

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

Diplomant	: Bc. Zdeněk Hejl
Vedoucí diplomové práce	: Ing. Libor Gála
Oponent diplomové práce	: Ing. Daniel Kafka

**Automatizovaná podpora procesu vývoje webu
s důrazem na SEO**

ROK: 2009

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 05. 05. 2009

.....
podpis

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, především mému vedoucímu panu Ing. Liboru Gálovi.

Abstrakt

Hlavním cílem této práce je navržení procesního modelu tvorby nového webu a návrh a implementace jeho automatizované podpory prostředky informačních technologií. Samotný návrh procesu bude vystavěn na identifikovaných nejlepších praktikách dostupných metodik pro tvorbu webových aplikací a bude respektovat a aplikovat principy SEO (optimalizace pro vyhledávače).

Největším přínosem této práce je navržený procesní model pro tvorbu webových aplikací, který vychází z toho nejlepšího, co nabízí metodiky pro tvorbu webových aplikací a navíc obsahuje i dodatečné činnosti, které je potřeba vykonávat, ale které nejsou součástí daných metodik. Dalším významným přínosem je namapování optimalizačního procesu na činnosti navrženého procesního modelu a implementace automatizované podpory těchto činností pomocí desktopové aplikace.

Práce je strukturovaná do pěti základních kapitol (mimo úvod a závěr). První kapitola se zaměřuje na definici základních pojmů spojených s vývojem aplikací a především na provedení analýzy vybraných metodik pro vývoj webových aplikací. Druhá kapitola navazuje na výstupy analýzy daných metodik a tvoří jádro celé práce. Jejím obsahem je návrh a popis procesního modelu pro vývoj webu. Třetí kapitola se zabývá problematikou SEO, konkrétně definicí pojmu SEO, představením optimalizačního procesu a především představením optimalizačních doporučení. Čtvrtá a pátá kapitola jsou zaměřeny na automatizovanou podporu procesu tvorby webu s důrazem na SEO. V rámci čtvrté kapitoly jsou identifikovány činnosti procesního modelu, který by mohly být aplikačně podpořeny a pro tyto činnosti jsou definovány uživatelské požadavky na funkcionalitu. Poslední pátá kapitola je zaměřena na představení implementované aplikace pro podporu vývoje webu s důrazem na SEO.

Klíčová slova: vývoj webových aplikací, procesní model, optimalizace pro vyhledávače, SEO, optimalizační proces

Abstract

Principal aim of this thesis is the design of new website development process model and the design and implementation of its automated support by information technologies instruments. Process design itself will be build on identified best practices of available methodics for web applications development and this process will also respect and apply SEO (Search Engine Optimization) principles

Biggest contribution of this thesis is designed web application development process model which is based on web application development methodics best practices and also contains additional activities which have to be done but they aren't part of these methodics. Mapping of the optimization process and designed process activities and implementation of automated support for these activities via desktop application are next important contributions of this thesis.

This thesis is divided into five main chapters (except introduction and conclusion). First chapter is focused on definition of application development basic concepts and mainly on analysis of web application development methodics. Second chapter takes up analysis outputs and it forms the basis of this thesis. Design and description of web development process model generate the content of this chapter. Third chapter deals with SEO problems, in the concrete, it deals with definition of SEO, presentation of optimization process and especially with presentation of optimization recommendations. Fourth and fifth chapters are focused on automated support for web development process model with emphasis on SEO. Within the scope of fourth chapter, process model activities which can be supported by application are identified and user requirements are defined for these activities. Last fifth chapter is focused on presentation of implemented application for web development with emphasis on SEO support.

Keywords: web applications development, process model, search engine optimization, SEO, optimization process

Obsah

1. ÚVOD	1
1.1. VYMEZENÍ TÉMATU	1
1.2. CÍLE PRÁCE	1
1.3. PŘÍSTUP KE ZPRACOVÁNÍ PRÁCE	1
1.4. ZAMĚŘENÍ PRÁCE	2
2. ANALÝZA METODIK PRO VÝVOJ WEBOVÝCH APLIKACÍ	3
2.1. DEFINICE ZÁKLADNÍCH POJMŮ	3
2.1.1. METODA, METODIKA A METODOLOGIE	3
2.1.2. ŽIVOTNÍ CYKLUS VÝVOJE IS	4
2.1.3. PROCES A PROCESNÍ MODEL	5
2.1.4. VZTAH METODIKY, ŽIVOTNÍHO CYKLU A PROCESNÍHO MODELU	6
2.2. ROZDÍLY VE VÝVOJI DESKTOPOVÝCH A WEBOVÝCH APLIKACÍ	7
2.3. ÚVOD DO METODIK PRO VÝVOJ WEBOVÝCH APLIKACÍ	8
2.3.1. CO JSOU TO HYPERMÉDIA?	8
2.3.2. CHARAKTERISTIKY METODIK	9
2.3.3. ZPŮSOB PREZENTACE METODIK	9
2.4. OOHDM	10
2.4.1. ZÁKLADNÍ INFORMACE A HISTORIE	10
2.4.2. VÝVOJOVÝ PROCES	11
2.5. SOHDM	13
2.5.1. ZÁKLADNÍ INFORMACE A HISTORIE	13
2.5.2. VÝVOJOVÝ PROCES	14
2.6. WSDM	15
2.6.1. ZÁKLADNÍ INFORMACE A HISTORIE	15
2.6.2. VÝVOJOVÝ PROCES	16
2.1. WEBML	18
2.1.1. ZÁKLADNÍ INFORMACE A HISTORIE	18
2.1.2. VÝVOJOVÝ PROCES	19
2.2. SHRUTÍ KAPITOLY	21
3. NÁVRH PROCESNÍHO MODELU PRO VÝVOJ WEBU	23

3.1. NEDOSTATKY SOUČASNÝCH METODIK	23
3.2. NOTACE BPMN	24
3.3. NÁVRH HRUBÉHO PROCESNÍHO MODELU	25
3.4. FÁZE ÚVODNÍ ANALÝZA PROJEKTU	28
3.4.1. VÝVOJOVÝ PROCES	28
3.4.2. IDENTIFIKACE DŮVODŮ PRO VZNIK PROJEKTU	28
3.4.3. VYMEZENÍ PŘEDMĚTU PROJEKTU	29
3.4.4. URČENÍ HLAVNÍCH VÝSTUPŮ PROJEKTU	29
3.4.5. IDENTIFIKACE CÍLOVÝCH UŽIVATELŮ	29
3.4.6. FORMULACE CÍLŮ A PŘÍNOSŮ PROJEKTU	30
3.4.7. ČASOVÁ, ZDROJOVÁ A NÁKLADOVÁ ANALÝZA PROJEKTU	31
3.4.8. STANOVENÍ RIZIK PROJEKTU	32
3.5. FÁZE ANALÝZA KONKURENČNÍHO PROSTŘEDÍ	33
3.5.1. VÝVOJOVÝ PROCES	33
3.5.2. IDENTIFIKACE KONKURENČNÍCH SEGMENTŮ	34
3.5.3. IDENTIFIKACE RELEVANTNÍCH KLÍČOVÝCH SLOV	35
3.5.4. IDENTIFIKACE KONKURENTŮ	35
3.5.5. ANALÝZA KONKURENTŮ	37
3.5.6. SWOT ANALÝZA PROJEKTU	38
3.6. FÁZE ANALÝZA UŽIVATELSKÝCH POŽADAVKŮ	40
3.6.1. VÝVOJOVÝ PROCES	40
3.6.2. IDENTIFIKACE TŘÍD A SKUPIN UŽIVATELŮ	40
3.6.3. SPECIFIKACE UŽIVATELSKÝCH POŽADAVKŮ	41
3.6.4. TVORBA SCÉNÁŘŮ POUŽITÍ	42
3.7. FÁZE DATOVÝ NÁVRH	42
3.7.1. VÝVOJOVÝ PROCES	42
3.7.2. IDENTIFIKACE OBJEKTOVÝCH TŘÍD	42
3.7.3. DEFINOVÁNÍ VZTAHŮ MEZI OBJEKTOVÝMI TŘÍDAMI	43
3.8. FÁZE HYPERTEXTOVÝ NÁVRH	45
3.8.1. VÝVOJOVÝ PROCES	45
3.8.2. NÁVRH STRUKTURY WEBOVÉ APLIKACE	45
3.8.3. KOMPOZICE JEDNOTLIVÝCH WEBOVÝCH STRÁNEK	47
3.8.4. TVORBA NAVIGAČNÍ SÍTĚ	48
3.8.5. INTEGRACE HYPERTEXTOVÉHO MODELU S FUNKCIONALITOU WEBOVÉ APLIKACE	49
3.9. FÁZE NÁVRH UŽIVATELSKÉHO ROZHRAŇÍ	51
3.9.1. VÝVOJOVÝ PROCES	51
3.9.2. NÁVRH LAYOUTU JEDNOTLIVÝCH WEBOVÝCH STRÁNEK	51

3.9.3. NÁVRH ZPŮSOBU TRANSFORMACE OBSAHOVÝCH JEDNOTEK	52
3.10. FÁZE IMPLEMENTACE	53
3.10.1. VÝVOJOVÝ PROCES	53
3.10.2. VÝBĚR IMPLEMENTAČNÍHO PROSTŘEDÍ	54
3.10.3. VÝBĚR WEBHOSTINGU A DOMÉNY	55
3.10.4. TRANSFORMACE DATOVÉHO MODELU	57
3.10.5. IMPLEMENTACE A TESTOVÁNÍ UŽIVATELSKÉHO ROZHRANÍ	59
3.10.6. IMPLEMENTACE A TESTOVÁNÍ FUNKCIONALITY	60
3.10.7. ZÍSKÁNÍ DAT A NAPLNĚNÍ DATABÁZE	61
3.11. FÁZE PROVOZ	62
3.11.1. VÝVOJOVÝ PROCES	62
3.11.2. VÝBĚR MONITOROVACÍHO SYSTÉMU	62
3.11.3. IMPLEMENTACE MONITOROVACÍHO SYSTÉMU	64
3.11.4. NASTAVENÍ MONITOROVACÍHO SYSTÉMU	64
3.11.5. ANALÝZA NÁVŠTĚVNOSTI	64
3.11.6. ANALÝZA UMÍSTĚNÍ VE VYHLEDÁVAČÍCH	65
3.11.7. ANALÝZA VYUŽÍVÁNÍ WEBOVÉ APLIKACE	66
3.11.8. ANALÝZA BYZNYS PŘÍNOSŮ	66
3.11.9. ZHODNOCENÍ ANALYTICKÝCH VÝSTUPŮ	67
3.12. SHRNUTÍ KAPITOLY	68
 <u>4. IMPLEMENTACE PRINCIPŮ SEO DO PROCESNÍHO MODELU</u>	 <u>69</u>
 4.1. ÚVOD DO PROBLEMATIKY SEO	 69
4.1.1. DEFINICE SEO	69
4.1.2. SEO JAKO MARKETINGOVÝ NÁSTROJ	70
4.1.3. OPTIMALIZAČNÍ PROCES	71
4.2. VÝBĚR DOMÉNOVÉHO JMÉNA	72
4.3. VYTVOŘENÍ WEBU PŘÍVĚTIVÉHO K VYHLEDÁVACÍM ROBOTŮM	74
4.3.1. STRUKTURA URL ADRES	74
4.3.2. SITEMAPY (MAPY WEBU)	75
4.3.3. SOUBOR ROBOTS.TXT	76
4.3.4. OMEZENÍ VYUŽÍVÁNÍ ŠPATNĚ INDEXOVATELNÝCH TECHNOLOGIÍ	76
4.4. KLÍČOVÁ SLOVA A JEJICH UMÍSTĚNÍ	77
4.4.1. TITULEK WEBOVÉ STRÁNKY	77
4.4.2. METATAGY DESCRIPTION A KEYWORDS	78
4.4.3. TĚLO WEBOVÉ STRÁNKY	79

4.5. LINK BUILDING (ZÍSKÁVÁNÍ ODKAZŮ)	80
4.5.1. ZÁKLADNÍ POJMY	80
4.5.2. VÝMĚNA RECIPROČNÍCH ODKAZŮ	82
4.5.3. POKROČILÉ TECHNIKY LINK BUILDINGU	83
4.6. SHRUTÍ KAPITOLY	86
<u>5. SPECIFIKACE ZADÁNÍ APLIKACE</u>	87
5.1. IDENTIFIKACE ČINNOSTÍ VHODNÝCH PRO APLIKAČNÍ PODPORU	87
5.2. DEFINICE UŽIVATELSKÝCH POŽADAVKŮ	88
5.2.1. IDENTIFIKACE RELEVANTNÍCH KLÍČOVÝCH SLOV	88
5.2.2. IDENTIFIKACE KONKURENTŮ	88
5.2.3. ANALÝZA KONKURENTŮ	89
5.2.4. VÝBĚR WEBHOSTINGU A DOMÉNY	89
5.2.5. OPTIMALIZACE ON-PAGE FAKTORŮ	89
5.2.6. OPTIMALIZACE OFF-PAGE FAKTORŮ	90
5.2.7. ANALÝZA UMÍSTĚNÍ VE VYHLEDÁVAČÍCH	90
5.3. SHRUTÍ KAPITOLY	90
<u>6. PREZENTACE IMPLEMENTOVANÉ APLIKACE</u>	92
6.1. CHARAKTERISTIKY IMPLEMENTACE	92
6.1.1. IMPLEMENTAČNÍ PROSTŘEDÍ	92
6.1.2. POUŽITÉ NÁSTROJE	92
6.2. ZÁKLADNÍ MODELY	93
6.3. APLIKAČNÍ PROSTŘEDÍ	95
6.4. SHRUTÍ KAPITOLY	98
<u>7. ZÁVĚR</u>	99
<u>POUŽITÉ ZDROJE</u>	101
<u>TERMINOLOGICKÝ SLOVNÍK</u>	107

1. Úvod

1.1. Vymezení tématu

V souvislosti s neustále větším rozšiřováním internetu pochopila většina podniků, organizací i jednotlivců, že k tomu, aby o nich bylo vědět, je nutné se na internetu prezentovat. Vzhledem k tomu, že na internetu existují miliardy webových stránek, samotné vytvoření webové prezentace ovšem většinou nestačí. Je nutné zajistit, aby se potenciální návštěvníci o daných webových stránkách vůbec dozvěděli a stránky mezi tisíci dalšími nezapadly.

