

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

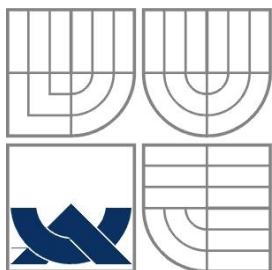
**ROZŠÍŘENÍ FUNKČNOSTI NÁSTROJE PROCESS
INSPECTOR**

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

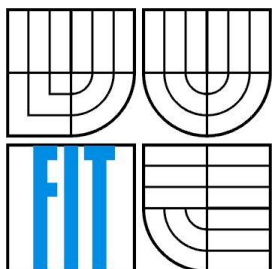
AUTOR PRÁCE
AUTHOR

Bc. MARTIN OPRŠAL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ROZŠÍŘENÍ FUNKČNOSTI NÁSTROJE PROCESS INSPECTOR

PROCESS INSPECTOR TOOL FUNCTIONALITY EXTENSION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Martin Opršal

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. Jitka Kreslíková, CSc.

BRNO 2011

Zadání diplomové práce

Řešitel: **Opršal Martin, Bc.**

Obor: Management a informační technologie

Téma: **Rozšíření funkčnosti nástroje Process Inspector
Process Inspector Tool Functionality Extension**

Kategorie: Softwarové inženýrství

Pokyny:

1. Prostudujte problematiku managementu procesů.
2. Seznamte se s nástrojem Process Inspector a s platformou SharePoint včetně technologií ASP a .NET.
3. Po konzultaci s vedoucí navrhněte rozšíření nástroje o nové funkce na platformě SharePoint na základě požadavků z jeho praktického použití. Vyhodnoťte přínos jednotlivých rozšíření.
4. Navrhněte realizaci a implementujte rozšíření s nejvyšším přínosem v praktickém použití.
5. Ověřte nové rozšíření na reálných datech.
6. Zhodnoťte dosažený přínos na základě praktického ověření a diskutujte další možná rozšíření výsledného produktu.

Literatura:

- Becker, J., Kugeler, M., Rosemann, M.: Process Management, Springer-Verlag, 2003, ISBN: 3-540-43499-2.
- Dyba, T., Dingsoyr, T., Moe, N. B.: Process Improvement in Practice, Kluwer Academic Publishers, 2004, ISBN: 1-4020-7869-2.
- <http://msdn.microsoft.com/cs-cz/library/default.aspx>
- Dokumentace stávající implementace nástroje Process Inspector na technologii .NET.
- Příspěvky projektu Knowledge Management Informační Společnosti zabývající se nástrojem Process Inspector

Při obhajobě semestrální části diplomového projektu je požadováno:

- Nepožaduje se.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

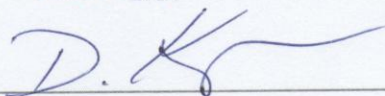
Vedoucí: **Kreslíková Jitka, doc. RNDr., CSc., UIFS FIT VUT**

Konzultant: Fiedler Zdeněk, Ing., Allium

Datum zadání: 20. září 2010

Datum odevzdání: 25. května 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Diplomová práce se zabývá problematikou managementu procesů, a to především obecnými principy, které vedou ke zdokonalení procesů ve společnosti. Hledá metody, jak usnadnit identifikaci a popis procesů. Je zde uvedena aplikace, která má právě identifikaci a popis procesů usnadnit. Následně je popsána praktická implementace této aplikace na platformu Microsoft SharePoint.

Abstract

This master's thesis deals with process management, especially the general principles that lead to improvements in company processes. It's looking for methods to facilitate the identification and description processes. There are those applications that have just the identification and description of the process easier. Following is a description of the practical implementation of this application to Microsoft SharePoint.

Klíčová slova

Process Inspector, proces, procesní řízení, řízení procesů, analýza procesů, Microsoft SharePoint 2010, Microsoft SharePoint Designer 2010, Microsoft Visual Studio 2010, asp.Net

Keywords

Process Inspector, process, process management, process analysis, Microsoft SharePoint 2010, Microsoft SharePoint Designer 2010, Microsoft Visual Studio 2010, asp.Net

Citace

Opršal Martin: Rozšíření funkčnosti nástroje Process Inspector, diplomová práce, Brno, FIT VUT v Brně, 2011

Rozšíření funkčnosti nástroje Process Inspector

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. RNDr. Jitky Kreslíkové, CSc.

Další informace mi poskytl Ing. Zdeněk Fiedler

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Opršal

24. 5. 2011

Poděkování

Děkuji tímto doc. RNDr. Jitce Kreslíkové, CSc. a Ing. Zdeňku Fiedlerovi za pomoc a podporu při řešení této práce

© Martin Opršal, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Management procesů	4
2.1 Definice základních pojmů	4
2.1.1 Proces	4
2.1.2 Procesní řízení.....	4
2.1.3 Role procesního řízení.....	5
2.2 Cyklus zdokonalení procesů	6
2.2.1 Identifikace procesů	6
2.2.2 Měření procesů.....	7
2.2.3 Shromažďování dat	9
2.2.4 Analýza chování procesu	10
2.2.5 Zlepšení procesu	12
2.3 Reengineering.....	12
2.3.1 Proč orientace na procesy	13
2.3.2 Úloha informačních technologií.....	14
3 Process Inspector	15
3.1 Získávání znalostí o procesech	15
3.2 Základní koncepce	16
3.2.1 Základ práce s nástrojem Process Inspector	16
3.3 Historie.....	18
3.4 Konkurenční produkty	19
4 SharePoint 2010.....	21
4.1 Představení aplikace SharePoint	21
4.1.1 Edice v SharePoint	22
4.2 Základní prvky.....	22
4.2.1 Listy a knihovny.....	22
4.2.2 Webové stránky.....	23
4.3 Správa SharePoint	23
4.3.1 Instalace SharePoint	23
4.3.2 Programy pro správu	24
4.4 Technologie asp.Net.....	25
4.5 Přístup k datům	26
4.5.1 CAML	26

4.5.2	LINQ	26
4.5.3	REST	27
4.6	Vývoj.....	27
4.7	Exporty a importy	28
4.7.1	Záloha a obnovení	28
4.7.2	Export a Import řešení webu.....	29
4.7.3	Import webu do Visual Studio	29
5	Návrh systému	31
5.1	Datová struktura.....	31
5.1.1	List procesů.....	32
5.1.2	List dokument	32
5.1.3	List se slovníkem.....	33
5.1.4	List otázek.....	33
5.1.5	List odpovědí	33
5.1.6	List typů procesu	34
5.2	Uživatelské rozhraní	34
5.2.1	Úvodní stránka	34
5.2.2	Rychlá navigace	36
5.2.3	Stránky detailu položek v listech	36
5.3	Chybějící dokumenty	37
5.4	Web part strom procesů.....	38
6	Implementace	39
6.1	Listy	39
6.1.1	Definice typu.....	40
6.1.2	Definice listů.....	41
6.1.3	Definice pohledů	42
6.2	Projekt typu Event Receiver	42
6.2.1	Editace a vkládání položek	42
6.2.2	Chybějící dokumenty	43
6.3	Webová komponenta strom procesů	44
6.3.1	Test rychlosti.....	45
6.4	Uživatelské rozhraní a ovládání.....	46
6.4.1	Úvodní stránka	46
6.4.2	Detailní pohled na proces	47
6.4.3	Detailní pohled na dokument	49
7	Závěr	50
7.1	Další rozšíření.....	50

1 Úvod

Při implementaci firemního informačního systému, je třeba analyzovat procesy, které ve společnosti probíhají. Právě analýza procesů, nejvíce ovlivňuje výsledný informační systém. Je třeba identifikovat procesy, které jsou klíčové pro výsledný produkt společnosti a neopomenout jejich podporu v informačním systému. Proto je třeba hledat automatizované postupy, které usnadní celou analýzu, která probíhá na konfrontaci mezi konzultantem a zákazníkem.

V úvodu práce se ale budeme zabývat obecnými postupy managementem procesů, bude zde představen plán jak zlepšovat procesy ve společnosti, včetně definic základních pojmů od kterých se dále v práci budeme odrážet.

Také se zde budeme zabývat aplikací Process Inspector v té podobě jako je teď nasazená ve společnosti Allium, včetně metodiky práce s ní. Poté navrhne a implementujeme novou verzi aplikace Process Inspector, která bude na platformě Microsoft SharePoint 2010. Platformě SharePoint bude věnována jedna kapitola, ve které bude platforma představena, a budou zde popsány základní principy. Tato nová verze bude vycházet z původní aplikace a podle nastudované problematiky a rozšířena o další prvky, které usnadní analýzu procesů.

Tato diplomová práce nenavazuje na semestrální projekt.

2 Management procesů

Tato kapitola se zabývá managementem procesů, definujeme zde základní pojmy, ukážeme jeden z možných přístupů jak procesy řídit. V krátkosti se zde podíváme i na fenomén Reengineering. Tato kapitola pochází z velké části z literatur [1,2,3], tamtéž se lze dočíst další podrobnosti.

2.1 Definice základních pojmů

V literaturách je možno najít mnoho různých definic ať už procesů (z anglického Process) nebo procesního řízení (z anglického Process Management), bylo vybráno jen pár, které nejlépe vystihují jednotlivé pojmy. Například v literatuře [13] je možné nalézt pěkný přehled různých definic jednotlivých pojmů.

2.1.1 Proces

První definice pochází od Gabriel A. Pall (z roku 1987) a je čerpána z literatury [13]:

„Proces může být definován jako logická organizace lidí, materiálů, energie, nástrojů a procedur do pracovních činností, jejichž cílem je výroba specifického koncového výsledku.“

Jako druhou definici přináším značně komplexnější definici, která tu předchozí rozšiřuje zejména v zavedení pojmu subprocess (podproces), definice pochází z literatury [13]:

„Proces je organizovaná skupina vzájemně souvisejících činností a/nebo subprocessů, které procházejí jedním nebo více organizačními útvary či jednou (podnikový proces) nebo více spolupracujícími organizacemi (mezipodnikový proces), které spotřebovávají materiál, lidské, finanční a informační vstupy a jejichž výstupem je produkt, který má hodnotu pro externího nebo interního zákazníka“

2.1.2 Procesní řízení

Z anglického originálu process management, někdy taky překládáno jako řízení procesů. Zde je uvedena definice, která pochází z literatury [13]:

„Procesní řízení představuje systémy, postupy, metody a nástroje trvalého zajištění maximální výkonnosti a neustálého zlepšování podnikových i mezipodnikových procesů, které vycházejí z jasně definované strategie organizace a jejichž cílem je naplnit stanovené strategické cíle.“

Vlastními slovy bych řekl, že se jedná o souhrn nejlepších praktik (ať už se jedná o postup, metodu, nástroj, atd.) z organizace potažmo celého oboru, tyto praktiky jsou definovány v podobě

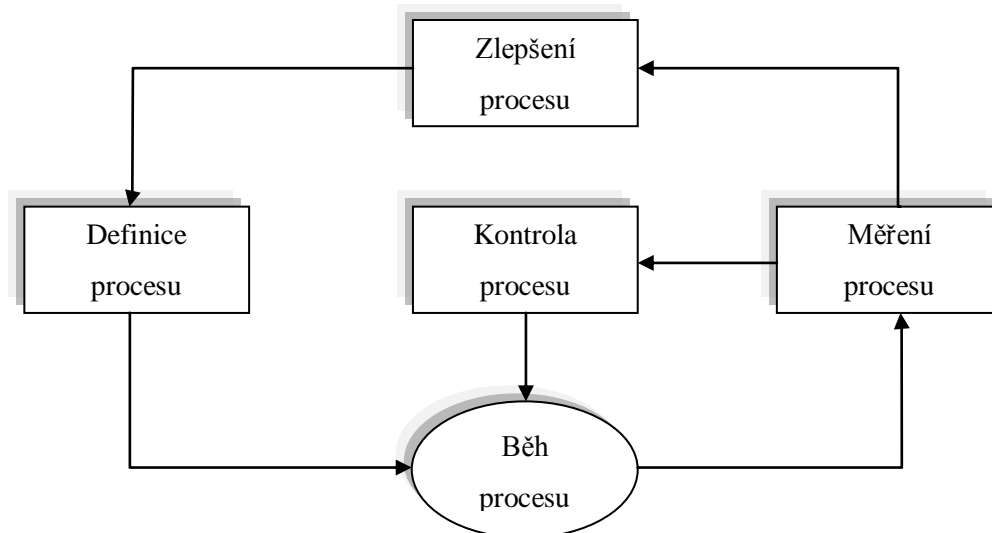
procesů. Podstatné je, aby k nim měl kdokoliv přístup, respektive jen ten, kdo s nimi může zacházet, a mohl na základě znalosti těchto procesů zkvalitnit řízení a organizaci společnosti. Souhrn praktik je neustále rozšiřován o nové procesy.

2.1.3 Role procesního řízení

Čtyři úkoly, které jsou zásadní pro řízení procesů, jsou následující (graficky zobrazeny na obrázku 1):

- Definice procesu
- Měření procesu
- Kontrola procesu
- Zlepšení procesu

Tyto čtyři kroky jsou odvozené z definice, kterou definoval W. Edwards Deming v tzv. Demingově principu PDCA kruhu (PDCA z anglického Plan, Do, Check, Act), jedná se o iterativní učení v cyklu, které napomáhá ke zlepšování procesů.



Obrázek 1: Čtyři klíčové úkoly procesního řízení [2]

Definice procesu – skládá se z dvou úkolů analýza a definice procesu, měly by být vydefinovány především ty procesy, které zlepší dosahování cílů společnosti.

Měření procesu, je určeno na odhalení odchylek od požadovaného výkonu, případně pro identifikaci příležitosti pro zlepšení procesu.

Kontrola procesu, znamená, aby proces neměl výkonnostní odchylky od normálu, tedy aby se proces choval vždy konzistentně.

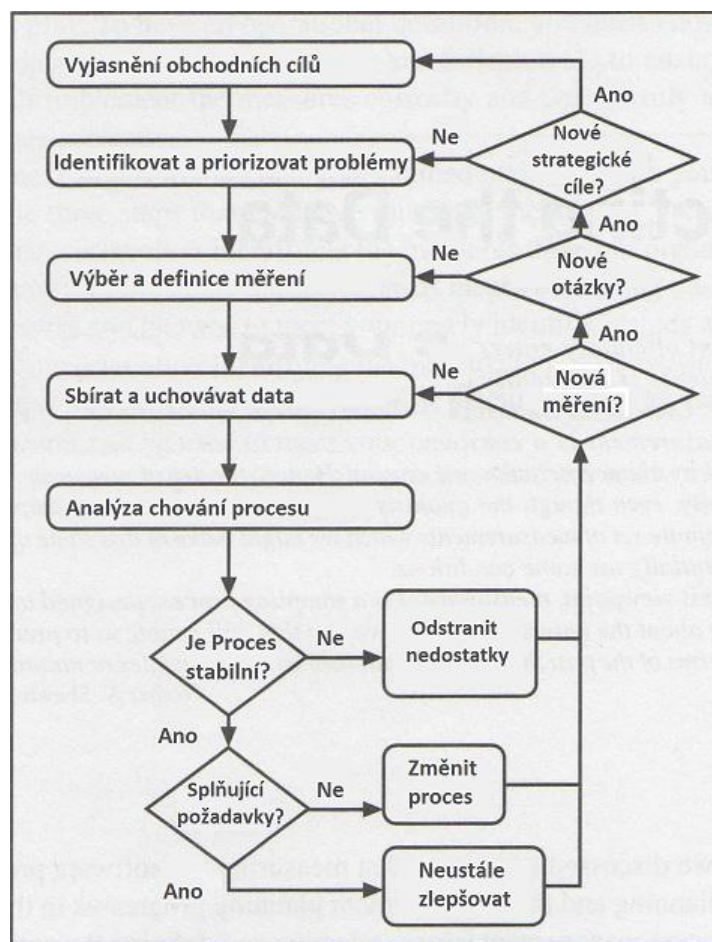
Zlepšení procesu

- Pochopení vlastností existujících procesů a faktorů, které je ovlivňují.
- Plánovat, ověřit a implementovat akce, které změni proces tak aby snáze dosáhl toho co je třeba.
- Posoudit dopady a získané přínosy a porovnat je s náklady na změnu procesu.

2.2 Cyklus zdokonalení procesů

Tato kapitola se zabývá blíže obecným způsobem jak procesy vylepšovat. Model, který zde představím je čerpán z literatury [2] a v ní je také možné nalézt podrobnější informace.

Na obrázku 2, jsou zobrazeny kroky, které vedou ke stálému zlepšování procesu, tento pracovní postup je postaven na dobře zmapovaných procesech a následně na jejich přesném měření, tak aby z těchto měření vyšly výsledky, které lze použít ke zlepšení procesu.



Obrázek 2: Management procesů [2]

2.2.1 Identifikace procesů

Ve zdrojové literatuře je neprávem pomíjena část věnována identifikaci procesu, nejdříve potřebujeme proces identifikovat a jasně říci co je jeden proces a co druhý, poté až máme takto identifikovaný proces je možné jakkoliv s tímto procesem zacházet, ať už ho měřit, zlepšovat nebo kontrolovat.

Identifikace procesu vyžaduje procesní způsob myšlení, se kterým často lidé nemají žádné zkušenosti. Musí se odtrhnout od hierarchického uspořádání společnosti a vidět svou činnost v globálním kontextu. Tato část je z celého managementu procesu intelektuálně nejnáročnější, právě

proto, že každý se musí odvázet od svých povinností a zodpovědností. Identifikace procesu stojí na vzájemné výměně znalostí mezi konzultantem, který zná obecné procesy společností a uživatelem, který zná procesy konkrétní společnost.

Nejrozsáhlejší procesy ve společnosti jsou hlavní procesy, jsou to procesy navržené pro uspokojení potřeb zákazníka. Hlavních procesů mívá společnost mezi čtyřmi až osmi. Důležité body pro identifikaci hlavních procesů jsou:

- Výchozím bodem definice hlavního procesu by měl být strategický cíl organizace a základní problémové oblasti konkurenceschopnosti
- Všechny funkce či informační toky, které ovlivňují hodnotu vytvářenou pro zákazníka, musí být vzaty v úvahu
- Hlavní procesy často obsahují i aktivity, které provádí zákazník nebo dodavatel

Hlavní procesy se skládají z několika menších procesů, které mohou být kategorizovány podle jejich povahy [17]:

- **Strategické procesy**, pomocí nichž společnost plánuje a realizuje rozvoj
- **Provozní procesy**, pomocí nichž společnost realizuje každodenní fungování
- **Podpůrné procesy**, které umožňují fungování strategických a provozních procesů

2.2.2 Měření procesů

Měření softwarových procesů začíná plánováním, které obsahuje tři kroky: určení oblastí procesního řízení, výběr a definice měření produktů a procesů a poslední krok integrace měření do organizace, jednotlivé kroky jsou popsány níže. Na výše uvedeném obrázku 2 jsou zobrazeny tyto kroky v kontextu celého procesního řízení.

