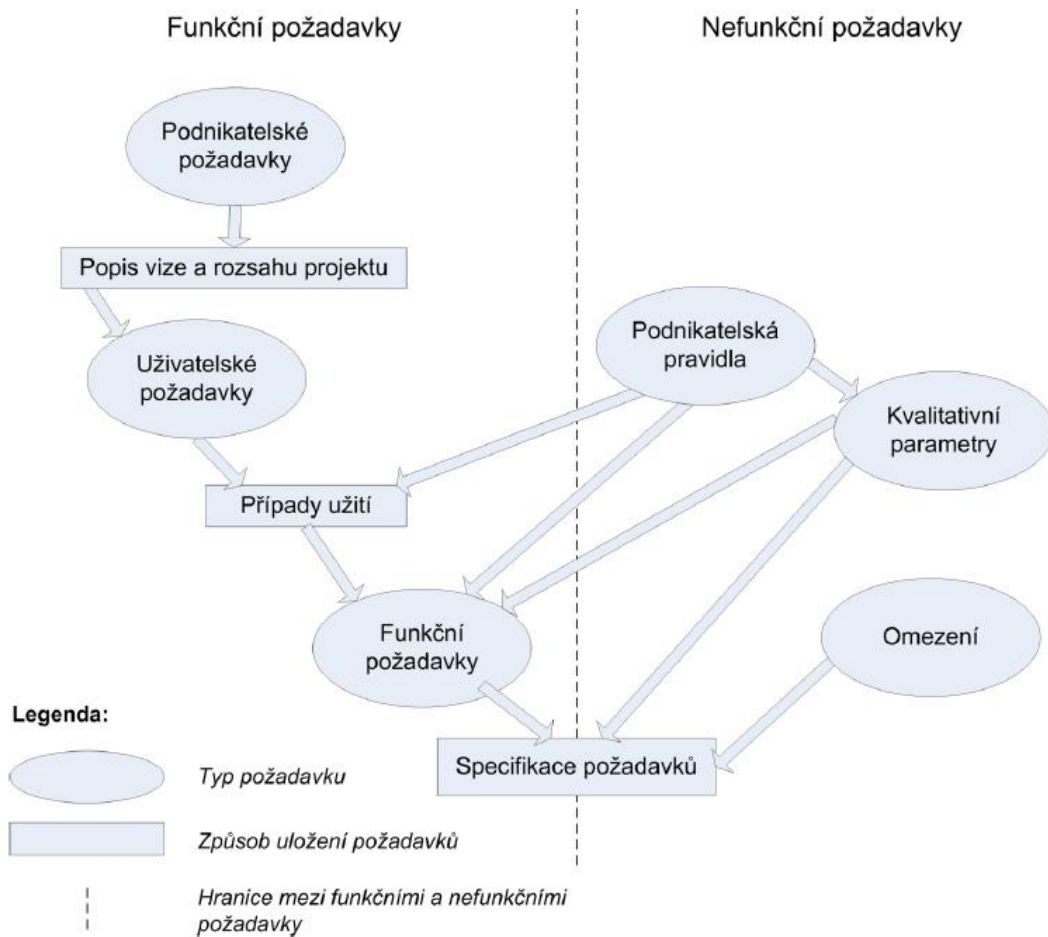
Požadavky na systém jsou rozděleny do tří různých skupin, a to na podnikatelské, uživatelské a funkční. Kromě těchto požadavků má každý systém dále zásobu parametrických požadavků. Parametrické požadavky jsou vyplývající nikoliv z funkce systému, ale jeho prostředí. Základními typy požadavků jsou požadavky funkční a nefunkční.

Základní typy požadavků [1], [12]:

- funkční požadavky – určují, jaké chování bude systém nabízet, funkce, které má systém obsahovat (zajišťovat), jedná se o popis systému z hlediska jeho požadované funkčnosti,
- parametrické (nefunkční) požadavky – specifikují vlastnosti nebo omezující podmínky systému kladené na systém, např. podmínky a termín dodání, požadavky implementační, bezpečnostní, legislativní, cenové, požadavky na spolupráci při provozu systému a na servis.

Na obrázku 1 jsou kromě funkčních a nefunkčních požadavků vyjádřeny také podnikatelské a uživatelské požadavky. Mezi podnikatelské požadavky jsou zařazeny cíle a vize organizace, která chce software vybudovat, tímto popisem je řečeno, proč je systém budován. Uživatelskými požadavky je popsáno, co uživatelé budou moci v rámci systému provádět. V podnikatelských pravidlech jsou zahrnuty předpisy a směrnice firmy, státní zákony, průmyslové standardy, tyto požadavky jsou nalezeny za hranicemi systému. Mezi kvalitativními parametry jsou zahrnuty kritéria systému důležitá pro uživatele nebo vývojáře. Omezení je popis základních cest při návrhu či vývoji systému.[27]

V literatuře je možné nalézt i jiné dělení požadavků. Dle použití je možné požadavky dělit například na základní a odvozené. Dle vztahu na požadavky týkající se obchodních cílů, produktu, návrhu, problémové oblasti a jiné.[2][16]



Obrázek 1: Vztahy mezi typy požadavků

Zdroj: převzato z [27]

Systémové požadavky jsou celkové požadavky na produkt složený z většího počtu podsystémů. Systém může být zcela softwarový, nebo může softwarové podsystémy kombinovat s hardwarovými. Součástí systému jsou i lidé.[27]

Technické prostředky

Termín technické prostředky (hardware, HW) vznikl z potřeby oddělit fyzický aspekt počítačů, telekomunikačních zařízení a dalších technických zařízení od programů, které jsou do těchto zařízení vkládány, a které s těmito zařízeními manipulují. Vybraná část IT je označována jako technologická infrastruktura IS. Jedná se o technické prostředky a část programového vybavení, které je nezbytné pro zajištění implementace funkcí IS a zajištění jeho provozu. Informační technologií je ovlivněn způsob uložení dat, přístup k nim i jejich prezentaci.[8]

Technické prostředky tvoří široká škála různých zařízení, zejména pak [9]:

- počítače, jejich přídatná zařízení a samostatné nosiče dat,

- technické prostředky komunikačních sítí (např. modemy, přenosové linky),
- kancelářské zařízení (např. skartovací stroje),
- specializované technické zařízení (např. linky pro balení listovních zásilek, generátory elektrického proudu),
- prostředky chránící systém před výpadky elektrického proudu a před kolísáním elektrického napětí,
- prostředky zajišťující požadované klimatické podmínky,
- prostředky protipožární ochrany,
- prostředky ochrany před neoprávněným vstupem.

Základními komponenty počítače jsou [8]:

- procesor,
- vnitřní paměť,
- vnější paměť,
- vstupní zařízení,
- výstupní zařízení,
- komunikační zařízení.

S růstem počtu technických prostředků, roste potřeba, aby byly vzájemně provázány, resp. integrovány a stejně tak jsou propojovány do dalších veřejných nebo privátních sítí. Vyjádřením integrace prostředků na úrovni osobní informatiky jsou tzv. osobní sítě, resp. PAN (Personal Area Network). S počtem instalovaných a aplikovaných prostředků osobní informatiky je zvyšována i složitost osobních sítí. Často jsou zahrnovány i prostředky, které jsou na pomezí informačních a komunikačních technologií a běžné spotřební elektroniky. [8]

Softwarové prostředky

Pro souhrn všech programů je používán termín programové vybavení (software, SW). Programové vybavení je tvořeno třemi základními skupinami, a to základní programové vybavení, prostředky vývoje a aplikační programové vybavení. Programové vybavení je tvořeno souhrnem všech počítačových programů a programových aplikací. Programová aplikace je tvořena vzájemně provázanou sadou programů. Každý program je představován formulací určitého algoritmu vyjádřeného pomocí programovacího jazyka. Algoritmem je potom označován přesný a jednoznačně určený postup, kterým je vyřešený daný typ úlohy.

Důležitou vlastností algoritmu je opakovatelnost. Programem je rozuměna část aplikace, která je spuštěna na jednom počítači, jedním příkazem, případně volbou jedné položky menu, přičemž program může být zároveň aplikací. U programu a aplikace je sledována jejich logická a fyzická struktura, rozhraní, která nabízí, architektura komunikace a další vlastnosti.[8]

Softwarové prostředky osobní informatiky nabývají postupně na své mohutnosti a širí sortimentu. V této práci jsou prezentovány ty nepoužívanější a pro práci uživatele i nejvýznamnější.[8]

Softwarové prostředky jsou pro lepší orientaci rozděleny do několika skupin [9]:

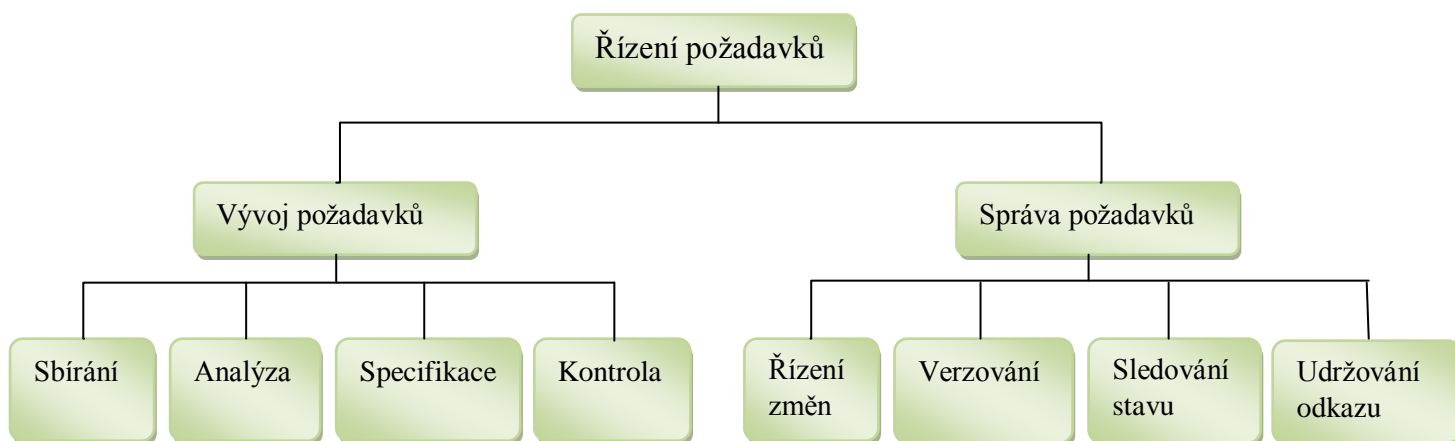
- základní kancelářské prostředky – zajišťují zpracování běžných dokumentů při individuální práci nebo administrativě, patří sem:
 - zpracování textů – textové editory (Word, WordPerfect),
 - zpracování tabulek – tabulkové kalkulátory (Excel, Quattro Pro),
 - zpracování schémat a prezentací – grafické editory (PowerPoint, Visio, Corel),
 - zpracování osobních databází (Access, FoxPro, Paradox),
 - zpracování www stránek – editory www stránek, HTML editory (FrontPage),
- komunikační prostředky – představují zejména elektronickou komunikaci mezi jednotlivci anebo komunikaci s dostupnými informačními zdroji, patří sem prostředky pro:
 - elektronickou poštu (Outlook, Lotus Notes),
 - prohlížení webových zdrojů – webové prohlížeče (Internet Explorer, Netscape),
- organizační prostředky pro plánování a organizování vlastní práce, případně rozsáhlejší kooperací, např. prostředky pro plánování a řízení projektů, ty jsou obvykle zahrnovány do softwaru pro skupinovou spolupráci, ale i na osobní úrovni se mohou efektivně využívat (MS Project, On Target),
- prostředky pro práci s grafikou, obrazy, fotografiemi, které zahrnují:
 - zpracování grafiky, ilustrací, obrazů (Corel),
 - zpracování digitálních fotografií,
 - zpracování videa,

- prostředky pro komplexní zpracování publikací – DTP (Publisher, Ventura Publisher, PageMaker),
- ostatní prostředky osobní informatiky, kde jsou zahrnuty multimédia, antivirové prostředky, počítačové hry.

2.2 Řízení požadavků

Řízení požadavků je založeno na stanovení služeb, které by měl vyvíjený systém poskytovat a omezení, za nichž je schopen pracovat. Dále je založeno na získávání požadavků, jaké na nový systém mají jeho uživatelé a v sestavování jejich priorit. Jedná se o proces vyjednávání, neboť obvykle je třeba vyřešit a vyvážit i mnoho protichůdných požadavků.[1]

Některými autory je řízení požadavků rozděleno na dva samostatné celky. V jednom celku je hovořeno o vývoji požadavků, v tom druhém je vysvětlena správa požadavků. Rozdělení je naznačeno na následujícím obrázku.[27]



Obrázek 2: Jednotlivé disciplíny práce s požadavky

Zdroj: převzato z [27]

Tato práce je zaměřena na vývoj požadavků, proto bude tato problematika více popsána. Jak je ukázáno na obrázku 2, vývoj požadavků je rozdělen na jejich sbírání, analýzu, specifikace a kontrolu. Těmito činnostmi jsou popsány všechny úkoly spojené se získáváním, vyhodnocováním a dokumentací požadavků na software nebo systém.[27]

Mezi úkoly těchto činností je zařazena [27]:

- identifikace tříd uživatelů, kteří budou systémem používat,

- získávání požadavků od zástupců jednotlivých tříd,
- pochopení jednotlivých uživatelských úkolů a podnikatelských cílů, které se tyto úkoly snaží splnit,
- analýza informací od uživatelů a odlišení jejich cílů od funkčních požadavků, parametrických požadavků, podnikatelských požadavků, navrhovaných řešení a nadbytečných informací,
- rozdělení části požadavků mezi softwarové moduly dané architekturou systému,
- seřazení kvalitativních parametrů podle důležitosti,
- vyjednání implementačních priorit,
- zpracování nasbíraných uživatelských potřeb do podoby modelů a psané specifikace požadavků,
- kontrola dokumentovaných požadavků, aby byla jistota, že uživatelským požadavkům rozumí všichni stejně, a aby se případné chyby našly dříve, než je převezme vývojářský tým.

Na vývoj softwaru má vliv jak prostředí, tak cílové prostředí. Proměnnými veličinami při vývoji softwaru jsou [6]:

- dostupnost kvalifikovaných specialistů,
- stabilita technologie pro implementaci,
- stabilita a schopnost nástrojů,
- efektivnost používaných metod,
- dostupnost expertů na věcnou oblast,
- nová funkcionalita a její vztah k existující funkcionalitě,
- metodika a její flexibilita,
- konkurence,
- čas,
- zdroje.

2.2.1 Definice požadavků

Shromažďování (získávání, sběr) požadavků na systém je činnost, která je provázána s několika etapami vývoje IS. V průběhu shromažďování požadavků na systém je sepsán dokument s tzv. specifikací požadavků na systém.[12]

Jako požadavek lze považovat cokoli, čím je ovlivňováno rozhodování při návrhu. Ve slovníčku softwarové terminologie je požadavek definován jako [27]:

- podmínka nebo funkce, kterou uživatel potřebuje pro řešení problému nebo dosažení nějakého cíle,
- podmínka nebo funkce, kterou musí systém nebo jeho část splňovat, aby bylo vyhověno smlouvě, standardu, specifikaci nebo jinému dokumentu, jenž je na něj formálně vztahován,
- dokumentovaná podoba některého z předchozích dvou bodů.

Požadavek je definován jako specifikace toho, co by mělo být implementováno. Požadavky jsou základem všech systémů. Jedná se v podstatě o vyjádření toho, co by měl systém dělat. Měly by být jediným vyjádřením, co by měl systém dělat, nikoli toho, jak by to mělo být prováděno. Tudiž by měly odpovídat na otázku „Co?“ uživatelé od systému potřebují, než na otázku „Jak“ dosáhnout toho, co potřebují od systému. Mezi požadavky nejsou zařazeny detaily o návrhu a implementaci softwaru, také informace o plánování projektu či testování softwaru. Tyto informace nejsou potřebné pro programování softwaru. [1][17][27]

Dokument specifikující požadavky na systém by měl popsat řešený systém z hlediska [12]:

- rozsahu řešení (co bude systém řešit, co již nikoliv, kde jsou hranice systému),
- požadovaných funkcí systému,
- dat zpracovávaných systémem (druhy dat, objemy datových toků),
- podmínek činností v reálném provozu (výkon, dostupnost, zabezpečení),
- spolupráce s neautomatizovanou částí a okolními systémy.

Mnoho firmami je vytvářena specifikace systémových požadavků tak, aby bylo přesně specifikováno, co budou softwarové systémy dělat. Tato specifikace je obvykle napsána v přirozeném jazyce, ale velkou výhodou bývá rovněž užití speciálních nástrojů pro tvorbu

požadavků. Toto definování požadavků je pouze počátkem procesu tvorby softwarového systému. Jedná se o počáteční vstup do analýzy a návrhu.[1]

V následující tabulce budou popsány vlastnosti specifikace požadavků a jejich stručný popis.

Tabulka 1: Vlastnosti a popis specifikace požadavků:

| Vlastnost | Popis |
|--------------------------------------|--|
| správná | každý požadavek je reprezentován něčím, co se od vyvíjeného systému požaduje, |
| jednoznačná | každý požadavek má pouze jednu interpretaci a pokrývá jen jednu skutečnost, |
| úplná | specifikací požadavků je obsaženo vše, co se očekává, že bude systém umět, |
| verifikovatelná | každý jeden požadavek lze ověřit, |
| konzistentní | žádný požadavek uvedený ve specifikaci není v rozporu s jinými předchozími dokumenty a žádný soubor požadavků uvedený ve specifikaci není konfliktní, |
| srozumitelná | specifikace je dobře pochopitelná a různé zájmové skupiny o ní mohou diskutovat, |
| modifikovatelná | strukturou a stylem specifikace je umožňováno děláním změn úplně a konzistentně, tj. specifikace musí být lehce modifikovatelná, aby se dala přizpůsobit změnám složitých systémů, |
| se zjištělým původem | specifikací je jasně identifikován zdroj každého požadavku, |
| sledovatelná | specifikace je zapsána tak, že umožňuje odkazovat na jeden konkrétní požadavek, |
| nezávislá od návrhu | specifikace není orientována na konkrétní softwarovou či hardwarovou architekturu, algoritmus apod., |
| anotovaná/vybavená poznámkami | specifikace může být díky poznámkám využita v procesu vývoje jako návod, |
| stručná | jednodušší specifikace je upřednostňována před rozvláchnou specifikací, |
| organizována | ve specifikaci je možné lehce vyhledat jednotlivé požadavky. |

Zdroj: upraveno podle [27]

Proces tvorby požadavků na systém [12]:

- začíná definicí klíčových požadavků na systém
 - stanovení hranic systému,
 - určení, co systém řeší a co již nikoliv,
 - určení hlavních funkcí systému,

- následuje definice typů uživatelů,
- pokračuje sběrem dalších požadavků a jejich přehledným zaznamenáním.

Během celého procesu definování požadavků je snaha o zachycení [12]:

- hranic a prostředí systému,
- typů uživatelů systému,
- klíčových požadavků na systém,
- požadavků typu „omezení“,
- požadavků na bezpečnost systému a jeho dat,
- požadavků na integraci systému do infrastruktury stávajících systémů.

2.2.2 Data

Data jsou hlavním předmětem operací v informatice. Jsou tedy zdrojem pro přípravu a zpracování informací. Mohou být různě strukturovaná i různě organizovaná.[8]

Nezbytnou součástí analýzy systému je získávání dat o požadavcích na systém. Mezi základní metody získávání dat o požadavcích na systém jsou zařazovány [12]:

- rozhovory s uživateli – spočívá v kladení otázek zástupcům uživatelů a v dokumentování jejich odpovědí,
- studium dokumentací a předpisů z dané oblasti – studium dokumentací a předpisů z dané oblasti řešení, dokumentace ke stávajícímu IS, technologické postupy práce podnikových útvarů, vstupní informace, výstupní sestavy,
- pozorování prostředí uživatele,
- dotazník,
- studium seznamu problémů, návrhů změn – v případě, že vyvíjenému systému předchází jiný,
- porovnání s obdobnými existujícími systémy,
- setkání uživatelů, workshopy.

Z hlediska práce s daty je možné rozlišovat [24]:

- strukturovaná data – explicitně zachycují fakta, atributy, objekty, přičemž významným rysem je existence určitých elementů dat, typickým příkladem je

ukládání dat pomocí relačních databázových systémů, díky strukturovanému uložení je možné snadno vybírat jen ta data, která jsou zapotřebí pro řešení nějakého informačního problému,

- nestrukturovaná data – jsou vyjádřena jako tok bytů bez dalšího rozlišení, jedná se např. o videozáznamy, zvukové nahrávky, obrázky nebo také textové dokumenty.