K tomu slouží SEO (Search Engine Optimization) neboli optimalizace pro vyhledávače. SEO lze chápat jako proces skládající se z množiny různých postupů a technik, které slouží ke zvýšení kvality a počtu návštěvníků webových stránek přicházejících pomocí výsledků vyhledávání relevantních klíčových slov ve vyhledávačích. Optimalizační proces ovšem nemůže existovat jen tak samostatně, ale musí být včleněn do celkového životního procesu aplikace. Životní cyklus aplikace zahrnuje činnosti od prvotního vytvoření požadavku na vznik webové aplikace přes analýzu konkurence a uživatelských požadavků, návrhové fáze až po samotnou implementaci a provoz aplikace.

1.2. Cíle práce

Hlavním cílem této práce je navržení procesního modelu tvorby nového webu a návrh a implementace jeho automatizované podpory prostředky informačních technologií. Samotný návrh procesu bude vystavěn na identifikovaných nejlepších praktikách dostupných metodik pro tvorbu webových aplikací a bude respektovat a aplikovat principy SEO (optimalizace pro vyhledávače).

1.3. Přístup ke zpracování práce

Přístup ke zpracování této práce je podřízen hlavnímu cíli práce tj. návrhu procesního modelu a návrhu a implementaci automatizované podpory částí tohoto procesního modelu. Samotný text práce (mimo úvodu a závěru) se skládá z pěti kapitol. První tři kapitoly se týkají návrhu procesního modelu a zbylé dvě sou zaměřeny na automatizovanou podporu procesního modelu.

První kapitola (*Analýza metodik pro vývoj webových aplikací*) se zaměřuje na definici základních pojmů spojených s vývojem aplikací, představení rozdílů mezi desktopovými a webovými aplikacemi a především na provedení analýzy vybraných metodik pro vývoj webových aplikací. Tato kapitola poskytuje teoretické zázemí pro následující kapitolu.

Druhá kapitola (*Návrh procesního modelu pro vývoj webu*) představuje jádro celé práce. V rámci této kapitoly je navržen a popsán procesní model pro vývoj webu. Podkladem pro návrh modelu jsou informace o jednotlivých metodikách pro vývoj webových aplikací a identifikované nedostatky těchto metodik.

Třetí kapitola (*Implementace principů SEO do procesního modelu*) je zaměřena na problematiku SEO (optimalizace pro vyhledávače). Po definování samotného pojmu a představení optimalizačního procesu je zbytek kapitoly věnován jednotlivým optimalizačním doporučením seskupeným do větších tematických celků.

Obsahem čtvrté kapitoly (*Specifikace zadání aplikace*) je identifikace jednotlivých činností z navrženého procesního modelu, které by mohly být aplikačně podpořeny a definování uživatelských požadavků na funkcionalitu pro vybrané činnosti.

Závěrečná pátá kapitola (*Prezentace implementované aplikace*) slouží k představení implementované aplikace pro podporu vývoje webu s důrazem na SEO. Kromě základních implementačních charakteristik jsou prezentovány základní návrhové modely použité při vývoji aplikace a také některé důležité obrazovky aplikace.

1.4. *Zaměření práce*

Obsah a výstupy této práce jsou určeny především pro tvůrce a správce webových aplikací. Tato práce jim poskytuje možný návod, jak postupovat při vývoji a provozu webových aplikací při respektování principů SEO. Postup lze využít při vývoji a provozu jakéhokoliv typu webové aplikace od jednoduchých webů skládajících se z několika stránek až po komplexní portály. V závislosti na typu aplikace může být kladen odlišný důraz na různé fáze navrženého procesního modelu, např. větší důraz na analýzu uživatelských požadavků, datový návrh a implementaci u aplikací poskytujících uživateli především funkcionalitu nebo naopak větší zaměření na hypertextový návrh u aplikací poskytujících převážně obsah.

2. Analýza metodik pro vývoj webových aplikací

Hlavním cílem této kapitoly je představení několika významných metodik, které se zabývají vývojem webových aplikací. Na základě těchto metodik bude poté v následující kapitole vystavěn procesní model pro tvorbu webu. Mezi další cíle lze zařadit definování základních pojmů a nastínění rozdílů mezi klasickými desktopovými aplikacemi a webovými aplikacemi.

2.1. Definice základních pojmů

2.1.1. Metoda, metodika a metodologie

Pojmy metoda, metodika a metodologie jsou často vzájemně zaměňovány a používány v nevhodných souvislostech. Tato situace je z velké míry zapříčiněna angličtinou, ve které pojmy metodika a metodologie často splývají. Ve slovníku lze totiž nalézt pro anglický výraz *methodology* české ekvivalenty *metodika* i *metodologie*.

Všechny tři výše uvedené pojmy je nejvhodnější definovat v obecné rovině s následným přenesením významu do konkrétní oblasti, tj. v tomto případě do oblasti vývoje informačních systémů (IS). Následující tři definice vychází z (Mareš, 2007).

Metoda

Postup, jak dosáhnout vytyčeného teoretického či praktického cíle, který zahrnuje všechna pravidla a návody k jednání a také prostředky, které umožňují snazší uchopení problému.

Metodika

Ustálené pravidlo použití určitých metod.

Metodologie

Souhrn metod určitého vědního oboru anebo disciplína, která se zabývá zkoumáním metod, cílů a uspořádání určitého oboru nebo skupiny oborů.

V oblasti vývoje IS lze za metodologii považovat *softwarové inženýrství*, které zastřešuje jednotlivé využívané metody vývoje IS. Příkladem metodik mohou být *vodopádový model*, *spirálový model*, *Rational Unified Process*, *SCRUM* či *extrémní programování*. Mezi metody

používané při vývoji IS lze zařadit například *YSM (Yourdon Structured Method)*, *JSP (Jackson Structured Programming)* nebo *JSD (Jackson System Development)*.

2.1.2. Životní cyklus vývoje IS

Životní cyklus je pojem, který původně pochází z oblasti marketingu - z teorie životního cyklu produktu. Tato teorie popisuje jednotlivé fáze produktu od příprav jeho vzniku až po jeho stažení z trhu. Podle (Kopecký, 2001) se životní cyklus produktu skládá ze šesti fází:

- výzkum a vývoj,
- uvedení na trh,
- růst prodeje,
- zralost,
- nasycení trhu,
- pokles prodeje.

Tyto fáze lze aplikovat také na IS, které jsou ve své podstatě produktem projektu vývoje IS. Pojem životní cyklus však bývá v informatice často chápán v užším smyslu jako *životní cyklus vývoje IS*. Například metodika MMDIS (Voříšek, 2003, s. 124) identifikuje šest základních etap životního cyklu vývoje IS:

- úvodní studie,
- globální analýza a návrh,
- detailní analýza a návrh,
- implementace,
- zavádění,
- provoz a údržba.

Problematika životního cyklu vývoje IS je často velice pevně spojena s metodikou, která je použita pro vývoj IS. Více o propojení životního cyklu a metodiky vývoje IS v podkapitole 2.2.4.

2.1.3. Proces a procesní model

Lze nalézt velké množství různých definic pojmu proces. Jedním příkladem může být definice z (Kacerovský, 2008):

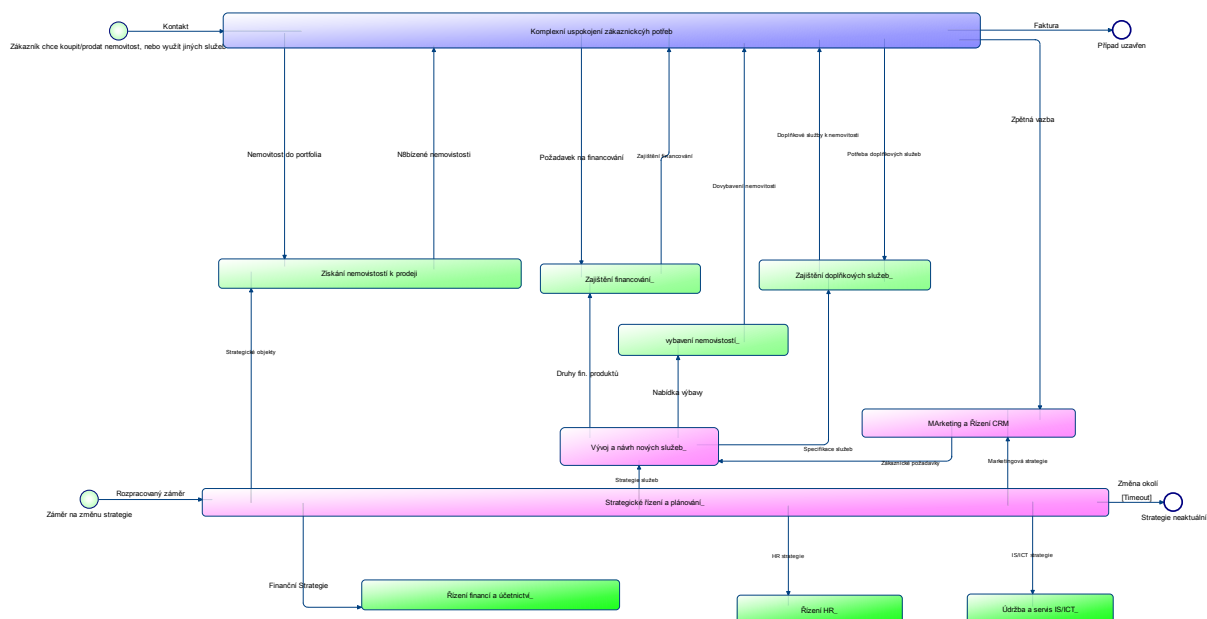
„Účelně naplánovaná a realizovaná posloupnost činností, ve kterých za pomoci odpovídajících zdrojů probíhá transformace vstupů na výstupy.“

Ačkoliv se jednotlivé definice mezi sebou liší, dají se vysledovat určité společné prvky.

Proces:

- je iniciován událostí a končí dosažením jednoho z cílových stavů,
- má svoje vstupy a výstupy,
- spotřebovává zdroje,
- je tvořen množinou vzájemně provázaných činností.

S procesy je spojen také pojem *procesní model*. Procesní model slouží k zachycení jednotlivých procesů a vzájemných vazeb mezi nimi. Schematické zobrazení procesního modelu nabízí obrázek 2.1. Na tomto obrázku představují kolečka na levé straně vstupní události a kolečka na pravé straně naopak cílové stavy. Mezi nimi se nachází procesy, které jsou vzájemně propojeny pomocí pojmenovaných vazeb.