2.2.2.1 Identifikace přínosu procesu

V této fázi, dohází k vyjasnění cílů společnosti, tak aby bylo možné tyto cíle přiřadit k firemním procesům. Z těchto procesů pak dochází k identifikaci kritického procesu, který se potýkal s problémy v minulosti, používá se nová technologie, nebo je více vytížen jak v minulosti. Dále je vytvářen seznam cílů, pro každý z kritických procesů. Výstupem této fáze je také seznam potenciálních problémů, vůči ostatním procesům.

2.2.2.2 Výběr a definice měření

Dalším krokem je definice měření, je potřeba definovat takové měření, které pomůže ke zlepšení produktu. Měřit můžeme vlastnosti výsledného produktu, který proces produkuje nebo proces samotný.

Příklady měřitelných atributů softwarových produktu jsou například funkce, velikost, rychlo běhu, samozřejmě do této kategorie patří i atributy kvality, které mohou udělat produkt atraktivnějším pro zákazníka. Obrázek 3 ukazuje, jaké vlastnosti lze měřit u procesů samotných. Příklad měřitelných

atributů procesu vynaložené úsilí, čas, velikost workflow, množství použitých zdrojů, množství typy defektu, oprav.

<i>Měřitelné vlastnosti objektů v softwarovém procesu</i>			
Věci přijaté, či užité	Aktivity a jejich části	Věci spotřebované	Věci produkované
Změny <ul style="list-style-type: none"> ◆ Typ ◆ Datum ◆ Velikost ◆ # obdrženo Požadavky <ul style="list-style-type: none"> ◆ Požadavky na stabilitu ◆ # identifikováno ◆ % v návrhu ◆ % v programování Zprávy o problému <ul style="list-style-type: none"> ◆ typ ◆ datum ◆ velikost ◆ původ ◆ striktnost ◆ # obdrženo Fondy <ul style="list-style-type: none"> ◆ peníze ◆ rozpočet ◆ stav Lidé <ul style="list-style-type: none"> ◆ roky zkušeností ◆ typ vzdělání ◆ % trénováno v XYZ Zařízení a prostředí <ul style="list-style-type: none"> ◆ stop čtverečních na zaměstnance ◆ hluk ◆ osvětlení ◆ # personálu v kanceláři ◆ investice do nábytku pro zaměstnance ◆ hodin používání počítače ◆ % využití kapacity 	Cesty <ul style="list-style-type: none"> ◆ čas zpracování ◆ výkon ◆ rozptýlení ◆ zpoždění ◆ rezervy Délka, velikost <ul style="list-style-type: none"> ◆ Fronty ◆ Zásobníky ◆ Vyrovnávací paměti 	Úsilí <ul style="list-style-type: none"> ◆ # hodin vývoje ◆ # hodin přepracování ◆ # hodin příprav ◆ # hodin schůzek Čas <ul style="list-style-type: none"> ◆ Východí čas/datum ◆ Konečný čas/datum ◆ Délka procesu/úlohy ◆ Čekací doba Peníze <ul style="list-style-type: none"> ◆ Cena k datu ◆ Cena povolení ◆ Cena přepracování 	Stav pracujících jednotek <ul style="list-style-type: none"> ◆ # vyrobeno ◆ # naprogramováno ◆ # otestováno Velikost pracujících jednotek <ul style="list-style-type: none"> ◆ # požadavků ◆ # funkčních bodů ◆ # řádků kódu ◆ # modulů ◆ # objektů ◆ # bytů v databázi Výstupní množství <ul style="list-style-type: none"> ◆ # akčních položek ◆ # schválení ◆ # nalezených chyb Výsledky testu <ul style="list-style-type: none"> ◆ # úspěšných případů testů ◆ % pokrytí testy Architektura programu <ul style="list-style-type: none"> ◆ fan-in ◆ fan-out Změny <ul style="list-style-type: none"> ◆ typ ◆ datum ◆ velikost ◆ vydané úsilí Problémy a defekty <ul style="list-style-type: none"> ◆ # zpráv ◆ hustota chyb ◆ typ ◆ původ ◆ rozšíření podle typu ◆ rozšíření podle původu ◆ # otevřených ◆ # uzavřených Kritické využití zdrojů <ul style="list-style-type: none"> ◆ % využití paměti ◆ % využití kapacity cpu ◆ % využití kapacity I/O

Obrázek 3: Příklady měřitelných entit v procesu [2]

Jakmile máme definováno, co chceme měřit, je třeba definovat měření. Je třeba, aby ti co budou měření provádět, věděli, co měření obnáší, a aby byli schopni sbírat a interpretovat data korektně. Důležité také je, aby lidé, kteří budou s daty pracovat a analyzovat je, věděli, odkud se data

berou a co představují, protože jinak hrozí, že data budou špatně interpretována. Na překonání těchto nedostatků, je dobré se ztotožnit se správně definovanými měřeními (z anglického well-defined measures), proto aby bylo měření dobře definované, musí splňovat tyto 3 kritéria, která sestavil W. Edwards Deming.

1. **Komunikace**, metody, které jsou použity pro definici měření nebo popis měření, musí ostatním precizně pochopit a vědět o čem měření je a co bude měření zahrnovat a co naopak měřeno nebude. Navíc každý kdo data bude používat, musí vědět, jak byla data shromážděna a díky tomu bude data chápat správně.
2. **Opakovatelnost**, je někdo další schopný opakovat měření a získat tak stejné výsledky?
3. **Sledovatelnost**, je původ dat identifikovatelný z hlediska času, zdroje, aktivity, produktu, prostředí, použitých měřících nástrojů a shromažďovacím agentem?

Sledovatelnost je především důležitá pro hodnocení a zlepšení výkonu procesu, protože měření výkonu může signalizovat nestabilitu procesu, je důležité, že kontext a okolnosti měření jsou uloženy v rámci měření. Díky tomu je možné určit důvody nestability.

2.2.2.3 Integrace s procesy

Třetím krokem v měření je jeho integrace s procesy samotnými. Zde jsou rozebrány tři kroky implementace.

Analýza především zkoumá, jaká měření probíhají v organizaci, a tím pokládá základy pro další dva kroky. Ve většině organizací nějaké měření probíhá, a proto je vhodné tato měření použít jako odrazový můstek pro nová měření, ať už dojde k úpravě anebo se jedná o nové měření.

Diagnóza znamená hodnocení dat, která jsou získávána u běžících měření. U těchto dat je posuzováno, jak jsou použitelná pro cíle nového měření, a to jestli se dají data použít, případně data upravit, nebo upravit měření tak aby odpovídalo datům, která chceme získat.

Akce převádí výsledky předešlých dvou kroků do implementace měření. Tedy hledá řešení jak měření provést a dohlíží, aby měření bylo provedeno správně.

2.2.3 Shromažďování dat

Operační (pracovní) činnosti, které jsou spojené se zdokonalením procesu, začínají právě sběrem dat. Procedury, které jsou definovány pro sběr dat, musí být integrovány do procesů a spuštěny. Tedy správní lidé, činnosti, senzory a nástroje musí být na správném místě a musí poskytovat výsledky. Samozřejmě shromažďování dat zahrnuje i samotný záznam a uchovávání dat. Úkoly, které je potřeba při sbírání a zaznamenávání dat provést jsou následující:

- Navrhnout metody a získat nástroje, které usnadní sběr dat.
- Vyškolit nebo najmout personál, který bude sběr dat provádět.
- Získávat a ukládat data pro všechny procesy, které jsou měřeny.
- Používat definované formáty pro předávání dat mezi lidmi, kteří je budou analyzovat.

- Monitorovat průběh a výkon aktivit, které sbírají data

Stejně jako ostatní procesy ve společnosti, tak i měření a sběr dat je proces a jako proces musí být kontrolován a samozřejmě musí být stabilní, proto aby dal správné výsledky.

Důležitým krokem před analýzou dat je jejich zkoumání a ověření. Proto, aby analýza byla užitečná, musí nasbíraná data splňovat kritéria pravdivosti (tedy například data jsou ve správném formátu, jsou kompletní, arytmiicky správně), konzistence (ověření správnosti dat, která jsou odlehlá nebo nepravděpodobná) a validity (hodnoty měření opravdu popisují atributy, které chceme měřit). Je důležité rozpoznat včas, jestli sbíraná data splňují tato kritéria. A v případě, že data kritéria nespĺňují začít měření od začátku.

2.2.4 Analýza chování procesu

Stabilita procesu, je důležitá proto, aby proces vytvářel vždy stejné výsledky a byl tedy smysluplně analyzovatelný.

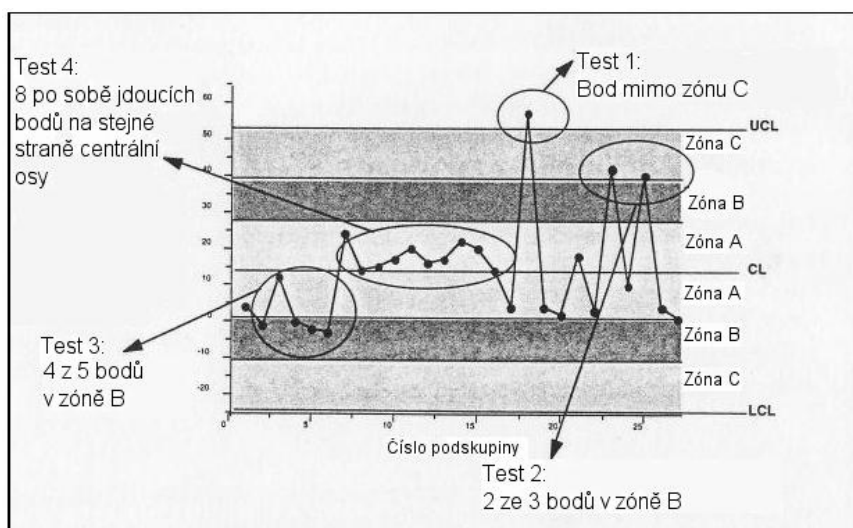
2.2.4.1 Kontrolní diagramy

Každý kontrolní diagram (z anglického Control Charts) má centrální osu a na obou stranách osy jsou kontrolní limity. Osa a limity reprezentují odhady, které jsou vypočítané ze souboru dat nashromážděného během provádění procesu. Středové a kontrolní limity nelze přiřadit libovolně, neboť jsou určeny k odhalení toho, co proces může skutečně udělat, ne toho, co někdo chce, aby udělal.

Kontrolní limity jsou ± 3 sigma od osy, kde sigma je směrodatná odchylka, která je zakreslená v grafu. V praxi se ukázalo, že používání limitu 3 sigma je adekvátně přesné k neobvyklým hodnotám, a vede k velmi málo falešným poplachům.

2.2.4.2 Test na nestabilní a nekontrolovaný proces

Existuje několik testů pro odhalování nestability procesu, ten co je představen níže, pochází z literatury [2], je určen převážně pro X-sloupcový graf. Pro ostatní typy grafu je použitelný s menšími úpravami, hledat lze v literatuře.



Obrázek 4: Jak poznat nestabilní proces [2]

Testy jak zjistit jestli se jedná o stabilní proces, graficky zobrazeny na obrázku 5.

- Test 1: jeden bod je mimo rozsah 3 sigma kontrolní limity
- Test 2: alespoň dva ze tří po sobě jdoucích bodu, které jsou na stejné straně od osy a dál jak 2 sigma (zóna C)
- Test 3: alespoň čtyři z pěti po sobě jdoucích bodů, které jsou na stejné straně osy a dál jak 1 sigma (zóna B)
- Test 4: alespoň osm po sobě jdoucích hodnot, které jsou na stejné straně osy (zóna A)

Testy 2,3 a 4 jsou nazývány testy běhu (z anglického run test) a jsou postaveny na základě domněnky, že distribuce vlastní variace je symetrické kolem průměru a po sobě jdoucí hodnoty jsou statisticky nezávislé. Údaje jsou vyneseny v časové posloupnosti.

2.2.4.3 Další grafy

Grafy pro průměr **X-sloupcový graf** (z anglického X-bar chart) a **vzdálenostní (R) graf** se používají, pokud je možné provést více měření v krátkém čase a ve stejných podmínkách. Graf zobrazuje chování procesu, jednotlivá měření jsou spojena do skupin u, kterých lze předpokládat změnu jen společné variace. X-sloupcový graf zobrazuje ústředí tendenci, tedy jak se hodnoty měnily mezi skupinami v průběhu času. R graf zobrazuje hodnoty uvnitř skupiny.

Graf **X-sloupcový (X-bar) a S graf** je obdobný jako předchozí graf, s rozdílem tím, že předchozí graf dává korektní výsledky jen pro podskupiny o velikosti 10 a méně, pro větší podskupiny se používá S graf.

Graf **individuální a pohyblivé rozmezí (XmR)** pro spojitá data i diskrétní data, pokud jsou měření déle časově od sebe a nelze tak použít X-graf a R-graf, používá se XmR. Navíc pokud je měřená hodnota diskrétní, může se použít obdobný výpočet jako pro data spojitá.

Graf **individuální a pohyblivé rozmezí mediánu** se používá jako snazší alternativa pro XmR.

Graf **pohyblivého průměru a rozmezí** jsou užitečně v případě, že se snažíme hlavně sledovat trend výkonu procesu namísto rozdílnosti hodnot v jednotlivém měření.

C graf a U graf jsou nejpoužívanější v softwarovém měření, používají se především pro měření počtu chyb, počtu řádků kódů a podobně.

2.2.5 Zlepšení procesu

Poté jak máme hotovou definici procesů, definované procesy máme změřené a měření je zpracované třeba formou grafu, přicházíme k poslednímu kroku a to prověřit výsledky a zhodnotit kde lze vytvořit jakékoliv zefektivnění procesu.

Základní tři kroky k vylepšení procesu jsou odstranění nedostatků, změna procesu a stále zlepšování.

1. Odstranění nedostatků, pokud proces je nestabilní, případně se blíží nestabilitě, je potřeba identifikovat příčinu nestability a podniknout kroky k prevenci této nestability.
2. Pokud je proces stabilní, ale není schopný splňovat nároky společnosti nebo zákazníka, je důležité vydefinovat, navrhnout a implantovat nezbytné změny, které změní proces tak aby nárokům od zákazníka nebo společnosti vyhověl. Podstatné je, že tato změna převádí starý proces do nového procesu a tento nový proces musí být pod kontrolou, tak jak je popsáno v předešlém bodě, před tím než začne splňovat definované požadavky.
3. Pokud je proces stabilní a schopný splňovat požadavky, pak poslední možnost jak proces vylepšit je neustále hledat možnosti jak zlepšit kvalitu a snížit potřebné náklady a čas. Změny, které se v procesu objeví, opět proces transformují do nového procesu a tak jako dřív, je nejdříve třeba se postarat o to, aby byl proces stabilní.

2.3 Reengineering

V této kapitole si popíšeme pojem reengineering (nechávám v anglickém tvaru právě proto, že v odborných českých literaturách se tento pojem používá), co to je, jaký je jeho přínos, tedy proč se s ním setkáváme a v neposlední řadě i návaznost na informační technologie.

Úvodem uvedu definici pocházející od Hammer, Chammpy a je uvedena v literatuře [11]:

„Reengineering znamená zásadní přehodnocení a radikální rekonstrukci podnikových procesů tak, aby bylo dosaženo dramatických zdokonalení z hlediska kritických měřítek výkonnosti, jako jsou náklady, kvalita, služby a rychlost“

V praxi pokud se firma rozhodne k reengineeringu znamená, že se musí od svých současných procesů odloučit a nechat vzniknout procesy nové, sjednotit některé procesy do jediného, některé procesy zrušit. Nejedná se tedy jen o zlepšení stávajících procesů. Vznik této oblasti má za následek začátek devadesátých let a pokles největších ekonomik světa, a nasycení trhu. Dnešní firmy jsou vytlačovány z trhu pomocí tří C (dle Hammera, uvedeno v literatuře [11]):

- Zákazníci (z anglického Customers)
- Konkurence (Competition)
- Změna (Change)

Dříve firmám stačilo vyrobit produkt a kupec se našel sám, pokud se mu něco nelíbilo, mohl se rozhodnout koupit anebo nekoupit. Byl snadno nahraditelný. Dnes se vše točí kolem zákazníka, který si může velmi rychle a levně zjistit potřebné informace a navíc trvá převis nabídky nad poptávkou. Samotný produkt rovněž doznal změn, dříve se vyráběly specializované produkty, dnes společnosti produkují univerzální produkty, tak aby jejich produkt vyhověl požadavkům co největší skupiny zákazníků.

2.3.1 Proč orientace na procesy

Na přelomu osmdesátých a devadesátých let se ukázalo, že starý způsob řízení firem je v nových podmínkách nevyhovující. Firmy nemůžou být řízeny pomocí předem a jasně dané hierarchické struktury, kde každý zaměstnanec má své bezprostřední nadřízené a má přesně určenou pozici, odpovědnost a pravomoci. U hierarchického řízení se předpokládá pevně definovaná struktura činností s neměnnou posloupností činností („technologie práce“), tedy každý z pracovníků je specialista na svůj díl práce a jeho výkon se projevuje velkou efektivností. Ale odděluje řízení od výkonu, tedy v krajním případě ten co řídí, nerozumí tomu, co řídí a naopak ti co vykonávají, nerozumí, proč to dělají. V takovém případě nelze předpokládat pružnost, variantnost postupů ani velkou nahraditelnost pracovníků, jak požaduje současný trh. V extrémním případě se pak může vytratit smysluplnost chování organizace jako celku, jediný ukazatel výsledku je jen finanční zisk či ztráta. V neposlední řadě je to tvořiví přístup pracovníků, v takto popsané organizaci nebude mít pracující dostatečný prostor a ani motivaci cokoliv měnit.