Z pohledu pevné struktury uložení dat jsou rozlišeny následující úrovně [19]:

- jednotlivé znaky (numerické, alfanumerické, atd.),
- položky – reprezentují určitou vlastnost sledované reality,
- záznamy – kolekce vzájemně souvisejících položek,
- soubory dat – kolekce vzájemně souvisejících záznamů,
- báze dat – je na nejvyšší úrovni hierarchie dat, kolekce integrovaných a vzájemně souvisejících souborů dat.

Z hlediska formy prezentace dat uživateli jsou data, tj. počítačově zachycený výsek sledované reality, prezentována jako obraz a zvuk. Obrazová data jsou tvořena texty a grafické objekty (obrázky, fotografie, schémata, atd.).[8]

Data je potřeba ukládat. Uložení dat je zajištěna jejich trvanlivost mezi dvěma spuštěními programu. Z hlediska způsobu uložení jsou využívána za úložiště dat nějakou formu souborů. Existují dva základní přístupy v uložení strukturovaných dat. Jeden je tradiční a druhý je označován jako databázový.[8]

Tradiční přístup k ukládání dat je stále využíván. Uživateli jsou stále ukládány dokumenty, obrázky a fotografie, hudba a jiné vytvořené pomocí autoringových nástrojů (např. aplikace kancelářského balíku, grafické programy atd.) formou souborů do adresářových struktur na disku.[9]

V databázovém přístupu je soubor dat označován pojmem entita, záznamy jsou výskytem entit a položky jsou označovány pojmem atribut entity. Množina vzájemně souvisejících entit je představována jako databáze. Organizaci entit a přístupu k datům je řízen specializovaným programovým vybavením, tzv. systémem řízení báze dat (SŘBD). Databáze a SŘBD dohromady tvoří databázový systém.[9]

2.2.3 Sběr požadavků

Ve fázi sběru požadavků je zahrnut soubor činností, kterými je umožněn vznik či objev požadavků. Jedná se o fázi, která je extrémně závislá na komunikaci a je využita metoda kvalitativního výzkumu. Typické činnosti mohou být rozděleny do těchto fází [2]:

- poznání dané oblasti – popis a porozumění vize, rozsahu a organizačních, politických, sociologických a dalších aspektů prostředí, ve kterém bude software používán,
- identifikace zdrojů požadavků – nalezení a popis existujících zdrojů požadavků v různých formátech, zdrojem může být investor, jednotlivé třídy uživatelů, stávající IS a používaná dokumentace,
- výběr metod, přístupů a nástrojů – zhodnocení vlastností projektu a výběr z vhodných alternativ, nejčastěji jsou odvozeny od metodik používaných ve vývojové firmě, úrovně složitosti a bezpečnosti softwaru a použitém životním cyklu,
- sběr požadavků od zúčastněných stran a ostatních zdrojů – podrobné zkoumání potřeb zúčastněných stran, hlavně uživatelů.

Pro ukončení fáze sběru požadavků není žádná definice. Jsou zaznamenány pouze doporučení, kterými jsou popsány situace, které tomu mohou napovídat. Pomyslná čára pro ukončení požadavků není nadefinována. Vždy se vyskytne situace, ve které je možné nalézt další funkční požadavek, nebo dojde k organizační či technologické změně v organizace. K ukončení sběru požadavků by mělo dojít, pokud jsou objevovány opakující se témata, požadavky s nízkou prioritou, požadavky neodpovídající vizi a rozsahu a jiné.[16][21][27]

2.2.4 Kontrola požadavků

Kontrolou je postaráno o to, aby požadavky splňovaly všechny žádoucí vlastnosti specifikace, aby byly rozděleny podle důležitosti, a aby byla výsledná specifikace úplná, jednotná, udržovatelná a související informace v ní šly dobře dohledat. Kontrola požadavků se snaží zajistit následující [27]:

- aby specifikace správně popisovala požadované vlastnosti systému, které jsou potřeba pro uspokojení jednotlivých účastníků projektu,
- aby byly požadavky na software správně odvozené ze systémových požadavků, podnikatelských pravidel a dalších zdrojů,

- aby byly požadavky úplné a kvalitní,
- aby si požadavky navzájem neprotiřečily,
- aby požadavky poskytovaly přiměřenou základnu pro návrh a implementaci systému.

Vyspecifikované požadavky si je možné představit jako popis modelu části světa, který je nutné před použitím ověřit, zda byl vytvořen správně. Správnost je ověřena shodou modelu s realitou a s očekáváním zákazníka. Je možné, že při sběru dojde ke špatnému pochopení požadavků zákazníka. Touto fází je zaručeno odstranění chyb, které vznikají při vývoji požadavků, a které by následně byly zahrnuty do systému při návrhu.

Kontrolu požadavků je možné rozdělit na dvě části. Jednou z částí je verifikace a druhou je validace. Pod pojmem verifikace požadavků je představována kontrola správnosti definování požadavků analytikem. Tato část je řešena bez účasti zákazníka. Jedná se o kontrolu správnosti, úplnosti, konzistence a dalších vlastností specifikace požadavků. Na druhé straně validace je kontrola požadavků zákazníkem. Kontroluje se zda došlo ke správnému a úplnému zachycení požadavků.[2][16][17]

2.2.5 Funkce a funkcionalita systému

Funkcí a funkcionalitou je představen statický pohled na obsah IS/ICT. Funkce je vymezena jako obsahově určená skupina operací s daty, vztahující se k určité definované potřebě uživatele. Funkcemi informatiky je chápána obsahová stránky činností nebo schopností informatiky, tj. co z hlediska potřeb uživatele umí nebo má umět. Funkcemi je rozuměno např. založení záznamu nového zákazníka, zpracování přehledu zákazníků a jejich zakázek, vytvoření nabídky, objednávky, faktury apod. Funkcionalita je hierarchicky uspořádaný souhrn poskytovaných, požadovaných nebo plánovaných funkcí.[8][19]

Funkce i funkcionalitu je možné rozlišovat na různých úrovních informatiky, a to [8]:

- na úrovních celého informačního systému nebo jeho části (oblasti), pak je hovořeno o funkcionalitě informačního systému nebo funkcionalitě řízení prodeje, řízení výroby, účetnictví apod.,
- u prostředků aplikačního softwaru nebo jejich modulů, takže např. je sledována a posuzována funkcionalita aplikačního softwaru pro řízení celého podniku, funkcionalitu modulu řízení prodeje, modulu řízení výroby, účetnictví apod.,

- u prostředků základního softwaru, je sledována např. funkce databázových systémů, operačních systémů apod., obdobně pak i u technických prostředků.

Každá z jednotlivých funkcí je obvykle definovaná celou řadou dílčích charakteristik, z nichž musí být jasné [8]:

- přesné vymezení obsahu funkce, tj. dílčích operací s daty, jejich charakteru, např. výpočtů, postupů apod.,
- jaké jsou vstupy a výstupy funkce (datové báze, dokumenty, zprávy apod.),
- komu je funkce určena, resp. jaký mají k ní jednotliví uživatelé nebo jejich skupiny přístup, těmito skupinami jsou především interní pracovníci firmy, ale i partneři – externí uživatelé, jako např. zákazníci, dodavatelé, veřejnost,
- kde je funkce realizována nebo jak je distribuována, např. pouze na centrále společnosti, na závodech, provozech,
- jaké jsou její specifické nároky na informační a komunikační technologie (nároky na objem a rychlost přenosových kanálů, na rozsah databází, jejich bezpečnost).

Základní kategorie funkcí z hlediska charakteru operací s daty jsou [8]:

- transakční funkce – slouží pro vytváření a aktualizace datovýchází, např. založení nového zákazníka do zákaznické báze,
- analytické a plánovací funkce – představují zpracování nejrůznějších přehledů, analýz nebo podnikových plánů, např. přehled tržeb za zboží podle zákazníků, tabulka vývoje prodeje podle druhů zboží za několik posledních měsíců,
- speciální, správní a provozní funkce – zajišťují obvykle archivace a zálohování dat, správu číselníků, např. číselníku zboží, zákazníků, dodavatelů.

3 NÁSTROJE ANALÝZY INFORMAČNÍHO SYSTÉMU

Nástroj je prostředkem k uskutečnění určité činnosti v procesech analýzy, vývoje a provozu IS a prostředkem k vyjádření výsledku této činnosti. Nástroj je často svázán s konkrétní technikou. Nástroje jsou vždy formalizovány vyjádřením, proto je možné a žádoucí, aby byly v maximální míře automatizovány.[22]

V literatuře [22] jsou nástroje rozděleny podle toho, do jaké etapy vývoje informačního systému jsou zahrnuty. Níže jsou uvedeny pouze nástroje, které se týkají fáze analýzy vývoje informačního systému a jsou využívány ve výuce předmětů PISR a KISR.

3.1 ER-diagram

Diagram entit a vztahů je grafický nástroj. Je používán k vyjádření datových objektů – entit (Entity), jejich podstatných vztahů (Relationship) a podstatných vlastností – atributů (Attribute) těchto objektů a vztahů.[22]

3.1.1 Základní prvky používané v ER modelu

Entita je rozlišitelný a identifikovatelný objekt reality. Na základě podobnosti jsou slučovány do entitních množin (typů). Každá entitní množina má uveden identifikátor. Identifikátor je minimální množina atributů, které zajišťují jednoznačnou identifikaci entit v této množině. Pojmu identifikátor entit je také přidělen pojem primární klíč.[4][20][22]

Jednotlivé entity vstupují do konkrétních vzájemných *vztahů*. Při modelování jsou zajímavé typy vztahů mezi jednotlivými typy entit. Jsou charakterizovány vztahy kardinalita, parcialita (volitelnost), exkluzivita (výlučnost) a externí identifikace (slabý vztah). Způsob zachycení vztahů je jednou z nejnápadnějších odlišností způsobů kreslení (symbolik) jednotlivých modifikací ER diagramů.[20][22]

Atribut je vlastnost entit nebo vztahu. Je to datový prvek, kterým je blíže charakterizována entita nebo vztah. Má různou volitelnost, a to buď totální atribut, nebo parciální atribut. Atributy jsou dále rozlišovány na základní, odvoditelný, identifikátor (primární klíč), cizí klíč, alternativní klíč a sekundární (nejednoznačný) klíč. Základní atribut je takový, který nelze odvodit z jiných atributů. Odvoditelný je možno získat odvozením z jiných základních či odvoditelných atributů. Cizí klíč je atribut nebo množina atributů, které jsou v jiné entitě primárními klíči a jsou využívány k vyjádření vztahů v datovém modelu. Alternativní klíč je minimální množina atributů, která je použita k zajištění jednoznačné identifikaci výskytů entit

a nebyla zvolena za primární klíč. Sekundární klíč je neprázdná množina atributů, které jsou důležité pro přístup datům reprezentovaným entitou.[4][20]

3.1.2 Postup tvorby ERD

Při tvorbě datového modelu je vycházeno [22]:

- z toho, jak je analytikem pochopena modelovaná realita, zda se již vyskytuje jiný model zkoumané výseče reálného světa – například DFD včetně popisu datových toků a data storů ve slovníku dat,
- z interview s uživateli informačního systému,
- z ostatních možných zdrojů zjišťování dat – oddělení správy dat, z dokumentace stávajících aplikací apod.

ERD se většinou nepovede navrhnout napoprvé a musí být neustále upravováno a zdrobňováno podle nově získaných poznatků. ER modely jsou vytvořeny na dvou úrovních podrobnosti [22]:

- hrubý (prvotní) konceptuální datový model (konceptuální schéma reality), který je vytvářen za účelem poznání zkoumané reality, rozpoznání základních datových objektů a jejich vztahů,
- podrobný (konečný) konceptuální datový model (konceptuální schéma dat), kterým jsou popsány všechny požadované datové struktury a jejich integritní omezení, a který je využit jako předobraz datové základny.

Vlastní postup [4]:

1. Výběr nejdůležitějších objektů – entit a vztahů mezi entitami.

Pro každou entitu je zkoumáno:

- jaký je hlavní účel objektu,
 - k čemu ještě slouží,
 - z čeho se skládá,
 - jaký je jeho identifikátor.
2. Nakreslení entit a jejich vztahů do ER diagramu. Je definována kardinalita vztahů a parciálních vztahů.

3. Přidání atributů k entitám, jsou označeny parciální atributy a je ověřen identifikátor entit.
4. Kompletace hierarchie – pro každý typ specializace/generalizace je hledán atribut specializace a je zkoumáno, zda je specializace úplná, nebo proč tomu tak není.
5. Odstranění transitivních vztahů (nepřímých, které je možné odvodit z vztahů jiných).
6. Zaznamenání omezujících podmínek (integritní omezení) k entitám.
7. Odstranění nadbytečných entit z datového modelu.
8. Ověření úplnosti datového modelu. Je prováděno porovnáním obsahu datového modelu s uživatelskými požadavky či funkčním nebo procesním modelem. Při porovnávání je zkoumáno, zda zjištěné požadavky či procesy mají odpovídající vyjádření v datovém modelu.

3.2 Data Flow Diagram

Diagram datových toků je využíván jako grafický prostředek návrhu a zobrazení funkčního modelu systému. Je tedy základním nástrojem konceptuálního funkčního modelu. Funkčním modelem je popsáno z jakých procesů a z jakých návazností je realita poskládána a i jakými procesy bude tvořen informační systém. Je vyvinut z tzv. „activity diagrams“.[4]

3.2.1 Základní prvky používané v DFD

Proces (transformace, funkce) je označena kolečkem (elipsou). Jedná se o informační procesy (zpracování dat), jimiž je modelováno reálné dění. Procesem je znázorněna transformace dat, kterou je směřováno k vyprodukování výstupu (transformace vstupu na výstup). Jsou rozlišovány datové a řídicí procesy. Datovým procesem (funkcí) je vyjádřena fyzická transformace dat. Hlavním úkolem funkce je zpracovávat, transformovat data. Řídicím procesem je vyjádřen algoritmus řízení (vzájemných časových návazností) procesů v určité části systému. Řídicí proces je používán k zachycení real-time charakteristik. Každý proces musí mít název a jednoznačné číslo. Toto číslo je složeno z čísla bezprostředně nadřazené funkce a přiděleného čísla v rámci úrovně.[4][22][23]


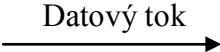

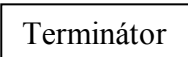
Datovým tokem (Data Flow) je reprezentován přesun dat/informací z jedné části systému do jiné, nebo z okolí systému do systému nebo ze systému do okolí. Je znázorněn šipkou. Musí být znám a pojmenován. Název musí daná data reprezentovat a jasně vyjadřovat jejich obsah. V těchto tocích jsou obsažena data, která jsou systémem zpracována a ukládána. Pro tvorbu konceptuálního funkčního modelu musí platit, že datový tok je abstrakcí jakékoliv

formy přesunu dat. Pouze je vyjádřen fakt, že data jsou přesouvána a není nic řečeno o konkrétní formě přesunu. Datové toky jsou jednou ze dvou základních forem komunikace procesů uvnitř systému. Jedná se tedy o abstrakci jakékoliv formy přesunu dat.[22][23]

Data storem (skladiště dat, datový sklad) je vyjádřen „depozitář“ dat. Data jsou uchovávány pro jejich pozdější použití. Je znázorněn pomocí dvou rovnoběžek, mezi nimiž je umístěn název. Tento symbol (symbol přerušení) je připomínkou, že uložení dat znamená přerušení toku dat v čase. Význam data store je tedy „místo dočasného uchování dat“. Může být implementován jako pole, soubor (obyčejný, databázový), nebo jako cokoli jiného, například šanon, kniha. Je využíván všude tam, kde je mezi procesy časově zpožděné předávání dat (asynchronní). Na konceptuální úrovni se jedná o abstrakci jakékoliv formy přesunu dat. Je vyjadřován pouze fakt, že data jsou uschována (jejich tok je v čas přerušen) a není řečeno nic o konkrétní formě tohoto uložení. Název skladiště dat by měl být v množném čísle. Pro každý data store musí existovat jeden datový tok dovnitř (uložení dat) a jeden ven (použití dat). Data do a z něj jsou vedena přes proces (transformaci dat). Data store je další formou komunikace procesů.[4][23]

Terminátorem (externí entita) jsou popsány objekty, které nejsou zařazeny do popisovaného systému, nýbrž do jeho podstatného okolí. Terminátorem (počátek, konec datového toku, zdroj dat, místo a účel spotřeby dat) je znázorněn externí zdroj, nebo místo určení dat. Je jím vyjádřen objekt v okolí systému, s nímž systém komunikuje. Fyzicky to může být člověk, skupina lidí, oddělení ve stejné organizaci, ale vždy se jedná o objekt vně modelovaného systému. Může se jednat i o systém, který není přímo v zájmu modelu, ale má na něj vliv. Graficky je znázorněn jako čtverec/obdélník. Terminátor by měl mít výstižný název, kterým je vyjádřen typ externího zdroje/místa určení. Při jisté míře abstrakce může být i terminátor považován za další formu komunikace procesů a to prostřednictvím okolí.[22][23]

Symbole prvků DFD [23]:

- proces 
- datový tok 
- datový sklad 
- terminátor 

3.2.2 Hierarchie DFD

Na vrcholu hierarchie se nachází pouze jeden DFD, tzv. kontextový diagram. V kontextovém diagramu je obsažen celý systém jako jedna funkce. V tomto diagramu je reprezentována hranice systému a všechny zdroje a místa určení dat (terminátory).[22]

Bezprostředním rozkladem kontextového diagramu je nižší úroveň – DFD úroveň 0. V této úrovni jsou obsaženy základní funkce systému a jejich vztahy jsou vyjádřeny prostřednictvím datových toků a data storů. Diagram úrovně 0 je posledním diagramem popisujícím celý systém. Diagramy následujících úrovní (1, 2, atd.) jsou popsány jen určité části systému (funkci z diagramu vyšší úrovně). Stejným způsobem pokračuje hierarchický rozklad nižších úrovní až na elementární úroveň. V nejnižší úrovni jsou obsaženy elementární funkce.[4][22]

3.2.3 Pravidla tvorby DFD

Číslováním procesů je identifikována jednak úroveň rozkladu, do níž jsou funkce zařazeny, a jednak i proces, jehož je daná funkce rozkladem. Číslování je prováděno shora dolů po úrovních a v rámci jedné úrovně. **Názvy procesů** (funkcí) by měli být stručné, výstižné, má vyjadřovat celou funkční náplň procesu. **Složitost DFD** by neměla být příliš vysoká. DFD by neměl být příliš komplexní. Musí být pochopitelný pro uživatele, analytika i konstruktéra systému. DFD nemůže obsahovat příliš velké množství procesů a toků. Jeden DFD by neměl mít méně než 3 a více než 9 funkcí. **Přehledné a esteticky uspořádané DFD** – musí být technicky správný (přesný, konzistentní), přijatelný a srozumitelný pro uživatele, úhledně nakreslený. **Konzistenci DFD** je označována logická soudržnost a bezrozpornost diagramu. Ta není samozřejmá, protože díky hierarchickému rozkladu je jedna skutečnost vyjadřována často ve více diagramech.[22]

3.3 Datový slovník

Data Dictionary (datový slovník) je místem centrálního popisu datových prvků v systému. Datové prvky jsou společné všem jednotlivým specifickým modelům. V datovém modelu jsou to struktury entit (atributů). Jinými strukturami z obsahově totožných dat jsou například datové toky nebo datové sklady ve funkčním modelu. Ve Structure Chart, popisujícím modulární programovou strukturu, jsou obsaženy tatáž data ve formě parametrů, předávaných při vzájemném volání modulů. Obsahově stejná data jsou zahrnuta v různých strukturách v různých diagramech. Tímto je vyvolána potřeba centrálního popisu, ze kterého bude zřejmé, v jakých všech formách (strukturách) je příslušný datový element v modelu systému

využíván. Tradičně je jazyk slovníku v textové podobě, která je založena na Backus-Naurovy formě.[22]

Jedná se o nástroj strukturovaného popisu s těmito vlastnostmi [22]:

- jednotlivé abstraktní prvky jsou definovány postupně shora dolů,
- jednotlivé definované struktury – obsahy abstraktních prvků jsou definovány vždy jako jeden z možných tří typů:
 - prvky v sekvenci jsou spojeny znaménkem „plus“,
 - varianty selekce jsou odděleny znaménkem „svislá čárka“,
 - iterace (opakování) prvku je vyjádřena složenou závorkou,
- kulaté závorky jsou využívány k účelovému vyjádření nepovinnosti prvku.