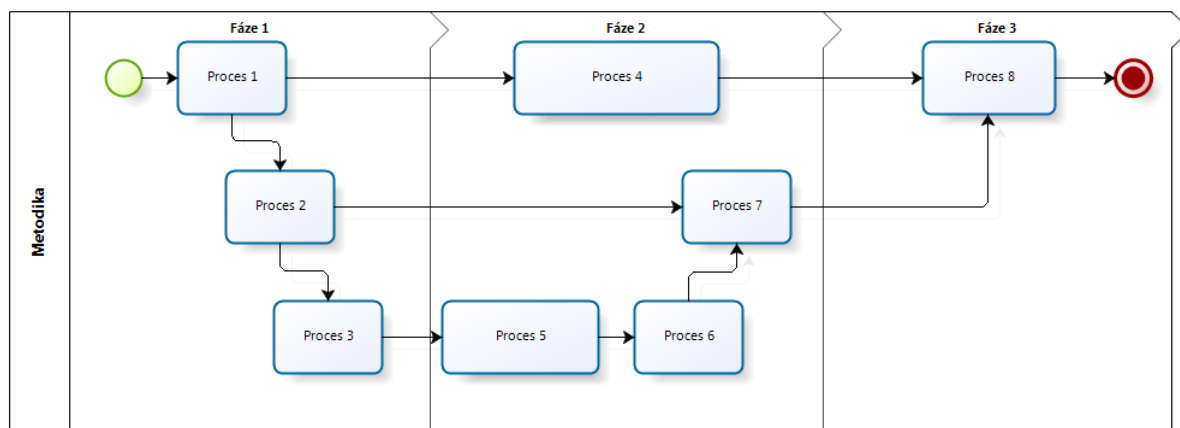


Obrázek 2.1: Schematické zobrazení procesního modelu

2.1.4. Vztah metodiky, životního cyklu a procesního modelu

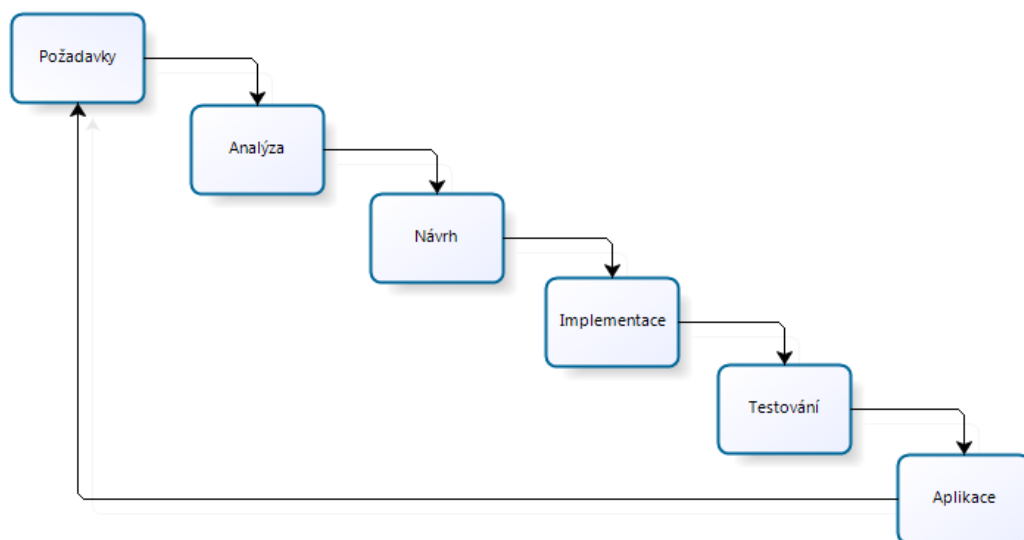
Cílem této podkapitoly je nastínění vztahů, které fungují mezi jednotlivými pojmy, které byly představeny výše, tj. konkrétně mezi metodikou, životním cyklem a procesním modelem. Vzájemné vztahy budou vysvětleny s pomocí schématu (viz obrázek 2.2).

Jak je patrné z tohoto obrázku, jednotlivé procesy z procesního modelu jsou přiřazeny minimálně jedné fázi životního cyklu. Při pohledu z druhé strany platí, že jednotlivé fáze životního cyklu shromažďují procesy, které mají podobné charakteristiky a slouží k dosažení podobných cílů. Shrňeme-li tato dvě tvrzení, můžeme tvrdit, že fáze životního cyklu představují hierarchicky nadřazenou úroveň k procesům, neboli že fáze životního cyklu se nachází na vyšší úrovni obecnosti oproti poměrně podrobným procesům.

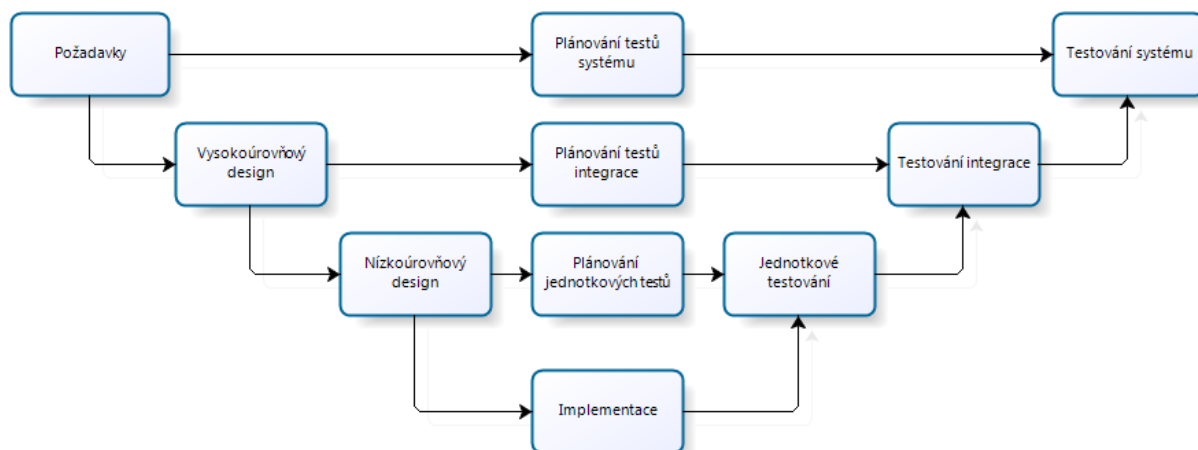


Obrázek 2.2: Vztah mezi metodikou, životním cyklem a procesním modelem

Druhým důležitým vztahem je vztah mezi metodikou a životním cyklem. Tento vztah je poměrně silný, jelikož většinou vybraná metodika ovlivňuje složení i obsah jednotlivých fází životního cyklu. Metodika mimo jiné definuje i tzv. *milníky*, které představují přechody mezi jednotlivými fázemi životního cyklu. Na obrázcích 2.3, resp. 2.4 vycházejících z (Lewallen, 2005) jsou uvedeny příklady modelů životního cyklu pro metodiku inkrementálního vývoje IS, resp. tzv. V-metodiku.



Obrázek 2.3: Inkrementální vývoj IS (Lewallen, 2005)



Obrázek 2.4: Model V (Lewallen, 2005)

2.2. Rozdíly ve vývoji desktopových a webových aplikací

Ačkoliv se odlišnost desktopových a webových aplikací v současnosti snižuje (vyvíjí se například v jednom vývojovém prostředí), stále mezi nimi existují rozdíly, které jsou již dány charakterem prostředí, ve kterém běží.

Desktopové aplikace běží na lokálním počítači, spotřebovávají více lokálních systémových zdrojů a jsou více závislé na konkrétním operačním systému. Na druhou stranu webové aplikace se spouští ve webovém prohlížeči, běží na vzdáleném serveru, spotřebovávají méně lokálních zdrojů a jsou minimálně závislé na operačním systému.

Pro účely této práce jsou ovšem důležité ještě jiné rozdíly. Jedná se o rozdíly v grafickém rozhraní aplikací a ve způsobu navigace v aplikacích. Desktopové aplikace nabízí bohatší možnosti tvorby uživatelského rozhraní a specifikace interakce s uživatelem. Možnosti navigace jsou však často omezeny a navigace je více řízena aplikací než uživatelem. Webové aplikace naopak nabízí menší množství prvků uživatelského rozhraní a horší interaktivitu, avšak poskytují větší možnosti navigace. Hypertextové odkazy, které propojují jednotlivé části aplikace, dávají větší část zodpovědnosti za navigaci do rukou uživatelů a umožňují nelineární využití aplikace.

Právě kvůli odlišným možnostem uživatelského rozhraní a především naprosto odlišnému navigačnímu modelu začaly vznikat metodiky určené speciálně pro vývoj webových aplikací. Několika významným webovým metodikám budou věnovány další podkapitoly.

2.3. *Úvod do metodik pro vývoj webových aplikací*

2.3.1. Co jsou to hypermédia?

Dříve než si představíme jednotlivé vybrané metodiky, je nutné seznámit se s metodikami pro vývoj webových aplikací obecně. Tyto metodiky nebývají často uváděny jako „*metodiky pro vývoj webových aplikací*“, „*metodiky pro tvorbu webu*“ nebo „*webové metodiky*“, ale lze je nalézt spíše pod označením „*metodiky pro vývoj hypermediálních aplikací*“, popř. „*hypermédia metodiky*“. Pojmy hypermediální a webové aplikace budeme pro účely této práce považovat za synonyma.

Podle (Dolog, Bieliková, 2002) si lze pod pojmem hypermediální aplikace představit aplikace, jejichž cílem je získávání, zobrazování, manipulace a přístup k informacím v různých formátech (např. texty, obrázky, zvuky či videa). Největšího rozmachu dosáhly tyto aplikace s nástupem internetu. Na rozdíl od klasických aplikací jsou vystavěny na možnosti nesequenční navigace, tj. nemusí být stanovena přesná posloupnost toho, jak mají být jednotlivé části aplikace využívány (např. jednotlivé kusy textu čteny), ale záleží na uživateli, kterou cestou se vydá. Navíc jsou hypermediální aplikace zaměřeny více na uživatelské rozhraní a možnosti personalizace obsahu a funkčnosti. Připojíme-li k tomu ještě častou práci s nestrukturovanými informacemi, je více než jasné, že pro vývoj těchto aplikací je nutné využít odlišné postupy než při vývoji klasických aplikací.

2.3.2. Charakteristiky metodik

Ve výše zmíněném článku (Dolog, Bielíková, 2002) se autoři pokusili vytvořit sadu tzv. *perspektiv*, pomocí kterých je možné charakterizovat jednotlivé metodiky. Tyto perspektivy jsou vzájemně provázané a představují různé úhly pohledu na metodiky. Mezi uvedené perspektivy patří:

- perspektiva aplikačních vrstev,
- perspektiva vývojového procesu,
- aspektová perspektiva,
- perspektiva formálnosti,
- perspektiva užitých technik a notací.

Perspektiva aplikačních vrstev se zabývá otázkou „*Co modelovat?*“. Nejčastěji jsou využívány tři vrstvy – aplikační doména, navigace a prezentace. Model aplikační domény určuje, jaké informace a v jaké struktuře budou nabízeny uživatelům aplikace. Navigační model poté definuje cesty a způsoby, kterými se uživatel může pohybovat mezi jednotlivými bloky informací. Prezentační model nakonec specifikuje uživatelské rozhraní pro práci s aplikací.

Perspektiva vývojového procesu se zaměřuje na otázku „*Kdy modelovat?*“, tj. které modelovací techniky použít v jednotlivých fázích vývoje (sběr požadavků, analýza, návrh a implementace). Aspektová perspektiva klasifikuje metodiky podle toho, zda slouží spíše pro modelování struktury systému a vztahů mezi částmi struktury nebo jsou cíleny více na modelování chování uživatelů systému a vzájemnou interakci mezi systémem a uživatelem. Perspektiva formálnosti umožňuje dělit metodiky a jednotlivé modelovací metody na formalizované, poloformalizované a neformalizované. Poslední perspektiva se zabývá technikami použitými při modelování a také notacemi, které jsou užity pro záznam modelování.

2.3.3. Způsob prezentace metodik

Jednotlivé metodiky budou uvedeny v pořadí od nejstarší (OOHDM) po nejnovější (WebML). Metodiky zařazené do této práce jsou pouze výběrem ze všech existujících metodik pro vývoj hypermediálních (webových) aplikací. Cílem tohoto výběru bylo zahrnout především metodiky, které jsou často citované a neustále aktivně vyvíjené.

U každé metodiky budou na začátku představeny základní informace ve struktuře dle tabulky 2.1.

Celý název	Kompletní název metodiky
Autoři	Seznam autorů první verze metodiky
Rok vzniku	Rok vzniku první verze metodiky
URL adresa metodiky	URL adresa věnovaná metodice a spravovaná tvůrci metodiky
Zdroje informací	Přehled nejdůležitějších zdrojů vztahujících se k metodice
Přínosy	Přehled nejvýznamnějších přínosů metodiky
Nedostatky	Přehled nejvýznamnějších nedostatků metodiky
Aplikační podpora	Názvy aplikací, frameworků či doplňků sloužících jako podpora dané metodiky

Tabulka 2.1: Charakteristiky metodik

Poté bude následovat shrnutí historického vývoje metodiky od jejího vzniku, přes důležité milníky až po současnost a také nastínění předpokládaných směrů vývoje v budoucnu. Zbytek kapitoly bude věnován představení vývojového procesu definovaného metodikou.

2.4. OOHDM

2.4.1. Základní informace a historie

Celý název	Object-oriented Hypermedia Design Methodology
Autoři	D. Schwabe, G. Rossi, S. D. J. Barbosa
Rok vzniku	1995
URL adresa metodiky	http://www.tecweb.inf.puc-rio.br/oohdm
Zdroje informací	(Schwabe, Rossi, Barbosa 1995), (Schwabe, Rossi, 1998), (Guell, Schwabe, Vilian, 2000), (Molhanec, 2001), (Lima, Schwabe, 2003), (Schwabe, et al., 2004)
Přínosy	• první objektově orientovaná metodika • zdroj, ze kterého vychází další metodiky

Nedostatky	• metodika původně určená pro vývoj hypermediálních aplikací (není specializovaná přímo na vývoj webových aplikací)
Aplikační podpora	OOHDM-Web

Tabulka 2.2: Základní informace o metodice OOHDM

Metodika OOHDM vznikla v roce 1995 na univerzitě PUC v Rio de Janeiru. Za první článek, který se metodikou zabýval a představil ji světu lze považovat (Schwabe, Rossi, Barbosa 1995). Následně v roce 1998 byla provedena první revize metodiky, která byla publikována v (Schwabe, Rossi, 1998). Druhá revize metodiky byla provedena v roce 2000 a zabývala se přidáním fáze sběru uživatelských požadavků do vývojového procesu, viz (Guell, Schwabe, Vilian, 2000).