Organizace tedy musí změnit především své fungování, tedy dochází k výrazné redukci mezistupňů řízení na pouhé dvě úrovně, strategické vedení podniku a chování podniku, tedy procesy. Procesy vždy vytváří hodnotu pro zákazníka, nebo jiný proces. Tím je dána i struktura, klíčové jsou ty procesy, které produkují hodnoty pro zákazníka, pro ostatní procesy je potřeba hledat jejich využití v klíčových procesech. Od této hierarchie procesů je odvezena organizační a komunikační struktura, informační systém a technologie. Navíc musí být tyto prvky pružné, tak aby mohli být i samotné procesy pružné a díky tomu lépe reagovali na trh. [12,13]

Vlastnosti procesů, které prošly reengineeringem [11]:

- Několik prací je spojeno do jedné (zobecnění, agregace činností)
- Pracovníci rozhodují (delegace pravomocí)
- Kroky procesu jsou vykonávány v přirozeném sledu (nepřízpůsobovat proces organizaci)
- Procesy mají různá provedení (potřeba postihnout v obecné rovině základní směry proměnlivosti procesu)

- Práce se provádějí tam, kde je to nejrozumnější (není pevná organizace, dochází k optimalizaci)
- Redukují se kontrolní nástroje a opatření (delegace odpovědnosti)
- Minimalizují se smírčí jednání (zaměření se na cíl, rozdělení odpovědnosti)

2.3.2 Úloha informačních technologií

Informační technologie bourají to, co v dřívějších dobách nebylo možné, dnes mají společnosti pobočku na druhé straně světa a není problém ji řídit, jako by to byla budova přes ulici. A je tu víc bariér, které byly zbourány, například složitou práci může dělat jen expert - řešením jsou expertní systémy, informace se vyskytuje jen na jednom místě - řešení sdílené databáze, a našly by se i další příklady například v literaturách [11,12,13].

Úloha informačních systémů není jen ve zbourání bariér, v dnešní době jsou k dispozici ERP systémy (z anglického Enterprise Resource Planning, volně přeloženo jako systémy na plánování podnikových zdrojů), které při správné implementaci a použití představují dobrý nástroj na podporu podnikových procesů, naopak pokud jsou nesprávně implementovány a používány dojde jen k upevnění starých procesů a tedy i přetrvávání starých problémů. Zásadní úloha informačních technologií spočívá v tom, že podporují informační procesy, respektive zrychlují, zefektivňují a brání vzniku chyb v procesech. [12,13]

3 Process Inspector

Proces inspektor je nástroj, který má usnadnit analýzu a definici procesů ve společnosti, je určen pro konzultanty i pro uživatele. Používá se při implementaci nového informačního systému ve společnostech. Podrobněji si tento nástroj popíšeme v této kapitole, také si zde představíme aktuální verzi, která se používá ve společnosti Allium.

3.1 Získávání znalostí o procesech

Nejdříve se podíváme na teoretický základ a důvody proč vznikl a je rozvíjen nástroj Process Inspector.

Informační systém (dále jen IS), který odráží podnikové procesy, je potřeba stále zdokonalovat tak aby pomáhal dosahovat cíle společnosti. Proto je potřeba mít stále zmapované podnikové procesy. Této činnosti se účastní zpravidla 2 týmy lidí, tedy dodavatel a uživatel. Jejich spolupráce často vykazuje následující klíčové nedostatky:

1. Informace prezentované jednou stranou jsou bez odpovídajících znalostí pro druhou stranu pouze data, jejichž pomocí druhá strana znovu získává znalosti, které měly být předány – to je velmi neefektivní postup.
2. Nevyslovené předpoklady typu „to se rozumí, že ..., o tom není třeba hovořit“ se nejčastěji projevují právě tam, kde dochází k výměně nejdůležitějších zkušeností a nejzajímavějších principů fungování organizace uživatele.
3. Nekompatibilní struktura informací umožňuje interpretovat slovní označení jedné strany jako úplně jiné pojmy ze slovníku druhé strany (rozdílný sémantický obsah)

A právě uživatel je ten kdo ví, v čem ho současný informační systém (dále jen IS) brzdí a co mu v systému chybí, prvotní úvahy o rozvoji IS lze tedy získat od jejich uživatelů. Důležitá je ale efektivita, se kterou se budou tyto znalosti získávat. Nejpodrobnější výsledky by přinesl osobní kontakt, ale ten mnohdy není z časových a tedy i finančních možností možný a navíc mnohdy jsou opomenuty nejdůležitější body, popřípadě je těm důležitým krokům kladena daleko nižší důležitost. Je proto potřeba najít jiný způsob získání znalostí od uživatelů, tak aby neutrpěla kvalita těchto znalostí.

Navíc uživatelé mají většinou požadavky na rozšíření funkčnosti systému, tedy nové funkce v systému, větší automaticnost systému, v takovém případě lze systém rozvíjet z kvantitativního hlediska. Největší potenciál změny se ale zpravidla skrývá ve změně podnikových procesů, tedy změny kvalitativní, na tuto znalost ale uživatelé sami téměř neukážou. K tomu účelu slouží procesní charakteristika společnosti, tedy popis vnitrofiremních procesů. K popisu procesů lze použít tzv. konceptuální model, který obvykle popisuje dokumenty, se kterými ve společnosti pracují, a tedy

práci s nimi musí podporovat i IS. Konceptuální model znázorňuje propojení jednotlivých procesů přes jejich vstupní a výstupní dokumenty s ohledem na jejich práci v IS.

Nejdůležitější na celém procesu získávání znalostí je efektivnost postupu, kterým se znalosti získávají, protože čím je postup efektivnější tím je i levnější a díky tomu je i výsledný IS levnější. Důležitým faktorem při popisu procesů je jeho podrobnost. Nejprve je proto vhodné identifikovat ty části popisu procesů, které nesou nejvyšší informační hodnotu a lze je nejsnáze využít při následné optimalizaci procesů. Současně je při optimalizaci podnikových procesů žádoucí hodnotit efektivnost jednotlivých optimalizací. Mírou pro takové hodnocení je přínos, který optimalizace procesu poskytne k celkovému fungování společnosti. Optimalizace by se tedy přednostně měla zaměřit na takové procesy, které nejvýznamněji ovlivňují dosahování cíle společnosti.

Aby byl popis vnitrofiremních procesů úspěšný, je potřeba brát ohled na dva důležité faktory, popis stávajících procesů musí zachytit dostatečně podrobně jednotlivé procesy, ale zároveň se nezabývat nedůležitými podrobnostmi. Druhým faktorem pak je výběr takových procesů k optimalizaci, které nejlépe podpoří dosahování cíle společnosti. A právě na první faktor sběru se používá aplikace Process Inspector. [9,10]

3.2 Základní koncepce

Tak jak je popsáno výše většinou se procesy ve společnosti popisují na základě osobního kontaktu s účastníkem procesu, který proces popíše. Pokud ale stejný proces má popsat jiný účastník procesu, vznikne jiný popis téhož procesu a to nejen kvůli víceznačnosti jazyka ale také proto, že každý z účastníků popisuje proces z vlastního pohledu. Navíc mezi popisy procesů, lze najít procesy, které jsou, známé z předchozích projektů, nemusí být zřejmé z ústního popisu účastníka procesu, ale měli by být zřejmé z formálního popisu, který tvoří konzultant.

Při popisu procesů je důležitá i vazba mezi procesy. Procesy jsou provázány pomocí dokumentů, výstupní dokument jednoho procesu je vstupním dokumentem jiného procesu. Důležitou součástí analýzy procesů je tak i analýza dokumentů. A stejně jako u procesů je i u dokumentů problém dvojího označení stejného dokumentu a je na konzultantovi aby toto názvosloví sjednotil. Ze zkušeností vyplývá, že pouze 28% vstupních dokumentů, lze najít mezi výstupními dokumenty jiných procesů.

Nabízí se možnost použít nástroje, které by usnadnily popis procesů a zároveň vedly účastníka procesu (tedy zákazníka) k formálnějšímu jazyku, tedy aby popis procesů byl formálnější a strukturovaný.

3.2.1 Základ práce s nástrojem Process Inspector

Aplikace umožňuje identifikaci jednotlivých uživatelů proto, aby bylo možné jednoznačně určit autora procesu, a také aby bylo možné určit konzultanta, který je zodpovědný za popis procesu.

V aplikaci je možné popsat proces včetně popisu četnosti jeho provádění, typu procesu, oblasti, do které spadá a podobně, rozhraní k zadání procesu je na obrázku 5.

Obrázek 5: Základní údaje o procesu

Následně jsou definovány vstupní a výstupní dokumenty a to tak, že pracovník přiřadí procesu nejdříve jeho výstupy a poté z dostupných dokumentů vybere vstupní dokument, pokud takový dokument nenalezne tak vloží nový dokument, který přiřadí jako vstupní. Právě tato funkčnost pomáhá řešit problém různých názvů stejného. Dokumenty, které nemají výstupní proces, jsou označené jako chybějící, to ale nutně nemusí znamenat, že popis procesů je nekompletní, protože některé dokumenty z této skupiny jsou vytvářeny vnějšími procesy (například dodavatelé, zákazníci) anebo jsou výstupními dokumenty procesů, které ještě nebyly analyzovány. Definici vstupních a výstupních dokumentů ukazuje obrázek 8. Každopádně takovýto popis dokumentů, umožňuje rychle identifikovat nedostatek v analýze procesů a má za následek, že průnik množin vstupních a výstupních dokumentů tvoří 75% všech dokumentů, oproti 28%, které jsou uvedeny dříve.

Název	Popis	Pracovník
Požadavek na změnu	Dokument formulující požadavek zákazníka nebo dodavatele na změnu v zadání úkolu.	Zdenek Oprsal

Název	Popis	Pracovník
Specifikace požadavku na změnu	Dokument přesně popisuje zadání požadavku na změnu zadání úkolu.	Zdenek Oprsal

Název procesu	Popis procesu
Návrh provedení změny	Na základě analýzy požadavku je proveden návrh změny zadání úkolu.

Obrázek 6: Vstupy a výstupy procesu

Procesy lze logicky propojovat také hierarchicky do stromové struktury. Hierarchickou strukturu lze chápat jako rozpad procesů na podprocesy, tedy zpřesňování jeho popisu, naproti tomu

propojení pomocí vstupních a výstupních dokumentů může jít napříč stromovou strukturou a odpovídá skutečným provázáním procesů.

Součástí definice procesu je určení jeho typu. Ke každému typu se váží otázky. Na otázky odpovídá pracovník slovně. Na základě odpovědí je možné proces přesněji zařadit a případně vyhodnotit jeho důležitost z hlediska celého podniku. Jedná se o doplňující informace, které rozšiřují popis procesu. Tyto informace jsou dále použity při analýze procesů vhodných k optimalizaci.

Funkcí, která ulehčí následnou formalizaci procesů, je slovník. Slovník obsahuje formální pojmy s definovaným významem, které byly použity v předešlých projektech. Pojmů ve slovníku je omezený počet, pracovník při definici procesu navíc vybírá pojem ze slovníku, který nejlépe vystihuje proces. Tento postup má pomoci k formálnějšímu popisu procesů a ulehčuje tak práci konzultanta, pokud pracovník nenalezne odpovídající pojem ve slovníku je to signálem, aby kontaktoval konzultanta a našel s ním řešení a to buď, že společnými silami najdou odpovídající pojem ze slovníku anebo se jedná o proces, který je specifický pro konkrétní společnost. V každém případě se můžou opřít o znalosti z předchozích projektů, díky čemuž dochází k eliminaci nedorozumění mezi konzultantem a pracovníkem. K slovníku se navíc váží otázky s předdefinovanou množinou odpovědí, které daný slovníkový pojem zpřesňují a je tedy možno mít pod jedním pojmem více slovníkových pojmů, které by se odlišovaly jen v maličkostech. Pokud pracovník není schopen na otázky zodpovědět, značí to, že vybral nevhodný pojem ze slovníku.

The screenshot shows a web application interface. At the top, there are navigation tabs: 'Odhlásit se', 'Nový proces', 'Nový výstup', 'Připojit vstup', and 'Uložit proces'. Below this, the interface is split into two main panels. The left panel, titled 'Seznam', contains a list of processes under the heading 'Chybějící dokumenty'. It has a filter section with fields for 'Identifikace', 'Typ' (set to 'Vše'), and 'Oblast' (set to 'Vše'). Below the filter, there are two items: 'Analýza požadavku' and 'Vstupy' (with a 'missed' status). The right panel, titled 'Editace/Přidávání procesu', shows the 'Slovník' tab. It displays a process entry: 'Položka odpovídá výrazu "Zachycení požadavku"'. Below this is a 'Popis' (description) and a list of 'Otázky' (questions) with dropdown menus for answers. The questions are: 'Mohou požadavky přicházet elektronickou poštou?' (answered 'Ano') and 'Je třeba potvrzovat přijetí požadavku?' (answered 'Ne').

Obrázek 7: Slovník, odpovědi na otázky

3.3 Historie

Po části věnované funkční stránce aplikace se v krátkosti podíváme na to, jak se aplikace vyvíjela a jak vypadá její současný stav.

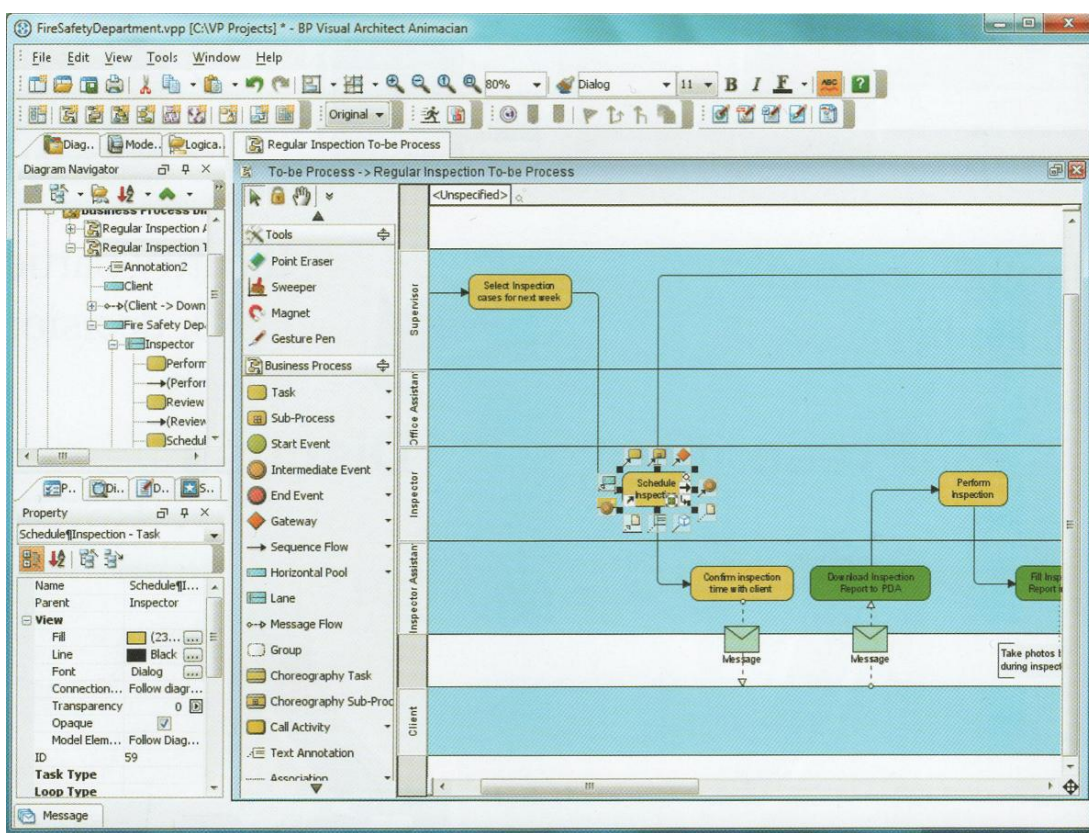
Aplikace Process Inspector byla vyvíjena v jazyce PHP a používala databázi MySQL. Tato aplikace byla použita pro přibližně 5 projektů. V roce 2006 se začala přepisovat do asp.Net a databázového serveru Microsoft SQL Server, celou práci v roce 2007 obhájil Michal Rozsypálek

jako svou Diplomovou práci. Bohužel jeho aplikace nakonec nepřinesla očekávaný efekt a proto je stále používána aplikace psaná v PHP, která od té původní verze doznala několik nepatrných změn převážně v uživatelském rozhraní.

Největší problém obou aplikací byl to, že jakákoliv změna, jako přidání sloupce v tabulkách, jiné zobrazení stránky vyvolala programové úpravy, které analýzu u specifického zákazníka prodražily a hlavně prodloužily, protože se konzultant s uživatelem k tématu museli vracet později. Tento jev by měla vyřešit technologie SharePoint.

3.4 Konkurenční produkty

Mezi produkty na analýzu a modelování procesů lze zařadit produkty například Enterprise Architect od společnosti Sparx Systems, Business Process Visual Architect od Visual Paradigm a Inubit Suite od Inubit. Všechny nástroje mají jedno společné, slouží pro konzultanty a analytiky, kteří na základě komunikace se zákazníkem, získají data, která použijí na vytvoření modelu procesů. Všechny tyto nástroje jsou graficky a uživatelsky příjemné a snadno ovladatelné. Ze základních stavebních bloků, které se mezi sebou propojují, vzniká model procesů, bohužel běžný uživatel (tedy zákazník) není schopen s takovýmto nástrojem pracovat. Na obrázku 9 je ukázka aplikace Business Process Visual Architect, ostatní jmenované aplikace vypadají obdobně.



Obrázek 8: Aplikace Business Process Visual Architect

A Proto obrovskou výhodou Process Inspectoru je jeho užití, aplikace slouží nejen konzultantům, kteří procesy na základě analýzy popisují. Slouží i běžným uživatelům (pracovníkům ve společnosti), kteří pomocí ní procesy popisují, opakovaně upřesňují a rozvíjejí (např.: podrobnými popisy pracovních míst a popisy postupů – návody, nebo požadavky na změny a úpravy nástrojů). A díky exportům bude možné provést export do výše popsaných nástrojů.

4 SharePoint 2010

V této kapitole se podíváme na produkt Microsoft SharePoint 2010 Server, na kterém bude aplikace vyvíjena. Také zde budou představeny i další produkty, které byly při vývoji použity anebo úzce souvisí s platformou SharePoint. V neposlední řadě budou popsány obecné principy úprav a vývoje v prostředí SharePoint.