Pro úplné vymezení by měla definice zahrnovat [22]:

- význam datového prvku v rámci uživatelské aplikace,
- složení datového prvku, pokud je složen ze smysluplných elementárních prvků,
- hodnoty, kterých může elementární datový prvek nabývat, případně i se specifikací formátu.

3.4 State Transition Diagram

K modelování chování systému, které je závislé na čase, je použit STD (State Transition Diagram, diagramy přechodu stavů). Pomocí tohoto diagramu jsou popisovány časové následnosti procesů (funkcí). Tyto diagramy jsou využity při návrhu real-time systémů, ale i v ostatních typech systémů existují obecná pravidla časových návazností procesů. Diagram stavů a přechodů je nástrojem popisu algoritmu řízení – nástrojem popisu struktury řídicího procesu.[10][22]

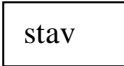
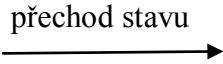
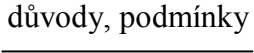
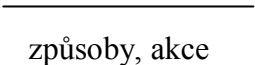
3.4.1 Základní prvky používané v STD

Diagram stavů a přechodů je definován [10][22]:

- stavy řízeného systému funkcí (obdélníky) – každý stav je východiskem k případným přechodům do dalších stavů,
- možné přechody mezi jednotlivými stavy (šipky),
- důvody (podmínky, signály) jednotlivých přechodů,

- způsoby (akce, vedoucí k uskutečnění) jednotlivých přechodů.

Symboly prvků STD [22]:

- stav 
- přechody 
- důvody 
- způsoby 

Stavem systému je vyjádřen interval mezi jednotlivými akcemi, které jsou uskutečněny v daném okamžiku. Změnou stavu je označován přechod modelovaného systému z jednoho důležitého rozpoznatelného stavu do dalšího. Změna stavu nastane při rozpoznání, že je splněna určitá podmínka. Systém je přemístěn ze stavu do stavu provedením patřičných akcí. Podmínky a akce jsou zachyceny v STD jako popis šipek (změn stavů). Modelovaný systém musí mít definován výchozí stav. Tímto je zajištěn jeden kořen – jeden vstupní obdélník. Konečných stavů systému může být více. Výchozímu stavu žádný stav nepředchází a po konečném stavu žádný další nenásleduje. Popisem stavů a přechodů je definována vnitřní struktura řídicího procesu – daný mechanismus řízení systému funkcí. Řídicí proces tím, že je znám aktuální stav řízeného systému funkcí, je schopen se v případě přijetí signálu příslušného řídicího toku kvalifikovaně rozhodnout o následující akci systému. [10][22]

3.4.2 Hierarchie STD

Reálný systém má obvykle desítky stavů, jež je třeba rozlišit. V tomto případě je použito členění a hierarchizace diagramů obdobná jako u DFD. Každý stav je pak popsán samostatným STD. Vazba mezi úrovněmi je zviditelněna číslováním stavů. Zásady číslování jsou analogické jako u DFD. Hierarchická dekompozice je dekompozicí jejich stavů a není dekompozicí řídicích procesů. Řídicí algoritmus je vždy jeden jediný. [22]

Konzistence uvnitř STD znamená že [22]:

- jsou definovány všechny stavy odpovídající realitě,
- jsou definovány všechny změny stavů v realitě přípustné,

- do každého ze stavů se lze nějak dostat,
- popisovaný systém reaguje v každém stavu na všechny podmínky v tomto stavu v realitě přípustné.

3.5 Případ užití

Případem užití (Use case) je specifikována část funkcionality systému, kterou využívá aktér, a která plní určitý cíl. Případy užití jsou prezentovány funkčními požadavky na vyvíjený systém. Název případu užití by tento cíl měl vyjadřovat a je pro něj používána slovesná vazba.[4]

Modelování případů užití je jednou z forem řízení požadavků a jedná se o iterativní proces, ve kterém je postupováno vpřed postupným upřesňováním. Nejprve je zahájeno s pouhým názvem případu užití, později jsou k názvu připojeny i další podrobnosti. Tyto podrobnosti jsou složeny z počátečních krátkých popisů, které jsou upřesněny do úplné specifikace.[1]

Modelování případů užití je složeno z následujících aktivit [1]:

- nalezení hranic systému,
- vyhledání účastníků (actors),
- nalezení případů užití,
- specifikace případů užití,
- tvorba scénářů.

3.5.1 Základní prvky používané v případě užití


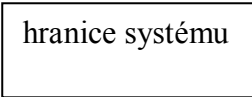
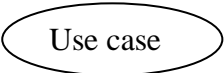
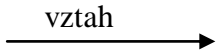
Výstupem uvedených aktivit je model případu užití. V tomto modelu jsou obsaženy čtyři komponenty [1]:

- účastníci (actors) – role přidělené osobám nebo předmětům používajícím daný systém,
- případy užití (use case) – činnosti, které mohou účastníci se systémem vykonávat,
- relace (relationships) – smysluplné vztahy mezi účastníky a případy užití,
- hranice systému (system boundary) – ohraničení zobrazené kolem případů užití, jež je vyznačením území nebo hranic modelovaného systému.

Vztahy případů užití [1]:

- zobecnění (generalizace) – je použita pouze v případech, kdy je jeden nebo více případů užití, jež jsou v podstatě určitou specifikací obecnějšího případu,
- include – jsou vyčleněny kroky společné několika případům užití do samostatného případu užití, který je do příslušných případů užití následně zahrnut,
- extend – je způsobem, jímž je možné do existujícího případu užití rozšířit o nové chování.

Symbyly prvků případu užití [27]:

- aktér 
- hranice 
- případ užití 
- vztahy 

3.5.2 Diagram případu užití

Diagramem případu užití je možné popsat chování systému z hlediska uživatele. Je zde specifikováno, jaké typy uživatelů (lidé i jiné systémy) jsou využívány systémem a jaké činnosti jsou vykonávány. Model případů užití je složen z diagramů případů užití a specifikací (slovních popisů) jednotlivých případů užití.[4]

3.5.3 Tvorba diagramu případu užití

V tomto diagramu je hranice systému vyjádřena rámečkem. Popiskem je určen jeho název. Účastníci jsou znázorněni vně hranice systému, zatímco případy užití, kterým je utvářeno chování systému, je umístěn uvnitř těchto hranic. Vztah mezi účastníkem a případem užití je znázorněn pomocí plné čáry, což je symbol přiřazení. U tohoto typu přiřazení je předpokládáno implicitní stereotyp „communication“, kterým je označováno, že účastník a případ užití spolu určitým způsobem komunikují. Zmiňovaný stereotyp v diagramu není nikdy explicitně zobrazen.[1]

3.5.4 Scénáře

Funkcionalita vyjádřená případem užití je realizována jako posloupnost interakcí mezi aktérem a systémem. Tato posloupnost je označena jako scénář případu užití. Scénářem je představována jedna instance případu užití, tedy jeden průchod případem užití od začátku do konce. Na začátku je spouštěcí událost, dále je pokračováno, dokud není dosaženo cíle případu užití anebo je přerušeno. Případ užití je množina možných scénářů. Specifikace případu užití je slovní popis, ve kterém je obsažen kromě základního, úspěšného, také popis alternativních scénářů. Slovní popis by měl být srozumitelný pro uživatele, protože je podkladem pro diskusi o požadavcích na systém.[4]

Scénáře jsou dalším způsobem, jímž je možné nahlížet na případy užití. Je vyjádřením jedné zvláštní cesty případem užití. Důležitým aspektem je skutečnost, že není větven. Znamená to, že každé eventuální rozvětvení může být nakonec vyústěním v samostatný scénář. Každý případ užití je obsažen přesně v jednom hlavním scénáři (primary scenario). V hlavním scénáři je předpokládáno, že vše půjde jako po másle a že během pouti nedochází k ničemu nepředpokládanému – k žádným chybám, k žádným odchylkám, přerušením nebo rozvětvením. Každý případ užití má ovšem i své vedlejší scénáře (secondary scenarios). Jsou to alternativní cesty tokem událostí. Případ užití je možné považovat za deltu obrovské řeky. V každém případě užití je nalezen jeden hlavní scénář, který vede hlavním korytem zmiňované delty. Další, o něco menší koryta jsou považovány za vedlejší scénáře. Vedlejšími scénáři jsou zachyceny chyby, rozvětvení a přerušení hlavního toku.[1]

3.6 Metoda IDEF

Metoda je věnována modelování funkcí informací. Byla vyvinutá pro potřeby ministerstva obrany USA. IDEF0 (Integrated Definition for Function Modeling) je určena pro modelování činností, rozhodnutí a akcí podniku. Činnosti jsou ve vztahu svými vstupy, výstupy, kontrolními signály a mechanismy. IDEF0 však není určena k použití pro modelování sledu činností, ani k pochopení toho, jak činnosti přidávají hodnotu ekonomickým zdrojům. Hlavní výhodou je, že jsou použita jako nástroj pro komunikaci mezi uživateli, odborníky a projektanty.[25][26]

Základem jsou následující metody [25]:

- IDEF0 – je určena pro účely sestavení funkčního modelu, který strukturovaným způsobem popisuje funkce modelované oblasti

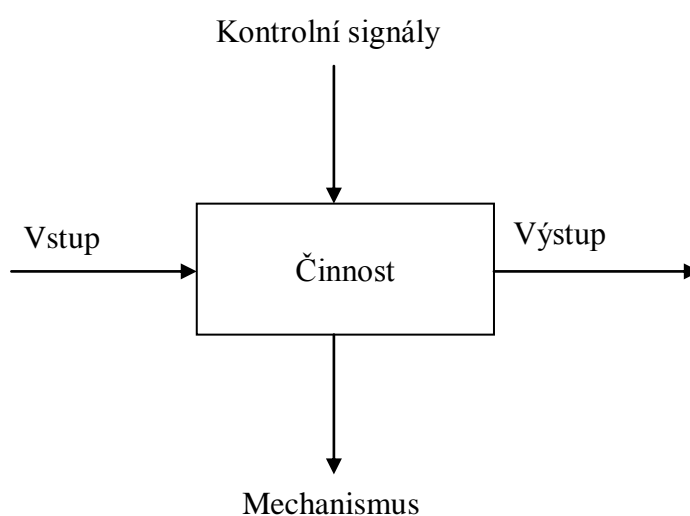
- IDEF1 – slouží k sestavení informačního modelu, kterým je reprezentována struktura a sémantika informací
- IDEF2 – popisující dynamiku systému, tedy jeho chování.

3.6.1 Základní prvky používané v IDEF0

Objekty IDEF0 jsou lidé, stroje nebo systémy, kterými je zajištěna transformace vstupů na výstupy. Na obrázku 3 jsou ukázány základní prvky IDEF0. Primárními modelovacími prvky jsou funkce a data (objekty), které vzájemně tyto funkce propojují.[26]

Základními prvky jsou [25]:

- funkce – popisuje činnost transformující vstup na výstup,
- vstup – data, která budou funkcí transformovány na výstup,
- výstup – data produkované funkcí,
- řízení – je dáno pravidly potřebnými k vytvoření požadovaného výstupu,
- mechanismus – je definován prostředky nutnými k realizaci funkce.



Obrázek 3: Základní prvky IDEF0

Zdroj: zpracováno podle[26]

3.6.2 Hierarchie IDEF0

Účelem použití IDEF0 je vytvoření modelu, který je složen z hierarchicky uspořádané sady diagramů a textů. Hierarchie je přesně vytvořený systém vzájemných odkazů popisujících funkce organizace nebo podniku. Každé funkci je přiděleno číselné označení a označení diagramu. Funkce je dále rozpracována do dalších podfunkcí. Jedná se o stromový přístup dekompozice funkce. Vrchol hierarchie je definován kontextovým diagramem označeným písmenem a číslem 0. Při sestavování diagramů jsou dodržovány zásady jejich řazení ve směru diagonály a diagram by neměl mít méně než tři a více než šest funkcí. Výstupy dané funkce mohou být vstupem, řízením nebo mechanismem jiných funkcí. Tímto způsobem je zajištěna vzájemná závislost mezi funkcemi.[25]

4 VYBRANÝ INFORMAČNÍ SYSTÉM VEŘEJNÉ SPRÁVY

Následující kapitola bude věnována informačnímu systému Munis. Tento systém je určen pro města a obce. Systém byl vybrán s ohledem na využití ve výuce předmětu Informační systémy regionů. Celá kapitola je čerpána z jednoho zdroje [11], a tím je stránka firmy Triada věnována IS Munis.

Při vývoji bylo dbáno na dodržování standardů ISVS, na bezpečnost dat proti ztrátě či zneužití. Řadu údajů je možné zadávat jednoduchým a rychlým výběrem z číselníků.

Systém je navržen jako modulární a otevřený a je složen z navzájem spolupracujících a propojených agend. Uživatelé si mohou vybrat jednotlivé moduly podle rozsahu agend, které zpracovávají. Systém je pravidelně, minimálně dvakrát ročně, aktualizován, a to na základě probíhajících legislativních změn a jako důsledek přirozeného vývoje IS, jako např. modernější technologie nebo přání uživatelů. Principem otevřenosti je zajištěno to, že k systému je možné připojovat aplikace třetích stran. Systémem Munis je umožněna oboustranná komunikace s okolím. Jedná se zejména o vstupy a výstupy v oblasti účetnictví, registru obyvatel, katastru nemovitostí, sociálních dávek, služeb pošty a bankovních ústavů.

V IS Munis je využívána územní identifikace založená na celostátním registru územní identifikace, ve kterém jsou obsaženy veškeré adresní body budov v ČR. Pro účely aktualizace jsou vydávány změnové soubory, které je možné načítat do systému adres a tím udržovat adresy v aktuálním stavu. Tento registr je používán v rozšířené podobě. Změnové soubory jsou stahovány prostřednictvím Internetu.

Mezi základní rysy IS Munis jsou zařazeny [11]:

- aktualizace zahrnující legislativní změny, zapracování modernějších technologií a přání uživatelů,
- serverová část je závislá na použitém SŘBD,
- přímá podpora tvorby dokumentů v prostředí MS Word a 602 Text za pomoci šablon a vzorů,
- integrace Elektronické podatelny a elektronického podpisu,
- modularita a otevřenost,
- jednotné ovládání dle standardů Windows,

- celostátní registr adres spolupracující s registrem Ministerstva práce a sociálních věcí,
- Český generátor tiskových sestav, společnosti SINEA, umožňující libovolné vytváření a úpravy tiskových sestav v uživatelsky jednoduchém prostředí,
- každou sestavu je možné uložit jako skupinu HTML souborů, tiskové výstupy lze navíc jednoduše převádět do formulářů RTF a PDF, které jsou dále nezávislé na vlastním systému Munis,
- systém přímo spolupracuje s elektronickou poštou,
- možnost přímého odesílání SMS.

Jak už bylo uvedeno výše IS Munis je navržen jako modulární, skládající se z navzájem spolupracujících a propojených agend. Je vybíráno z jednotlivých modulů podle rozsahu agend, které jsou zpracovávány.

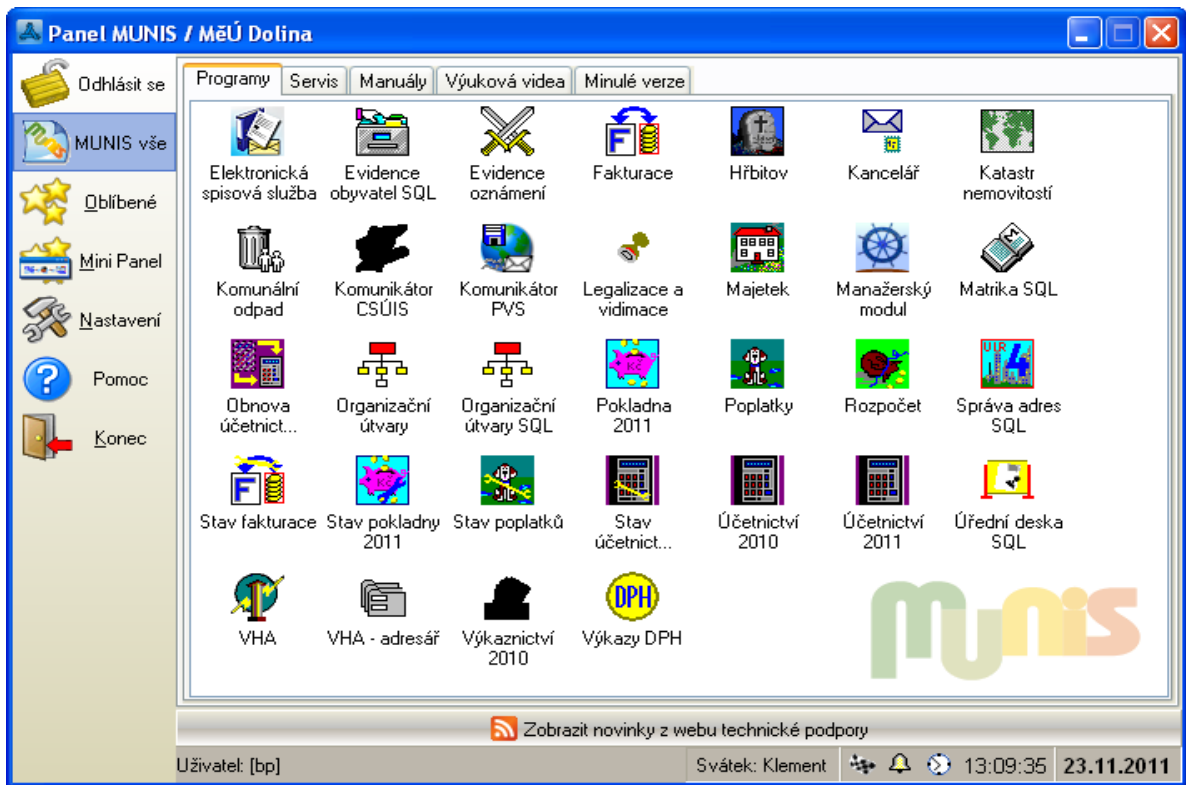
Moduly je možné rozdělit následujícím způsobem do dvou skupin. Jednou ze skupin jsou evidenční a správní agendy. Do této skupiny patří například moduly správa adres, evidence obyvatel, matrika a další. Druhou skupinou jsou ekonomické agendy. V této skupině jsou začleněny moduly evidence hřbitova, komunální odpad, poplatky a jiné.

Na obrázku 3 jsou prezentovány moduly, které jsou podporovány IS Munis. Tato práce bude zaměřena pouze na jeden z těchto modulů. Tímto modulem je evidence obyvatel. Tímto modulem jsou poskytovány údaje i jinému modulu, jako například modulu poplatky, pokladna, hřbitov a dalším. Jediným modulem, se kterým není možné spolupracovat, je modul matrika. Tato spolupráce je zakázána legislativou.

V modulu evidence obyvatel je umožněno zpracování živé kartotéky, ve které je sledována řada údajů o občanovi s trvalým bydlištěm v dané obci. Změny jsou evidovány v tzv. mrtvé kartotéce.

Z pořízených dat je možné získat přehled o pohybu obyvatelstva v obci během časového období – počty narozených, zemřelých lidí, stěhování v rámci obce i za hranice katastru. Do statistického sledování jsou zahrnovány například počty obyvatel v jednotlivých částech obce, sledování věkového složení obyvatel, rodinné vztahy, přehled jubilantů.

Aktualizace je prováděna pomocí základního i aktualizacího výdeje dat předávaného obcím.



Obrázek 4: Moduly IS Munis

Zdroj: vlastní zpracování

5 PŘÍPADOVÁ STUDIE DEFINOVÁNÍ POŽADAVKŮ NA EVIDENCI OBYVATEL INFORMAČNÍHO SYSTÉMU MUNIS

Tato případová studie bude poskytnuta studentům předmětu PISR a KISR jako ukázkový příklad definování požadavků. Jako informační systém byl vybrán IS Munis, se kterým jsou studenti tohoto předmětu seznámeni během výuky. V následující kapitole budou ukázány nástroje, se kterým je pracováno, a studenti jsou s nimi seznámeni.

Jak už bylo zmíněno, IS Munis je systém, se kterým je pracováno na obecních a městských úřadech. Jelikož tento systém má mnoho modulů, pro tuto studii byla vybrána evidence obyvatel, jelikož od této agendy občan požaduje nejvíce dokumentů, a se kterou spolupracují a jiné agendy, kterým jsou poskytnuty údaje z této evidence.

Podle magistrátu města Hradec Králové je rozsah činností úřadu spojeného s evidencí obyvatel následující [18]:

- vydávání výpisu údajů z IS evidence obyvatel,
- přijímání žádostí od občana – těmito žádostmi jsou vyřizovány osobní doklady občana, čili občanský průkaz a cestovní doklady,
- vydávání správních rozhodnutí o zrušení údajů o místu trvalého pobytu,
- hlášení změn trvalého pobytu,
- správní řízení ve věci změny jména a příjmení,
- navrácení přecházejícího příjmení po rozvodu.