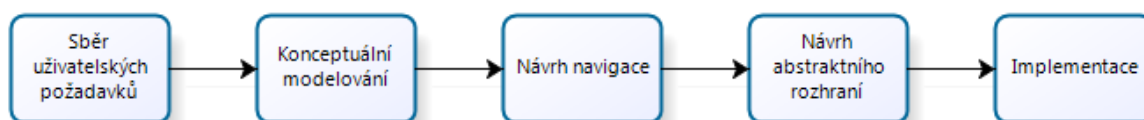
V posledních letech se tvůrci metodiky zaměřují na dvě hlavní větve vývoje:

- využití standardního frameworku Java 2 (J2EE),
- rozvoj nové metodiky SHDM.

Metodika SHDM (Semantic Hypermedia Design Method) je postavená na základech OOHDM a snaží se implementovat principy sémantického webu. Více o SHDM se lze dozvědět z (Lima, Schwabe, 2003) a (Schwabe, et al., 2004).

2.4.2. Vývojový proces

Vývojový proces popsaný metodikou OOHDM se dle (Schwabe, Rossi, 1998) skládá ze čtyř fází. Doplníme-li do tohoto procesu i sběr uživatelských požadavků z (Guell, Schwabe, Vilian, 2000), dostáváme pětifázový proces zobrazený na obrázku 2.5.



Obrázek 2.5: Vývojový proces dle OOHDM

Fáze sběru uživatelských požadavků se skládá z několika částí:

- identifikace rolí a úloh,
- specifikace scénářů,
- specifikace případů užití,
- specifikace diagramů uživatelských interakcí,
- validace případů užití a diagramů uživatelských interakcí.

Cílem následující fáze, konceptuálního modelování, je vytvoření schématu, které bude reprezentovat vybrané objekty, jejich vztahy a spolupráci. Samotné schéma je tvořeno třídami, vztahy a subsystémy, přičemž notace je velmi podobná notaci UML. Lze také využívat možnosti agregace a generalizace/specializace tříd. Třídy na rozdíl od UML mohou také obsahovat vícetypové atributy¹. Pro dokumentaci jsou využívány *Class and Relationship Cards*, které jsou podobné jako CRC karty².

Na základě sběru požadavků a konceptuálního modelu se provádí návrh navigace. Prvním krokem je vyřešení navigace v rámci jednotlivých případů užití. Poté, co jsou navržena všechna dílčí navigační schémata, pokračuje se návrhem celkové navigace. Navigační schémata představují ve skutečnosti různé pohledy na konceptuální schéma v závislosti na tzv. *kontextu*. Kontext bývá nejčastěji vyjádřen jako určitý druh filtru, pomocí kterého lze klasifikovat položky (více viz (Guell, Schwabe, Vilian, 2000)).

Předposlední fází je návrh uživatelského rozhraní. V OOHDM se používají tzv. *Abstract Data View (ADV)*, které umožňují oddělení navigace a prezentace a zobrazovat tak stejnou navigační strukturu různými způsoby. Technicky jsou ADV objekty, které si uchovávají svůj stav, rozhraní, organizaci a chování. Jednotlivé ADV lze navíc skládat z jiných ADV, a to i hierarchicky.

¹ Jedná se atributy, jejichž typ není stanoven jednoznačně, ale je definován výčtem možných typů. Například hypotetický atribut *PrirazenaMedia* třídy *Clanek* může mít nastaven jako typ hodnoty *Obrazek*, *Zvuk*, *Video*. Znamená to, že tento atribut může obsahovat hodnotu jakéhokoliv výše uvedeného typu.

² *Class-Responsibility-Collaboration* karty – využívané při návrhu objektově-orientovaných aplikací.

Poslední fází je implementace dříve navrhnutých modelů. V rámci implementace je nutné:

- namapovat položky z jednotlivých modelů na reálné položky (soubory, databáze),
- implementovat uchovávání stavových informací (např. pomocí cookies),
- implementovat uživatelské rozhraní pomocí technologií umožňujících generování dynamických stránek (např. PHP, ASP.NET, JSP).

2.5. SOHDM

2.5.1. Základní informace a historie

Celý název	Scenario-based Object-oriented Hypermedia Design Methodology
Autoři	Ch. Lee, H. Lee
Rok vzniku	1996
URL adresa metodiky	není
Zdroje informací	(Lee, et al., 1996), (Lee, Lee, 2000)
Přínosy	• vývoj postaven na využití scénářů chování uživatelů
Nedostatky	• málo informačních zdrojů o metodice
Aplikační podpora	SHDMTool

Tabulka 2.3: Základní informace o metodice SOHDM

Metodika SOHDM³ byla původně vytvořena roku 1996 na Korea Advanced Institute of Science and Technology (KAIST) a publikována v (Lee, et al., 1996). V roce 2000 byla částečně přepracována v (Lee, Lee, 2000), aby odpovídala tehdejším požadavkům na vývoj hypermediálních aplikací.

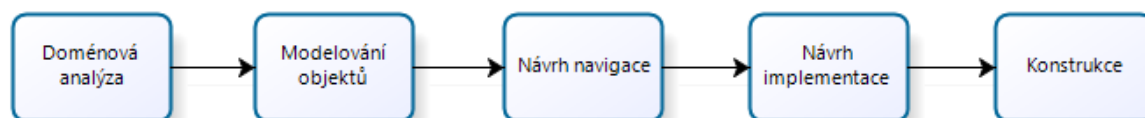
Tvůrci první verze metodiky společně s dalšími kolegy z KAIST mimo jiné vytvořili ještě další dvě metodiky, které se zabývají vývojem hypermediálních aplikací. Jedná se o:

- Workflow-based Hypermedia Development Methodology (Lee, Suh, 1998),
- View-based Hypermedia Design Methodology (Lee, et al., 1999).

³ Někdy též užívána zkratka SHDM, která je však zaměnitelná s metodikou Semantic Hypermedia Design Method, která je jednou z vývojových větví metodiky OOHDM.

2.5.2. Vývojový proces

Aktuální verze metodiky SOHDM prezentovaná v (Lee, Lee, 2000) definuje pět základních fází vývojového cyklu (viz obrázek 2.6). Oproti původní verzi metodiky ubyla samostatná fáze tzv. *návrhu pohledů*, jelikož byla připojena na konec fáze modelování objektů.



Obrázek 2.6: Vývojový proces dle SOHDM

Doménová analýza si klade za cíl vymezit hranice modelované aplikace a shromáždit byznys požadavky na tuto aplikaci. Prvním krokem je využití *System Scope Diagramu (SSD)*⁴, který umožňuje definovat hranice aplikace, určit externí entity a události, které tyto entity mohou vyvolávat při interakci s aplikací. Na základě informací získaných ze SSD jsou poté vytvářeny tzv. *scénáře*, které umožňují zachytit klíčové byznys procesy. Tyto scénáře jsou zaznamenávány přirozeným jazykem do předpřipravených tabulek (viz tabulka 2.4), přičemž pro každý scénář je vytvořena jedna tabulka.

Číslo	Aktor	Aktivita	Informace	Vstupní podmínka	Výstupní podmínka	Popis
Číslo aktivity	Entita, která spouští aktivitu	Operace spuštěná aktorem	Informace potřebné pro spuštění aktivity	Podmínka, která musí být splněna pro spuštění aktivity	Podmínka, která musí být splněna, aby mohla být aktivita ukončena	Případný dodatečný popis aktivity

Tabulka 2.4: Vzor tabulky pro záznam scénářů v SOHDM

Ve fázi modelování objektů se provádí transformace scénářů z předchozí fáze do podoby *Class-Responsibility-Collaboration (CRC)* karet. CRC karta obsahuje informace o:

- názvu a atributech třídy,
- odpovědnostech a kolaboracích třídy,
- nadtřídách, podtřídách, asociacích a komponentách třídy.

⁴ Diagram SSD je založen na Data Flow Diagramu.

Kromě samotných CRC karet se také využívá *Class Structure Diagram (CSD)*, který slouží k přehlednějšímu zobrazení vazeb mezi jednotlivými třídami.

Z CRC karet a CSD poté vychází návrh navigace. Základními navigačními jednotkami jsou pohledy, které jsou vytvořeny nad třídami a jejich vztahy. Tyto pohledy jsou poté seskupovány do *Access Structure Nodes (ASN)*. Mezi různými pohledy a ASN je následně vytvořena navigační síť na základě vytvořených scénářů. Závěrečným krokem v rámci návrhu navigace je vytvoření struktury jednotlivých stránek a specifikace toků mezi stránkami.

Poslední návrhovou částí je návrh implementace aplikace. V rámci této fáze je specifikováno mapování tříd vyjádřených CRC kartami na databázové tabulky a také mapování navigačních struktur na prvky uživatelského rozhraní (např. elementy HTML). Jakmile je navržena implementace, je možné postoupit do fáze konstrukce, ve které se provede implementace vytvořených návrhů v konkrétním implementačním prostředí.

2.6. WSDM

2.6.1. Základní informace a historie

Celý název	Web Site Design Method
Autoři	O. De Troyer, C. Leune
Rok vzniku	1998
URL adresa metodiky	http://wsdm.vub.ac.be/
Zdroje informací	(De Troyer, Leune, 1998), (De Troyer, Casteleyn, 2001a), (De Troyer, Casteleyn, 2001b), (De Troyer, Casteleyn, 2003), (Plessers, Casteleyn, De Troyer, 2005), další zdroje ze seznamu publikací ⁵
Přínosy	• vývoj postaven na detailní analýze uživatelů • velké množství informačních zdrojů o metodice
Nedostatky	• žádné významné nedostatky
Aplikační podpora	není

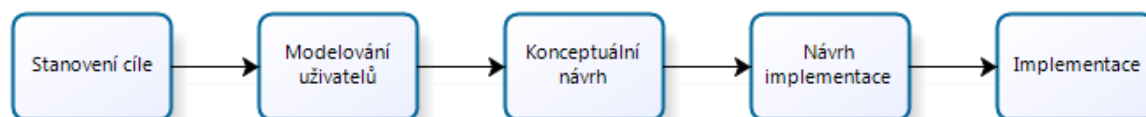
Tabulka 2.5: Základní informace o metodice WSDM

⁵ <http://wsdm.vub.ac.be/Research/publications.php>

Metodika WSDM vznikla na Tilburg University v roce 1998 a byla poprvé představena v (De Troyer, Leune, 1998). V roce 2001 byly možnosti metodiky a vývojový proces demonstrovány na příkladu aplikace pro správu příspěvků zasílaných autory na konferenci, viz (De Troyer, Casteleyn, 2001a) společně s přílohou (De Troyer, Casteleyn, 2001b). V (De Troyer, Casteleyn, 2003) byla poté publikována aktualizovaná verze metodiky, která měla za cíl umožnit lepší podporu pro modelování komplexních byznys procesů. V současné době probíhají především práce na zavádění principů sémantického webu do metodiky WSDM, viz (Plessers, Casteleyn, De Troyer, 2005).

2.6.2. Vývojový proces

Při popisu vývojového procesu definovaného metodikou WSDM budeme vycházet z (De Troyer, Casteleyn, 2003). V tomto článku je představen proces, který se skládá z pěti základních vývojových fází (viz obrázek 2.7).



Obrázek 2.7: Vývojový proces dle WSDM

První a druhá fáze procesu jsou velice těsně propojeny. V rámci první fáze je stanoven cíl webu, předmětná oblast, kterou web pokrývá, a také jsou specifikováni cíloví uživatelé webu. V návaznosti na závěry z této fáze je provedeno dvoukrokové modelování uživatelů webu. Prvním krokem je provedení identifikace a klasifikace možných uživatelů webu, přičemž platí, že členové stejné třídy mají stejné informační i funkční požadavky. V druhém kroku jsou poté jednotlivé třídy uživatelů podrobněji charakterizovány. Výstupem fáze modelování uživatelů je hierarchie uživatelských tříd společně s jejich charakteristikami a požadavky na informace, funkce, navigaci a použitelnost.

Fáze konceptuálního návrhu je rozdělena na dvě hlavní části:

- modelování úloh,
- návrh navigace.

Cílem modelování úloh je detailní analýza informačních a funkčních požadavků jednotlivých tříd uživatelů, které byly definovány ve fázi modelování uživatelů. K tomu jsou využívány modely úloh a tzv. *Object Chunks*. Prvním krokem je specifikace úlohy pro každý požadavek a následná dekompozice této úlohy na elementární úkony. K tomu je využíváno notace *ConcurTaskTrees (CTT)*, která byla vyvinuta v kontextu *Human-Computer Interaction*⁶. Výstupem CTT je hierarchický strom skládající se ze čtyř typů úkonů⁷:

- uživatelské,
- aplikační,
- interakční,
- abstraktní.