4.1 Představení aplikace SharePoint

SharePoint je webová aplikace. Typické použití pro ni je správa obsahu a správa dokumentů. Je navržena jako náhrada více webových aplikací a tak aby podporovala rozšiřitelnost o požadavky uživatelů.



Obrázek 9: SharePoint [18]

Jak ukazuje obrázek 9, SharePoint podporuje 6 okruhů a o další funkcionalitu lze dále rozšířit.

- **Internetový portál** – poskytuje na role orientovaný přístup k informacím, lidem, k tomu slouží nástroje jako moje stránka, sociální síť, šablony firemního portálu.
- **Obchodní formuláře** (z anglického Business Forms) - možnost získávat důležitá podniková data a automatizovat procesy.
- **Spolupráce** - nabízí jednoduchou spolupráci různých typů aplikací nebo funkcí, jako mohou být Microsoft Outlook, listy, úkoly, kalendář, wiki stránky, projekty a další.
- **Business Intelligence** (nechávám v anglickém tvaru v české odborné literatuře se tento název používá) – dělat informovaná rozhodnutí na základě znalosti obchodních dat, k tomu účelu slouží reporty, Excel na straně serveru, vizualizace.

- **Prohledávání** – tak aby výsledky prohledávání byly rychle a systematicky přístupné.
- **Správa obsahu** – správa celého životního cyklu obsahu, jako je správa dat, správa dokumentů, správa webu se spoluprací s workflow, politikou pro správu.

SharePoint umožňuje na jednom serveru provozovat weby (prostory, z anglického site) pro více společností, každá ze společností pak může mít definovaných i více webů na kterých jsou uložena data. V terminologii SharePoint se jedná o kolekci prostorů (z anglického site) a každá z nich pak obsahuje kolekci webů.

Část grafického uživatelského rozhraní s funkčními tlačítky se jmenuje Ribbon a je obdobné jako v řadě MS Office 2010, pomocí něho lze provádět úkony podle kontextu aktuální stránky. Obecně poskytuje možnos:

- Kopírovat, vytvářet, mazat a přejmenovávat seznamy a knihovny, stránky, weby, a webové komponenty
- Správu uživatelských oprávnění a zobrazit historii verzí dokumentu nebo stránky
- Správu definice a vlastnosti seznamů a knihoven, stránek, lokalit, a webových komponent
- Manipulovat s obsahem v seznamech, knihovnách, na stránkách a webech.

4.1.1 Edice v SharePoint

Microsoft SharePoint 2010 se nabízí v několika edicích, základní edice je SharePoint 2010 Foundation, která je volně ke stažení k Windows Server 2008, nabízí základní funkčnost a většinou se používá jako platforma pro vývoj aplikací třetích stran. Oproti tomu SharePoint Server 2010, je plně placená verze, která obsahuje několik modulů, rozlišeny jsou tu dvě verze a to Standart nebo Enterprise. Verze Enterprise obsahuje vše co Standart a navíc další rozšiřující moduly. Co přesně každá z edic nabízí je k dispozici v literatuře [7]. Aby to ale s edicemi nebylo tak lehké, tak Microsoft změnil od verze SharePoint 2007 názvosloví, SharePoint 2010 Foundation se ve verzi 2007 jmenuje SharePoint 2007 Services.

4.2 Základní prvky

V této kapitole se zaměříme na specifika aplikace SharePoint. Vysvětlíme zde pouze ty prvky, které jsou potřeba k pochopení realizovaného řešení. Více informací lze například získat v literatuře [7,14].

4.2.1 Listy a knihovny

Listy a knihovny jsou základním stavebním kamenem SharePoint. Jaký je mezi nimi ale rozdíl? Na první pohled téměř žádný, oba typy dokážou obsahovat jak dokumenty, tak i volitelná pole (atributy). Důležité je jak výslednou strukturu chápat.

Knihovna pracuje s dokumenty a dokáže z nich vytvářet stromovou strukturu včetně složek, obdobnou jako operační systém, ale oproti němu má jednu obrovskou výhodu a to jsou atributy. V SharePoint je možné definovat pole, které lze vytáhnout do knihovny jako další sloupec. Tato pole mohou být navíc editovaná i v kancelářském balíčku Microsoft Office, popřípadě přímo zadána v dokumentu. Díky tomu lze například udělat knihovnu objednávek a do knihovny vytáhnout z dokumentu nejen to kdo objednávku vytvořil a kdy, ale i například částku na kterou objednávka je.

Oproti knihovně lze zjednodušeně list chápat jako tabulku z relační databáze, u které se definují atributy, a také relace mezi listy. Navíc ke každému řádku lze také připojit soubor, ale to pouze ve formě přílohy. A podobně jako v seznamech lze data hierarchicky vkládat do složek. Nad listy je možné definovat pohledy, které zobrazují jen část sloupců a popřípadě jen položky, které splňují filtry, které jsou v pohledu nastaveny.

4.2.2 Webové stránky

Někdy také pojmenované jako wiki stránky, jsou HTML stránky, které lze jednoduše editovat pomocí WYSIWYG editoru, nové ve verzi 2010 je i možnost vkládat webové komponenty přímo do textu, tedy není potřeba mít webovou komponentu na editaci textu a kolem ní poskládané ostatní webové komponenty, tak jak je to obvyklé ve většině CMS aplikací. Tento přístup je alespoň z mého pohledu hodnocen velmi kladně díky tomu, že daleko víc podporuje logiku, jak stránku chápou uživatelé. Navíc stránky umožňují vkládat odkazy na neexistující stránky (podobně jako například na známém serveru wikipedia.org), pokud uživatel na takový odkaz klikne je vyzván k vytvoření stránky. Samozřejmostí jsou grafické šablony, do kterých se stránka vypisuje a díky tomu tvoří ucelenou grafickou prezentaci.

4.3 Správa SharePoint

O správě SharePoint by se dalo napsat hodně, je tomu věnována kniha [14]. Zde je vybráno jen to, co je nezbytné pro tuto diplomovou práci.

4.3.1 Instalace SharePoint

V první řadě se zde podíváme, co je potřeba před tím, než je samotná instalace spuštěna.

První ze všeho jsou hardwarové požadavky na server, minimum, se kterým musíme počítat je 8GB operační paměti, pro vývojáře mají menší požadavek jen na 4GB paměti. Další požadavky jsou 80GB volného místa na disku a k tomu 64bitový procesor se čtyřmi jádry, nejlépe s osmi. Softwarové požadavky jsou pouze 64bitový operační systém kde minimum je Windows Server 2008 R2 (případně Windows Server 2008 SP1, který automaticky bude převeden na SP2), 64bitový Microsoft SQL Server 2008 R2 (nebo SQL Server 2008 SP1 kde je potřeba doinstalovat ještě některé opravy) a vyšší

verze, v případě vývojáře se dá SharePoint rozjet i na Windows 7 a i Windows Vista. Zbytek aplikací už dokáže instalační průvodce doinstalovat sám. Sama instalace je pak jednoduchá, stačí ověřit, že systém má všechny předpoklady, popřípadě předpoklady splnit. Poté pomocí průvodce spustit instalaci, která trvá kolem hodiny.

S hardwarovými požadavky mám tu zkušenost, že pro vývojáře stačí i něco kolem 2,5GB a procesor se dvěma jádry. Diplomovou práci jsem začal dělat na virtuálním stroji, a dokud nebylo v databázi hodně dat nebo se neprováděli složité výpočty, tak SharePoint měl odezvu v přijatelném čase. Další poznámka, SharePoint si udělá i vlastní instanci SQL serveru, většinou běží na (local)\SHAREPOINT.

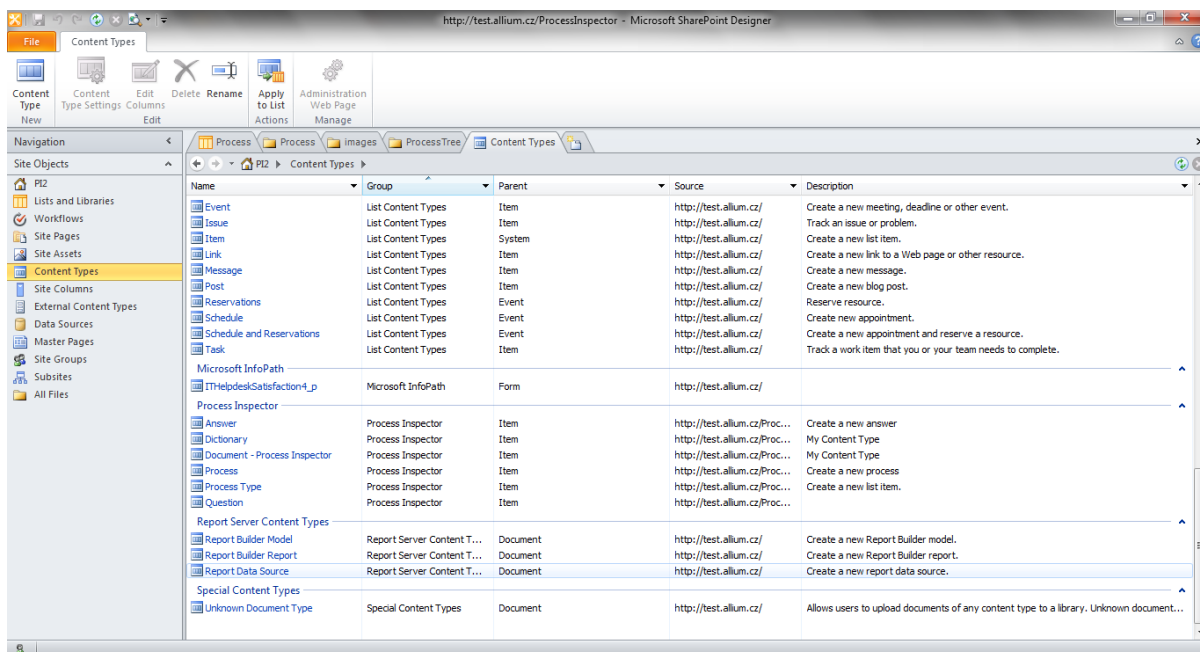
Problematická je instalace SharePoint na virtuální počítač tak, aby byl jednoduše přenositelný anebo proto, že není k dispozici 64bitový procesor. Microsoft Virtual PC vůbec nedovolí založit 64bitový počítač, i přes to, že máte 64bitový procesor. Další produkt na virtualizaci je Hyper-V server, ten ale potřebuje Microsoft Server 2008 a ten taky kde kdo na svém počítači nemá. Je možné použít konkurenční varianty a to buď VMware, který dokáže udělat z 32bitového procesoru 64bitový, anebo VirtualBox, pro který jsem se rozhodl já, protože se dá jednoduše spustit i na běžném PC a celkově na mě nepůsobil složitě, přesto šlo v něm nastavit vše, co jsem potřeboval.

4.3.2 Programy pro správu

SharePoint nabízí sám o sobě dostatečné nástroje, jak upravovat jednotlivé moduly pro běžné uživatele, pokud ale chceme detailnější, případně specifičtější nastavení, je potřeba se podívat i po ostatních nástrojích.

4.3.2.1 SharePoint Designer

Oproti nastavení v SharePointu, je Designer daleko detailnější co se týče nastavení, vše ale stejně jako v SharePointu je pomocí průvodců, formulářů a nemusí se psát žádný kód. Jedinou výjimkou jsou editace stránek, kde se samozřejmě otevře HTML editor a je možné upravovat stránky pomocí HTML. Je zde přístupné i nastavení, ke kterému se přes aplikaci SharePoint nelze dostat. Ukázka aplikace je na obrázku 11.



Obrázek 10: Ukázka aplikace SharePoint Designer

4.3.2.2 Visual Studio 2010

Ve Visual Studiu je úplná kontrola nad definicemi prvků v SharePoint, vše lze definovat pomocí souboru v XML struktúře, každý prvek, ať už se jedná o definici listu, definici stránky a další. Každý prvek má jasně danou XML strukturu a v literatuře lze najít, co jaký atribut v XML struktúře dělá. Ukázky jak co vytvořit lze najít v kapitole 6, Implementace.

4.3.2.3 Office InfoPath 2010

SharePoint je propojen i s produktem InfoPath z řady Microsoft Office 2010, je možné upravovat formuláře, ať už se jedná o formulář editace položky v listu anebo definovat vlastní formulář, který bude data sbírat.

4.4 Technologie asp.Net

Microsoft .Net je platforma na vytváření aplikací ať už pro desktopy (Windows), webové aplikace, nebo například pro mobilní zařízení. Základem je .Net Framework, který zajišťuje potřebné prostředí pro běh aplikací. .Net Framework je postaven na CLR (z anglického Common Language Runtime), který překládá kód do CIL (z anglického Common Intermediate Language), poté jak je kód spuštěn tak dochází k překladu z CIL do nativního jazyku operačního systému. Díky tomu, je kód psaný v .Net přenositelný mezi různými platformami.

Asp.Net je pak rozhraní pro vytváření webových aplikací. Pro vývoj aplikací je možné použít libovolný jazyk z rodiny .Net. Výhodou Asp.Net je, že se nejedná o skriptovací jazyk, tedy nemusí být vždy skripty rozpoznávány a znovu překládány, díky tomu, je asp.net rychlejší jak například PHP.

Nejnovější verze .Net Frameworku je verze 4, která byla uvedena v roce 2010, pro SharePoint 2010 je potřeba alespoň verze 3.5 SP 1.

4.5 Přístup k datům

SharePoint nabízí několik možností jak přistupovat k datům, která jsou uložena v listech. Jsou zde uvedeny jen ty nejdůležitější, která jsou dále použity v této práci, komplexnější popis je k nalezení v literatuře [4].

4.5.1 CAML

CAML je zkratka anglického Collaborative Application Markup Language, je základní rozhraní k přístupu k datům v SharePoint 2010. Nevýhodou je, že je slabě typované.

K datům se přistupuje přes třídy SPSite, SPWeb, SPList a SPListItem. Samozřejmě, že kromě těchto tříd jsou k dispozici i další třídy pro manipulaci s daty.

- Třída SPSite reprezentuje celý pracovní prostor, tedy skupinu webů
- Třída SPWeb reprezentuje jeden web, který obsahuje listy, knihovny, stránky atd.
- Třída SPList reprezentuje list v SharePoint
- Třída SPListItem reprezentuje řádek v listu

Pokud chceme vybrat název procesu s ID 5 z aktuálního webu, přečtu ho pomocí sekvence příkazů:

```
SPWeb Web = SPContext.Current.Web;  
SPList List = Web.Lists["Process"];  
SPListItem Process = List.Items.GetItemById(5);  
string Title = Process["Title"];
```

Dotazy na výběr konkrétních dat (příkaz where v SQL) je potřeba psát v XML struktuře, struktura je popsána třeba v literatuře [4].

4.5.2 LINQ

LINQ je zkratka anglického Language Integrated Query, jedná se o nástroj jak do zdrojového kódu psát dotazy podobné jako v SQL. Je novinkou v SharePoint 2010, poprvé byl představen ve verzi .Net Framework 3.5 a Microsoft Visual Studio 2008. LINQ je mapující objekt v jazyce .NET. Výhodou je, že můžeme přebírat relace v databázových tabulkách, v případě SharePoint relace mezi listy. Pro to aby LINQ fungoval, potřebujeme poskytovatele, který vezme objekty a přeloží je do nativních příkazů. Pro SharePoint 2010 LINQ překladač mapuje objekty a operace do CAML nativního přístupu k datům.

Pokud chceme používat, LINQ jazyk v kódu, je potřeba vygenerovat třídu reprezentující objekty v SharePoint a tuto třídu přidat k projektu. Poté je nutné k projektu přidat reference na knihovny System.Xml.Linq a Microsoft.SharePoint.Linq.

Třidu vygenerujeme pomocí utility SPMetal.exe, který je většinou ve složce C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\BIN. SPMetal.exe se spouští s parametry:

```
SPMetal /web:http://site /code:path /language:csharp
```

Kde parametr /code, říká kam se výsledný *.cs soubor vygeneruje. Soubor obsahuje třídu s entitami reprezentujícími listy. Parametr /web, určuje web ze SharePoint, ze kterého se mají listy vzít, poslední parametr /language je pro jazyk, který má být použit ve vygenerovaném souboru, v ukázce bude použit jazyk C#.

Jak je vše připraveno, je možné napsat příkaz do kódu jako například vypsání nedokončených dokumentů:

```
ProcessInspectorDataContext dc = new ProcessInspectorDataContext("http://romeo-  
virtual/ProcessInspector");  
var documents = from document in dc.Document  
                where document.Unfinished == true  
                select document;
```

Popřípadě jiná syntaxe, která pracuje pomocí třídy List:

```
SPList list = Web.Lists["Document"];  
List<SPListItem> results = list.Items.OfType<SPListItem>().Where(i=>  
(bool)i["Unfinished"] == true).ToList();
```

Oproti CAML má LINQ jednu výhodu a to, že se jedná o silně typované entity. Navíc se daleko intuitivněji (pro znalé SQL) píší dotazy nad daty a je možné pomocí where psát i složitější dotazy nad více tabulkami.

4.5.3 REST

REST pochází z anglického Representational State Transfer. Jedná se o dotazovací jazyk na straně klienta, který se na data dotazuje přes protokol http, pomocí metody GET se pošle příkaz na výběr dat se syntaxí podobnou jako je URL a SharePoint vrátí data ve struktuře XML. Tato metoda je vhodná pro výběr dat na straně klienta, například v Microsoft Silverlight. Tento typ výběru dat jsem nadále v práci nepoužil, a proto zde nebudu uvádět podrobnosti, ty lze nalézt v literatuře [4,6,7].

4.6 Vývoj

Jedinou možností jak vyvíjet nové funkce pro SharePoint je přes Microsoft Visual Studio 2010. Kde jsou předdefinované projekty, případně typy souborů podle toho, co chceme vytvořit. V této práci jsou použity hned několik typů projektů, které si v krátkosti představíme dále. Více se lze dočíst v literatuře [4].

Nové v SharePoint 2010 je možnost výběru, jak se bude projekt přenášet na server a od toho je pak odvozeno i to co je možné v projektech vytvořit.