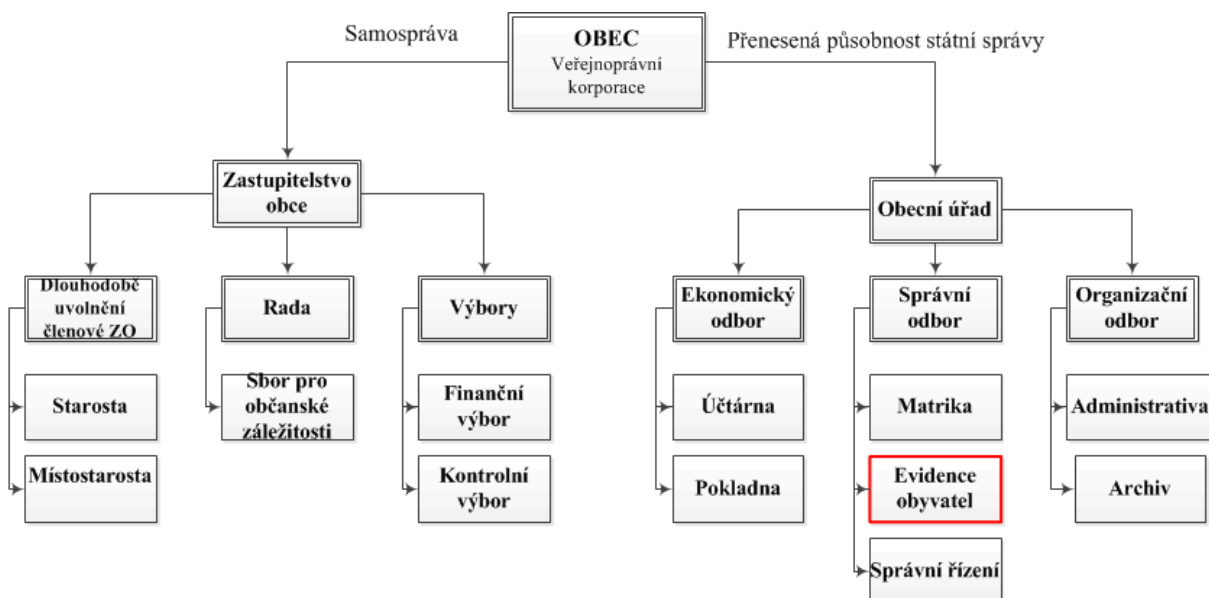
Rozsah těchto činností je možné pokládat za požadavky, které jsou vyžadovány od občana, aby systém provedl. Tyto požadavky nejsou do systému zavedeny občanem, ale příslušnou úřednicí obecního úřadu, která má k tomuto úkonu oprávnění.

Informační systému musí být připraven na různé situace, se kterými občan přichází na úřad a je očekáváno, že tyto situace budou vyřešeny.

Externím dodavatelem, firmou Triada, informačního systému Munis je prováděna aktualizace dvakrát ročně.

Obecní úřad je institucí, ve které jsou poskytovány služby veřejné správy. Jsou zde zajištěny agendy, které jsou úřadu svěřeny státem v rámci tzv. přenesené působnosti a rovněž agendy samosprávy v rozsahu, které mu jsou uloženy zastupitelstvem. Úřad je členěn na

odbory a ty jsou dále členěny na oddělení. Do čela obce je volen starosta. Podrobnější členění je ukázáno na následujícím obrázku.



Obrázek 5: Organizační struktura obce

Zdroj: vlastní zpracování

5.1 Popis informačního systému

Účastníci:

- úřednice,
- občan.

Informace:

- údaje pro zpracování požadavku občana,
- evidence obyvatel,
- registr obyvatel.

Technologie:

- HW,
- SW (IS Munis).

Procesy:

- přijetí žádosti od občana,
- zpracování žádosti,
- vydání občanského průkazu,

- vydání cestovních dokladů,
- změna trvalého pobytu,
- vytištění výpisu údajů z evidence obyvatel,
- poskytování údajů jiné oprávněné agendě,
- archivování údajů o občanovi.

Produkty, služby:

- výpis údajů z evidence obyvatel,
- občanský průkaz,
- cestovní doklad,
- hlášení o změně trvalého pobytu,
- změněné údaje na žádost občana,
- archivace údajů o občanovi.

Zákazníci:

- občané.

Prostředí – okolí:

- stát,
- jiná obec nebo město,
- legislativa,
- ministerstvo vnitra.

Infrastruktura:

- komunikační,
- technická,
- informační.

5.2 Nástroje definování požadavků

5.2.1 Hierarchie funkcí

Hierarchie funkcí je statický pohled, kterým je popsán hierarchický rozpad funkcí informačního systému. Na nejnižší úrovni rozpadu jsou definovány elementární funkce, které jsou uživatelům k dispozici. Funkce vyšší úrovně jsou pojmenováním určité množiny souvisejících funkcí. Každá funkce by měla mít svůj stručný popis.

V této studii je proveden rozklad informačního systému, který se zabývá evidencí obyvatel. Jak už bylo výše naznačeno, jednou z funkcí tohoto systému je vyřizování žádostí, které jsou podávány občanem. Toto vyřizování žádostí je rozčleněno na příjem a výdej

konkrétních žádostí od občanů. Další funkcí, která je žádána od systému, je poskytování údajů. Tyto údaje jsou poskytnuty jiné úřednici, kterou je o tyto informace požádáno, a která tyto údaje potřebuje k výkonu své práce. Jedná se o informace pro další moduly, jako jsou komunální odpad a hřbitovní poplatky. V těchto modulech jsou údaje o občanovi přiřazeny jejich konkrétnímu poplatku. Poslední funkcí, kterou je možné od systému vyžadovat, je archivace údajů. Tato archivace je daná legislativou. Archivace údajů při úmrtí je vymezena legislativou na dobu 75 let. Hierarchie funkcí informačního systému, podporujícího evidenci obyvatel, je na následujícím obrázku.

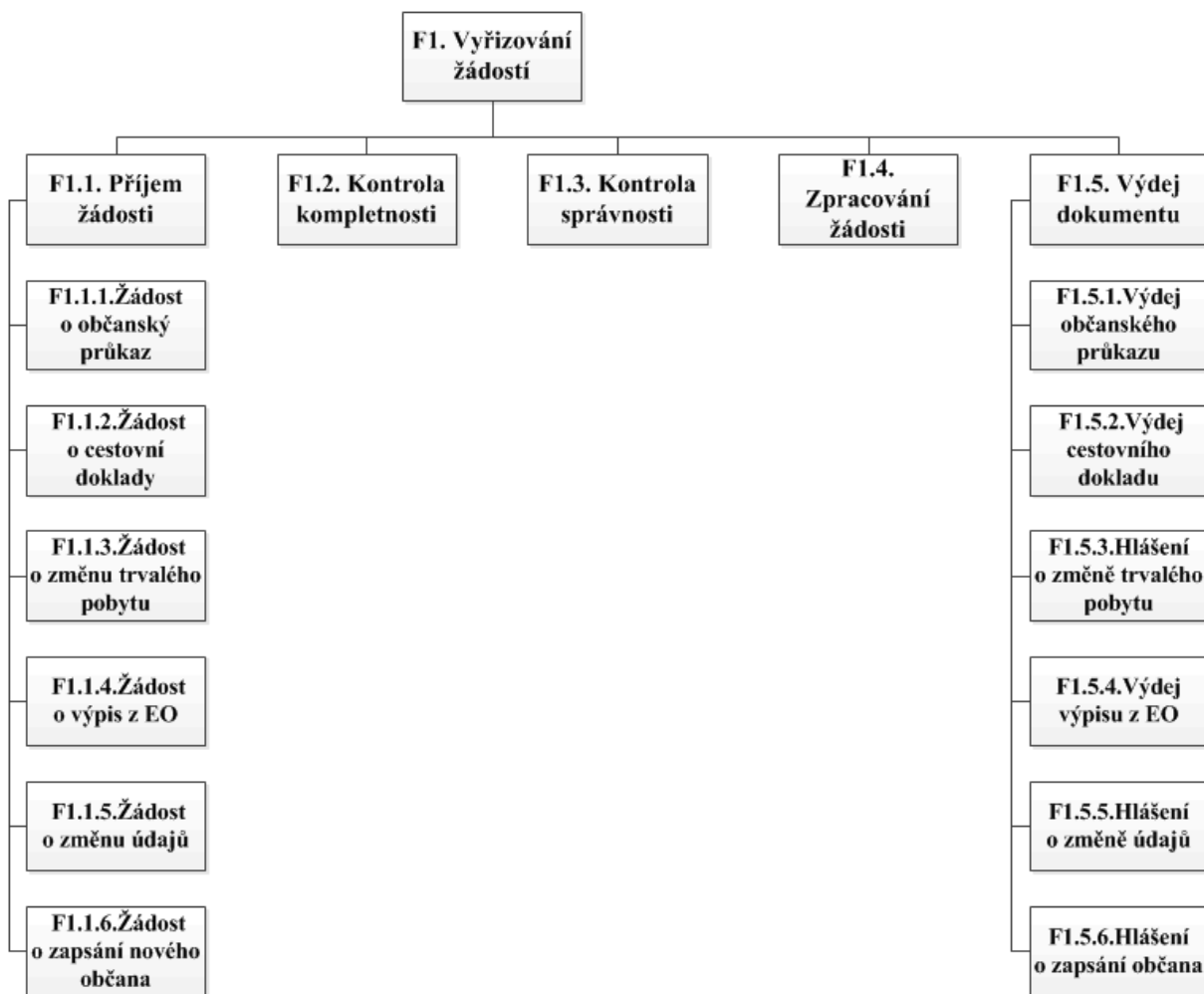


Obrázek 6: Hierarchie funkcí

Zdroj: vlastní zpracování

Jelikož je hierarchie funkcí rozsáhlá, pro další zpracování je vybrána pouze část tohoto rozkladu. V následujícím textu bude zpracována pouze funkce vyřizování žádostí, jejíž hierarchie je ukázána na obrázku 7.

Hierarchický rozklad je složen z pěti funkcí a to příjem žádosti, kontrola kompletnosti, kontrola správnosti, zpracování žádosti a výdej dokumentu. Příjem žádosti je dále rozložen podle toho, co od úřadu občan požaduje. Kontrolou kompletnosti je žádost přezkoumána z pohledu vyplnění povinných údajů. Kontrolou správnosti je zkontrolována, jestli občan vyplnil údaje na žádosti správně a je-li zařazen do evidence obyvatel. Žádost je zpracována úřednicí příslušného úřadu. Výdej dokumentů je dále rozložen podle toho, jaký dokument úřednice občanovi vydává.



Obrázek 7: Hierarchie funkce Vyřizování žádostí

Zdroj: vlastní zpracování

V tabulce je ukázáno, jak by mohl vypadat popis funkcí informačního systému.

Tabulka 2: Popis funkcí

| Číslo funkce | Název | Popis |
|--------------|---------------------------|---|
| 1 | Vyřizování žádostí | Zpracování jednotlivých žádostí, poskytování údajů jiné agendě, archivace údajů. |
| 1.1. | Příjem podaných žádostí | <ul style="list-style-type: none"> příjem podaných žádostí, výdej vyhotovených dokumentů. |
| 1.1.1. | Žádost o občanský průkaz | <ul style="list-style-type: none"> příjem podané žádosti, vydání potvrzení. |
| 1.1.3. | Žádost o cestovní doklady | <ul style="list-style-type: none"> převzetí žádosti, vydání potvrzení. |
| 1.2 | Kontrola kompletnosti | <ul style="list-style-type: none"> kontrola kompletnosti úřednicí. kontrola povinných údajů |
| 1.5. | Výdej dokumentu | <ul style="list-style-type: none"> výdej dokumentů, potvrzení o převzetí |

| | | |
|--------|--------------------------|---|
| 1.5.1. | Výdej občanského průkazu | <ul style="list-style-type: none"> • výdej dokumentu - OP, • kontrola občanem, • podepsání převzetí. |
| 1.5.2. | Výdej cestovního dokladu | <ul style="list-style-type: none"> • výdej dokumentu - CP, • kontrola občanem, • podepsání převzetí. |



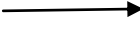
Zdroj: vlastní zpracování

5.2.2 Kontextový diagram datových toků

Tento kontextový diagram je dělaný z pohledu datových toků. Diagramem je objasněno vnější chování systému. Diagramem je popsána nejvyšší úroveň systému k jeho okolí. Jsou jím popsány požadavky, které chce od systému uživatel.

V této situaci občan podá žádost na vyřízení. Žádostí je vyvolán přenos informací. Žádost je přijatá úřednicí, která ji zpracuje. Po zpracování žádosti je vyhotoven doklad a žádost je považována za vyřízenou. Úřednicí je vydán dokument, o který bylo občanem zažádáno.

Použité symboly:

- terminátor 
- systém 
- tok informací 



Obrázek 8: Kontextový diagram funkce vyřizování žádosti

Zdroj: vlastní zpracování

5.2.3 Diagram datových toků




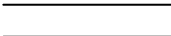
Tímto diagramem je navázáno na kontextový diagram. Je jím popsána vnitřní funkcionalita systému na základě analýzy toků dat. Tato analýza je zachycena mezi interními funkcemi systému navzájem a s okolím systému. DFD diagram je hierarchický a jeho nejvyšší

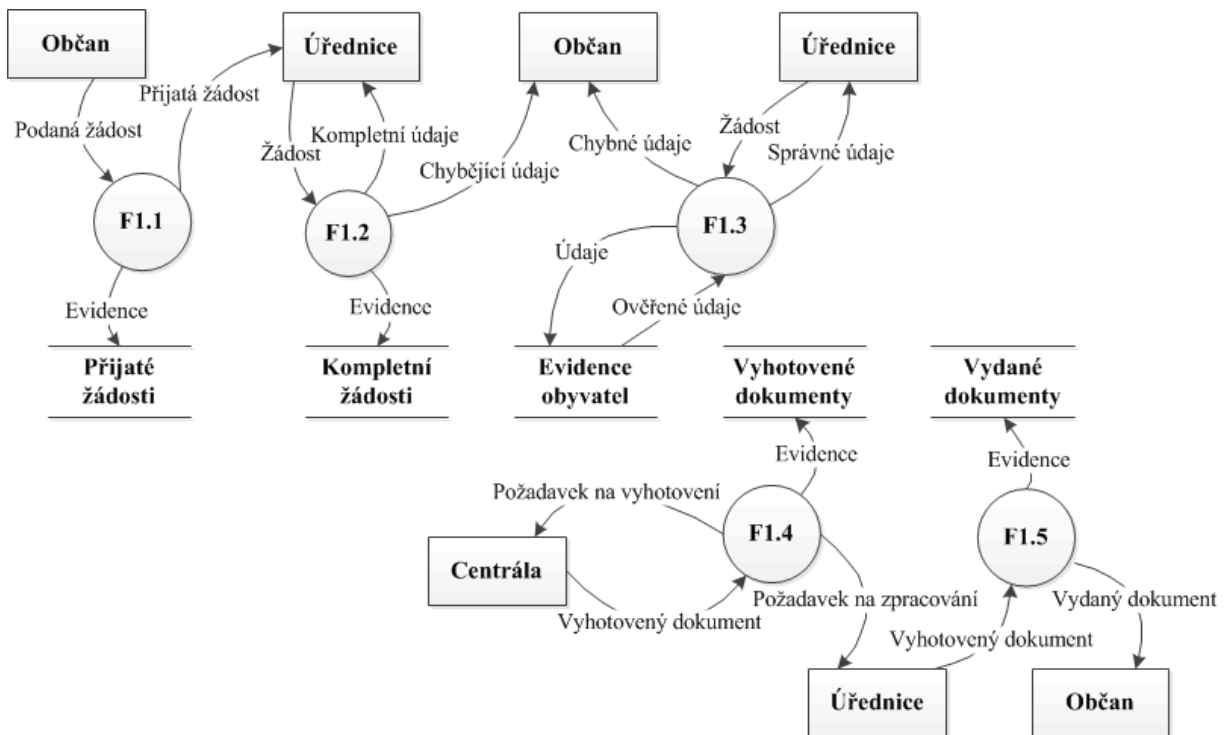
úroveň vychází z kontextového diagramu ukázaného na obrázku 9. Jelikož je postup zpracování totožný i v ostatních funkcích, jsou na následujícím obrázku popsány pouze funkce příjem a výdej občanského průkazu.

Občanem je zažádáno o nový občanský průkaz. K vyřízení žádosti je nutné mít vyplněný formulář. Takto vyplněná žádost je předána úřednici, kterou je občanovi vydáno potvrzení a datum, kdy je nutné se dostavit pro vyhotovený občanský průkaz. Pracovníci úřadu je provedena kontrola kompletnosti a správnosti údajů. Pokud jedna z těchto kontrol není správná, je žádost vrácena občanovi. Kontrola správnosti je provedena v modulu evidence obyvatel. Občan je zkontrolován v evidenci obyvatel. Dále je žádost zpracována. Žádost na vyhotovení dokumentu je poslána do centrály, kterou je vyhotovený dokument poslán zpět úřadu. Do evidence obyvatel je zaneseno nové číslo občanského průkazu. Poté co je žádost zpracována, občanovi je předán vyhotovený dokument, v tomto případě se jedná o občanský průkaz. Před vydáním občanského průkazu je občanem tento průkaz zkontrolován a následně je podepsáno převzetí o vydání tohoto dokumentu.

Diagram datových toků pro vyřizování žádosti, konkrétně se jedná o občanský průkaz, je ukázán na následujícím obrázku.

Použité symboly:

- terminátor 
- systém 
- tok informací 
- datové úložiště 



Obrázek 9: DFD příjmu a výdeje žádosti o občanský průkaz

Zdroj: vlastní zpracování

5.2.4 IDEF0

Metoda IDEF0 je hierarchicky uspořádaná sada diagramů s přesně vytvořeným systémem vzájemných odkazů popisujícími funkce. V této metodě jsou zachyceny vstup, výstup, funkce, řízení a mechanismus. Funkce je popis činnosti transformující vstup na požadovaný výstup. Vstupem jsou data nebo objekty, které budou funkcí převedeny na výstup. Výstupem jsou chápána data nebo objekty produkované funkcí. Řízení je dáno pravidly potřebnými k vytvoření požadovaného výstupu. Poslední složkou je mechanismus, kterým jsou definovány prostředky nutné k realizaci funkce.

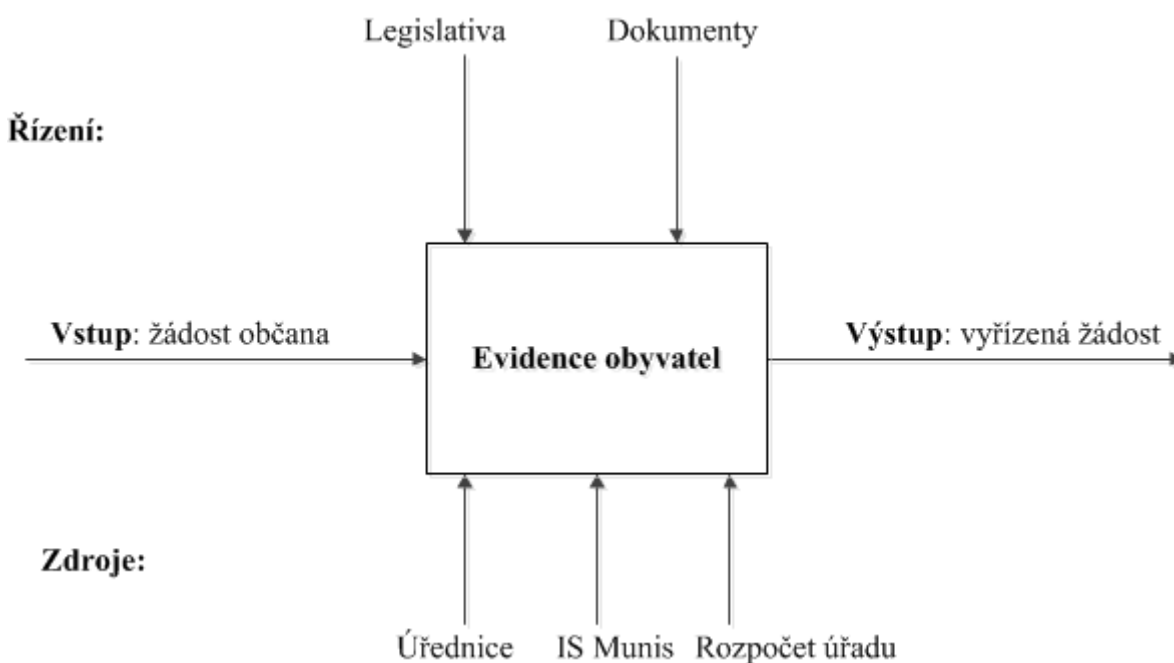
Funkce je zde zastoupena evidencí obyvatel. Do této evidence jsou zapisovány změny nebo vyřizovány činnosti, které byly požadovány od občana. Vstupem je samotná podaná žádost od občana. Vyřízení této žádosti je podmínováno dodáním dokumentů nebo je omezena legislativou. Toto je v tomto modelu řízení.