Na CTT poté navazuje tvorba výše zmíněných *Object Chunks*. Pro každý elementární úkon definovaný v CTT je vytvořen právě jeden *Object Chunk*, který modeluje informační anebo funkční požadavky daného úkonu. Při menším rozsahu lze *Object Chunks* modelovat graficky, jinak je možné využít tabulkové formy či pseudokódu.

Druhou částí konceptuálního návrhu je návrh navigace. Cílem návrhu navigace je vytvoření konceptuální struktury webu a modelování toho, jak se budou moci členové jednotlivých uživatelských tříd pohybovat v rámci webu. K modelování se využívá tzv. *Navigation Track*, který je vytvořen pro každou třídu uživatelů a snaží se uspokojit jejich navigační požadavky. *Navigation Track* lze chápat jako určitou podmnožinu webu, která obsahuje informace a funkcionalitu požadovanou určitou třídou uživatelů. Na závěr je vytvořena konceptuální struktura webu, která zohledňuje všechny *Navigation Tracks* a vazby mezi nimi.

Čtvrtou fází vývojového procesu definovaného metodikou WSDM je návrh implementace. V této fázi je navržena struktura a design jednotlivých webových stránek. Návrh je založen na požadavcích na použitelnost a na charakteristikách jednotlivých uživatelských tříd, které byly vytvořeny ve fázi modelování uživatelů. Poslední fází je samotná implementace návrhů z předchozích fází.

⁶ Obor zabývající se problematikou interakce mezi člověkem a počítačem.

⁷ Podrobnější popis CTT a jednotlivých typů úkonů viz (De Troyer, Casteleyn, 2003)

2.1. WebML

2.1.1. Základní informace a historie

Celý název	Web Modeling Language
Autoři	S. Ceri, P. Fraternali
Rok vzniku	1998
URL adresa metodiky	http://www.webml.org
Zdroje informací	(Ceri, Fraternali, Bongio, 2000), (Ceri, Fraternali, Matera, 2002), (Brambilla, et al., 2007), další zdroje ze seznamu publikací ⁸
Přínosy	• velice propracovaný návrh navigace a uživatelského rozhraní • velké množství informačních zdrojů o metodice • podpora vývojového procesu aplikací WebRatio
Nedostatky	• žádné významné nedostatky
Aplikační podpora	WebRatio

Tabulka 2.6: Základní informace o metodice WebML

Metodika WebML vznikla v roce 1998 na Polytechnice v Miláně. První komplexní pojednání o této metodice lze ovšem nalézt až v (Ceri, Fraternali, Bongio, 2000) a poté v (Ceri, Fraternali, Matera, 2002). Metodika během svého vývoje prošla čtyřmi verzemi:

- *WebML 1* – zaměření na specifikaci uživatelského rozhraní a navigace pro informační weby⁹,
- *WebML 2* – přidána podpora byznys operací umožňujících např. správu obsahu, autorizaci, autentizaci,
- *WebML 3* – zavedení konceptu pluginů, které umožnilo otevření jazyka pro potřeby jednotlivých webových analytiků a návrhářů,
- *WebML 4* – aktuální verze, která přidává podporu pro *Model-driven Development (MDD)*¹⁰, stejně tak jako možnost interakce s webovými službami či možnost specifikování workflow pro webovou aplikaci.

⁸ <http://www.webml.org/webml/page93.do>

⁹ Weby, které poskytují pouze informace a nenabízí uživateli funkcionalitu pro interakci se systémem.

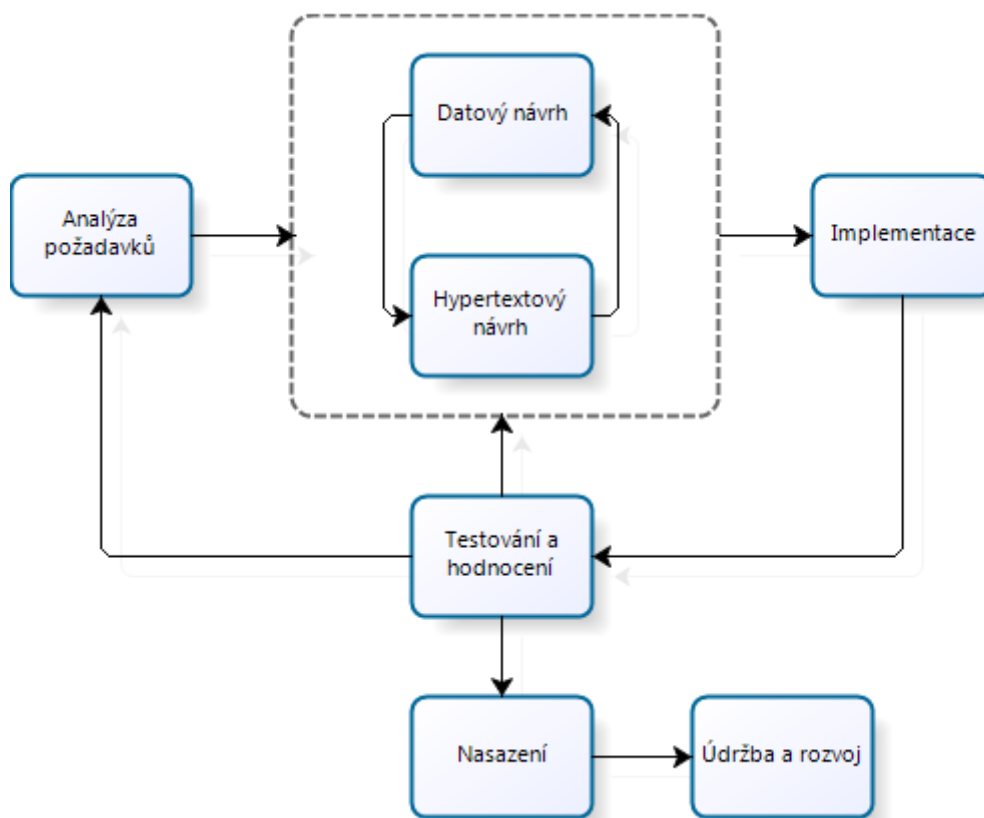
¹⁰ Přístup k vývoji aplikací, který je postaven a klade důraz na využití modelů.

Aktuální verze byla publikována v (Brambilla, et al., 2007). V posledních letech je úsilí tvůrců metodiky zaměřeno především na následující problematiku:

- implementace principů sémantického webu,
- umožnění vývoje webových aplikací založených na byznys procesech,
- podpora webových aplikací typu *Web 2.0*¹¹ a *Rich Internet Applications (RIA)*¹².

2.1.2. Vývojový proces

Zde uvedený proces vývoje webu dle metodiky WebML (viz obrázek 2.8) vychází z (Brambilla, et al., 2007). Vývojový proces se skládá ze sedmi fází a je založen na iterativním a inkrementálním přístupu k vývoji aplikací.



Obrázek 2.8: Vývojový proces dle WebML (Brambilla, et al., 2007)

¹¹ Populární pojem užívaný pro označení nového směru vývoje internetu, který je charakterizován především sociálními sítěmi a sdílením informací.

¹² Nový přístup k vývoji webových aplikací, který se snaží zkombinovat výhody desktopových a webových aplikací. Základní myšlenkou je přenesení části zodpovědnosti ze serveru na webový prohlížeč. Využívány jsou technologie typu AJAX, Flash či Silverlight.

Cílem první fáze, analýzy požadavků na webovou aplikaci, je vymezení zaměření webové aplikace a analýza požadavků na funkcionalitu. Na vstupu celého vývojového procesu se nachází byznys požadavky, které vedly k rozhodnutí o vývoji webové aplikace. Hlavními výstupy této fáze jsou:

- identifikace skupin uživatelů, kteří mají stejné charakteristiky nebo hrají stejnou roli v rámci byznys procesů,
- specifikace požadavků na funkcionalitu pro jednotlivé skupiny uživatelů,
- identifikace základních informačních objektů,
- dekompozice webové aplikace na tzv. *pohledy*, které představují podmnožiny webové aplikace, které slouží k uspokojení požadavků jednotlivých uživatelských skupin.

Není stanovena žádná specifická notace, pro záznam jednotlivých výstupů. Mohou být použity jak neformální způsoby (textový popis, tabulky), tak i formalizované způsoby zápisu (např. UML diagramy).

Následující fází je datový návrh, který slouží k uspořádání základních informačních objektů identifikovaných v předchozí fázi do srozumitelného a konzistentního datového schématu, které může být navíc obohaceno o odvozené objekty. Odvozené objekty či atributy se zapisují pomocí jazyků typu *OQL*¹³ nebo *OCL*¹⁴. Pro záznam lze využít *E-R diagramy*¹⁵ nebo *UML diagramy tříd*.

V rámci třetí fáze, hypertextového návrhu, se provádí návrh uživatelského rozhraní, které je zobrazeno ve webovém prohlížeči. Hypertextový návrh vychází z pohledů definovaných v první fázi. Tyto pohledy jsou děleny na oblasti a oblasti dále na jednotlivé stránky nebo jiné podoblasti. Samotné stránky jsou poté složeny z tzv. *obsahových jednotek*, které mohou být pěti typů:

- *Data Unit* – obsahuje vybrané atributy vybrané entity,
- *Multidata Unit* – obsahuje vybrané atributy celé sady entit,

¹³ Object Query Language - dotazovací jazyk pro objektově-orientované databáze.

¹⁴ Object Constraint Language – deklarativní jazyk pro popis pravidel v UML.

¹⁵ Entity-Relationship diagramy - diagramy užívané pro modelování databází.

- *Index Unit* – obsahuje seznam entit s možností výběru,
- *Scroller Unit* – umožňuje procházení uspořádané sady entit,
- *Entry Unit* – formulář pro vstup uživatelských údajů.

Mezi jednotlivými obsahovými jednotkami i stránkami je poté vytvořena síť odkazů. Odkazy mezi obsahovými jednotkami jsou nazývány *kontextové*, jelikož jsou přenášeny parametry, naopak odkazy mezi stránkami jsou *nekontextové*. Mimo přenášených parametrů lze také pracovat s globálními proměnnými. Při potřebě rozsáhlejší interakce mezi uživatelem a aplikací lze využít tzv. *jednotek činností*, které reprezentují vytváření, úpravu a mazání položek či vytvoření nebo zrušení vazby mezi položkami. Pro podrobnější informace o obsahových jednotkách a jednotkách činností viz (Brambilla, et al., 2007).

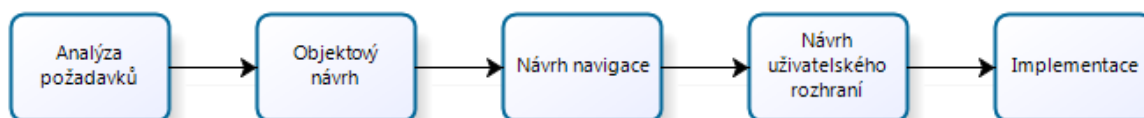
V následující fázi, při implementaci, je často využíván nástroj WebRatio¹⁶, který byl vytvořen přímo tvůrci metodiky. Tento nástroj umožňuje vizuální modelování datového i hypertextového návrhu, návrh layoutu uživatelského rozhraní i propojení modelů na fyzické datové zdroje. Následně je schopen z výše uvedených modelů a informací vygenerovat samotný kód webové aplikace, který je poté možné dále upravovat.

Pátou fází je testování a hodnocení implementované webové aplikace. Pokud jsou nalezeny nedostatky, vývojový proces se podle typu a vážnosti nedostatků přesouvá buď do fáze analýzy požadavků, nebo do fází datového a hypertextového návrhu. Je-li aplikace schválena, je poté možné ji nasadit přímo do provozního prostředí. Poslední fází vývojového procesu je následně údržba a rozvoj webové aplikace

2.2. Shrnutí kapitoly

Hlavním cílem této kapitoly bylo představení několika významných metodik pro vývoj webových aplikací. Ze všech možných metodik byly nakonec vybrány čtyři, které jsou často citované, neustále jsou aktivně vyvíjeny, přináší nové pohledy na vývoj webových aplikací a jsou použitelné i v praxi. Jedná se o metodiky OOHDM, SOHDM, WSDM a WebML. Ačkoliv vývojové procesy definované jednotlivými metodikami se liší, lze zde vypořádat určitý společný rámec. Jedná se o základní kostru procesu, kterou lze vyjádřit následujícím schématem (viz obrázek 2.9).

¹⁶ <http://www.webratio.com>



Obrázek 2.9: Zobecněný vývojový proces

V rámci analýzy požadavků se provádí sběr byznys požadavků na webovou aplikaci, identifikace skupin uživatelů a hrubá specifikace informačních a funkčních požadavků. Při objektovém návrhu jsou jednotlivé informační a funkční požadavky transformovány do datových schémat, která vyjadřují vztahy mezi jednotlivými entitami. Na základě těchto schémat a byznys požadavků je pak následně vybudována navigační struktura webové aplikace. Nad datovým a navigačním schématem je poté vystavěna prezentační vrstva. Ve fázi implementace jsou všechny výše uvedené modely a informace implementovány do výsledné webové aplikace.