- **Farm solution** – nemá žádné omezení instrukcí nebo konstrukcí ve zdrojovém kódu, ale jsou potřeba administrátorská oprávnění
- **Sendbox solution** – má omezení přenesených dat za den, v kódu se nesmějí vyskytovat některé instrukce nebo konstrukce, ale nepotřebuje administrátorská práva

Nejdůležitějším typem procesu co lze vytvořit je **webová komponenta** (z anglického web part), na výběr jsou dvě možnosti a to projekt s názvem *Web part*, který je obdobný jako webová komponenta z asp.Net anebo projekt *Visual Web Part*, který je upravený pro snadnější programování a lze do něj vkládat i více předdefinovaných ovládacích prvků z nabídky SharePoint.

Dalším typem projektu je **event receiver**, pokaždé když dojde ke změně stavu některé položky, ať už se jedná o položku v listu nebo změna stavu ve workflow a další, je vždy generována událost (z anglického Event) a poté může být zpracována právě kódem, který je v tomto typu projektu.

Veškeré prvky v aplikaci SharePoint (tedy například listy, knihovny, Ribbon) jsou definovány pomocí definice ve formátu XML, pokud chceme jakoukoliv definici prvku změnit, vytvoříme prázdný projekt ve Visual Studiu a do něj vložíme XML soubor, kde vytvoříme novou definici, nebo řekneme, jak se má současná definice změnit.

4.7 Exporty a importy

Složitě po diskusních fórech a v literatuře [4,6,7] jsem zjišťoval jak nejen přenášet celý web mezi servery, ale také jak dostat definici webu do Visual Studia tak, abych mohl web modifikovat, jak je potřeba. Proto jsem se rozhodl vložit tuto kapitolu, kde budou popsány základní typy souborů, a jak s nimi zacházet. A poslední poznámka Power shell, který se používá pro správu SharePointu je odlišný od Power Shellu ve Windows je nutný ho spustit z nabídky start a pak ze složky se SharePointem (většinou to je start/všechny programy/Microsoft SharePoint 2010)

4.7.1 Záloha a obnovení

Provádí se záloha (obnovení) pouze celé kolekce webů, jednotlivých webů to nelze. Záloha (obnovení) se provádí přes Power Shell, který se musí spustit s administrátorskými oprávněními, pak lze napsat příkaz `Backup-SPSite` a kolekce webů je vyexportovaná. Ukázka jak takový příkaz může vypadat:

```
Backup-SPSite -Identity http://romeo-virtual/ -Path C:/temp/backup_20110506
```

S obnovením je to obdobné, použije se příkaz `Restore-SPSite`, který má obdobné parametry pouze je přidán parametr `-Force`, který přepíše starý web, opět ukázka:

```
Restore-SPSite -Identity http://site/ -Path C:/temp/backup_20110506 -Force
```

Jak vypadají celé příkazy včetně všech parametrů i s krátkým popisem jak je použít, lze najít v literatuře [5], tam lze také najít možnost zálohy přes centrální administraci SharePoint, která zde není uvedena.

4.7.2 Export a Import řešení webu

Při exportu webu jsou dvě možnosti a to použít centrální administraci nebo Power shell, při importu je jediná možnost a to použití Power shell. O obou možnost se lze více dočíst v literatuře [5]. K přenosu webu je použit soubor s příponou cmp, který obsahuje šablonu webu. Tato šablona může obsahovat i data, která jsou na webu uložena.

4.7.2.1 Export pomocí Centrální administrace SharePointu

V Centrální administraci, která se spustí přes nabídku start ve Windows, se přes odkaz *Backup and Restore* dostaneme na stránku zálohování a obnovení, zde v sekci *Granular Backup* přejdeme na odkaz *Export Site or List* a dál přes jednoduchého průvodce vybereme, jak chceme web exportovat.

4.7.2.2 Použití power shellu

K exportu slouží příkaz `Export-SPWeb`, povinné parametry jsou url webu a cesta kam má být výsledný export uložen, tedy například:

```
Export-SPWeb -Identity http://site -Path C:\Temp\Export.cmp
```

Pro Import webu je obdobný příkaz tedy `Import-SPWeb`, opět jsou stejné parametry a i příkaz vypadá podobně.

```
Import-SPWeb -Identity http://site -Path C:\Temp\Export.cmp
```

Ostatní parametry obou příkazů se lze dočíst z nápovědy příkazu nebo z webu [5].

4.7.3 Import webu do Visual Studio

Pokud chceme upravit definic prvků, které jsou již vytvořené je třeba exportovat web do formátu wsp a ten pak importovat do Microsoft Visual Studio, kde je možné definice opravit.

4.7.3.1 Export wsp souboru

Oproti předchozím exportům, zde není možnost exportu pomocí Power shellu, ale jen přes SharePoint. Export probíhá ze dvou kroků vytvoření šablony a samotný export.

Na webu, který chceme exportovat, vybereme *Site Action* v horním menu vybereme možnost *Site settings*, poté na další stránce ve skupině *Site Actions*, je možnost *Save site as template*, tu vybereme a vyplníme formulář jak se má šablona jmenovat.

Pokud už máme šablonu vytvořenou tak v *Sollution Galery* (url: `~/_catalogs/solutions/Forms/AllItems.aspx`) vybereme šablonu, kterou chceme exportovat, klikneme na jméno šablony a SharePoint nám nabídne kam se má wsp soubor uložit.

4.7.3.2 Import wsp do Visual Studio 2010

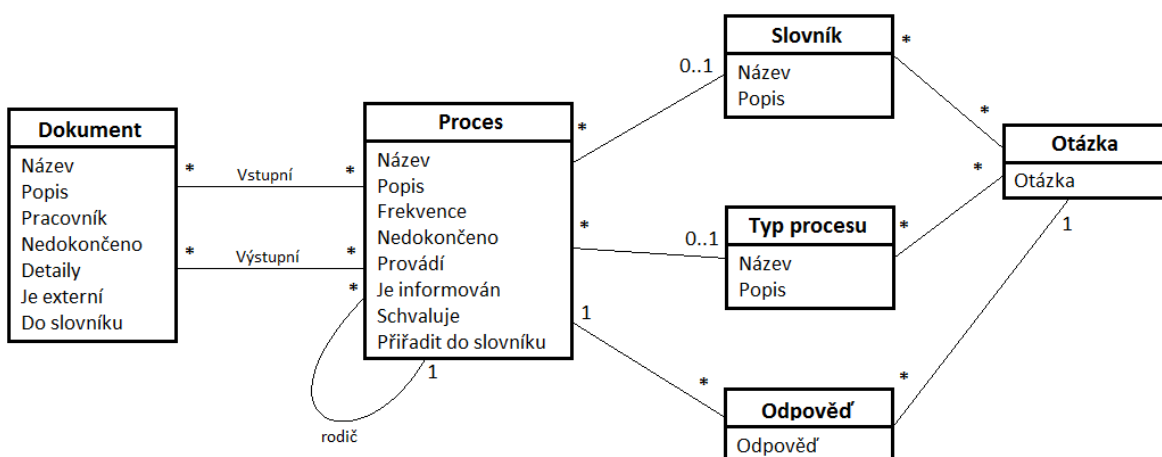
Důležité je pustit Microsoft Visual Studio jako administrátor. K importu wsp souboru slouží projekt s názvem *Import SharePoint Solution Package*. Po založení takového projektu budeme vyzváni k zadání adresy webu, ke kterému se projekt vztahuje a také k výběru wsp souboru z disku. Tímto dostaneme celou definici webu do Visual Studia a můžeme změnit co je potřeba.

5 Návrh systému

Při návrhu systému jsem vycházel z aplikace, kterou v současnosti používají ve firmě Allium, s.r.o. a je popsána v konceptu v kapitole 3 a ze závěru, které jsou k dispozici v diplomové práci Michala Rozsypálka [8]. Je třeba navrhnout takové řešení, které bude co nejvíce používat funkcionalitu, která je již v aplikaci SharePoint implementována, proto je nezbytná znalost celé aplikace SharePoint před samotným návrhem.

5.1 Datová struktura

Na obrázku 12 je uveden ER diagram mezi listy v SharePoint. Tento diagram zobrazuje databázi jak jí lze vidět přes SharePoint pomocí listů. Skutečná databáze na Microsoft SQL serveru má jinou strukturu, SharePoint díky své pevné datové struktuře uchovává veškerá data v jediné tabulce a do pomocných tabulek si ukládá relace mezi entitami.



Obrázek 11: ER diagram

V tomto diagramu ani níže neuvádím standardní pole listů, jako jsou informace o vytvoření, editaci, přílohy a další. U těchto polí nelze nijak upravit definici, z velké části jsou pod správou SharePoint, který si je sám plní a upravuje.

Velkou výhodou SharePoint je, že uživatelé si mohou do listů jednoduše přidávat nová pole (listem se rozumí struktura podobná relační tabulce), samozřejmě jen tehdy pokud na to mají oprávnění, tedy výsledná struktura se v praxi může ještě rozšířit, třeba i o různá pole pro různé projekty.

5.1.1 List procesů

Základní list celé aplikace obsahuje procesy. Na základě analýzy byly do návrhu přidány na rozdíl od původní aplikace pole: provádí, je informován, schvaluje, přiřadit do slovníku. Výsledný list procesů obsahuje tato pole:

- Název – jednořádkový název procesu
- Popis – víceřádkové popis procesu
- Frekvence – výběr z možností jak často se proces opakuje
- Nedokončeno – zaškrtačací políčko, určuje, zdali proces potřebuje revizi
- Provádí – identifikace pracovníka nebo role, která daný proces vykonává
- Je informován – identifikace pracovníka nebo role, která získává zprávy o stavu procesu
- Schvaluje – identifikace kompetentního pracovníka nebo role, která výsledek procesu osvědčuje
- Rodičovský proces – výběr rodičovského procesu z existujících procesů, toto pole slouží na vytváření hierarchické struktury procesů
- Výstupní dokument – množina dokumentů, které vystupují z procesu
- Vstupní dokument – množina dokumentů, které vstupují do procesu
- Typ procesu – výběr jednoho typu z listu typů procesu
- Přiřadit do slovníku – zaškrtačací políčko, má-li být proces zařazen do slovníku

5.1.1.1 Vkládání a editace procesů

Při vkládání nebo editaci procesů, pokud dojde ke změně slovníkového pojmu, je třeba k procesu vložit otázky s odpověďmi, které patří ke slovníku. Staré otázky se smažou, kromě těch, které už obsahují vyplněné odpovědi, proto aby se odpovědi, které uživatel vyplnil se neztratily a mohly být použity i nadále.

Pro snazší pořízení celé struktury je ergonomicky upraveno uživatelské rozhraní tak, aby mohl uživatel rychle vložit subprocess nebo proces na stejné úrovni stromu – tedy bratrský. Tato ergonomická změna byla vyžádána kompletně jinou ergonomií práce v SharePoint na rozdíl od původní aplikace v PHP.

5.1.2 List dokument

Obsahuje vstupní a výstupní dokumenty, které slouží k propojení procesů. Opět z analýzy byly přidány nová pole: je externí, přiřadit do slovníku. Dokument obsahuje tato pole:

- Název – jednořádkový název procesu
- Popis – víceřádkové popis procesu

- Pracovník – určuje konzultanta, který spolupracuje na definici procesu, do tohoto pole je možné osoby vyhledat z uživatelů SharePoint
- Detail – více řádková text, detailní popis dokumentu
- Nedokončeno – zaškrťovací políčko, určuje, jestli dokument potřebuje revizi
- Je externí – zaškrťovací políčko, říká, jedná-li se o externí dokument, pokud je zatrženo, není tento dokument vytvářen vnitřním (modelovaným) procesem a nejedná se v důsledku ani o chybějící dokument
- Přiřadit do slovníku – zaškrťovací políčko, má-li být dokument zařazen do slovníku

5.1.2.1 Mazání dokumentů

Pokud je dokument přiřazen k libovolnému procesu jako vstupní nebo výstupní dokument, není možnost tento dokument smazat tak, aby nevznikaly nekonzistence popisu procesů.

5.1.3 List se slovníkem

Tento list slouží k udržování slovníkových pojmů, které se poté přiřazují procesu, ke každému slovníkovému pojmu se váže skupina otázek. Slovníkové pojmy by měli definovat konzultanti na základě znalostí z dřívějších projektů. Obsahuje tato pole:

- Název – jednořádkový název
- Popis – víceřádkový popis
- Otázky – výběr množiny otázek z listu otázek

5.1.3.1 Editace slovníku

Pokud dojde ke změně otázek, které jsou definované u slovníku, je potřeba tyto změny přenést i k procesům tak, aby byla databáze konzistentní.

5.1.4 List otázek

Otázka obsahuje jediné pole a to samotná jednořádková otázka.

Na listu otázek, je potřeba zakázat mazání a editaci jestliže otázka je použita v odpovědích. Tak aby nedocházelo k nekonzistenci mezi otázkou a odpovědí.

5.1.5 List odpovědí

Odpověď obsahuje tato pole:

- Proces – výběr procesu, ke kterému se odpověď vztahuje
- Otázka – výběr otázky z definic otázek
- Odpověď – víceřádková odpověď na otázku

List odpovědí je plněn automaticky pomocí typu procesu a slovníku, který vždy založí prázdnou odpověď k danému procesu a otázce. Aby nevznikala duplicitní odpověď je potřeba kontrolovat, jestli už proces neobsahuje otázku, která se právě vkládá. Editaci jednotlivých položek je možné z editačního formuláře procesu, proto není potřeba, aby tento list byl v rychlé navigaci na webu.

5.1.6 List typů procesu

Posledním listem je list s typy procesu. Typ procesu klasifikuje proces z hlediska objektu i subjektu procesu – tedy kdo a co zpracovává. Tento list obsahuje následující pole:

- Název – jednořádkový textový název, který určuje typ procesu
- Popis – víceřádkový popis
- Otázky – výběr množiny otázek z listu otázek

Dílejší upřesňující otázky uvedené výše, mohou být definovány a jsou přiřazovány právě na základě typu procesu.

5.2 Uživatelské rozhraní

V této kapitole se podíváme na návrh uživatelského rozhraní, kromě toho, že zde budu nadále řešit jak a kam se má jaký prvek zobrazovat. Je nutné dávat důraz i na ergonomii ovládání, tak aby se ovládání co nejvíce usnadnilo a také aby se celá koncepce ovládání nerozcházel s konceptem pospaným v kapitole 3.

5.2.1 Úvodní stránka

V původní aplikaci byla úvodní stránka po nalogování prázdná, až na seznam kořenových procesů, je třeba využít potenciál toho, že tuto stránku uvidí každý, kdo se na web přihlásí. A přinést mu informace, které uživatele zajímají.

Úvodní stránka je to co uživatel uvidí poprvé, když na web vstoupí, je proto nutné mu podat základní informace o tom kde se zrovna nachází a co má dělat, popřípadě takového uživatele přesměrovat na stránku s nápovědou tak, aby se dokázal co nejrychleji zorientovat. Na druhé straně je toto stránka, na kterou přichází uživatelé, kteří už s aplikací mají zkušenosti a mají v nich už nějaké data zadané, ti naopak potřebují vědět, co se změnilo a nejlépe co je potřeba dále udělat.

Konečný návrh zobrazuje obrázek 12, stránka je rozdělena na 2 sloupce a dále je rozdělena na 3 menší zóny (červeně ohraničené). Každá ze zón je určena nějakému typu uživatelů a podle toho se v nich budou zobrazovat prvky.

Process Inspector

Začněte vytvořením nového procesu pomocí odkazu New process v záhlaví stránky. Nápověda k aplikaci je k dispozici [zde](#).

Neúplné procesy

Tyto procesy nemají definované vstupní nebo výstupní dokumenty, zkontrolujte prosím jejich definic.

Title	Input document	Output document	Parent process	Dictionary
P4 - proces	Dokument 266			asdasd
Testa			P10 - proces	
xxx	Dokument 1		Proces 12.5	asdasd
ZOP P16		Dokument 10; Dokument 103	P16 - proces	
zop8	Dokument 103			

Chybějící dokumenty

Tyto dokumenty nemají proces, který by je vytvářel, prosím doplňte takový proces, případně pokud se jedná o proces zvenčí označte dokument jako externí.

Title	Unfinished
Dokument 113	
Dokument 119	
Dokument 123	
Dokument 124	

P1 - proces

- Dokument 16
- Dokument 16
- Proces 12.5
- Proces 16.19
- Proces 13.36
- Proces 13.86
- Proces 7.110
- Proces 10.113

P2 - proces

- Dokument 258
- Dokument 258
- Proces 7.10
- Proces 8.41
- Proces 14.57
- Proces 16.99
- Proces 13.146
- Proces 8.261

P3 - proces

- Dokument 103

Obrázek 12 Návrh úvodní stránky

První zóna hned pod nadpisem slouží pro nové návštěvníky, zde může být umístěn odstavec se základními informacemi, také zde bude uveden odkaz na stránku s nápovědou, kde uživatel zjistí podrobnější informace o celé aplikaci.

Druhá zóna v levém sloupci je určena pro uživatele, kteří už se s aplikací pracovali, naleznout zde seznam chybějících dokumentů, to jsou dokumenty, které nemají proces, který by je tvořil, to může znamenat hned několik skutečností, dokument pochází z externího procesu, nebo k dokumentu chybí proces, který by jej vytvářel, v obou případech se od uživatele očekává, že „problém“ vyřeší. Více o tomto prvku je popsáno v další kapitole. Druhou webovou komponentou v této zóně, je list neúplných procesů, obdobně jako u chybějících dokumentů, tak i v tomto případě procesy, které jsou zde vypsané, potřebují revizi, nebo doplnit. Neúplný proces je takový proces, který nemá definovaný alespoň jeden vstup anebo alespoň jeden výstup. Podle definice v kapitole 3, musí mít každý proces vstupy a výstupy, proto pokud je nějaký proces nemá je třeba je doplnit. Na toto zobrazení bude použit prvek webová komponenta list procesů, která je již v SharePoint implementována, pouze bude nastavena pro naše účely.

Třetí zóna, respektive pravý sloupec, slouží jako přehled toho jak procesy vypadají, tedy je zde zobrazen strom procesů s možností exportu stromu do aplikace Enterprise Architect. Více o stromu procesů v další kapitole. Část exportu je vytvářena v diplomové práci s názvem Rozšíření nástroje proces inspektor o export dat od Jana Brodzáka.

Je také potřeba říct, že SharePoint umožňuje jednoduše jakoukoliv stránku změnit, proto i tato stránka může být pro libovolné projekty/uživatele upravena a zobrazovat jiné informace.