Do řízení jsou zařazeny následující dokumenty nebo omezení:

- žádost občana,
- hlášení o přestěhování,
- rodný list,

- oddací list,
- úmrtní list
- legislativa a jiné předpisy.

K tomu, aby byla žádost zpracována a vyřízena, je zapotřebí mít k dispozici i zdroje. V tomto případě zdroji je rozuměno prostředky, čili mechanismus, kterými je dosaženo zpracování žádosti. Prvním zdrojem je úřednice, kterou je žádost přijata, zpracována a vydána. Dalším zdrojem je IS Munis, ve kterém úřednice podanou žádost zpracovává. Toto zpracování probíhá jako změna údajů, ověření údajů, tisk údajů z evidence obyvatel. Jako poslední zdroj je uveden rozpočet úřadu.



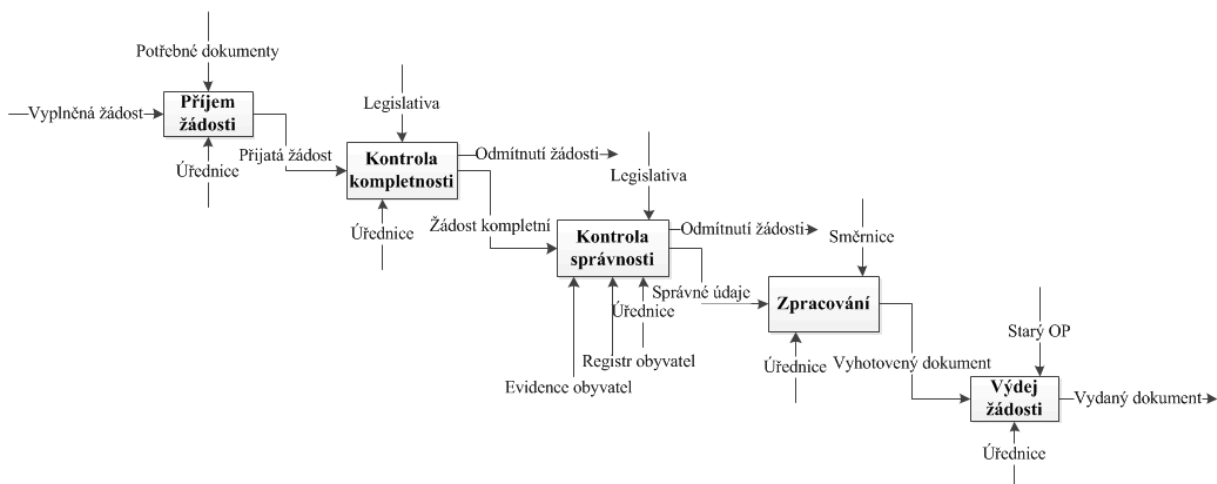
Obrázek 10: IDEF0 evidence obyvatel

Zdroj: vlastní zpracování

IDEF0 evidence obyvatel je brán jako kontextový diagram metody IDEF. K diagramu je možné vytvořit dekompozici funkce na podfunkce. Tento proces je ukázán na obrázku 11. Při sestavování diagramů jsou dodržovány zásady jejich řazení ve směru diagonály. Diagram by neměl mít méně než tři a více než šest funkcí. Platí zde důležitá vlastnost, kdy výstup dané funkce může být vstupem, řízením nebo mechanismem jiných funkcí. Takto jsou definovány vzájemné závislosti mezi funkcemi.

Evidence obyvatel je tvořena pěti dalšími funkcemi. Funkce Příjem žádosti má na svém vstupu objekt Vyplněná žádost, která byla již definována v kontextovém diagramu výše.

Regulátorem (řízením) jsou potřebné dokumenty, které jsou poskytnuty úřednicí. Výstupem z této funkce je přijatá žádost. Tento výstup je vstupem do další funkce, kterou je Kontrola kompletnosti. Regulátorem je zde legislativa. Tato žádost je kontrolována úřednicí, kterou je přezkoumána kompletnost údajů. Poté je posouzeno, jestli je žádost vrácena z důvodu chybějícího povinného údaje, nebo jestli je žádost poslána dále. Výstupem z této funkce je buď žádost odmítnuta, nebo žádost kompletní, která slouží jako vstup do další funkce. Další funkcí je Kontrola správnosti. Vstupem do této funkce je kompletní žádost. Regulátorem je legislativa. Mechanismy, které jsou potřebné k této funkci, jsou evidence obyvatel, registr obyvatel a úřednice. V evidenci obyvatel je provedena kontrola občana, jestli je tento občan součástí evidence. To samé platí i u registru obyvatel, ve kterém je nalezen občan, který podává žádost o změnu trvalého pobytu. Tato kontrola je provedena úředníkem. Úředníkem jsou kontrolovány také údaje, které jsou vyplněny občanem, jestli souhlasí s evidencí. Na úředníkovi je posouzení žádosti. Výstupy z této funkce jsou dva. Jedním je odmítnutí žádosti a druhým, že na žádosti jsou vyplněny správně údaje. Další funkcí je Zpracování. Vstupem do této funkce je správné údaje na žádosti. Regulátorem funkce jsou směrnice úřadu. Zpracování je provedeno úřednicí. Po zpracování požadavku je vyhotoven dokument. Poslední funkcí je tudíž Výdej žádosti. Vstupem je stále vyhotovený dokument. Funkci vykonává úředník. Výstupem této funkce je vydaný dokument.



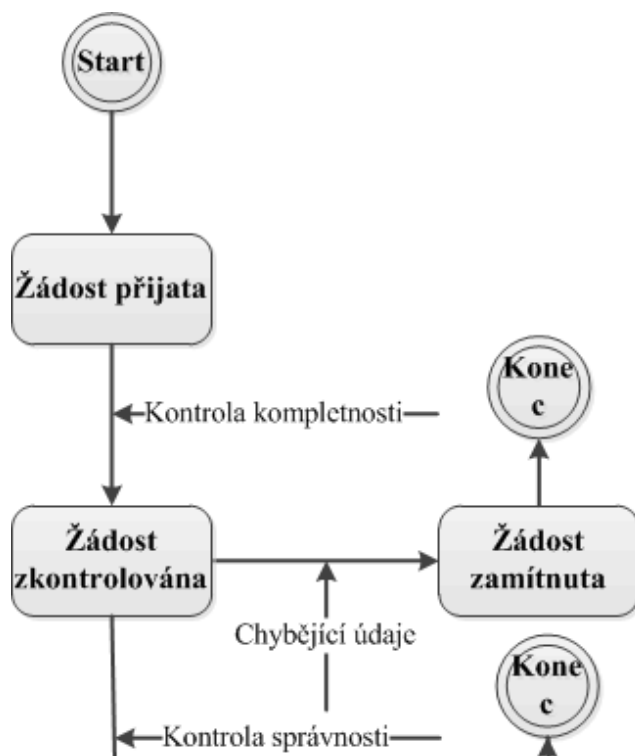
Obrázek 11: Dekompozice IDEF0 vyřizování žádosti

Zdroj: vlastní zpracování

5.2.5 Stavový diagram

Tímto diagramem je zachyceno, v jakých stavech se systém nebo jeho části nacházejí. Uživatelem není sdělen algoritmus, ale jsou jím známy stavy. Název stavu ho musí výstižně definovat. Stavy jsou definovány jako statické, přechody jsou realizovány jako následek události, což je zastoupeno splněním podmínky. Je nutné zapsat podmínku přechodu.


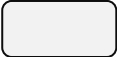

Prvním stavem je podaná žádost. Tento stav je vyvolán občanem a to tak, že je podána vyplněná žádost a jsou přiloženy potřebné dokumenty. Poté následuje kontrola kompletnosti. Kontrola je provedena úředníkem. Úředníkem je zkontrolováno, jestli jsou vyplněné všechny povinné údaje. Pokud je zjištěno, že povinný údaj není vyplněn, je definován stav Žádost zamítnuta a je vrácena občanovi. Diagram zde končí. V opačném případě následuje stav Žádost zkontrolována. Dále je pokračováno kontrolou správnosti. Kontrola je založena na vyhledání občana v evidenci obyvatel. V případě nesprávně vyplněného údaje je opět definován stav Žádost zamítnuta. Jinak následuje stav Žádost zkontrolována. Po kontrole je žádost zpracována. Je zpracována úředníkem za pomoci evidence obyvatel. Po tomto přechodu je definován stav Žádost zpracována. Po zpracování a zanesení všech údajů do evidence obyvatel je definován poslední stav. Tímto stavem je Výdej dokumentu občanovi. Občanem je dokument převzat, je zkontrolována správnost údajů občanem a poté podepsáno vydání dokumentu. Celý diagram je uveden v příloze C.



Obrázek 12: Ukázka části stavového diagramu vyřizování žádosti

Zdroj: vlastní zpracování

Použité symboly:

- začátek, konec 
- stav 
- přechod ze stavu do stavu 




5.2.6 EPC

Metodou EPC jsou funkce realizovány pomocí aktivit. Diagramem je popsána posloupnost aktivit, případně, které z aktivit budou realizovány souběžně. Tato metoda je založena na řetězení událostí a aktivit do posloupností realizujících požadovaný cíl. Událostí je definována vstupní podmínka uskutečnění aktivity. Ukončení aktivity je definováno další událostí – výstupní podmínkou, na kterou jsou navázány další aktivity. Každá aktivita je vymezena dvěma událostmi, tím je jednoznačně definován začátek a konec.

Událost Podaná žádost je vstupní podmínkou aktivity Posouzení kompletnosti žádosti a zároveň je to spouštěcí událostí celého procesu. Pokud je žádost správně vyplněná, kompletní, je generována další událost Správně vyplněná žádost. V opačném případě je výstupní událostí Chybně vyplněná žádost. Zde byla použita logická spojka XOR. Touto spojkou je posloupnost rozvětvena na dvě části. Při použití této spojky je nutné rozhodnutí, kterou větví bude pokračováno. Je možné pokračovat pouze jednou větví. V případě správně vyplněné žádosti je provedena aktivita Posouzení správnosti žádosti. Touto aktivitou je vyvoláno větvení s použitím logické spojky XOR. Pokud nejsou údaje o občanovi správné, je výstupní událostí Chybné údaje. V opačném případě je pokračováno událostí Správné údaje. Dále je následováno aktivitou zpracování žádosti. Tato aktivita je ukončena událostí Vyhotovená žádost. Touto událostí je vyvolána aktivita Výdej dokumentů. Výstupní podmínkou této aktivity je událost Převzetí občanem.

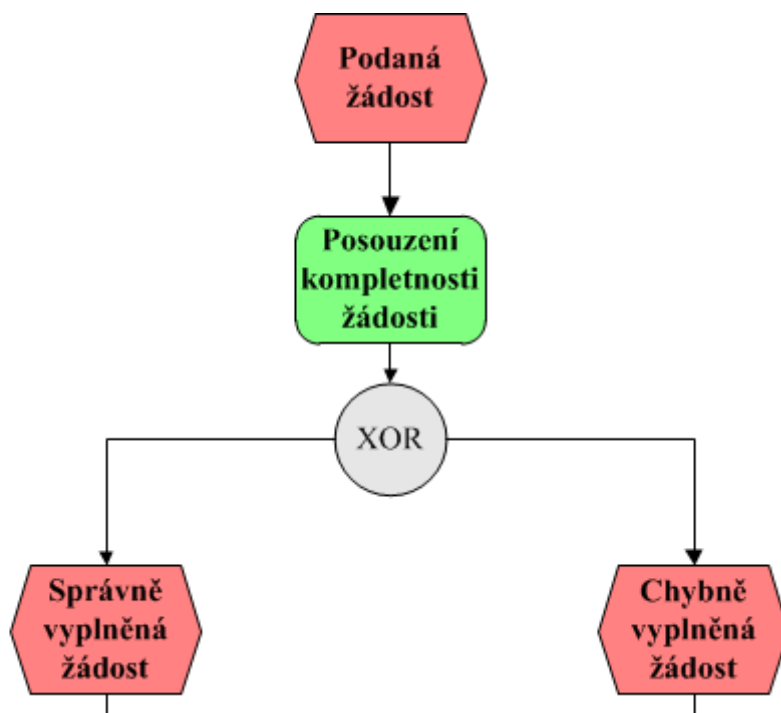
Použité symboly jsou uvedeny v následující tabulce.

Tabulka 3: Symboly EPC diagramu

| Popis | Symbol |
|----------------|---|
| Událost |  |
| Aktivita |  |
| Logická spojka |  |

Zdroj: vlastní zpracování

Na obrázku 12 je prezentována ukázka části kostry diagramu EPC bez přiřazení dokumentů. Celý diagram je uveden v příloze C.






Obrázek 13: Ukázka části EPC digramu vyřizování žádosti



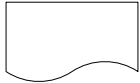
Zdroj: vlastní zpracování

EPC je pouze zjednodušený diagram, který neobsahuje vazbu na dokumenty. Jedná se pouze o přehled událostí a aktivit. Na druhé straně eEPC je rozšířený diagram, který integruje statický pohled. Jsou v něm přidána data, informace, dokumenty, SW, lidské zdroje. Na následujícím obrázku je prezentována pouze ukázka diagramu eEPC, celý diagram je v příloze C. Dokumentem je žádost a vydaný dokument. Organizační jednotkou je úřednice a informace je brána z evidence obyvatel.

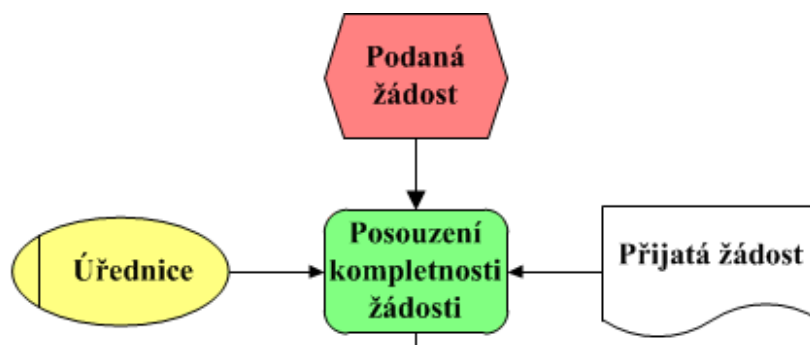
Použité symboly jsou ukázány v následující tabulce.

Tabulka 4: Symboly eEPC diagramu

| Popis | Symbol |
|----------------|---|
| Událost |  |
| Aktivita |  |
| Logická spojka |  |

| | |
|----------------------|---|
| Organizační jednotka |  |
| Informace |  |
| Dokument |  |

Zdroj: vlastní zpracování



Obrázek 14: Ukázka diagramu eEPC





Zdroj: vlastní zpracování

5.2.7 Use Case

Funkční specifikace je řešena prostřednictvím diagramů případu užití, jinak řečeno Use Case. Diagramy případů užití explicitně identifikují procesy a jejich okolí. Use Case je definován jako posloupnost akcí, které jsou realizovány úřadem v interakci se specifickými aktéry s cílem vytvořit požadované hodnoty. Kromě případu užití je definován i další element a tím je aktér. Aktérem je míněna osoba nebo něco, co stojí mimo proces specifikovaný daným případem užití. Účelem tohoto diagramu je dokumentování interakce mezi službami, které jsou poskytovány úřadem a těmi, kterými jsou tyto služby požadovány. Takto vytvořeným modelem je identifikováno, co je účelem úřadu a jaké funkce jsou nabízeny svému okolí.

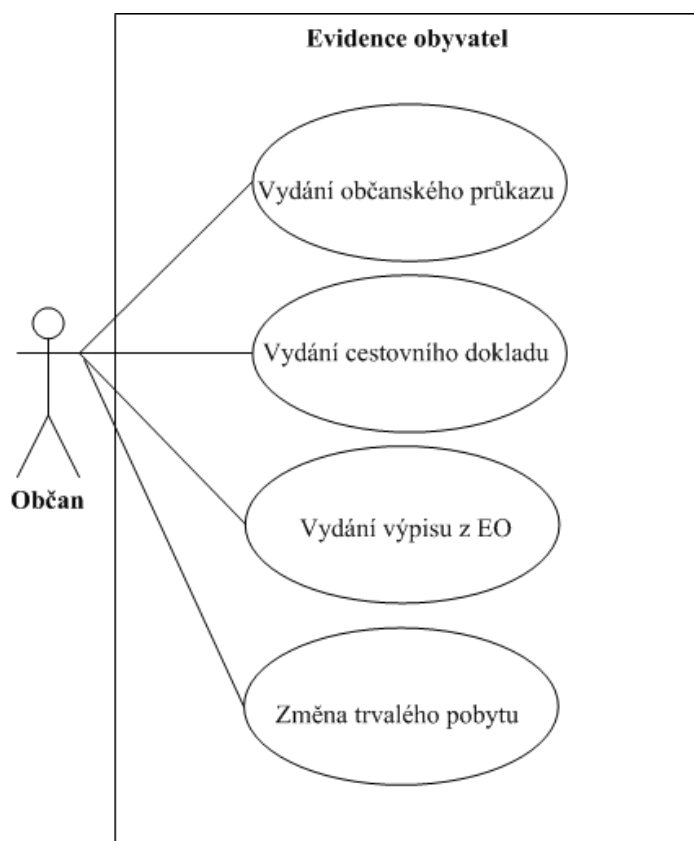
V tomto modelu jsou identifikováni dva aktéři. Jedním je občan, který požaduje od úřadu vyřízení své žádosti a dostání vyžadovaného dokumentu. Druhým aktérem je úřednice, která je zaměstnána na úřadu, jsou jí přijímány, zpracovány a vydávány žádosti od občana. Zpracování žádosti je založeno na spolupráci s informačním systémem úřadu. V tomto systému je nutné zkontrolovat, upravovat, archivovat nebo vytvořit nové údaje o občanu.

Použité symboly:

- aktér 
- případ užití 
- hranice systému 
- vztah 

Následující obrázek je řešen z pohledu občana. Jsou zde ukázány případy užití, které jsou občanem vyžadovány po úřadu. Občanem jsou řešeny různé životní situace, které by tento diagram měl zachytit. Rovněž se jedná o elementární funkce hierarchického rozkladu.

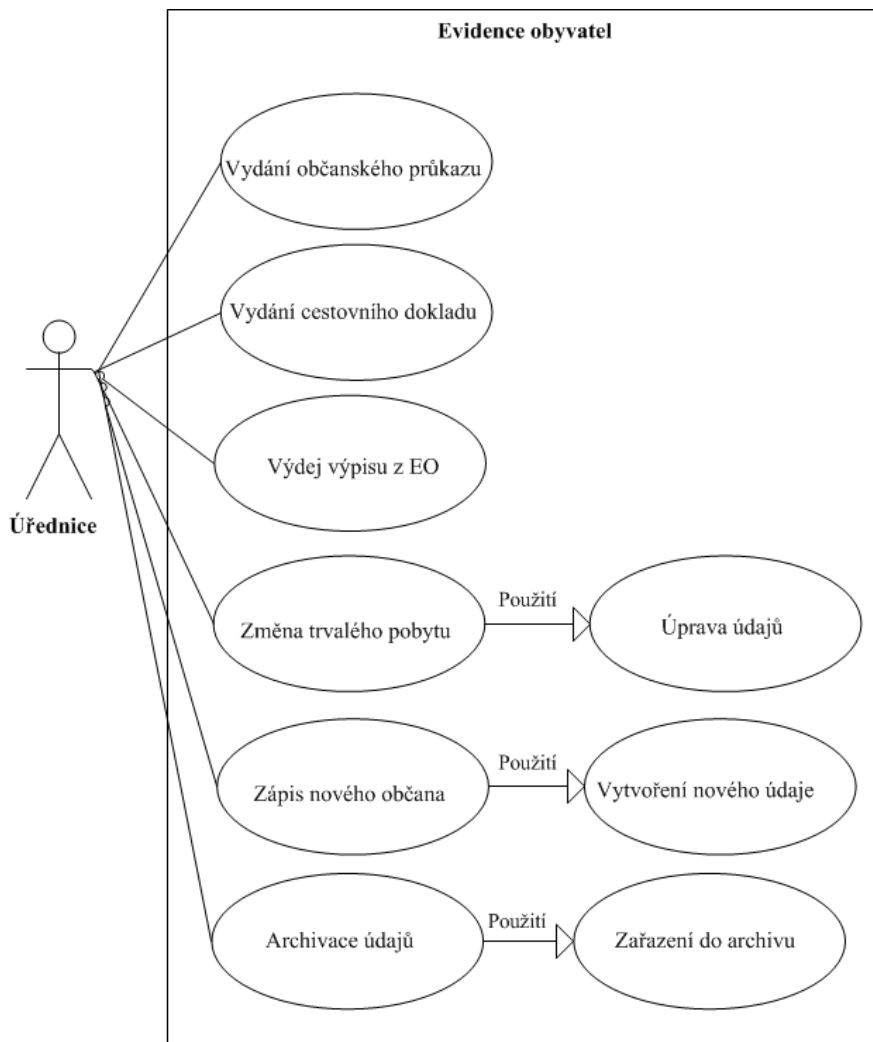
Aktér, v tomto případě občan, vstupuje do jednotlivých procesů, které jsou zastoupeny jednotlivými případy užití.