Analyzujeme-li přínosy a nedostatky jednotlivých metodik, získáme výsledky, které jsou prezentovány v tabulce 2.7.

Metodika	Přínosy	Nedostatky
OOHDM	• první objektově orientovaná metodika • zdroj, ze kterého vychází další metodiky	• metodika původně určená pro vývoj hypermediálních aplikací (není specializovaná přímo na vývoj webových aplikací)
SOHDM	• vývoj postaven na využití scénářů chování uživatelů	• málo informačních zdrojů o metodice
WSDM	• vývoj postaven na detailní analýze uživatelů • velké množství informačních zdrojů o metodice	• žádné významné nedostatky
WebML	• velice propracovaný návrh navigace a uživatelského rozhraní • velké množství informačních zdrojů o metodice • podpora vývojového procesu aplikací WebRatio	• žádné významné nedostatky

Tabulka 2.7: Přínosy a nedostatky jednotlivých metodik

3. Návrh procesního modelu pro vývoj webu

Hlavním cílem této kapitoly je návrh procesního modelu pro vývoj webu (webové aplikace), na základě poznatků získaných analýzou metodik pro vývoj webových aplikací. Mezi sekundární cíle lze zařadit identifikaci nedostatků současných metodik z byznys pohledu a představení notace BPMN pro modelování procesů.

3.1. *Nedostatky současných metodik*

Ačkoliv by se na první pohled mohlo zdát, že metodiky uvedené v kapitole 2, jsou dostačující pro celý proces vývoje webu, existují určité aspekty, které tyto metodiky obecně nepokrývají či pokrývají minimálně. Při identifikaci těchto nedostatků budeme vycházet ze zobecněného vývojového procesu představeného na obrázku 2.9, který se skládá z pěti fází:

- analýza požadavků,
- objektový návrh,
- návrh navigace,
- návrh uživatelského rozhraní,
- implementace.

Jak je z těchto fází patrné, jádrem jednotlivých metodik je především samotný návrh a implementace webových aplikací. Takovýto proces je ovšem vhodný pouze pro webové aplikace, které jsou užívány vymezeným okruhem uživatelů a nečelí přímým konkurenčním tlakům, např. firemní intranetové portály. Jakmile se však jedná o web, jehož cílem je docílení co nejvyšší návštěvnosti a který se pohybuje v konkurenčním prostředí, je výše uvedený proces nedostatečný. Co je tedy potřeba v daném procesu změnit?

Do daného procesu je nutné implementovat především následující aktivity:

- analýza konkurence,
- zajištění technologického zázemí pro běh webové aplikace (např. výběr domény či implementačního prostředí),
- provoz webové aplikace s důrazem na monitoring návštěvnosti a využívání aplikace.

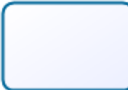
3.2. Notace BPMN

Dříve než bude proveden samotný návrh procesního modelu pro vývoj webových aplikací, je nutné představit notaci, pomocí které bude samotný procesní model vyjádřen. Vybrána byla notace *Business Process Modeling Notation (BPMN)*¹⁷, která představuje současný standard pro modelování byznys procesů. Tato notace byla vyvinuta organizací *Business Process Management Initiative (BPMI)* a v současnosti je spravována organizací *Object Management Group (OMG)*¹⁸. Aktuální verze je BPMN 1.2, přičemž probíhají práce na vývoji BPMN 2.0.

Při modelování pomocí BPMN jsou využívány čtyři typy elementů:

- Flow Objects,
- Connection Objects,
- Swimlanes,
- Artifacts.

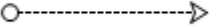
Přehled základních BPMN elementů vycházející z (White, 2004) je uveden v tabulce 3.1. Grafická reprezentace elementů pochází z aplikace BizAgi Process Modeler¹⁹.

Element	Popis	Grafická reprezentace
Flow Objects		
Start Event	Událost, která způsobuje start procesu	
Intermediate Event	Událost či stav, které nastávají v průběhu procesu	
End Event	Cílový stav procesu	
Task	Atomická činnost	
Sub-process	Činnost, která se skládá z jiných činností	

¹⁷ <http://www.bpmn.org>

¹⁸ <http://www.omg.org>

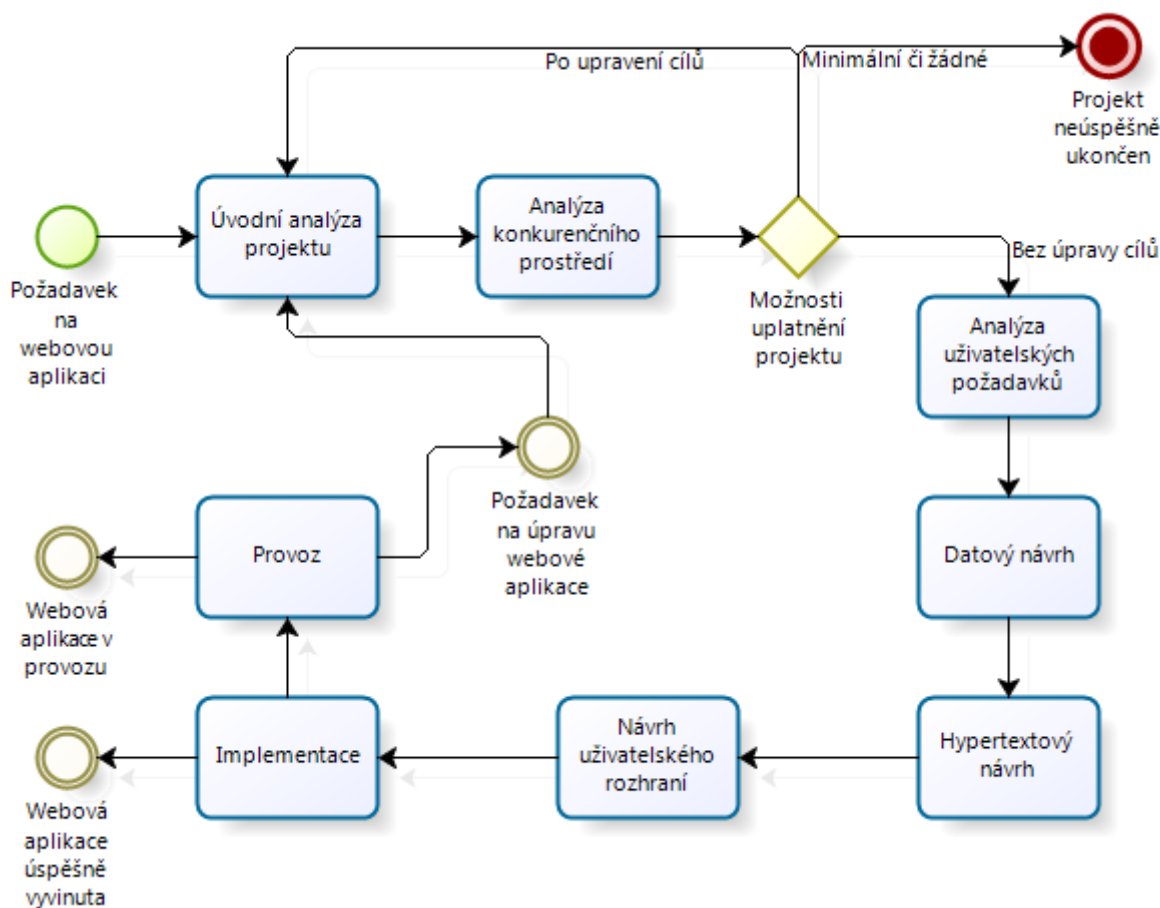
¹⁹ <http://www.bizagi.com>

Gateway	Kontrola procesu pomocí slučování a rozdělování jednotlivých procesních toků	
Connection Objects		
Sequence Flow	Slouží k propojení jednotlivých činností a určení pořadí vykonávání činností	
Message Flow	Slouží k určení toku zpráv mezi jednotlivými částmi procesu	
Association	Slouží k připojení vstupů a výstupů k aktivitám	
Swimlanes		
Pool	Představuje samostatný proces, který odpovídá jednomu účastníkovi (např. podniku) v rámci celého procesního modelu. Různé Pools mohou být propojeny pouze pomocí Message Flow.	
Lane	Představuje podmnožinu poolu (např. oddělení v rámci podniku). Mezi jednotlivými Lane mohou probíhat pouze Sequence Flow.	
Artifacts		
Data Object	Data použitá jako vstup nebo výstup činnosti	
Group	Slouží k seskupení činností pro dokumentační nebo analytické účely. Neovlivňuje žádným způsobem tok procesu.	
Annotation	Umožňuje připojení dodatečných textových informací k jednotlivým elementům.	

Tabulka 3.1: Přehled základních BPMN elementů (White, 2004)

3.3. Návrh hrubého procesního modelu

Na základě informací získaných z analýzy metodik pro vývoj webových aplikací a také s využitím identifikovaných nedostatků daných metodik jsem pro účely této práce navrhl procesní model pro vývoj webových aplikací. Schéma hrubého procesního modelu (procesní model na vysoké úrovni obecnosti) lze nalézt na obrázku 3.1.



Obrázek 3.1: Návrh procesního modelu pro vývoj webových aplikací

Jednotlivé fáze procesu budou podrobněji strukturovány a popsány v samostatných podkapitolách. U jednotlivých dílčích činností bude vždy uveden obsah činnosti, důležité výstupy a popř. také použitelné metody a doplňující zdroje informací.

Na tomto místě je potřeba nyní pouze vysvětlit význam zbylých elementů modelu. *Požadavek na webovou aplikaci* představuje vstupní bod celého procesního modelu, startovací událost, která vyjadřuje vznik potřeby jednotlivce, podniku či organizace vytvořit webovou aplikaci. Ačkoliv je požadavek na webovou aplikaci prvním elementem celého zde prezentovaného procesu, ve skutečnosti je vznik požadavku pouze jedním z možných výstupů předcházející byznys analýzy, která se snaží odpovědět na vzniklé byznys problémy či požadavky (např. nalezení nových prodejních kanálů, zlepšení propagace značky). Popis zmíněné byznys analýzy ovšem není předmětem této práce.

Element *Možnosti uplatnění projektu* vyjadřuje rozhodnutí, které by mělo určit na základě stanovených cílů a analýzy konkurence, zda se může daný projekt uplatnit v konkrétním konkurenčním prostředí. Výsledkem mohou být tři možná rozhodnutí:

- projekt se může uplatnit, aniž by bylo potřeba měnit stanovené cíle,
- projekt má možnost se uplatnit, ale musí být přeformulovány některé cíle,
- projekt má minimální či žádnou šanci na uplatnění, i kdyby byly některé cíle přeformulovány.

Projekt je uplatnitelný v konkurenčním prostředí bez nutné změny cílů (větev *Bez úpravy cílů*) obecně tehdy, jestliže:

- se zaměřuje na málo konkurenční oblast,
- se zaměřuje na úzce specifikovanou podoblast více konkurenční oblasti,
- přináší nové služby, informace či přístupy k řešení problémů.

Jestliže projekt nespadá ani do jedné z výše uvedených tří kategorií, avšak je možné částečným přeformulováním cílů dosáhnout toho, že se do jedné z kategorií zařadí, pokračuje se větví *Po upravení cílů*. Pokud existuje minimální či žádná šance na uplatnění projektu, aniž by byly všechny cíle kompletně přeformulovány, pokračuje se větví *Minimální či žádné*, která ústí v neúspěšné ukončení projektu neboli cílový stav *Projekt neúspěšně ukončen*.

V hrubém procesním modelu se poté vyskytují ještě tři stavy. Průběžný stav *Webová aplikace úspěšně vyvinuta* vyjadřuje úspěšné dokončení implementace webové aplikace, průběžný stav *Webová aplikace v provozu* vyjadřuje situaci, kdy je aplikace naimplementována, provozována a je prováděn její monitoring a průběžný stav *Požadavek na úpravu webové aplikace*, který vyjadřuje situaci, kdy výsledky webové aplikace neodpovídají stanoveným cílům, a je nutné webovou aplikaci upravit.