5.2.2 Rychlá navigace

Vzhledem k tomu, že nástroj Process Inspector obsahuje velké množství procesů a dokumentů, ale jen podmnožina z nich zajímá uživatele a to především ty procesy, potažmo dokumenty, se kterými chce nadále pracovat, budou v rychlé navigaci, dva odkazy a to na list procesů a na list dokumentů. Tyto odkazy zobrazí procesy (dokumenty), u kterých je uživatel nastaven jako pracovník, díky čemuž má rychlý přístup k procesům, která má pod svou kontrolou. List může také obsahovat procesy, které byly vytvořeny uživatelem. Existuje tedy rychlý pohled na procesy, které se týkají uživatele, a může je díky tomu rychleji spravovat.

5.2.3 Stránky detailu položek v listech

Na detailních stránkách jednotlivých položek, je třeba podávat ucelenou informaci o konkrétní položce a nejen výčet všech atributů položky. Navíc je kladen důraz na logičnost a jednoduchost ovládání a přecházení mezi položkami. I detailní a editační stránky položek se dají jednoduše změnit a rozšířit o další webové komponenty.

5.2.3.1 Detailní stránka procesu

Detailní stránka procesu by kromě všech polí z procesu měla obsahovat webové komponenty, které zobrazují strukturu stromu všech podprocesů aktuálního procesu včetně vstupních a výstupních dokumentů. Aby uživatel dostal lepší přehled, o tom jakou strukturu proces má, a mohl se také mezi procesy pohybovat, tedy otevřít detail libovolného podprocesu slouží strom procesů. Bude použita webová komponenta strom procesů, která bude zobrazovat jen určitý podstrom podle aktuálního procesu.

Dále by na této stránce měly být zobrazeny otázky s odpověďmi, které se týkají procesu, s možností na tyto otázky odpovědět. K tomu se použije standardní webová komponenta list odpovědí, na které bude nastaven filtr podle aktuálního procesu a navíc u tohoto prvku musí být možnost, přejít do editace odpovědi.

Na této stránce má být možnost vytvořit bratrský proces, tedy proces, který má stejného předka, a podproces, který má předka aktuální proces. Obě tyto volby otevrou nové okno k zadání nového procesu, s tím, že tyto procesy budou mít vyplněný rodičovský proces. Díky těmto změnám bude snazší vytvářet hierarchii procesů. Tyto dvě volby by měly být přístupné z rozhraní Ribbon, abychom dodrželi koncepci ovládání portálu SharePoint.

Aby se usnadnilo procházení stromu procesů, byla možnost vložení jednoduchých tlačítek, která by posunovala kontext okna na rodičovský proces nebo podproces, nakonec bylo od toho návrhu upuštěno a to z několika důvodů. Ten nejdůležitější je, že to co je na stránce detailu toto procházení umožňuje, ať už pomocí stromu procesů (přechod na podprocesy) nebo pomocí detailu

procesu, kde je odkaz na rodičovský proces. Navíc nelze jednoznačně určit podproces, na který by měla stránka přejít, v případě, že podprocesů je víc.

5.2.3.2 Detailní stránka dokumentu

U dokumentu nás kromě atributů zajímají i procesy, do kterých dokument vstupuje anebo ze kterých vystupuje a možnost na tyto procesy přejít a podívat se na jejich detaily. Nakonec tento problém bude vyřešen tak, že na stránku budou vloženy webové komponenty list procesů, jedna bude zobrazovat procesy, do kterých dokument vstupuje a druhá procesy ze kterých dokument vystupuje. U těchto komponent je důležité, aby se přes odkaz bylo možné přejít na detail procesu, taktéž zde budou zobrazeny nejdůležitější informace o procesu. Na zobrazení procesů budou použity standardní webové komponenty list procesů, s tím, že na těchto procesech bude nastaven filtr, podle aktuálního dokumentu.

5.3 Chybějící dokumenty

Důležitým prvkem celé aplikace je identifikovat dokumenty, které jsou potenciálně chybějící, buď se může jednat o dokumenty, které vstupují z venku, anebo se jedná opravdu o dokumenty, které v popisu procesů chybí a tím je popis nekompletní. Dokument u sebe ve výchozím stavu nemá informaci o tom, jestli je přiřazen k procesu, popřípadě počet procesů ke kterým je přiřazen. A proto nejsem schopni na webové komponentě list dokumentů vydefinovat takový filtr, aby zobrazoval dokumenty, které nejsou přiřazeny jako výstupní k nějakému procesu. Je několik možností jak chybějící dokumenty zobrazit, pokaždé se ale jedná o dodělaný některé z částí.

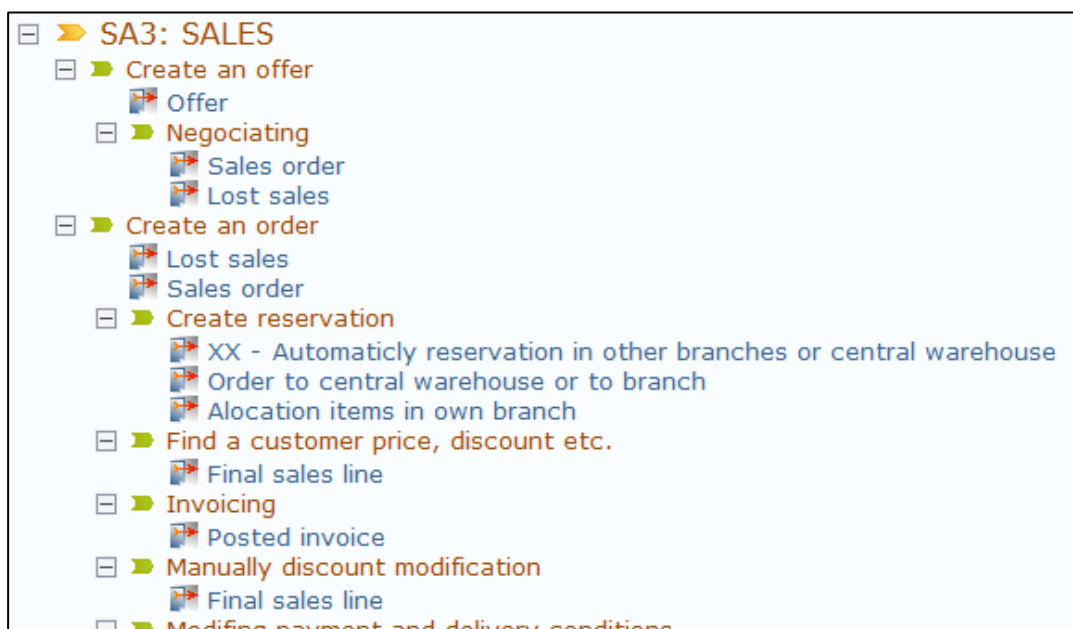
První návrh řešení je vytvořit novou webovou komponentu a v ní podle vlastního algoritmu chybějící dokumenty zobrazit. Potřebujeme ale, aby tato nová komponenta vypadala obdobně jako list dokumentů, aby zapadala do grafického rozhraní a uživatelského ovládání. Tento požadavek je časově náročný, museli bychom vytvořit celou takovou komponentu sami.

Druhou možností je, založit nové pole na dokumentu, které bude určovat počet procesů, u kterých je dokument výstupní. Díky tomuto poli by pak bylo možné vytvořit filtr nad listem dokumentů, který je součástí SharePoint. Problém je, udržovat takové pole, pokaždé když se nějakému procesu změní vstupní a výstupní dokumenty je potřeba znovu pole přepočítat. Důležité je, aby toto nezabralo příliš mnoho času a systém se díky tomu nezpomalil.

Nakonec je zvolena druhá možnost, z důvodu toho, že se použije co nejvíc prvků a metod, které jsou již hotové, a díky tomu bude práce jednodušší, ale také proto, aby ovládání zůstalo jednotné a výhoda je i možnost vytvořit si vlastní filtr nad dokumenty pro každého uživatele, tento filtr bude zahrnovat počet procesů, který dokument vytváří.

5.4 Web part strom procesů

Webová komponenta strom procesů zobrazuje procesy v hierarchické struktuře, včetně jejich vstupních a výstupních dokumentů. Původní strom procesů zobrazuje obrázek 13. Nová webová komponenta bude grafickým návrhem vycházet od té stávající. Nová komponenta proto, že SharePoint neobsahuje žádné předdefinované prvky, který by šli použít na vykreslení stromu procesů.



Obrázek 13 Původní strom procesů

Hierarchii procesů určuje pole rodičovský proces v listu procesů. Pokud proces nemá definovaného rodiče, jedná se o kořenový proces. Každý proces ve stromě je odkazem na detail procesu. V případě dokumentů, které musí být od sebe odlišeny pomocí obrázku vstupní a výstupní dokument a taktéž jako u procesů je i dokument odkazem na detail dokumentu.

Uživatel musí mít možnost jednotlivé větve zobrazovat (popřípadě skrývat), a tím získávat detailnější (respektive méně detailní) pohled na procesy, které strom zobrazuje.

Původní strom procesů byl časově náročný na zobrazení, na to, že se jedná o často používanou část aplikace, je potřeba aby se strom zobrazoval rychle, to je jeden z cílů, které jsou na implementaci kladeny. Také musí být schopen zobrazit jen určitý podstrom tak, aby byla komponenta použitelná i na detailní stránce procesu tak jak je popsáno v kapitole výše.

6 Implementace

V této kapitole se podíváme na samotné řešení, budou zde popsány postupy jak definovat a vytvářet to, co bylo navrženo v kapitole 5. Také zde budou objasněny algoritmy, které jsem použil pro vykreslování dat, tvorbu webových komponent a správu dat.

6.1 Listy

Listy jsou řešeny pomocí obsahových typů (z anglického Content type), předem se nadefinují veškeré atributy, které položky v listech obsahují, a poté lze na základě těchto typů vytvořit samotné listy.

Listy mohou obsahovat pouze následující typy sloupců:

- **Jeden řádek textu** – dovoluje uživateli zadání text až do délky 255 znaků
- **Více řádků textu** – dovoluje uživatelům zapsat víceřádkový text, včetně možnosti tento text formátovat
- **Výběr z možností** – lze vybrat hodnotu z předdefinovaných hodnot
- **Číslo** – zadání čísla lze zde nastavit maximální a minimální číslo, které je povoleno, také počet desetinných míst a případně má-li se zobrazovat číslo jako procenta
- **Měna** – podobně jako číslo, pouze ve formátu pro měny
- **Datum a čas** – uživatel může vybrat datum a případně i čas pomocí výběru z kalendáře
- **Vyhledávací pole** (z anglického lookup) – toto pole slouží jako vyhledávací pole z jiného listu nebo knihovny, uživatel tak může vybrat hodnotu, která je definována jinde ale pouze na stejném webu.
- **Ano/Ne** (zaškrtávací pole) – pole, které ukládá jen bitovou hodnotu.
- **Uživatel nebo skupina** – pole, které pomocí výběru z osob, lze vybrat osobu nebo skupinu, z Active Directory.
- **Odkaz nebo obrázek** – dovoluje uživateli zadat URL stránky (ať už externího nebo interního webu), SharePoint pak podle nastavení tuto URL zformátuje jako obrázek (v případě URL obrázku) nebo jako odkaz.
- **Vypočítané pole** – toto pole používá hodnoty z ostatních polí ve stejném listu a podle předem definované výpočtu vypočítá hodnotu, kterou zobrazí
- **Externí data** – nové pole ve verzi 2010, musí být povoleno v administraci. Dovoluje procházet externí obsahové typy a používat je jako metadata v listu.
- **Spravovat metadata** – opět nové pole, dovoluje vybírat hodnoty ze slovníku hodnot metadat

Z kapitoly návrh datové struktury je pak téměř jasné, jaké typy sloupců se mají pro jednotlivé listy použít. Pole, u kterých se požaduje zadání osoby, je možné řešit dvěma způsoby a to buď jako

další list, který bude uchovávat osoby anebo použít pole uživatel/skupina. Já se rozhodl pro řešení pomocí pole uživatel/skupina, kde se dá nastavit, aby byly vyhledávány jen osoby. Navíc pokud zvolím toto pole, je možné nad tímto polem nastavovat filtr podle aktuálního uživatele, pomocí makra [Me].

6.1.1 Definice typu

Obsahový typ (přeloženo z anglického Content type), je typ entity v listech. List může obsahovat více typů a ty dokonce mohou obsahovat různé atributy. V listu se pak zobrazí jen ty atributy, které jsou u obsahových typů definované. Výhodou je, že pak stejné entity můžeme zobrazovat v různých listech (to ale není náš případ). Každý obsahový typ musí dědit z již existujícího typu, nejjednodušší je s názvem *Item*, který má jediný povinný atribut a to název.

Definici pomocí obsahového typu usnadňuje přidávat další atributy do definovaných entit, když se přidá nový atribut do obsahového typu, pak lze jej jednoduše promítnout do všech listů, u kterých je použit. Navíc je potřeba zakládat položky, které nemají název, jako jsou otázky a odpovědi a jediný způsob jak toho docílit je přes obsahový typ.

Obsahový typ se nejnázve definuje v aplikaci SharePoint Designer. Postup je takový, že nejdříve se vydefinují atributy (sloupce) pomocí záložky *Site Columns*, a posléze se tyto atributy přidají do vlastního obsahového typu.

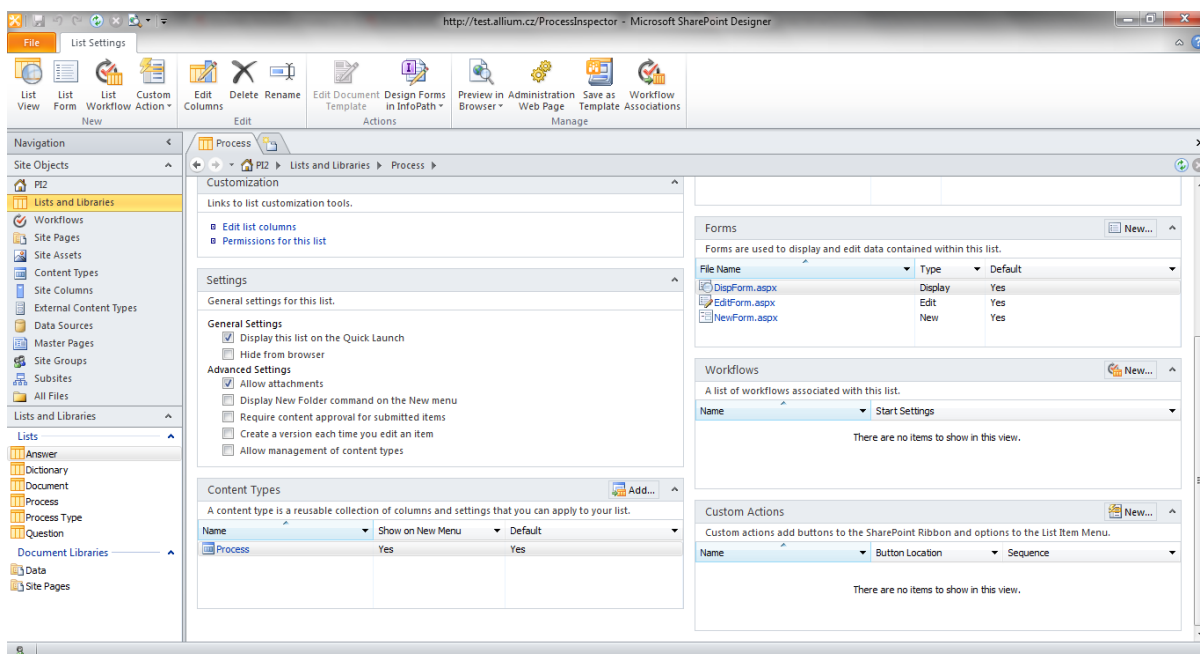
Jak vyřešit otázky a odpovědi, které nemají atribut název? Tady je potřeba tento obsahový typ definovat pomocí aplikace Visual Studio, kde pomocí XML vytvoříme typ, u kterého přerušíme dědění, a tím pádem bude obsahovat jen ty atributy, které jsou v definici. Ukázka jak takový typ nadefinovat, jedná se o definici otázky:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <ContentType ID="0x010088A9D4BD24F6473D8D88DE9271617915" Name="Question"
    Inherits="TRUE" Group="Process Inspector" Overwrite="TRUE"
    xmlns="http://schemas.microsoft.com/sharepoint/">
    <Folder TargetName="_cts/Question" />
    <FieldRefs>
      <FieldRef ID="{0030883f-d544-43cd-9cd2-ab5093e66541}"
        Name="Question" Required="TRUE" />
    </FieldRefs>
  </ContentType>
</Elements>
```

Když se podíváme na konstrukci XML souboru, tak lze vypořozovat nejdůležitější tagy a to *ContentType*, kde se definuje jak se má obsahový typ jmenovat a jaký má mít parametry. Druhý důležitý tag je *FieldRef*, který ukazuje na definici pole, která typ obsahuje, tento tag musí vždy mít identifikaci již vytvořeného pole pomocí atributu *Id* a název.

6.1.2 Definice listů

Poté, jak jsou nadefinované obsahové typy (Content type), je na řadě vytvoření listů. Listy jsou opět vytvářeny pomocí aplikace SharePoint Designer nejprve se vytvoří list, tento krok je možný i v aplikaci SharePoint. Poté je potřeba listu přiřadit obsahový typ, to už jde pouze v aplikaci SharePoint Designer. Je třeba se dostat na stránku definice listu (obrázek 14), zde zatrhnout možnost *Allow management of content types*, a ve spodní sekci *Content types* pomocí tlačítka *Add* přidat obsahový typ, který definuje list a zbytek smazat. Pokud máme jen jeden obsahový typ v seznamu, tak je možné zrušit označení u pole *Allow management of content types*. Výsledek jak by celá definice měla vypadat, je vidět na obrázku 14, kde je zobrazena definice procesu.



Obrázek 14: Definice listu procesů

Obsahové typy jsou definované bez návazností na jiné listy, proto je potřeba ještě tyto návaznosti vytvořit. Ty se tvoří pomocí vyhledávacího pole (z anglického lookup field), který vždy směřuje na jiný list. Na obrazovce kde jsme definovali listy (obrázek 14) je možnost *Edit list columns*, ta nám otevře novou obrazovku, kde je přehled všech sloupců v listu. Pak pomocí funkce *Add new column* v menu aplikace je možné přidat vyhledávací pole. Poté co vybereme na jaký list má pole směřovat a co se má v listu zobrazovat, jsou v editačním formuláři dvě možnosti povolit prázdné hodnoty a povolit více hodnot, druhá jmenovaná slouží na vytváření vazeb N:N. Tímto postupem lze vydefinovat všechny listy, které jsou v diplomové práci použity.

Při definici vyhledávacích polí je potřeba kontrolovat i nastavení, jak se má SharePoint zachovat v době mazání položky. K tomuto nastavení se lze dostat přes volbu *Administration Web Page*, která otevře webovou stránku a zde je pak možné definovat, jestli se mají smazat i položky z odkazového listu (volba *Cascade delete*) nebo pokud existuje položka v listu tak nemazat (volba

Restrict delete) a poslední možnost je, aby se nic nekontrolovalo a položka byla normálně smazána a odstraněna z vyhledávacích polí.

6.1.3 Definice pohledů

Pohled je zobrazení dat, u kterých se vyberou atributy (sloupce), které se mají zobrazit a lze definovat filtry, které zobrazí jen určité data. Pohled se definuje na listu pomocí funkce přístupné z menu Ribbon v záložce *List*, buď *Create View* pro definici nového pohledu nebo *Modify View*, pokud chceme definici pohledu změnit. Mezi pohledy pak lze přepínat pomocí *Current View*.

V této diplomové práci jsou pohledy použity na chybějící dokumenty, neúplné procesy, vlastní procesy, vlastní dokumenty. Každý uživatel si navíc může definovat své vlastní pohledy.

6.2 Projekt typu Event Receiver

V tomto projektu, jsou vybrány z množiny události jen tyto události:

- *ItemAdding* – přidávání položky do listu, předtím než je uložena do databáze, zde probíhají kontroly, jestli položka splňuje vlastnosti, které má a případně je ukládání rušeno. Také je zde možnost některé hodnoty upravit.
- *ItemAdded* – položka je přidána do databáze, většinou se zde provádí výpočty nebo úpravy jiných položek, které se mají při vkládání upravovat.
- *ItemEditing* – úprava položky, opět jako u vkládání se volá před samotnou změnou v databázi
- *ItemEdited* – úprava položky po uložení do databáze
- *ItemDeleting* – mazání položky, poslední možnost mazání zastavit anebo spustit jiné metody s hodnoty mazané položky
- *ItemDeleted* – položka je smazána a už nelze nijak přečíst její hodnoty

6.2.1 Editace a vkládání položek

Každá z metod ve třídě *EventReceiver* má parametr typu *SPItemEventProperties*. Z něj lze dostat informace o aktuální položce, listu, webu a vše co je potřeba. Díky tomu, že do stejné metody spadnou události z jakéhokoli listu, je proto nutné, nejdříve rozlišit o jaký typ položky se jedná, lze k tomu použít obsahové typy. To má tu výhodu, že podchytíme změnu v různých listech. Na zjištění jakého typu je položka, která je upravována (respektive přidávána, mazána) lze použít metodu *properties.ListItem.ContentType.Name*. K položce se pak lze dostat přes pole *properties.AfterProperties[]*, které obsahuje novou hodnotu pole. Proměnná *properties* obsahuje i *properties.BeforeProperties[]*, která by podle názvu měla obsahovat vlastnosti před vložením, tedy původní, v listech je bohužel tato proměnná prázdná a lze jí použít pouze v knihovnách, jak se lze dočíst v článku [15]. Jediná šance jak se dozvědět původní hodnoty, je tak

pomocí `properties.ListItem`, před uložením do databáze, tedy v metodách `ItemAdding`, `ItemEditing`, `ItemDeleting`.

6.2.1.1 Vkládání odpovědi

Pomocí LINQ dotazu zjistíme, jestli již existuje odpověď, která má přiřazený stejný proces a otázku, pokud takovou odpověď najdeme, vložení zrušíme pomocí konstrukce:

```
properties.Cancel = true;  
properties.ErrorMessage = "Item exists";
```

6.2.1.2 Editace procesu

Pokud dojde ke změně slovníkového pojmu anebo typu procesu, je potřeba nejdříve smazat prázdné odpovědi, opět pomocí LINQ vybereme odpovědi, které se mají smazat. Bohužel, LINQ neobsahuje žádnou metodu jak smazat víc položek naráz, takže se musí mazat v cyklu po jedné.

Jak jsou odpovědi smazané, je na řadě přiřazení nové odpovědi k procesu, postupně se vkládají všechny odpovědi s otázkami, které jsou u slovníku a typu definované, pokud již proces odpověď obsahuje, odpověď se nevloží díky kontrole při vkládání odpovědi do listu, tady je potřeba odchytit výjimku pomocí `try`, `catch` bloku tak aby neskončila s chybou celá úprava procesu.

Celý algoritmus je vložen do metody `ItemUpdated`, tak aby se vykonal v době, kdy už je proces uložen do databáze.

6.2.1.3 Editace slovníku, typu

Pokud dojde ke změně otázek u slovníku (respektive u typu procesů) tak se pro každý proces, který je slovníku přiřazen vykoná stejný kód jako při editaci jednotlivých procesů.

6.2.1.4 Mazání položek

Je zajištěno správným nastavením listů a vyhledávacích polí a není proto potřeba jej jakkoliv upravovat. Kdyby to však bylo potřeba, je možnost vytvořit metodu `ItemDeleting` anebo `ItemDeleted`.

6.2.2 Chybějící dokumenty

Jak bylo nastíněno v návrhu, u dokumentů přibude nový atribut, který bude říkat ke kolika procesům je dokument přiřazen jako výstupní. Toto pole se musí měnit, vždy když dojde ke změně procesu, proces je vytvořen nebo pokud je proces smazán.

První krok je princip jak získat počet procesů. K tomu nám pomůže LINQ dotaz, poté jak vybereme dokument tak proměnná `ProcessProcess0.Count` obsahuje počet procesů, ze kterých dokument vystupuje. Poté je třeba udělat změnu na dokumentu, kde naplníme sloupec počet procesů. Pokud bychom chtěli použít proměnnou `ProcessProcess0` ve výběru dat (ve `where` klauzuli), SharePoint toto nedovolí, tato proměnná je přístupná až poté co proběhne výběr dat.

Přepočítat počet procesů se musí pro všechny dokumenty, které byli k procesu přiřazení původní a nové. Jenže, v metodě `ItemUpdated`, ve které bezpečně víme, že proces už je v databázi uložen, není možné se dostat k původním hodnotám. Je tedy potřeba volat funkci ještě před tím, než je uložena do databáze, tedy v metodě `ItemEditing`. Díky tomu, že v databázi je stále původní proces, tak LINQ dotaz vrátí starý počet procesů u dokumentu, tento počet je proto nutné opravit o 1. Podstatné je, aby kód `base.ItemUpdated(properties)`, který provádí kontroly vstupních polí, byl před naším voláním. Poté v `ItemUpdated` vypočítáme hodnotu pro nové dokumenty, tím máme úpravu počtu dokumentů jak u nových tak u starých. U mazání je obdobný problém jako při úpravě a vše se musí provést předtím, než se proces z databáze smaže.

Pomocí pole počet procesů, je možné nastavit filtr tak aby se brali pouze dokumenty, které nemají žádný proces a nejsou zároveň externími dokumenty a v kombinaci se webovou komponentou list dokumentů jsem schopni zobrazit chybějící dokumenty.

6.3 Webová komponenta strom procesů

Webová komponenta bude vykreslena pomocí prvku `TreeView` z .Net Framework, který vykresluje stromové menu včetně tlačítek plus (respektive minus) na rozbalení (respektive sbalení) celé části podstromu. Při použití tohoto prvku je tedy potřeba správně datově vygenerovat strom s hierarchií, která odpovídá definicím jednotlivých procesů a nemusíme se tolik starat o zobrazení stromu a funkčnost v prohlížeči, protože je o ně již postaráno použitím výše zmíněného prvku. Jednotlivé uzly ve stromu jsou typu `TreeNode`, kde lze definovat název, obrázek, odkaz a další atributy, podstatný je atribut `Nodes`, který obsahuje dětinské uzly.

U webové komponenty jsou vytvořeny parametry, pomocí kterých, je možné v nastavení webové komponenty na platformě SharePoint určit název listu a sloupců, které se používají pro vykreslení stromu. Pokud parametry zůstanou prázdné, automaticky budou použity názvy potřebné pro `Process Inspector`.

Algoritmus na vykreslení ve výsledku vypadá tak, že prvně se do paměti načtou všechny dokumenty a ukládají se do mapujících struktur. U každého takto vloženého dokumentu se jako klíč vloží jeho ID. To má za následek zrychlení, jak ukazuje následující kapitola, oproti tomu kdy se na každý dokument ptalo pomocí CAML jednotlivě až byl potřeba k přiřazení k určitému procesu. Pak se vytvoří všechny uzly ve stromu a to tak, že se postupně prochází všechny procesy, vytvoří se uzel a pokud proces obsahuje vstupní nebo výstupní dokumenty tak se vytvoří i uzly s těmito dokumenty a rovnou se přiřadí k uzlu procesu jako potomci. Pokud se jedná o kořenový proces tak se rovnou vloží do stromu. Proces se ukládají opět do mapující struktury s tím, že klíč je `Id` procesu a do hodnoty se uloží vytvořený uzel a `id` rodičovského procesu. Tím máme celý strom v paměti a je třeba jej správně hierarchicky zobrazit. Na to slouží třetí část, kde jsou postupně procházeny všechny prvky ze struktury s procesy a na základě hodnot přiřazují rodičovským uzlům jejich potomky. V celém

algoritmu není kontrola, jestli strom obsahuje nějaký cyklus mezi procesy. Pokud by měl, tak díky tomu, že v tomto cyklu nebude prvek, který nemá předka (tedy je rodičovský), tak se takovýto cyklus ani nemůže vykreslit.

Navíc tato komponenta se používá i na stránce detailu procesu. Na stránce detailu je třeba aby se vykreslil jen podstrom aktuálního procesu. Pro tento účel si webová komponenta vezme ID procesu z URL a provede celý algoritmus stejně, tedy opět vezme všechny dokumenty i procesy. Samozřejmě by šlo vybírat jen ty procesy, které do stromu patří, ale pokládali bychom tak SQL databázi několik dotazů místo jednoho, který vybere vše. Každý dotaz si vezme čas na zpracování. Po výběru dat strom vykreslí jen tu část, která odpovídá procesu. Tedy jako kořenový proces bude určen proces se zadaným ID, ostatní větve stromu, nebudou vykresleny, protože na svém začátku neobsahují proces, který by byl vložen do struktury zobrazovaného stromu.

6.3.1 Test rychlosti

Byl proveden test se 110 procesy, rozdělených do stromu tak, že v kořenovém uzlu je 10 procesů a každý proces obsahuje 10 procesů. Navíc v listu dokumentů jsou 2 dokumenty a každý proces obsahuje jeden z nich jako vstupní a druhý jako výstupní.

Výsledek prvního testu byl hrozivý, celá webová komponenta se vykreslila za 2,5 sekundy, z toho 99% času trvalo získávání dat. V následujícím testu byla komponenta vykreslena bez vstupních a výstupních dokumentů a čas celého zobrazení se snížil na desítky milisekund. Bylo tedy jasné, že je třeba zlepšit přiřazování dokumentů.

Po zlepšení práce s databází v úrovni dokumentů, jak je popsáno v kapitole výše, byl čas na vykreslení celé komponenty v průměru 143ms.

110 procesů je vzorek z velmi malé firmy, průměrně se v Process Inspectoru nachází kolem 300 procesů a dalších 350 dokumentů. Byly vygenerovány procesy v náhodné hierarchii a s náhodnými dokumenty. A poté znovu zpuštěn strom procesů, výsledný čas na výpočet byl 401 ms, i přes to, že procesy byly v náhodné struktuře tak vytvořit takovou strukturu trvá jednotky milisekund, právě proto, že při tvorbě stromové hierarchie jsou veškerá data už v paměti a snadno dostupná pomocí identifikátoru. Celý strom není nijak závislý na tom, jak složitá je stromová struktura, ale pouze na tom kolik je v něm procesů a dokumentů. Testy měření shrnuje tabulka 1.

	Před úpravou	Po úpravě
110 procesů se 2 dokumenty	2 544 ms	143 ms
110 procesů bez dokumentu	104 ms	113 ms
300 procesů a 350 dokumentů	<i>Neměřeno</i>	401 ms

Tabulka 1: Rychlost vykreslení stromu procesů

Mimo byl proveden i test rychlosti vykreslení stromu v původní aplikaci, tam byl měřen čas potřebný na vykreslení celé stránky, protože nebylo možné měřit čas na zpracování stromu. Průměrná

doba na vykreslení takové stránky je 3,44 sekund, pro přibližně 370 procesů a 375 dokumentů. Do měřeného času je započítané jen vygenerování HTML stránky a stáhnutí na klienta. Pro porovnání bylo provedeno obdobné měření i na SharePoint, kde průměrný čas byl 1,13 sekundy, samotné vykreslení je tedy rychlejší, jenže v případě aplikace SharePoint navíc klient stahoval další součásti stránky, jakou jsou skripty, obrázky a díky tomu, se čas protáhl o přibližně 2-3 sekundy. Tento čas záleží na tom, co vše má prohlížeč uloženo na disku.

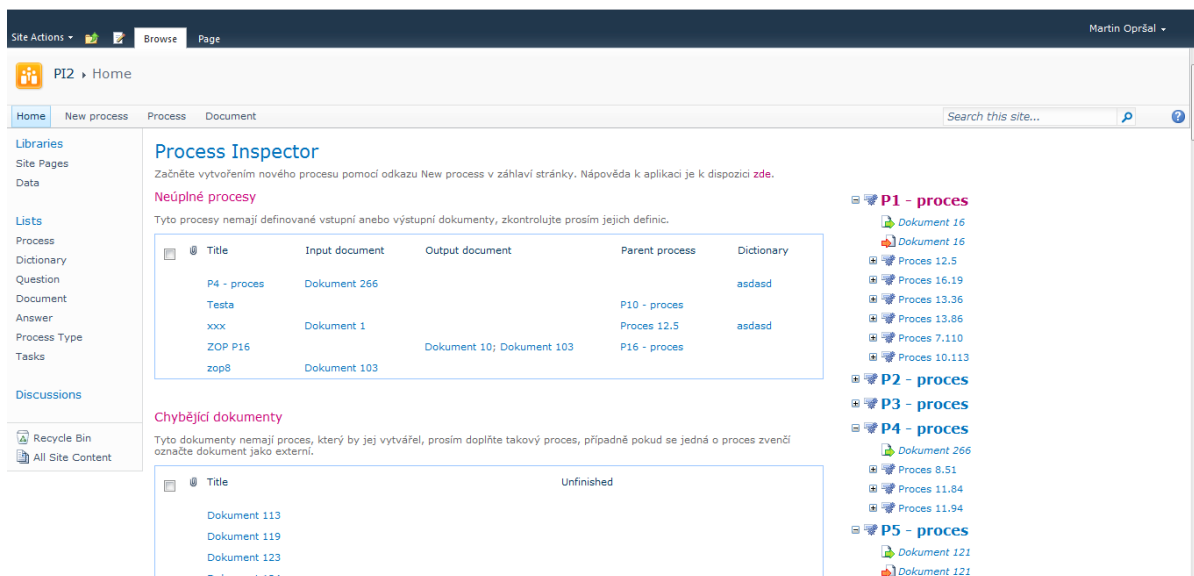
Pro představu server, na kterém byly dělány testy, je virtuální stroj, který běží na čtyř jádrovém procesoru Intel Xeon, společně s ním na hostitelském stroji běží i tři další stroje. Virtuální stroj má přiděleno 6GB paměti. Důležité je i to, že databáze i SharePoint jsou na stejném stroji.

6.4 Uživatelské rozhraní a ovládání

V této kapitole se podíváme na to, jak jednotlivé prvky, propojit tak, aby odpovídali návrhu. Použijeme zde, v jedné řadě prvky, které jsou popsány výše, ale i standardní prvky z aplikace SharePoint.

6.4.1 Úvodní stránka

Úvodní stránka obsahuje webovou komponenty strom procesů, list procesů a list dokumentů. List procesů je webová komponenta, na které je nastaven pohled neúplných procesů, tedy takových, které nemají vstupní nebo výstupní dokument. List dokumentů je webová komponenta, která zobrazuje chybějící dokumenty. Celá úvodní stránka je k náhledu na obrázku 15. V tomto případě bylo potřeba definovat pohledy, zobrazovací prvky byly použity standardní a nebylo třeba nic programovat.



Obrázek 15: Domovská stránka

6.4.1.1 Rychlá navigace

Rychlá navigace se definuje v *Site Settings*, které lze najít vlevo nahoře pod menu *Site Action*, na stránce *Site Settings* pak vybrat odkaz *Top Link Bar*. Zde je možné upravovat, mazat anebo vytvářet nové záložky v rychlé navigaci, u záložky se zadává jméno a adresa kam záložka směřuje.

Adresy listů nebo položek v listech jsou v aplikaci SharePoint definované podle konkrétního klíče a to: *adresa webu/Lists/název listu/* a dále se liší podle toho, co chceme zobrazit, to může být:

- Pohled nad daty – v takovém případě přidáme pouze *název listu.aspx*, tedy například <http://test.allium.cz/ProcessInspector/Lists/Process/AllItems.aspx>, pokud chceme zobrazit pohled AllItems nad listem procesů.
- Zobrazení detailu nebo formuláře – v takovém případě přidáme *název formuláře.aspx*, který chceme zobrazit (formuláře mohou být NewForm, EditForm, DispForm), také je důležité určit id položky, kterou chceme ve formuláři zobrazit, tedy například: <http://test.allium.cz/ProcessInspector/Lists/Process/EditForm.aspx?id=126> zobrazí editační formulář pro proces s id 126