Obrázek 15: Případu užití z pohledu občana

Zdroj: vlastní zpracování

Další diagram případu užití je zaměřen na pohled úřednice, zaměstnané na úřadě, která přímo spolupracuje s informačním systémem. V tomto diagramu je využit vztah použití (uses). Tento vztah je deklarován nutností zahrnout do vykonávání procesu proces jiný, vložený. Vložený proces je definován, jako spolupráce s IS Munis. Jedná se o změnu, kontrolu, zařazení do archivu nebo vytvoření nových údajů o občanech.



Obrázek 16: Příklad užití z pohledu úřednice

Zdroj: vlastní zpracování

Ke každému případu užití je přiřazen i scénář. Jsou rozlišeny dva druhy scénářů. Jedním je hlavní, kterým je zachycena komunikace aktéra a systému od počáteční podmínky až k výstupu. Druhý druh scénáře je alternativní. Alternativní scénář je vyhotoven v případě, že dojde k nesplnění nějaké podmínky. Následující tabulky uvedou případ hlavního scénáře vyhotoveného pro případ užití Zapsání nového občana. K hlavnímu scénáři budou napsány i alternativní scénáře.

Pro vydání občanského průkazu je občan zkontrolován v evidenci obyvatel. Kontrola je provedena z důvodu správnosti údajů. Úřednicí je zapsáno nové číslo občanského průkazu

v modulu evidence obyvatel. V modulu evidence obyvatel je třeba občana vyhledat a otevřít si formulář s jeho uloženými údaji. Úřednice provede kontrolu údajů na žádosti s údaji, které jsou uloženy v modulu evidence obyvatel. Po přijetí vyhotoveného občanského průkazu je jeho číslo zapsáno do evidence obyvatel.

Tabulka 5: Hlavní scénář případu užití vydání občanského průkazu

| | | | |
|----------------------|---|-------------|---|
| Use Case | Vydání občanského průkazu | | |
| Aktér | Úřednice | | |
| Úroveň | Uživatelská | | |
| Popis | Zkontrolovat občana v evidenci obyvatel a zapsat nové číslo OP do této evidence | | |
| Vstupy | K dispozici je žádost občana a jeho starý občanský průkaz | | |
| Výstup | Vydání občanského průkazu | | |
| Spouštěč | Přijátá žádost na vydání občanského průkazu | | |
| Hlavní scénář | 1 | Role | Akce |
| | 1.1 | A | zadá přihlašovací údaje do IS Munis |
| | 1.2 | S | přihlásí aktéra |
| | 1.3 | A | poklepe myší na modul evidence obyvatel |
| | 1.4 | S | otevře modul evidence obyvatel |
| | 1.5 | A | poklepe na vyhledání občana v evidenci |
| | 1.6 | S | otevře seznam občanů |
| | 1.7 | A | vyplní příjmení občana |
| | 1.8 | A | klikne myší na tlačítko vyhledat |
| | 1.9 | S | vyhledá občana |
| | 1.10 | A | dvakrát klikne na občana |
| | 1.11 | S | otevře formulář s údaji |
| | 1.12 | A | zkontroluje údaje s žádostí |
| | 1.13 | A | vyplní nové číslo OP |
| | 1.14 | A | klikne na tlačítko Uložit |
| | 1.15 | S | uloží data do evidence obyvatel |
| | 1.16 | A | zavře modul evidence obyvatel a odhlásí se ze systému |

Zdroj: vlastní zpracování

Alternativní scénář je možné zhotovit pro více případů z hlavního scénáře. Scénář bude vyhotoven pro body 1.1, 1.5 a 1.14. U bodu 1.5 nemusí systém nalézt občana v evidenci obyvatel. Tato situace může nastat v případě, že občan není přihlášen v dané obci nebo

úřednice špatně napsala příjmení. U bodu 1.14 může nastat situace, že úřednice napíše špatný číselný kód u občanského průkazu nebo bude špatná délka kódu.

Tabulka 6: Alternativní scénář k případu užití vydání OP

| | | |
|----------------------------|----------|--|
| Alternativní scénář | 1 | Alternativa k 1.1. Aktér vyplní špatně přihlašovací údaje, v tomto případě systém vrátí aktéra na stánku s přihlašovacími údaji a napíše oznámení o chybně zadaných údajích. |
| Alternativní scénář | 2 | Alternativa k 1.5. Pokud aktér nenalezne občana v evidenci obyvatel, musí tohoto občana zadat jako nového. |
| Alternativní scénář | 3 | Alternativa k 1.14. Pokud úřednice zadá špatnou délku kódu OP, systém tento formulář neuloží. |
| Alternativní scénář | 4 | Aktér může vyplňování formuláře ukončit volbou Zrušit. |

Zdroj: vlastní zpracování

5.3 Funkční požadavky na informační systém

Funkční požadavky na informační systém jsou posledním krokem ve specifikaci. Jedná se o písemné definování a shrnutí požadavků.

V následujícím textu jsou definovány funkční požadavky jedné větve hierarchického rozkladu. Jedná se o specifikaci požadavků na vyřizování žádostí. Níže jsou uvedeny jednotlivé funkce, které jsou potřebné k vyřizování žádostí. Tyto funkce jsou prezentovány na obrázku 9, na kterém je ukázán rozšířený IDEF0. V tomto diagramu jsou tyto funkce popsány číslem procesu, pracovní rolí, cílem uživatele, potřebnými daty, datovým úložištěm, stavem prvku, přístupovým právem a odkazem na diagram, ve kterém je funkce popsána.

1) Příjem žádosti

Číslo procesu: F1.1.

Pracovní role: úřednice

Cíl uživatele: přijmout žádost od občana

Výstup: přijatá žádost od občana připravená ke kontrole

Potřebná data: vyplněná žádost, potřebné dokumenty k vyřízení žádosti

Datové úložiště: přijaté žádosti

Stav prvku: žádost přijata

Přístupová práva: úřednice

Odkaz: stavový diagram, IDEF0, DFD

2) Kontrola kompletnosti

Číslo procesu: F1.2.

Pracovní role: úřednice

Cíl uživatele: kontrola vyplnění povinných údajů

Výstup: zkontrolovaná žádost - kompletně vyplněné povinné údaje; vrácená žádost

Potřebná data: vyplněná žádost

Datové úložiště: Kompletní žádosti

Stav prvku: žádost zkontrolována

Přístupová práva: úřednice

Odkaz: stavový diagram, IDEF0, EPC

3) Kontrola správnosti

Číslo procesu: F1.3.

Pracovní role: úřednice

Cíl uživatele: ověření občana v evidenci

Výstup: zkontrolovaná žádost - správné údaje na žádosti; vrácená žádost

Potřebná data: vyplněná žádost, údaje o občanova

Datové úložiště: evidence obyvatel

Stav prvku: žádost zkontrolována

Přístupová práva: úřednice

Odkaz: stavový diagram, IDEF0, EPC

4) Zpracování

Číslo procesu: F1.4.

Pracovní role: úřednice

Cíl uživatele: zpracování požadavků občana

Výstup: vyhotovený dokument

Potřebná data: vyplněná žádost, údaje o občanovi

Datové úložiště: vyhotovené dokumenty

Stav prvku: žádost zpracována

Přístupová práva: úřednice

Odkaz: stavový diagram, IDEF0, EPC

5) Výdej žádosti

Číslo procesu: F1.5.

Pracovní role: úřednice

Cíl uživatele: vydat občanovi dokumenty, které požadoval

Výstup: vydaný dokument

Potřebná data: potvrzení přijaté žádosti, vyhotovené dokumenty

Datové úložiště: vydané dokumenty

Stav prvku: vydán dokument

Přístupová práva: úřednice

Odkaz: stavový diagram, IDEF0, EPC

5.4 Shrnutí případové studie

Napřed je nezbytné při definování požadavků vytvořit celkový popis úřadu. Jedná se o slovní charakteristiku jak z pohledu okolí, tak i pohledu vnitřního uspořádání. Vnitřní uspořádání je definováno svojí organizační strukturou, vnitřní komunikací, čili předáváním informací mezi jednotlivými odbory.

Popis systému je vyhotoven na základě sběru dat od uživatelů. Jsou zde definovány účastníci, okolí, služby poskytované úřadem, technické prostředky, informace, zákazníci, procesy. Uživatel si musí uvědomit, kdo spadá do jeho okolí, co vše od IS očekává, kdo bude jeho uživatel.

V předcházejícím textu se jednalo o slovní popis úřadu a systému. Následně jsou využity diagramy. Prvním diagramem je hierarchie funkcí. Tímto rozkladem je ukázána nadřazenost a podřazenost funkcí. V poslední úrovni jsou definovány elementární funkce. Každá funkce by měla mít svůj slovní popis, co tato funkce dělá. Na nejvyšší úrovni v této studii je IS evidence obyvatel.

Po hierarchickém rozkladu byl použit další nástroj a tím byl DFD. Nejdříve byla vypracována 0. úroveň a tou je kontextový diagram. Kontextový diagram je brán z celkového pohledu. Jedná se o vztah s okolím, jeho vstupy a výstupy z dané funkce. DFD je hierarchicky rozložen na další úrovně. V první úrovni byl řešen příjem a výdej žádosti o občanský průkaz. Ostatní diagramy DFD vyřizování žádostí na první úrovni by byly vypracovány stejným způsobem.

Diagram IDEF0 byl vypracován taktéž v nulté úrovni. Jedná se tedy o kontextový diagram, který je možné dále hierarchicky rozkládat. Tento rozklad má ovšem svá pravidla. Do funkce vchází vstup, který je transformován na procesy na vyžadovaný výstup. K změně vstupu na výstup slouží mechanismy a řídicí pravidla. Ke kontextovému diagramu byla vytvořena i dekompozice IDEF0, která se skládá z pěti funkcí.

Stavovým diagramem je studentům ukázáno, jaké stavy daný objekt má. Jsou zde naznačeny i koncové stavy tohoto diagramu. Jedná se zejména o chybně napsané nebo nesprávné údaje. Následující stav je vyvolán podmínkou přechodu.

Předposledním nástrojem ve výuce je diagram EPC. Tento diagram představuje posloupnost jednotlivých aktivit, které jsou ohraničeny událostmi. V práci je ukázána pouze kostra diagramu, která je bez příslušných dokumentů, legislativy a zdrojů. Tyto rozšiřující prvky jsou uvedeny v diagramech eEPC a IDEF0.

Případem užití byla definována funkční specifikace. Jsou zde vyobrazeny elementární funkce hierarchie rozkladu. Jsou to funkce, které jsou vyžadovány od informačního systému. Diagram je vypracován z pohledu dvou různých aktérů, kteří říkají, co chtějí od systému. Jedním je pohled občana a druhým pohled úředníka.

Po vyhotovení předcházejících diagramů a slovního popisu následují funkční požadavky na informační systém. Tyto požadavky vycházejí z předešlých diagramů a slovního popisu. Je to souhrn těchto informací, které jsou důležité. Jsou tam vypsaní odpovědní uživatelé, procesy kterých se funkce týká. Dále jaké je datové úložiště, stav prvku a další informace.

Po vypracování kompletního seznamu funkčních požadavků, příjem, kontrola, zpracování a výdej, bude následovat další fáze vývoje informačního systému.

ZÁVĚR

Řízení požadavků na informační systém je velmi složitý proces. Počínaje sběrem dat od budoucích uživatelů, přes specifikaci, kontrolu ke konečnému definování funkčních požadavků. Je nutné získat od uživatelů, co nejvíce informací, které chtějí, aby systém uměl. Jelikož tato fáze je zaměřena na to „CO“ má systém dělat.

V práci byla vytvořena ukázková případová studie pro studenty předmětu Informační systémy regionů. Tato studie jim poslouží jako příklad definování požadavků. Vybraným systémem pro specifikaci byl určen IS Munis. Systém je součástí výuky předmětu PISR a KISR. Jelikož v IS Munis je mnoho modulů, byl vybrán modul evidence obyvatel. S tímto modulem se lidé setkávají nejčastěji, aniž by o tom věděli. Nástroje uvedené v případové studii jsou využívány při výuce. Studenti jsou s nimi seznámeni.

Evidence obyvatel je agendou samosprávy, se kterou je občan nejvíce v kontaktu. Existuje mnoho životních situací, které jsou spjaty s touto agendou, ať už se jedná o žádosti o osobní doklady, změnu trvalého pobytu, zapsání dítěte, archivaci údajů nebo výpis z evidence. Evidencí obyvatel jsou poskytnuty údaje o občanovi i jiné oprávněné agendě.

Pro rozsáhlou problematiku rozkladu funkcí byly všechny diagramy vyhotoveny pro větev vyřizování žádostí. Tento postup je popsán diagramy z různých pohledů podle notace diagramu. Bylo vytvořeno 12 diagramů popisujících problém požadavků na informační systém. V textu práce jsou jen ukázky některých využívaných diagramů. Celá případová studie je v příloze.

Jelikož se jednalo o ukázkovou případovou studii, bylo vycházeno pouze z informací na stránkách magistrátu města Hradec Králové. Pro práci nebylo nutné komunikovat s pracovníky na úřadu. Diagramy byly vyhotoveny pro studenty a jejich lepší pochopení symboliky a práce s těmito diagramy. V práci byla uvedena symbolika, která je využívána při výuce předmětu Informační systémy regionů.

POUŽITÁ LITERATURA

- [1] Arlow Jim, Neustadt Ila. UML a unifikovaný proces vývoje aplikací. Brno: Computer Press, 2003. ISBN 80-7226-947-X.
- [2] Aurum Aybüke, Wohlin Claes. Engineering and Managing Software Requirements. Heidelberg: Springer, 2005. ISBN 978-3-540-25043-2.
- [3] Balík Stanislav. Komunální politika: Obce, aktéři a cíle místní politiky. Praha: Grada Publishing, 2009. ISBN 978-80-247-2908-4.
- [4] Bruckner Tomáš, Voříšek Jiří, Buchalceková Alena. Tvorba informačních systémů. Praha: Grada Publishing, 2012. ISBN 978-80-247-4153-6.
- [5] Buchalceková Alena. Metodiky budování informačních systémů. Praha: Oeconomica, 2009. ISBN 978-80-245-1540-3.
- [6] Buchalceková Alena. Metodiky vývoje a údržby informačních systémů. Praha: Grada Publishing, 2005. ISBN 80-247-1075-7.
- [7] Doucek Petr. Řízení projektů informačních systémů. Praha: Professional Publishing, 2004. ISBN 80-86419-71-1.
- [8] Gála Libor, Pour Jan, Toman Prokop. Podniková informatika. Praha: Grada Publishing, 2006. ISBN 80-247-1278-4.
- [9] Gála Libor, Pour Jan, Šedivá Zuzana. Podniková informatika. 2. Praha: Grada, 2009. ISBN 978-80-247-2615-1.
- [10] Grady Jeffrey O. System Requirements Analysis. Burlington, USA: Academic Press, 2010. ISBN 978-0-12-088514-5.
- [11] IS Munis [online]. 2014 [cit. 2014-4-1]. Munis informační systém pro města a obce. Dostupné z WWW: <www.munis.cz>.
- [12] Kajzar Dušan. Tvorba informačních systémů II: proces vývoje informačního systému. Opava: Slezská univerzita, 2005. ISBN 80-7248-288-2.
- [13] Kočí Roman. Obecní samospráva v České republice. Praha: Legas, 2012. ISBN 978-80-87576-28-1.
- [14] Kolektiv autorů. Úvod do regionálních věd a veřejné správy. Plzeň: Aleš Čeněk, 2008. ISBN 80-85963-69-8.

- [15] Koudelka Zdeněk. Právní předpisy samosprávy. Praha: Linde Praha, 2001. ISBN 80-7201-274-6.
- [16] Laplante Phillip A. Requirements Engineering for Software and Systems.: CRC Press, 2009. ISBN 978-1-4200646-7-4.
- [17] Leffingwell Dean, Widrig Don. Managing software requirements: A unified approach. Indianapolis: AddisonWesley, 2003. ISBN 978-0-321-12247-6.
- [18] Oddělení osobních dokladů a evidence obyvatel. Hradec Králové: oficiální stránky statutárního města [online]. 2013 [cit. 2014-04-27]. Dostupné z: <http://www.hradeckralove.org/urad/oddeleni-osobnich-dokladu-a-evidence-obyvatel>
- [19] Pour Jan. Informační systémy a technologie. Praha: Vysoká škola ekonomie a managementu, 2006. ISBN 80-86730-03-4.
- [20] Ricardo Catherine. Databases Illuminated. Sudbury, Canada: Cathleen Sether, 2011. ISBN 978-1-4496-0600-8.
- [21] Robertson Suzanne, Robertson James. C. Mastering the Requirements Process. Addison-Wesley Professional, 2006. ISBN 978-0-321-46797-3.
- [22] Řepa Václav. Analýza a návrh informačních systémů. Praha: Ekopress, 1999. ISBN 80-86119-13-0.
- [23] Shelly Gary B., Rosenblatt Harry J. Systems Analysis and Design. Boston: Cengage Learning, 2009. ISBN 978-0-324-59766-0.
- [24] Sklenák Vilém. Data, informace, znalosti a Internet. Praha: C. H. Beck, 2001. ISBN 80-7179-409-0.
- [25] Vondrák Ivo. Metody byznys modelování. Ostrava, 2004.
- [26] Vymětel Dominik. Informační systémy v podnicích - teorie a praxe projektování. Praha: Grada Publishing, 2009. ISBN 978-80-247-3046-2.
- [27] Wiegers Karl Eugene. Požadavky na software. Brno: Computer Press, 2008. ISBN 978-80-251-1877-1.
- [28] Zákon č. 1/1993 Sb., Ústava České republiky. In: Sbíрка zákonů. 1993.
- [29] Zákon č. 128/2000 Sb., o obcích. In: Sbíрка zákonů. 2000.
- [30] Zákon č. 129/2000 Sb., o krajích. In: Sbíрка zákonů. 2000.
- [31] Zákon č. 133/2000 Sb., o evidenci obyvatel a rodných číslech. In: Sbíрка zákonů. 2000.