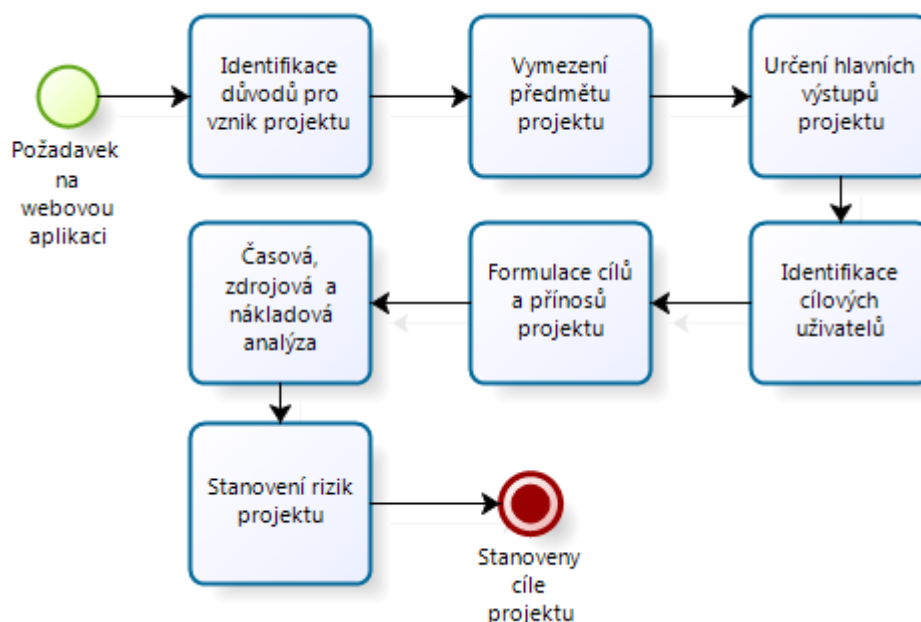
Jak je patrné ze schématu navrženého procesního modelu (viz obrázek 3.1), jediným konečným stavem procesu je *Neúspěšné ukončení projektu*. Neznamená to ovšem, že každý projekt končí automaticky neúspěšně, ale vychází to z charakteru životního cyklu webové aplikace. Webové aplikace totiž na rozdíl od desktopových aplikací musí být neustále monitorovány a upravovány tak, aby odpovídaly definovaným cílům, a proto je zde využit

cyklický proces, který tuto skutečnost vyjadřuje. *Neúspěšné ukončení projektu* představuje jediný únikový bod z tohoto cyklu a vyjadřuje skutečnost, že webová aplikace ani po provedených úpravách není schopná dosáhnout stanovených cílů.

3.4. Fáze Úvodní analýza projektu

3.4.1. Vývojový proces

První fází celého vývojového procesu je úvodní analýza projektu. V rámci této fáze je nutné zhodnotit současnou situaci, specifikovat předmět a zaměření projektu a také nastínit očekávané cíle a přínosy projektu. Proces *Úvodní analýza projektu* je vyjádřen na obrázku 3.2.



Obrázek 3.2: Vývojový proces fáze Úvodní analýza projektu

3.4.2. Identifikace důvodů pro vznik projektu

Prvním krokem je identifikace důvodů, které vedly k požadavku na vznik projektu, tj. k požadavku na vývoj webové aplikace. Mezi tyto důvody může patřit například v současnosti neexistující či zastaralá webová prezentace podniku, snaha o získání nového prodejního kanálu pro vlastní produkty a služby či nutnost vytvoření vnitropodnikového portálu. V závislosti na důvodu vzniku požadavku na webovou aplikaci se poté mírně liší i úhel pohledu na další kroky vývojového procesu, jelikož je zřejmé, že existují určité rozdíly např. mezi vývojem internetového obchodu a webových stránek střední školy. Hlavním výstupem je seznam důvodů pro vznik projektu.

3.4.3. Vymezení předmětu projektu

V návaznosti na identifikaci důvodů pro vznik projektu je nutné vymezit předmět projektu, který v obecnější formě vyjadřuje nejdůležitější činnosti prováděné v průběhu vývojového procesu. Společně s určením předmětu projektu je vhodné též explicitně vyjádřit, co předmětem projektu není, např. projekt se nezaměřuje na školení uživatelů vnitropodnikového portálu nebo projekt nepokrývá monitorování návštěvnosti webových stránek. Výstupem je seznam hlavních činností pokrytých projektem a seznam činností, které explicitně nejsou součástí projektu.

3.4.4. Určení hlavních výstupů projektu

Na základě informací získaných z předchozích dvou kroků je možné určit hlavní výstupy projektu. Mezi hlavní výstupy bude vždy patřit vyvíjená webová aplikace. Společně s ní mohou do hlavních výstupů patřit například:

- analýza konkurenčního prostředí,
- detailní analýza uživatelů a jejich požadavků,
- analýza návštěvnosti webových stránek.

Jednotlivé hlavní výstupy projektu je možné dále strukturovat a vytvořit tak hierarchii výstupů. Výstupem tohoto kroku je hierarchický přehled výstupů společně s popisem jednotlivých výstupů.

3.4.5. Identifikace cílových uživatelů

Dalším krokem je základní identifikace cílových uživatelů. V rámci tohoto kroku jsou na obecné úrovni identifikovány možné externí skupiny uživatelů webové aplikace. Nejsou prozatím detailně analyzovány uživatelské požadavky či podrobně charakterizovány jednotlivé skupiny. Budeme-li například vyvíjet webovou aplikaci zaměřenou na zprostředkování prodeje nemovitostí, lze jako základní cílové uživatele identifikovat subjekty nabízející nemovitosti a subjekty, které naopak nemovitosti poptávají. Uživatelských skupin či rolí reálně využitých ve webové aplikaci bude pravděpodobně více (např. administrátoři, content manažeři), ale v současné fázi vývojového procesu je dostatečná tato klasifikace. Hlavním výstupem tohoto kroku je seznam skupin cílových uživatelů společně s jejich základní charakteristikou.

3.4.6. Formulace cílů a přínosů projektu

Velice důležitým krokem je formulace cílů a přínosů projektu. Rozdíl mezi cílem a přínosem tkví v tom, že cíl je určitý očekávaný stav, kterého by měl projekt dosáhnout v okamžiku svého úspěšného dokončení, zatímco přínos je určitý očekávaný stav, který by měl nastat až v časovém období po ukončení realizace projektu. Cíle a přínosy by měly vycházet z poznatků získaných v předchozích krocích.

Formulované cíle a přínosy by měly splňovat tzv. *SMART kritéria*. Zkratka SMART je tvořena úvodními písmeny jednotlivých kritérií. Složení těchto kritérií není konzistentní, jelikož různé zdroje uvádí různá kritéria. Přehled možných používaných kritérií lze nalézt v tabulce 3.2.

Písmeno	Nejpoužívanější kritérium	Další užívaná kritéria
S	<i>Specific</i> (specifický)	<i>Significant</i> (významný), <i>Stretching</i> (flexibilní), <i>Simple</i> (jednoduchý)
M	<i>Measurable</i> (měřitelný)	<i>Meaningful</i> (smysluplný), <i>Motivational</i> (motivační), <i>Manageable</i> (řiditelný)
A	<i>Attainable</i> (dosažitelný)	<i>Appropriate</i> (odpovídající), <i>Achievable</i> (dosažitelný), <i>Agreed</i> (odsouhlasený), <i>Assignable</i> (přiřaditelný), <i>Actionable</i> (motivující), <i>Action-oriented</i> (zaměřený na akci), <i>Ambitious</i> (ambiciózní)
R	<i>Relevant</i> (relevantní)	<i>Realistic</i> (realistický), <i>Results/Results-focused/Results-oriented</i> (výsledkově orientovaný), <i>Resourced</i> (zajistitelný zdroji), <i>Rewarding</i> (užitečný)
T	<i>Time-bound</i> (časově ohraničený)	<i>Time framed</i> , <i>Timed</i> , <i>Time-based</i> , <i>Timeboxed</i> , <i>Timely</i> , <i>Timebound</i> , <i>Time-Specific</i> , <i>Timetabled</i> (časově vymezený), <i>Trackable</i> (sledovatelný), <i>Tangible</i> (konkrétní)

Tabulka 3.2: SMART kritéria (Wikipedia, 2007b)

Kritérium specifičnosti stanovuje, že cíl musí být jednoznačně formulován, aby nemohlo dojít k vícenásobným interpretacím. Specifický cíl musí být srozumitelný každému, kdo má alespoň základní povědomí o projektu. Každý cíl by měl být také měřitelný, aby mohlo být v každém okamžiku projektu stanoveno, z kolika procent byl cíl splněn. K zajištění měřitelnosti cílů jsou využívány tzv. *metriky*. Více o metrikách a procesech měření hodnot metrik se lze dozvědět z (Maryška, 2006).

Dalším kritériem, které je třeba uvažovat při formulování cílů, je dosažitelnost cíle. Toto kritérium tvrdí, že každý stanovený cíl musí být reálně dosažitelný s dostupnými zdroji. Cíle musí být také relevantní k zaměření projektu a musí být napojeny na cíle celého podniku či organizace. Posledním nejpoužívanějším kritériem pro formulaci cílů je časová ohraničenost cílů. Toto kritérium se zaměřuje na časové vymezení, které je určené pro naplnění jednotlivých cílů.

Při formulování cílů je také velmi vhodné vyvarovat se použití některých slov či slovních spojení. Přehled 47 výrazů, jimž je lepší se vyhnout při správném formulování cílů, lze nalézt v (van den Broek, 2007). Mezi tyto výrazy patří např. *pokusit se, doufat, přát si, připravit, brzy, přibližně nebo někdy*.

Výstupem tohoto kroku je seznam definovaných cílů a přínosů se stanovenými termíny splnění a nastavenými metrikami.

3.4.7. Časová, zdrojová a nákladová analýza projektu

Dalším velmi důležitým krokem v úvodní fázi procesu je časová, zdrojová a nákladová analýza projektu. Časová analýza spočívá v naplňování projektu z hlediska jeho časového vymezení. Je nutné stanovit termíny pro zahájení a ukončení projektu, určit délky trvání projektových fází i jednotlivých činností a v neposlední řadě také stanovit časové rezervy²⁰ projektu. Pro časovou analýzu lze využít metod založených na teorii grafů, např. *CPM*²¹, *PERT*²² nebo *MPM*²³. Pro grafické znázornění časové analýzy je možné využít tzv. *Ganttovy diagramy*.

²⁰ Časové rezervy jsou chápány jako možné odchylky od naplánovaných termínů.

²¹ Critical Path Method – deterministická metoda využívající hranově ohodnocené grafy, jejímž cílem je stanovení doby trvání projektu na základě nalezení tzv. kritické cesty grafem.

²² Program Evaluation and Review Technique – stochastická metoda využívající hranově ohodnocené grafy založená na odhadu délky trvání činností na základě tří odhadů (optimistický, nejpravděpodobnější a pesimistický).

²³ Metra Potential Method – deterministická metoda využívající uzlově ohodnocené grafy, která umožňuje definovat čtyři druhy možných vazeb mezi činnostmi. Často využívána v aplikacích pro podporu plánování projektů, např. Microsoft Project.

Z časové analýzy vychází druhá analytická část, zdrojová analýza. Cílem zdrojové analýzy je nalezení optimálního rozložení využití dostupných zdrojů v průběhu projektu. Poslední částí je analýza nákladů na projekt. Tyto náklady lze rozdělit na přímé, tj. takové, které jsou založeny na časové a zdrojové analýze (např. mzdové náklady), a nepřímé, neboli náklady, které přímo nevyplývají z časové náročnosti projektu (např. softwarové licence, webhosting či finanční rezervy).

Mezi hlavní výstupy tohoto kroku patří:

- časový harmonogram projektu,
- přehled využívaných zdrojů,
- strukturovaný přehled nákladů na projekt.

Více informací o analýze projektů lze nalézt v (Beranová, 2008).

3.4.8. Stanovení rizik projektu

Formulování rizik projektu je poslední krok v rámci první fáze *Úvodní analýza projektu*. Dle (LBMS, 2006) lze identifikovat pět základních druhů rizik:

- *organizační* – spojena s nutností organizačních změn k úspěšnému dokončení projektu,
- *plánovací* – spojena s naplánovanými termíny a s využitím zdrojů,
- *technická* – spojena s náročností a jednoznačností technických a technologických požadavků na projekt,
- *věcného rámce* – spojena s překračováním rozpočtu a neujasněním přínosů projektu,
- *externí závislosti* – spojena se závislostí na externích dodavatelích a nedostatkových zdrojích.

Identifikace rizik je ovšem pouze jedna z fází komplexního řízení rizik. Celý proces řízení rizik se dle (LBMS, 2006) skládá z těchto fází:

- plánování řízení rizik,
- identifikace rizik,
- kvalitativní analýza rizik,

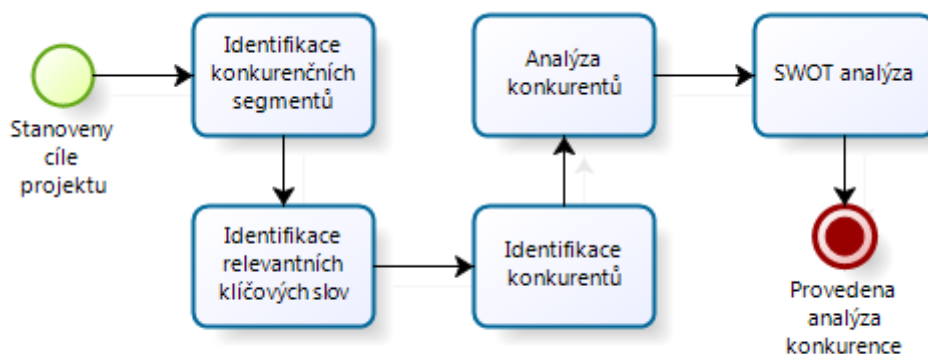
- kvantitativní analýza rizik,
- plánování reakcí na rizika,
- sledování a kontrola rizik.