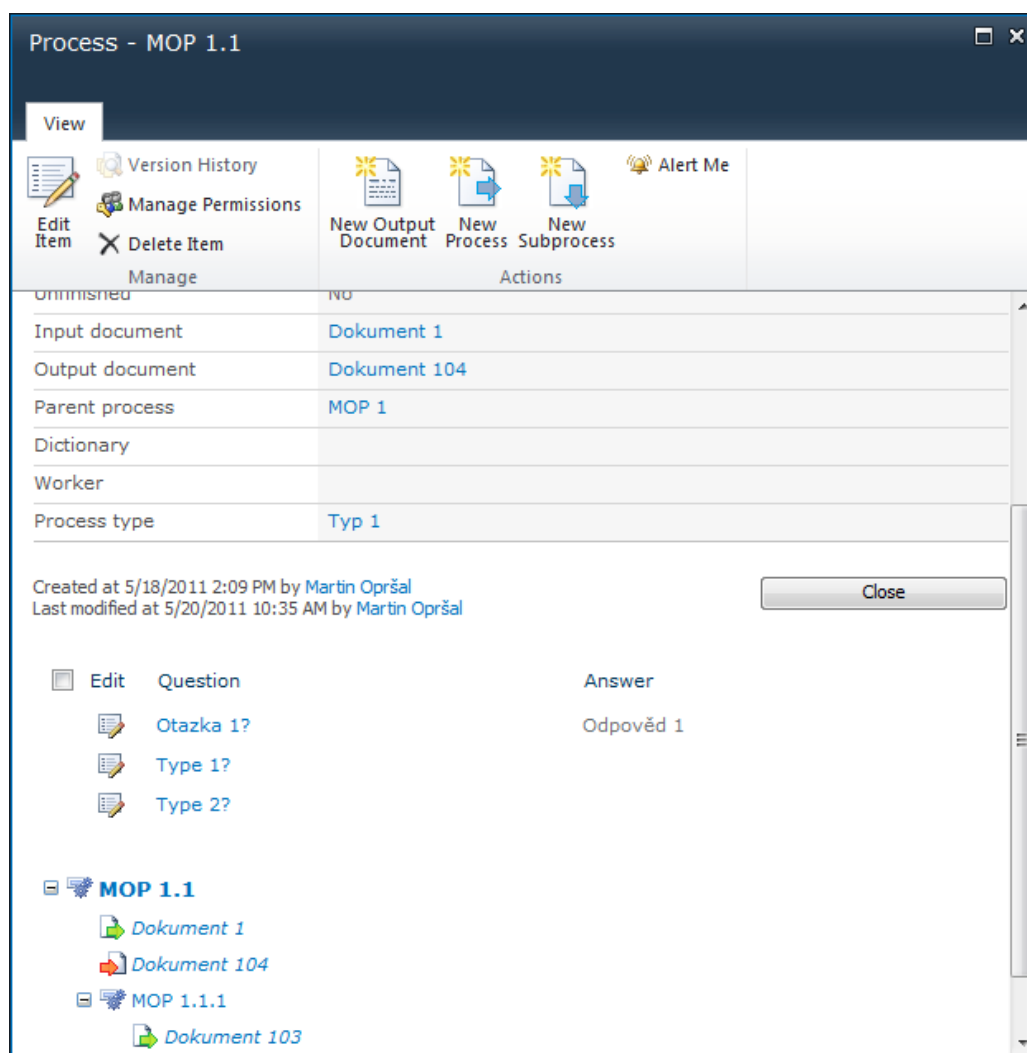
V rychlé navigaci se zobrazují odkazy na domovskou stránku webu a vlastní procesy a dokumenty. Na vlastní procesy (respektive dokumenty) jsou vytvořeny nové pohledy se jménem MyProcess (respektive MyDocuments), na kterých je nastaven filtr na pole vytvořil na makro [Me], které vrací přihlášeného uživatele, také je nastaven filtr na pole pracovník (worker) a v případě procesů i pole schvaluje, provádí, je kontrolován.

6.4.2 Detailní pohled na proces

K editaci formuláře detailu položky (respektive nové položky, editace položky) se dostaneme z rozhraní Ribbon, vybereme záložku *List* z něj *Modify Form Web Parts* po kliknutí a poté vybereme *Default Display Form*, tím se otevře nová stránka, kde je možné editovat detailní stránku procesu.

Detailní stránka procesu (stejně jako nový formulář, nebo editační formulář) ve výchozím stavu obsahuje webovou komponentu, která zobrazuje atributy procesu. Tato webová komponenta navíc poskytuje rozhraní, pomocí kterého na ní lze připojit další webové komponenty, které si z ní vezmou data. Lze tak vytvářet vztah poskytovatel - příjemce (z anglického provider – consumer). V případě stránky procesu vložíme novou webovou komponentu list odpovědí, na které v nastavení vybereme *Connections - Get Filter Values From - Process* a nastavíme, který sloupec v nové webové komponentě má filtrovat a podle které hodnoty z procesu. V případě webové komponenty odpovědi vybereme v poli *Provider Field Name* hodnotu *Title* a v poli *Consumer Field Name* hodnotu *Process*.

Poté ještě přidáme webovou komponentu strom procesů, u kterého není nutné připojovat jí k webové komponentě detail procesu, protože si sama vezme ID procesu z URL stránky a podle toho identifikátoru vypíše strom.



Obrázek 16: Detailní pohled na proces

6.4.2.1 Ribbon

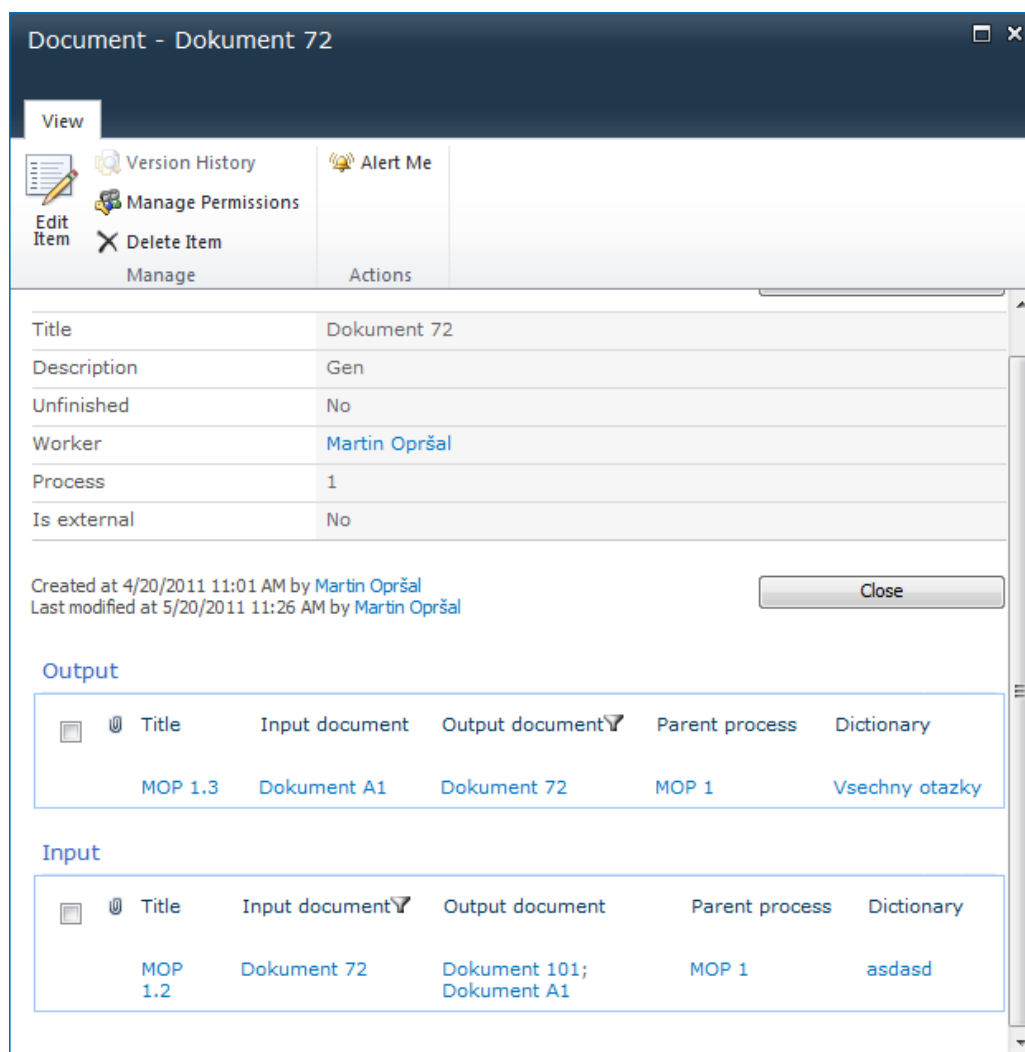
Přidání tlačítek do Ribbonu lze dvěma způsoby pomocí XML a pomocí aplikace SharePoint Designer.

V této práci je zvolena metoda definice v aplikaci SharePoint Designer, ukázkou XML lze najít v literatuře [5]. V aplikaci SharePoint Designer na nastavení listu (obrázek 14), v Ribbonu aplikace je možnost *New Custom Action*, tam lze vybrat na jaký formulář se má tlačítko vložit, a poté se vyplní jednoduchý formulář, kde se vybere odkaz kam má tlačítko ukazovat, obrázek, popis, práva, skupina ve které je umístěno a tak dále. Nová tlačítka jsou přesměrována na nový formulář a novému formuláři jako parametr je předán rodičovský proces, v případě podprocesu název aktuálního procesu, v případě bratrského procesu rodičovský proces aktuálního procesu. Na novém formuláři je pak dodělán JavaScript, který je čerpal z [19]. Tento skript vyplní pole, podle parametru v URL. Díky tomu pak může uživatel jakékoliv pole změnit, případně se rozhodnout že proces nechce vložit, v tom je výhoda oproti řešení když by se proces vkládal a předával jen k editaci.

Na základě testování jsem bylo navíc přidáno tlačítko *New output document*, které zatím směřuje pouze na vytvoření nového dokumentu, ale do budoucna by se měl dokument po vytvoření přidat k procesu jako výstupní.

6.4.3 Detailní pohled na dokument

Na stránku dokumentu byly přidány dvě webové komponenty s listem procesů. Tyto komponenty byly propojeny k hlavní webové komponentě a nastaven filtr na výstupní, respektive vstupní, dokument. K tomu účelu byly použity webové komponenty ze SharePointu. Výsledná obrazovka je na obrázku 17.



Obrázek 17: Detailní stránka dokumentu

7 Závěr

V první části této práce jsme se zabývaly managementem procesů, položili jsme základní teoretickou rovinu, kterou jsme dále použili, pro návrh aplikace. Na základě analýzy stávajícího řešení v jazyce PHP a řešení Ing. Rozsypálka byly navrženy změny stávající aplikace. U návrhu bylo třeba brát v úvahu i to co nám platforma SharePoint nabízí a co naopak je velmi problematické v ní udělat.

Výsledek této práce je první verze aplikace Process Inspector v prostředí SharePoint. Ať se jedná z velké části o přepis aplikace, bylo třeba vyřešit celou řadu specifických požadavků plynoucích ze změny platformy, mimo to, byly doporučeny a implementovány vlastnosti nad rámec prostého přepisu uvedeného výše, na základě mých osobních analýz návrhů a doporučení. Aplikace byla otestována na projektu popisu procesů u klienta společnosti Allium a to ELIT originální autodíly v Rumunsku.

Při vývoji aplikace byla použita předloha předchozí verze společnosti Allium, a v souladu s touto skutečností poskytují práva k užití, šíření a rozvoji i společnosti Allium.

7.1 Další rozšíření

Prvním možným rozšířením je podpora více jazyků, SharePoint nabízí jazykové balíčky, které přeloží uživatelské rozhraní do příslušného jazyka, ale navíc je potřeba vyřešit překlad listů. Několik stránek je tomuto problému věnováno v literatuře [14].

Další možností kde stávající aplikaci rozšířit je úprava uživatelského rozhraní a to především v editačních formulářích procesu, kde například přiřazování dokumentu k procesu, je pomocí výchozího pole, kde se ze seznamu dokumentů vybere jeden a pomocí tlačítka přidat se přiřadí k procesu. V praktickém ověření se ukázalo, že dokumentů v reálném projektu je opravdu hodně a vybírat takto dokument je zdlouhavé. Rychlejší by bylo pomocí vyhledávání dokument vyhledat a poté přiřadit, případně ho vybrat ze stromu procesů.

Další možností úpravy aplikace je přeměrování slovníku. Původní aplikace v PHP používala slovníkové struktury mimo vlastní popis procesů. Tento systém zůstal s ohledem na již existující data zachován. Je k zamyšlení, jestli pro budoucí použití bude vhodné připravit datové struktury tak, aby slovníkové pojmy byly prakticky totožné s procesy a dokumenty v jiném webu. Takováto organizace by konkrétně v ověřovacím projektu společnosti ELIT snadněji umožnila přenos procesů s mateřské společnosti v České republice do dceřiné společnosti v Rumunsku.

Literatura

- [1] BECKER, Jörg; KUGELER, Martin; ROSEMANN, Michael. *Process Management : A Guide for the Design of Business Processes*. Berlin : Springer-Verlag, 2003. 337 s. ISBN 3-540-43499-2.
- [2] FLORAC, William; CARLETON, Anita. *Measuring the Software Process : Statistical Process Control for Software Process Improvement*. [s.l.] : Addison Wesley, 2001. 250 s. ISBN 0-201-60444-2.
- [3] DYBÅ, Tora; DINGSOYR, Torgeir; MOE, Nils Brede. *Process Improvement in Practice : A Handbook for IT companies*. [s.l.] : Kluwer Academic Publisher, 2004. 114 s. ISBN 0-4020-7869-2.
- [4] RIZZO, Tom; ALIREZAEI, Reza; FRIED, Jeff; SWIDER, Paul; HILLIER, Scot; SCHAEFER Kenneth. *Professional SharePoint 2010 Developmen*. Indianapolis, Indiana : Wiley Publishing, 2010. 696 s. ISBN 978-0-470-52942-3.
- [5] *Microsoft TechNet* [online]. 2011 [cit. 2011-05-17]. Dostupné z WWW: <<http://technet.microsoft.com/en-us/library/default.aspx>>
- [6] *Microsoft SharePoint* [online]. 2011 [cit. 2011-05-17]. Dostupné z WWW: <<http://msdn.microsoft.com/cs-cz/sharepoint>>.
- [7] *Microsoft SharePoint 2010* [online]. 2011 [cit. 2011-05-17]. Dostupné z WWW: <<http://sharepoint.microsoft.com/>>.
- [8] ROZSYPÁLEK, Michal. *Rozvoj nástroje na podporu vývoje interaktivních aplikací*. 2007. 46 s. Diplomová práce. VUT.
- [9] FIEDLER, Zdeněk; ZENDULKA, Jaroslav; OPRŠAL, Zdeněk: *Jak získávat a uplatňovat znalosti při rozvoji IS*, příspěvek na konferenci DATAKON, 2006
- [10] FIEDLER, Zdeněk; OPRŠAL, Zdeněk; MATÝSKA, Martin: *Efektivní popis a optimalizace podnikových procesů*, příspěvek MOPP, 2007
- [11] HAMMER, Michael; CHAMPY, James. *Reengineering - radikální proměna firmy: manifest revoluce v podnikání*. 3.vyd. Praha : Management Press, 2000. 212 s. ISBN 80-726-1028-7.
- [12] ŘEPA, Václav: *Podnikové procesy – Procesní řízení a modelování*. 2.vyd. Grada Publishing, a.s., 2007.281s ISBN 978-80-247-2252-8
- [13] ŠMÍDA, Filip: *Zavádění a rozvoj procesního řízení ve firmě*. 1.vyd. Grada Publishing, a.s., 2007. 293 s. ISBN 978-80-247-1679-4

- [14] KLINDT, Todd; YOUNG, Shane; CARAVAJAL, Steve. *Professional SharePoint 2010 Administration*. Indianapolis, Indiana : Wiley Publishing, 2010. 800 s. ISBN 978-0-470-53333-8.
- [15] *C# Corner* [online]. 29.1.2011 [cit. 2011-05-18]. Before Properties, After Properties and ListItem in SharePoint 2010 Event Receivers. Dostupné z WWW: <<http://www.c-sharpcorner.com/Blogs/4223/>>.
- [16] *Microsoft Office* [online]. 2011 [cit. 2011-05-20]. Dostupné z WWW: <<http://officeimg.vo.msecnd.net/en-us/>>.
- [17] *LBMS* [online]. 2011 [cit. 2011-05-22]. Dostupné z WWW: <<http://www.lbms.cz/>>.
- [18] *Microsoft SharePoint* [online]. 4.1.2009 [cit. 2011-05-23]. To the Tech. Dostupné z WWW: <<http://www.tothetech.com/microsoft/microsoft-sharepoint.html>>.
- [19] *JQuery for Everyone: Pre-populate Form Fields* [online]. 20.4.2009 [cit. 2011-05-20]. EndUserSharePoint.com. Dostupné z WWW: <<http://www.endusersharepoint.com/2009/04/20/jquery-for-everyone-pre-populate-form-fields/>>.

Seznam příloh

Příloha 1. Instalace aplikace

Příloha 2. DVD s aplikací

Příloha 1: Instalace aplikace

Na přiloženém DVD jsou k dispozici zdrojové kódy a to formou exportů ze SharePointu a projektu ve Visual Studiu 2010.

Postup jak aplikaci zprovoznit, není jednoduchý a je časově náročný, proto bych doporučil, aby byla kontaktována společnost Allium, u které je celé řešení zprovozněno. I přes to, kdyby bylo potřeba aplikaci rozběhnout postup je následující:

1. Instalace Microsoft Windows 2008 R2 (64bitová verze)
2. Instalace Microsoft SQL Server R2 (64bitová verze)
3. Instalace Microsoft SharePoint Server 2010
4. Instalace Microsoft Visual Studio 2010
5. Vytvořit web v SharePoint
6. Přiložené projekty nahrát pomocí aplikace Visual Studio na server
7. Import řešení pomocí utility SPSite-Import jak je popsáno v kapitole 4.7.2

Tento postup navíc předpokládá alespoň základní znalost technologií společnosti Microsoft. Během instalace aplikace můžou vzniknout problémy, které je potřeba řešit a přímo nesouvisí s vyvinutou aplikací, ale například nastavením nebo instalací některého z produktů společnosti Microsoft.

Příloha 2: DVD s aplikací

Obsah DVD:

- Uživatelský návod.pdf – uživatelský návod
- DP.docx – zdrojový text práce
- DP.pdf – text práce
- Projekty
 - EventReceiverProcessInspector
 - Složka s projektem event receiver
 - WebPartProcessTree
 - Složka s projektem strom procesů
 - Images
 - Složka s obrázky
- Exporty
 - Export.cmp – export webu ve formátu cmp
 - Export.wsp – export webu ve formátu wsp