SEZNAM PŘÍLOH

Příloha A: Modely životního cyklu projektu IS/ICT..... -71-

Příloha B: Metodiky vývoje informačního systému..... -77-

Příloha C: Diagramy případové studie..... -81-

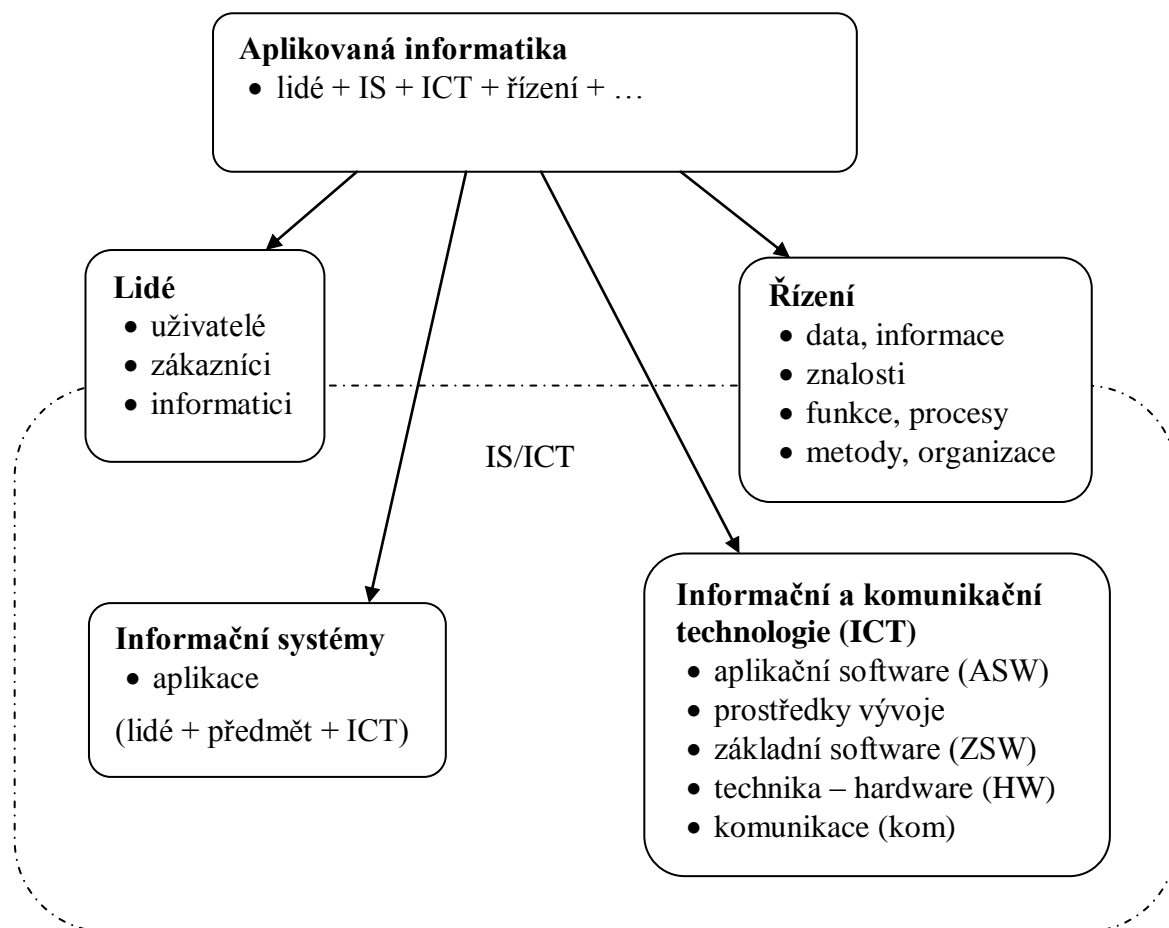
Příloha D: Případová studie

Případová studie D je elektronická a je přiložena na datovém nosiči.

Příloha A: Modely životního cyklu projektu IS/ICT

Za model životního cyklu je pokládán formalizovaný popis činností a aktivit, kde jsou definovány vazby mezi nimi a je dána časová posloupnost jejich plnění. Model životního cyklu je vymezen svým začátkem, předem nadefinovanými částmi a jejich hierarchickou strukturou a svým koncem.[7] Při řízení konkrétního projektu je třeba vybrat příslušné procesy z referenčního modelu procesů, zvolit určitý model životního cyklu a vzájemně je provázat. Model životního cyklu je významnou charakteristikou každé metodiky budování IS/ICT.[5]

Aplikovanou informatikou jsou chápány principy a pravidla práce s informacemi a charakteristiky s nimi spojených systémů a jejich prvků, které jsou významné pro její užití ve vymezené oblasti lidské činnosti. Následující obrázek je zaměřen na aplikovanou informatiku v ekonomice, na její uplatnění v řízení a provozu ekonomických subjektů.[8]



Obrázek 17: Aplikovaná informatika v ekonomice

Zdroj: převzato z [8]

Procesy v životním cyklu

V mezinárodní normě ISO/IEC 12207 Procesy v životním cyklu softwaru jsou definovány procesy, činnosti a úkoly, které je třeba provádět při dodávce, vývoji, provozu, údržbě a odstranění softwarového produktu a také procesy pro definování, řízení a zlepšování procesů životního cyklu softwaru.[5] Jinými slovy se jedná o návrh soustavy procesů a činností pokrývajících životní cyklus programového produktu včetně aplikačního programového vybavení od poptávky, stanovení koncepce jeho řešení, přes vlastní vývoj, kódování, testování, předání výsledného produktu k užívání, údržby až po vyřazení produktu z provozu.[7] Norma definuje 43 procesů v 7 procesních skupinách. V normě je uveden u každého procesu název, účel, výstup, činnosti a úkoly.[5]

Procesy životního cyklu produktu jsou rozděleny do těchto základních kategorií [7]:

- primární,
- podpůrné,
- organizační.

Primární procesy

Proces akvizice je výběr produktu nebo služby tak, aby byly splněny požadavky zadavatele. Začátkem procesu je zjištění požadavků zadavatele a koncem je výběr služby nebo produktu.[5] Akvizice je definování činností firmy, která pořizuje programový produkt. **Proces dodání** je vymezeno činnostmi dodavatele, který dodává programový produkt.[7] Účelem procesu dodání je dodat zadavateli produkt nebo službu tak, aby byly splněny dohodnuté požadavky.[5] **Proces vývoj** je definován činnostmi projektanta, který definuje a vyvíjí programový produkt.[7] Do tohoto procesu je zařazeno plánování projektu, definování požadavků a také hodnocení a řízení projektu.[5] **Proces provozu** je vymezen činnostmi firmy provozující programový produkt nebo zajišťující provoz jako službu formou outsourcingu. **Proces údržba** je definován činnostmi správce systému nebo subjektu zajišťujícího údržbu jako službu formou outsourcingu.[7]

Podpůrné procesy

Dokumentace je vymezena činnostmi pro záznam informací vyprodukovaných během životního cyklu.[7] Předmětem procesu **řízení konfigurací** je zajištění a udržování integrity všech produktů a jejich zpřístupnění všem zainteresovaným stranám. Výstupem tohoto procesu je definovaná strategie řízení konfigurací a jsou určeny prvky, které podléhají konfiguračnímu řízení.[5] Účelem procesu **zajištění jakosti** jsou činnosti pro objektivní

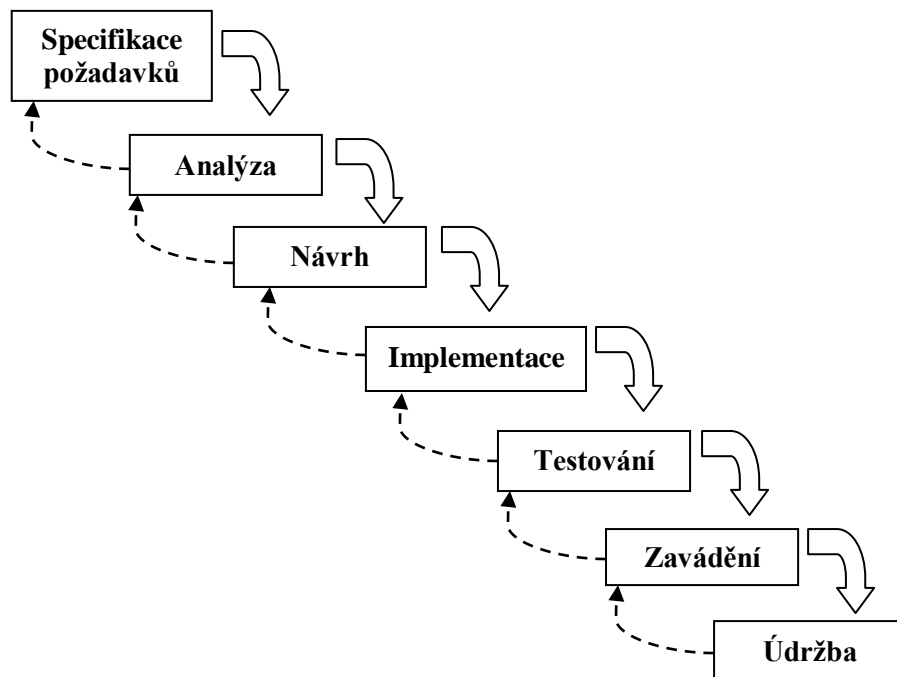
zjišťování shody programových produktů a procesů se specifickými požadavky (společná přezkoumání, prověrky, ověřování, validace). Součástí procesu **testování** jsou činnosti pro ověřování programových produktů na různých úrovních pro různé typy testů. Proces **validace** je definován činnostmi pro ověření shody finálního produktu se specifikovanými požadavky. Předmětem **společného přezkoumání** jsou činnosti pro hodnocení stavů a výsledků projektu. Proces **prověrka** je vymezen činnostmi pro hodnocení shody reality dosažené dodavatelem projektu s požadavky, plánem a kontraktem. **Řešení problémů** definuje proces pro analýzu a odstraňování problémů a neshod, které byly odhaleny během vývoje, provozování a údržby programového produktu.[7]

Organizační procesy

Součástí procesu řízení jsou základní činnosti při řízení projektu vývoje programového produktu během životního cyklu. Předmětem procesu tvorby infrastruktury je definování základních činností, které jsou vykonávány při vytváření, měření, řízení a zdokonalování procesu životního cyklu. Proces školení personálu definuje činnosti pro školení, tyto činnosti i upravuje.[7]

Vodopádový model

Vodopádový model představoval v době svého vzniku významný pokrok. Tímto modelem byl rozdělen celý proces vývoje do samostatných fází a byl umožněn systematický a opakovaný postup vývoje. Celý proces je zahájen fází specifikace požadavků, dále následuje fáze analýzy, návrhu, implementace, testování a zavedení. Poslední fází je provoz a údržba systému.



Obrázek 18: Vodopádový model

Zdroj: převzato z [5]

Předpoklady pro nasazení vodopádového modelu [7]:

- jeho jednotlivé etapy jsou vykonávány postupně po ukončení etapy předchozí, jejíž výsledky se při ukončení etapy verifikují a slouží jako vstupy etapy další,
- nejlepšího výsledku při řešení projektu IS/ICT se dosáhne při výše uvedeném pořadí etap,
- při žádném jiném pořadí vykonávání etap se nedosáhne výsledku lepšího.

Tento model přináší problém v případech, kdy je nutné provádět během vývoje změny. Nevýhodou vývoje tohoto modelu je, že zákazník je do vývoje zapojen pouze na začátku a konci procesu, a nemá tak kontrolu nad průběhem projektu.[5]

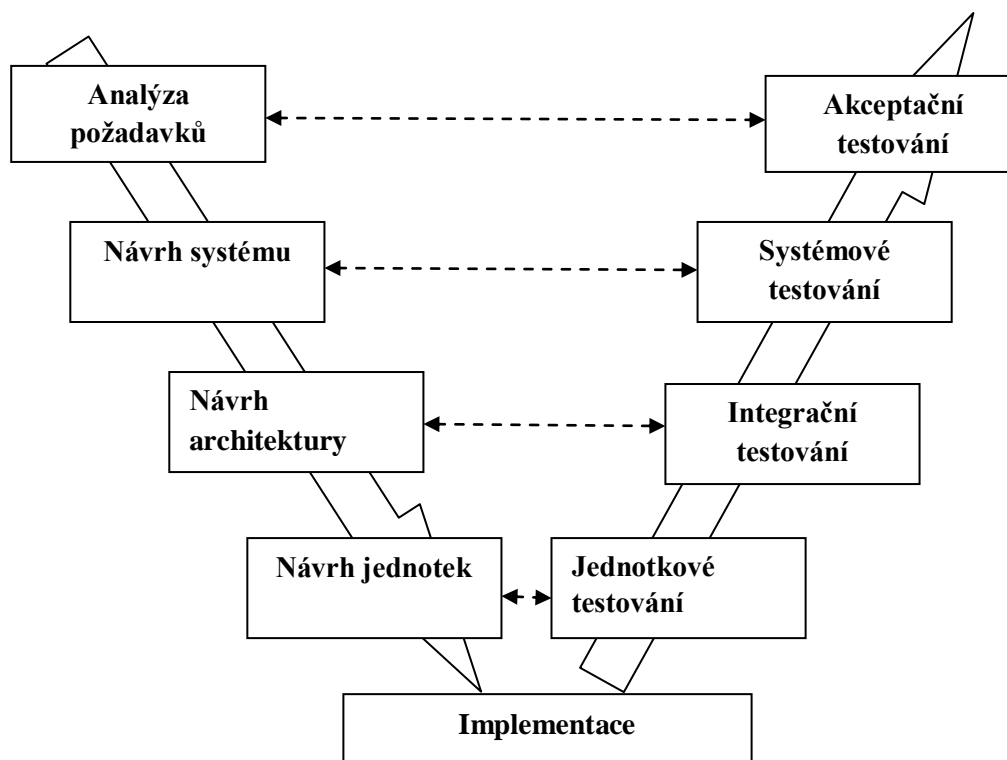
Největší problémy vývoje podle vodopádového modelu [5]:

- model předpokládá, že se všechny požadavky specifikují na začátku projektu, což v realitě v současném měnícím se prostředí není možné,
- iterace jsou možné jen v rámci jedné fáze, případně je možné se vrátit k předchozí fázi, tento způsob vede k tomu, že řešení problémů se odsouvá do pozdějších fází,

- tím, že je možné jen jednou projít všemi fázemi, není zpětná vazba od zákazníka, a tak často dochází k tomu, že po předání systému nastává dlouhá fáze údržby, kdy je systém upravován,
- hlavním problémem je pozdní integrace systému, která nastává až po naprogramování všech modulů, teprve tehdy jsou zjištěny problémy, jež často vedou ke změnám návrhu, přeprogramování a celkovému zpoždění projektu.

V-model

Jedná se modifikaci vodopádového modelu. Tímto modelem jsou vzájemně provázány fáze, v nichž se specifikuje a implementuje systém, s fázemi integrace a testování. V-model vznikl v době, kdy vzrostl význam testování a byly specifikovány různé techniky a druhy testů. Ve V-modelu jsou jednotlivým fázím návrhu, či analýzy, přiřazeny jednotlivé typy testů. Analýza, návrh a implementace jsou zakresleny v levé části shora dolů. Testování probíhá v pravé části zdola nahoru. Nejprve jsou testovány jednotky, potom následuje integrační testování a systémové testování. Jako poslední probíhá akceptační testování, které ověřuje, zda byly splněny definované požadavky na systém. [5]



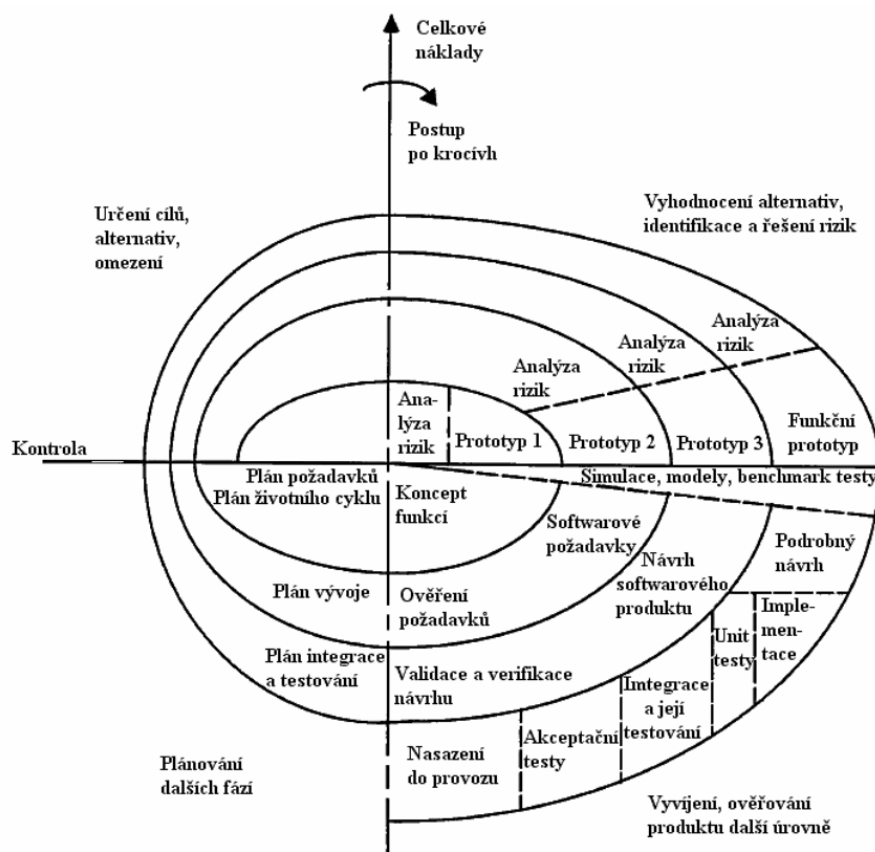
Obrázek 19: V-model

Zdroj: převzato z [5]

Spirálový model

Spirálový model je zařazen k modelům, které jsou pevně svázány s obdobím renesance řízení projektů IS/ICT a metodou prototypování. Projekty, tvořené pomocí spirálového modelu, jsou založeny na schopnosti řešitele vytypovat oblast vhodnou pro podporu prostředky IS/ICT a odkrýt existující latentní poptávku po informačním systému nebo jeho části.[7]

Model definuje čtyři iterace. V každé iteraci se opakují fáze definice cílů, analýza rizik, návrh řešení, ověření, vývoj, testování a plánování. Předmětem první iterace je identifikace globálních rizik projektu a stanovení základních východisek pro řešení. Druhá iterace je zaměřena na specifikaci požadavků. V průběhu třetí iterace se vytváří detailní návrh řešení. Čtvrtá iterace slouží k implementaci navrženého řešení a jeho testování.[5]



Obrázek 20: Spirálový model

Zdroj: převzato z [5]

Příloha B: Metodiky vývoje informačního systému

Následující metodiky jsou pouze výběrem z mnoha dalších. Vybrané metodiky byly vybrány a popsány z literatur [5][6][22]

Metodika tvorby informačního systému je doporučený souhrn etap, přístupů, zásad, postupů, pravidel, dokumentů, řízení, metod, technik a nástrojů pro tvůrce informačních systémů, který pokrývá celý životní cyklus.[22] Metodika je tvořena obecně uznávanými postupy a návody, které popisují činnosti při analýze, návrhu, vývoji, nasazování softwaru stejně jako činnosti spojené s řízením projektu. Cílem metodiky je formalizovat postupy, definovat zodpovědnosti a pravidla komunikace.[6]

Obecné principy metodik tvorby informačního systému (IS) jsou [22]:

- orientace na cíle a problémy – základním východiskem při tvorbě IS je hledání prvotních cílů a problémů v organizaci a z toho odvození cílů IS, realizace tohoto principu je hlavním účelem prvních etap vývoje IS,
- účast zadavatele projektu – vedení organizace musí být aktivně zapojeno po celou dobu vývoje IS, musí být informováno o všech problémech a řešení těchto problémů musí být společným rozhodnutím, bez splnění tohoto předpokladu nemůže být vývoj IS úspěšně realizován,
- klíčové dokumenty a jejich schvalování – bez schválení a potvrzení správnosti obsahu klíčových dokumentů nelze ve vývoji systému dále pokračovat,
- zapojení uživatele do návrhu – zástupci budoucích uživatelů jsou členy projekčních týmů ve všech etapách vývoje systému a jsou spoluodpovědní za řešení,
- modelování a abstrakce, princip tří architektur systému – vytváření tří úrovní modelů systému (konceptuální, technologické, implementační), realizace tohoto principu je zajištěna prostřednictvím používaných metod návrhu IS,
- ověřování a testování návrhu během celého vývoje – na konci každé činnosti je ověřeno, zda výsledky odpovídají cílům organizace, IS a požadavkům uživatelům, a zda jsou správně po formální i logické stránce,
- v každé etapě probíhá analýza i návrh – v jednotlivých etapách jsou analyzovány požadavky na systém a jsou zpodrobnovány na takovou úroveň, aby bylo možné na základě takto provedené analýzy navrhnout systém tak, aby mohla být zahájena další etapa,

- vývoj probíhá z hlediska všech úhlů pohledu na systém – v každé etapě je nutné analyzovat a navrhovat na odpovídající úrovni podrobnosti o datech, funkcích, organizaci, sociálním a psychologickém hledisku, technologii, ekonomickém hledisku,
- otevřenost metodiky – metodika tvorby IS by měla být postavena na běžně používaných a ověřených metodách a technikách vývoje IS a měla by být schopna vstřebávat veškeré nové poznání.

Metodiky lze také rozdělit do dvou skupin, a to na metodiky agilní a rigorózní. Tyto metody vychází z rozdílného chápání procesu vývoje softwaru. Tím jsou dány rozdílné předpoklady pro jejich zavedení, rozdílný obsah těchto metodik a rozdílné oblasti použití.[5] Rigorózní metodiky jsou založeny na přesvědčení, že budování IS/ICT lze popsat, plánovat, řídit a měřit. Jejich snahou je podrobně a přesně definovat procesy, činnosti a vytvářené produkty. Zpravidla bývají založeny na vodopádovém vývoji. Oproti tomu agilní metodiky jsou schopny vytvořit řešení velmi rychle a pružně jej přizpůsobovat měnícím se požadavkům. Jsou založeny na myšlence, že jedinou cestou, jak prověřit správnost navrženého systému, je vyvinout jej co nejrychleji, předložit zákazníkovi a na základě zpětné vazby jej upravit. Každá z agilních metodik je svým způsobem specifická, ale všechny jsou postaveny na stejných principech a hodnotách.[6]

Metodika SDM

System Development Methodology je klasickým reprezentantem strukturovaných metodik, které uvažují vodopádový přístup k tvorbě IS. Výrazným rysem je snaha být metodikou pro vývoj jakéhokoli IS obecně. SDM je velmi důkladně věnován způsobům řízení projektů, prostředkům zajišťování kvality výsledného IS, jsou zvažovány vlivy, které jsou ohrožující pro efektivnost vývoje IS. Jsou zvažovány dopady jednotlivých typů rozhodnutí na IS samotný, ale i na postup jeho tvorby ve fázích, které budou po tomto rozhodnutí následovat. Obecnost SDM je považována za její nespornou výhodu, která však v sobě nese i požadavek na to definovat si konkrétní nástroje, které budou využívány při vlastní tvorbě IS. SDM je vytvářen robustní rámec, do kterého je zasazena konkrétní tvorba, toto může fungovat jako kontrola, zda bylo vše potřebné uděláno.[22]

Metodika OPEN

Jedná se o veřejně přístupnou metodiku podporující celý životní cyklus vývoje. Je zaměřena zejména na vývoj objektově orientovaných a komponentových aplikací. Touto metodikou je definován procesní rámec známý pod názvem OPEN Process Framework

(OPF). Jde o procesní metamodel, ze kterého mohou být generovány instance specifické pro organizaci. Každá taková instance je vytvořena výběrem činností, úloh a technik a jejich specifickou konfigurací. Tento proces je označován jako konstrukce procesu. Touto metodikou je rozšířeno přizpůsobení procesu, kterým jsou přizpůsobovány činnosti a úlohy tak, aby bylo maximálně vyhověno problémové doméně. Tímto způsobem je OPEN flexibilní, může být použit pro malé i velké klíčové, projekty.[6]

Metodika Rational Unified Process

Metodika Rational Unified Process (RUP) je standardem mezi metodikami. Tato metodika byla původně zařazena do rigorózních metodik, pro svou velkou podrobnost, v současnosti je představována jako rámec, ve kterém je možné vytvořit metodiky pro všechny typy projektů od formální metodiky až po velmi lehkou, agilní metodiku. RUP je softwarový inženýrský proces, který je představován jako disciplinovaný přístup přiřazující úkoly a odpovědnosti v organizaci zabývající se vývojem softwaru. Je založena na nejlepších praktikách softwarového vývoje, kterými jsou iterativní vývoj, řízení požadavků, použití komponentové architektury, vizuální modelování, kontrola kvality softwaru a řízení změn. Proces vývoje je popsán v rámci dvou dimenzí, které odpovídají osám. Horizontální osa je tvořena dynamickým pohledem na proces, který je vyjádřen pomocí cyklů, fází, iterací a milníků. Vertikální osa je reprezentována statickým hlediskem procesu, popisu činností, artefaktů, pracovníků a pracovních toků.[5]

Metodika Feature-Driven Development

Jedná se o agilní metodiku, která zachovává procesní řízení a zdůrazňuje úlohu modelování při vývoji. Je založena na iterativním vývoji, který je řízen užitnými vlastnostmi produktu. Začátkem vývoje je vytvoření celkového modelu a je pokračováno posloupností dvoutýdenních iterací, ve kterých je prováděn návrh a realizace pro jednotlivé užité vlastnosti. Užité vlastnosti jsou malé výsledky užitečné z pohledu zákazníka, který je srozumitelný, měřitelný a realizovatelný spolu s dalšími v rámci dvoutýdenní iterace.[6]

FDD se skládá z 5 procesů [6]:

- vytvoření celkového objektového modelu,
- sestavení seznamu užitečných vlastností,
- plánování pro užitečnou vlastnost,
- návrh pro užitečnou vlastnost,

- realizace pro užitnou vlastnost.

Metodou FDD není přeceňován význam procesů při vývoji. Cílem je fungující produkt, nikoli, že je splněn předepsaný proces. FDD je i přesto definován lehkými procesy, které jsou popsány na více stránkách textu a které umožňují škálovat metodikou na větší projekty, rychleji zapojit nové zaměstnance, stanovit priority, zaměřit se na přínosy.[6]

Metodika Scrum

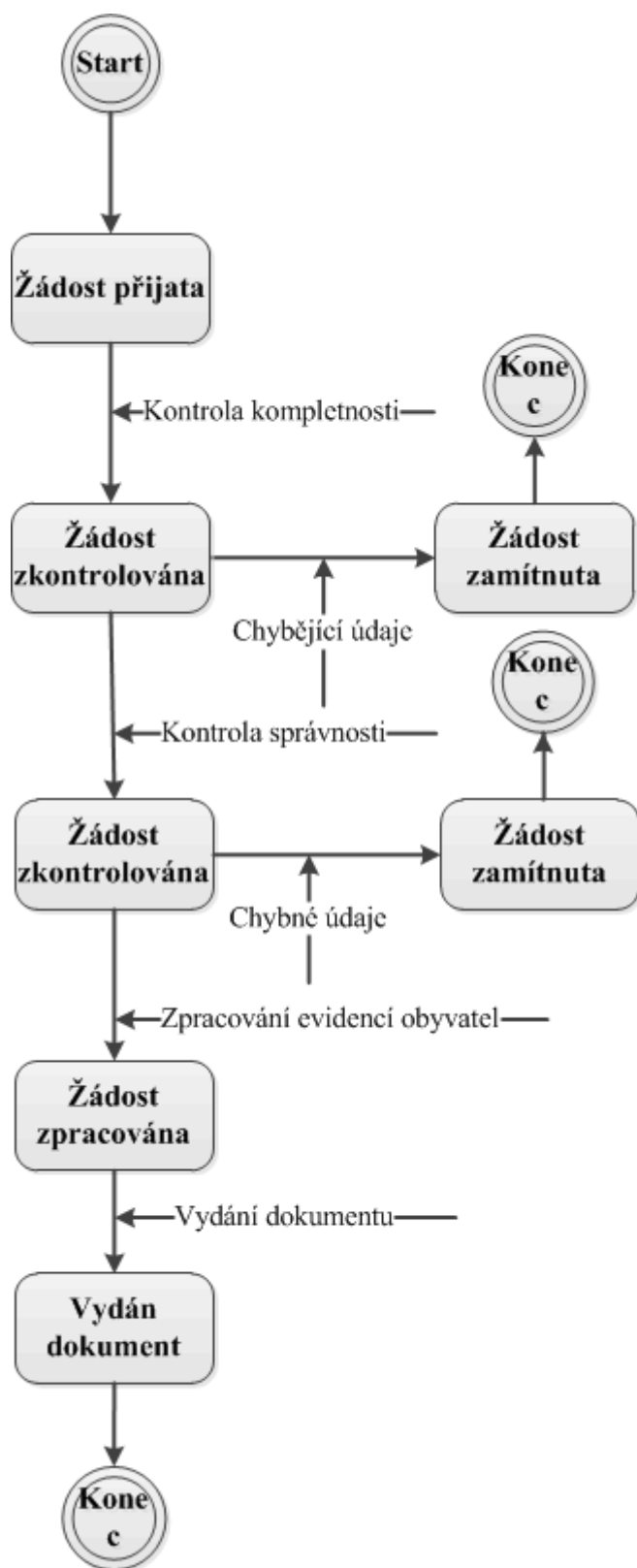
Tato metodika je založena na přesvědčení, že vývoj softwaru není definovaný proces, ale empirický proces, a proto je vyžadován úplně odlišný styl řízení. Metodika Scrum je adaptivní, rychlá a samoorganizující. Je zaměřena především na řízení projektu.[6] Vývoj je veden v 30denních iteracích nazývaných Sprint, ve kterých je dodávána vybraná množina užitných vlastností. Klíčovou praktikou jsou denní porady, které slouží pro koordinaci a integraci prací. Členové týmu se zapíší k úkolům, každý pracuje na splnění cíle Sprintu a každý se účastní denních porad. Denními poradami je monitorován stav projektu a jsou identifikovány problémy a překážky, které jsou ohrožující pro úspěch projektu. Sprint je konán každý den ve stejný čas a na stejném místě a doby trvání jsou maximálně 30 minut. Každý člen je povinen zodpovědět tři otázky: které položky dokončil od minulé rady, které nové úkoly má řešit a jaká vidí omezení a překážky pro řešení úkolů.[5] Metodikou Scrum jsou definovány čtyři fáze životního cyklu. V první a poslední fázi jsou vymezeny procesy, ve kterých jsou určeny vstupy, výstupy a transformace vstupu na výstupy. Těmito procesy je vyjádřena explicitní znalost a jejich provádění je lineární. Fáze vývoje je naproti tomu empirický proces, jehož činnosti nelze zpravidla identifikovat a řídit. Vstup do této fáze je jako černá skříňka, ve které je vyžadováno vnější řízení.[6]

Fáze metodiky Scrum jsou [6]:

- plánovací fáze – specifikují se funkční požadavky, plán dodávek, architektonická a byznys vize,
- vynášecí fáze – do souboru požadavků jsou připojeny nefunkční požadavky jako bezpečnostní, výkonnostní,
- fáze vývoje – jeden nebo více týmů dodává funkcionalitu s nejvyšší prioritou každých 30 dní, na konci každé iterace tým předvede výsledek,
- fáze dodávky – předání produktu uživatelům.

Příloha C: Diagramy případové studie

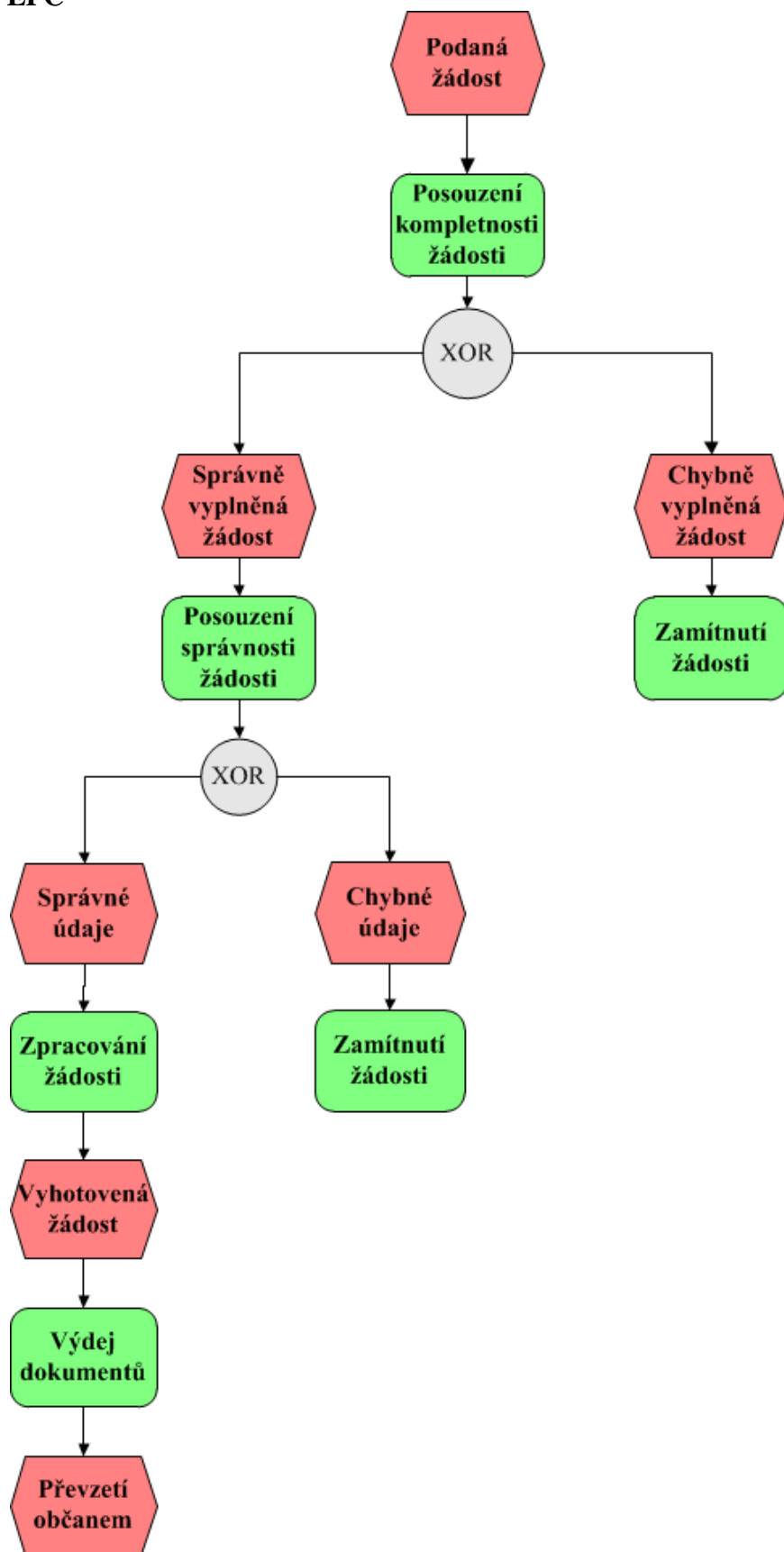
Stavový diagram



Obrázek 21: Stavový diagram vyřizování žádosti

Zdroj: vlastní zpracování

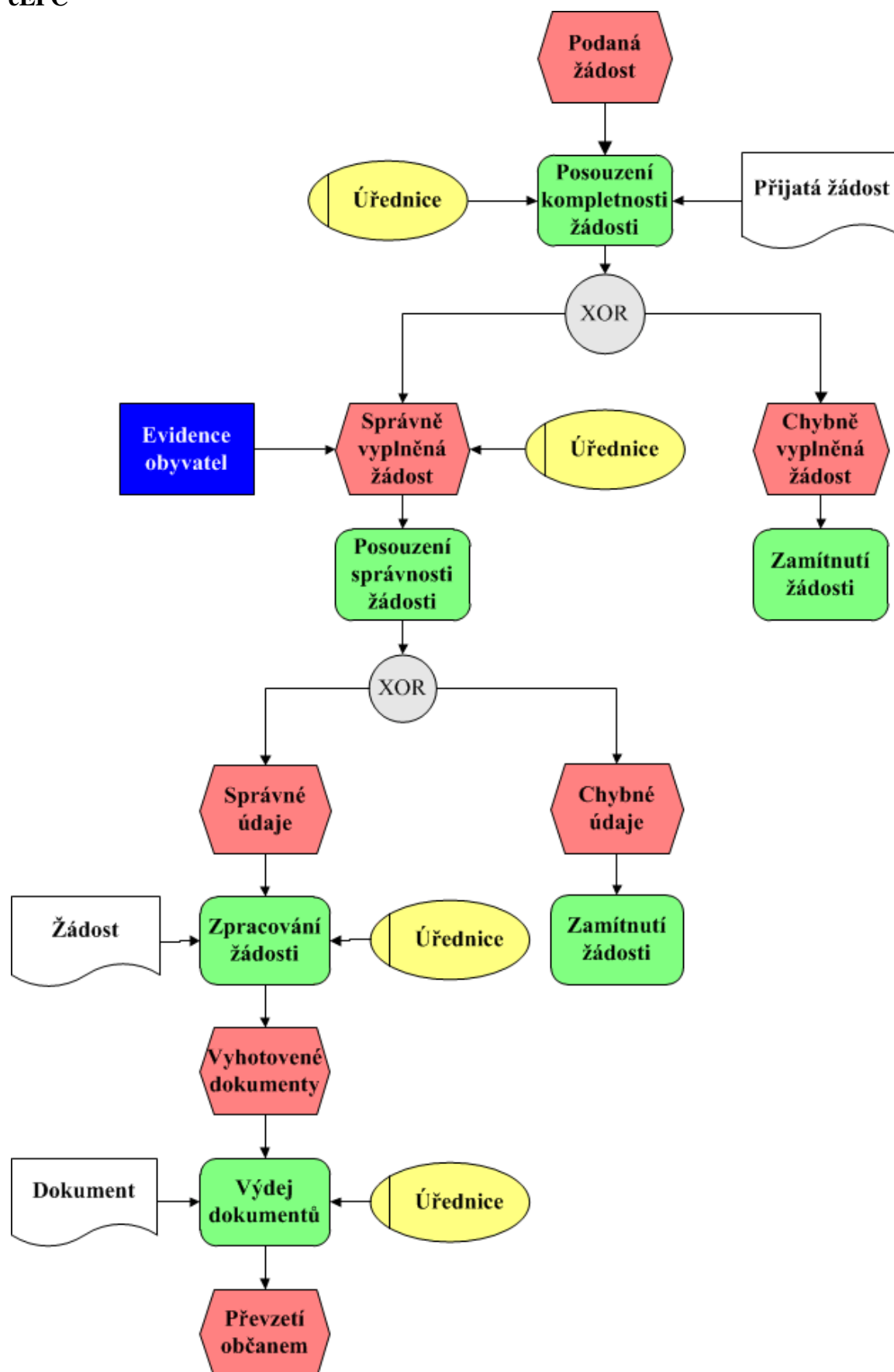
EPC



Obrázek 22: EPC diagram vyřizování žádosti

Zdroj: vlastní zpracování

eEPC



Obrázek 23: eEPC diagram

Zdroj: vlastní zpracování

Případ užití

Pro další ukázkou je uveden případ užití zapsání nového občana. Zapsání nového občana, v tomto případě narozeného dítěte, je provedeno v takzvané živé kartotéce. Pro zadání údajů je v modulu evidence obyvatel připraven formulář, ve kterém je nutné zapsat povinné údaje. Jsou tu k nalezení také nepovinné údaje, které je dobré mít vyplněné. Jsou to zejména občanství, číslo občanského průkazu, stav, jméno za svobodna a další. Povinnými údaji jsou rodné číslo, jméno a příjmení, datum narození, pohlaví, adresa trvalého bydliště, datum přihlášení a důvod registrace. Důvody registrace jsou různé, například přestěhování nebo pořízení dat o novém občanovi.

Tabulka 7: Hlavní scénář případu užití Zapsání nového občana

| | | | |
|----------------------|---|-------------|---|
| Use Case | Zapsání nového občana | | |
| Aktér | Úřednice | | |
| Úroveň | Uživatelská | | |
| Popis | Zapsat nově narozeného občana do evidence obyvatel | | |
| Vstupy | K dispozici je žádost občana s příloženým rodným listem nově narozeného dítěte a informace o dítěti | | |
| Výstup | Dítě je zapsáno do evidence obyvatel | | |
| Spouštěč | Přijatá žádost o zapsání nového občana | | |
| Hlavní scénář | 1 | Role | Akce |
| | 1.1 | A | zadá přihlašovací údaje do IS Munis |
| | 1.2 | S | přihlásí aktéra |
| | 1.3 | A | poklepe myší na modul evidence obyvatel |
| | 1.4 | S | otevře modul evidence obyvatel |
| | 1.5 | A | poklepe na přidání nového občana – živá kartotéka |
| | 1.6 | S | otevře formulář k vyplnění |
| | 1.7 | A | vyplní údaje o občanovi |
| | 1.8 | A | klikne myší na tlačítko uložit |
| | 1.9 | S | uloží data do modulu evidence obyvatel |
| | 1.10 | A | zavře modul evidence obyvatel a odhlásí se ze systému |

Zdroj: vlastní zpracování

Alternativní scénář je možné zhotovit pro více případů z hlavního scénáře. Scénář bude vyhotoven pro body 1.1, 1.8. Přičemž u bodu 1.8 může nastat více situací, které jsou dosaženy

tím, že úřednice nevyplní povinný údaj, některý údaj nebude v potřebném formátu, například rodné číslo nebo číslo OP nebudou mít potřebný počet číslic.

Tabulka 8: Alternativní scénář

| | | |
|----------------------------|----------|---|
| Alternativní scénář | 1 | Alternativa k 1.1. Aktér vyplní špatně přihlašovací údaje, v tomto případě systém vrátí aktéra na stránku s přihlašovacími údaji a napíše oznámení o chybně zadaných údajích. |
| Alternativní scénář | 2 | Alternativa k 1.8. Pokud aktér vyplní některý z požadovaných údajů nedostatečným způsobem, pak systém neuloží formulář a označí chybný údaj. |
| Alternativní scénář | 3 | Aktér může vyplňování formuláře ukončit volbou Zrušit. |

Zdroj: vlastní zpracování