Hlavním výstupem tohoto kroku je přehled rizik spojených s projektem. U jednotlivých rizik je nutné uvést zhodnocení dopadu rizika na projekt, určit preventivní opatření, stanovit způsob reakce na vzniklé riziko a určit zodpovědnou osobu.

3.5. Fáze Analýza konkurenčního prostředí

3.5.1. Vývojový proces

Druhou fází celého procesu vývoje webu je analýza konkurenčního prostředí, ve kterém se web bude pohybovat. Tuto fázi je možné vynechat u webových aplikací, které jsou zaměřeny na konkrétní, předem definované uživatele a nečelí žádné přímé konkurenci, tj. například webové aplikace určené pro interní užití v rámci podniku či organizace. U ostatních webových aplikací má tato fáze naopak velký význam. Proces v rámci fáze *Analýza konkurenčního prostředí* je zobrazen na obrázku 3.3.



Obrázek 3.3: Vývojový proces fáze Analýza konkurenčního prostředí

3.5.2. Identifikace konkurenčních segmentů

Prvním krokem analýzy konkurenčního prostředí webu je identifikace a vymezení konkurenčních segmentů s využitím informací získaných v předchozí fázi. Základem pro určení konkurenčních segmentů jsou především informace o stanovených cílech projektu a o cílových uživateli. Nejvyšší pozornost je nutné věnovat následujícím čtyřem kritériím:

- *charakter webu* – např. prezentace podniku, e-shop, web poskytující informace či web poskytující služby,
- *komerčnost webu* – weby poskytující informace a služby zdarma či za úplatu,
- *obsahové zaměření webu* – např. ubytování v Praze, vývoj aplikací v .NET, české hory nebo fotbalové dresy,
- *cíloví uživatelé webu* – např. francouzští cestovatelé do ČR, studenti českých středních škol, angličtí sportovní fanoušci či vrcholoví manažeři.

Tato kritéria nemusí být vztahována pouze k celému webu, ale lze je aplikovat také i na různé podmnožiny daného webu, např. různé kategorie e-shopu nebo web poskytující články zdarma i placené služby. Kombinací výše uvedených kritérií lze poměrně jasně identifikovat konkurenční segmenty. Budeme-li vycházet z příkladů uvedených u jednotlivých kritérií, můžeme například stanovit tyto hypotetické segmenty:

- webové prezentace společností zaměřených na nabídku ubytování v Praze,
- e-shopy zaměřené na prodej fotbalových dresů anglickým sportovním fanouškům,
- weby zaměřené na poskytování bezplatných informací o českých horách francouzským cestovatelům do ČR,
- webové aplikace zaměřené na poskytování placených služeb spojených se zajištěním ubytování v Praze.

Hlavním výstupem je přehled konkurenčních segmentů, které se týkají webu či jeho jednotlivých částí.

3.5.3. Identifikace relevantních klíčových slov

V prostředí internetu obecně existují tři různé způsoby, jak se návštěvník může dostat na webové stránky:

- pomocí odkazu z jiné webové stránky,
- přímým zadáním URL adresy do adresního řádku webového prohlížeče,
- pomocí vyhledávačů webových stránek – např. Google, Yahoo! Search, Live Search.

Přímá konkurence mezi jednotlivými webovými stránkami však panuje pouze ve výsledcích vyhledávání webových stránek. Je tedy nutné přetransformovat konkurenční segmenty identifikované v předchozím kroku do podoby seznamu klíčových slov, která mohou tvořit vyhledávací dotazy formulované uživateli. Dále uvedený postup identifikace relevantních klíčových slov vychází z kapitoly věnované výběru klíčových slov z (Planet Ocean, 2009). Postup je prezentován na příkladu webové aplikace zaměřené na nabídku ubytování v apartmánech v Praze. Důležité je ovšem nahlížet na výběr klíčových slov z pozice potenciálního návštěvníka webu a ne z pozice tvůrce webu.

1. *Určení obecných podstatných jmen a frází týkajících se daného segmentu* – např. apartmá, ubytování, ubytování v apartmá.
2. *Vytvoření variant vybraných klíčových slov (překlepy, jednotná a množná čísla, synonyma, nespisovné výrazy)* – např. apartmány, pronájem apartmá.
3. *Geografické cílení klíčových slov (má-li význam)* – např. apartmány v Praze, ubytování Praha, pronájem apartmánů v Praze.
4. *Přidání přívlasků k vybraným klíčovým slovům* – např. levné ubytování v Praze, prostorné apartmány Praha.

Výstupem z výše uvedených kroků jsou seznamy klíčových slov²⁴ charakterizujících jednotlivé identifikované konkurenční segmenty.

3.5.4. Identifikace konkurentů

Chceme-li správně identifikovat největší konkurenty v rámci jednotlivých konkurenčních segmentů, je nutné vycházet z klíčových slov získaných v předchozím kroku. Není vhodné

²⁴ Klíčová slova hrají významnou roli při procesu optimalizace pro vyhledávače (SEO). Klíčovým slovům a optimalizačnímu procesu je věnován větší prostor v rámci kapitoly 4.

například využít výstupy z konkurenční analýzy provedené mimo prostředí internetu, jelikož konkurence na internetu a mimo něj může být diametrálně odlišná. Na internetu se mohou objevovat subjekty, které se ve světě mimo internet vůbec nevyskytují a naopak významní konkurenti ze světa mimo internet nemusí na internetu představovat žádnou hrozbu.

Před samotným hledáním konkurentů je nutné položit si dvě zásadní otázky:

- Jaké je geografické cílení webové aplikace?
- Výsledky kterého vyhledávače jsou pro nás relevantní?

Pořadí zodpovězení těchto dvou otázek je důležité, jelikož v závislosti na geografickém cílení se mění relevance výsledků různých vyhledávačů. Není-li webová aplikace konkrétně geograficky cílena, lze pro určení vhodnosti využití výsledků jednotlivých vyhledávačů vycházet z podílů nejvýznamnějších vyhledávačů na celosvětovém trhu (viz tabulka 3.3).

Vyhledávač	Tržní podíl v % leden 2008	Tržní podíl v % leden 2009
Google	52,14	54,31
Yahoo!	10,94	8,76
MSN/Live	5,78	4,47
AOL	1,86	1,69
Ask	1,09	0,74

Tabulka 3.3: Tržní podíly nejvýznamnějších vyhledávačů (deGeyter, 2009)

Při cílení na český trh je ovšem situace jiná, jelikož se liší struktura trhu vyhledávačů. Dle statistik monitorovací služby Navrcholu.cz z října 2008 (Internet Info, 2008) jsou tržní podíly vyhledávačů následující:

- *Seznam* – 60,85 %,
- *Google* – 32,61 %,
- *Centrum* – 2,61 %,
- *Atlas* – 0,97 %,
- *Jyxo* – 0,14 %.

Po výběru vyhledávače/vyhledávačů je poté nutné pomocí nich vyhledávat dříve specifikovaná klíčová slova. Webové stránky, které se zobrazí na první straně výsledků,

představují potenciální konkurenci. Mimo samotné identifikace konkurence je důležitý také údaj o celkovém počtu vyhledaných stránek, popř. o počtu zobrazovaných reklam ve výsledcích vyhledávání. Tyto hodnoty umožňují určit konkurenčnost daných klíčových slov.

Mezi výstupy toho kroku patří:

- seznam relevantních vyhledávačů,
- přehled konkurenčních webových stránek pro jednotlivá klíčová slova,
- vyhodnocení konkurenčnosti jednotlivých klíčových slov.

3.5.5. Analýza konkurentů

Dalším krokem fáze *Analýza konkurenčního prostředí* je provedení samotné analýzy identifikovaných konkurentů. Při analýze konkurenčních webových stránek je třeba věnovat pozornost především následujícím aspektům:

- nabízené informace, služby anebo produkty,
- cílové skupiny uživatelů,
- použitelnost²⁵,
- partnerské vztahy,
- zdroje návštěvnosti.

Zatímco první tři aspekty je možné vysledovat přímo z webových stránek konkurence, pro zjištění partnerských vztahů a zdrojů návštěvnosti je nutné připojit dodatečnou analýzu ve vyhledávačích.

Jedním z možných zdrojů návštěvnosti jsou výsledky vyhledávání. Pro odhad množství návštěvníků plynoucích z vyhledávačů je nejdříve nutné identifikovat klíčová slova, na která je web zacílen, poté zjistit pozice, na kterých se web umísťuje ve výsledcích vyhledávání a v neposlední řadě také zjistit vyhledávanost daných klíčových slov. Pro zjištění

²⁵ Kvalitativní vlastnost webové aplikace, která určuje, jak je aplikace jednoduchá na použití.

vyhledávanosti klíčových slov lze využít např. *Google AdWords Keyword Tool*²⁶ nebo *Google Search-based Keyword Tool*²⁷.

Druhým možným zdrojem návštěvnosti jsou odkazy vedoucí z jiných webových stránek. Některé z těchto odkazů mohou vyjadřovat i silnější partnerské vazby mezi jednotlivými weby. Pro zjištění odkazů vedoucích na webové stránky konkurence nabízí všechny tři největší světové vyhledávače speciální operátory:

- *link:www.adresa.cz* – Google,
- *linkdomain:www.adresa.cz* – Yahoo!,
- *linkdomain:www.adresa.cz* – Live.

Hlavním výstupem tohoto kroku je hodnocení identifikovaných konkurenčních stránek na základě pěti výše uvedených aspektů.

3.5.6. SWOT analýza projektu

Poslední krokem v rámci analýzy konkurenčního prostředí je vypracování SWOT analýzy projektu vývoje webové aplikace. Cílem SWOT analýzy v rámci zde představovaného procesního modelu je shromáždění a analýza doposud získaných informací o projektu a o konkurenčním prostředí. Výstupní analýza představuje podklady pro následné rozhodnutí o dalším toku procesu (pokračování v procesu, přeformulování některých cílů, ukončení projektu).

SWOT analýza představuje zhodnocení silných (*Strengths*) a slabých (*Weaknesses*) stránek projektu a příležitostí (*Opportunities*) a hrozeb (*Threats*) projektu. Zatímco silné a slabé stránky vychází z charakteru projektu, příležitosti a hrozby mají svůj původ v konkurenčním prostředí. V tabulce 3.4 jsou uvedeny aspekty, které mohou tvořit silné a slabé stránky projektu stejně jako příležitosti a hrozby.

²⁶ <https://adwords.google.com/select/KeywordToolExternal>

²⁷ <http://www.google.com/sktool/>

Silné stránky • poskytování unikátních informací, služeb či produktů • nový způsob prezentace informací, služeb či produktů	Slabé stránky • poskytování pouze převzatých informací • poskytování alternativ k již zavedeným službám, které nenabízí žádnou přidanou hodnotu
Příležitosti • zaměření na méně konkurenční trh • zaměření na rozvíjející se trh • oslovení nových skupin uživatelů	Hrozby • zaměření na velmi konkurenční trh • zaměření na upadající trh • zaměření na špatně identifikovaný trh

Tabulka 3.4: Aspekty s dopadem na SWOT analýzu

Závěrečným krokem SWOT analýzy je určení převládajících charakteristik projektu. Výsledkem mohou být čtyři stavy:

- *převaha silných stránek a příležitostí* – např. webová aplikace poskytující unikátní služby zaměřená na rozvíjející se trh,
- *převaha silných stránek a hrozeb* – např. web poskytující unikátní informace určené pro velmi konkurenční trh,
- *převaha slabých stránek a příležitostí* – např. webové stránky nabízející alternativu k již existujícím službám bez přidané hodnoty cílené na málo konkurenční trh,
- *převaha slabých stránek a hrozeb* – např. web poskytující pouze převzaté informace zaměřený na velmi konkurenční trh.

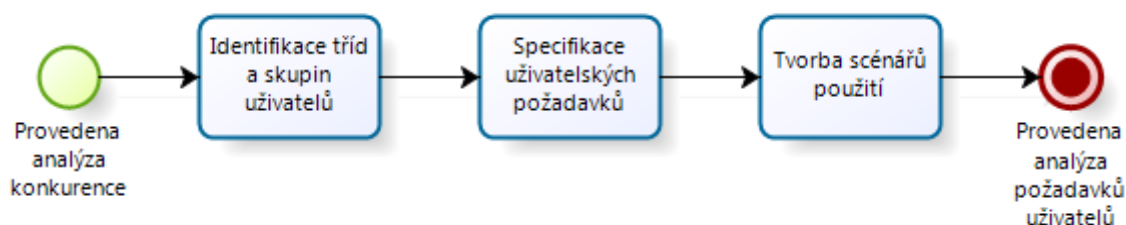
Na základě převažujících charakteristik projektu je poté možné rozhodnout o dalším směřování celého procesu vývoje webové aplikace. Převaha silných stránek a příležitostí představuje ideální situaci pro další pokračování procesu vývoje. Naopak převaha slabých stránek a hrozeb vybízí k ukončení projektu, popř. k výraznému přeformulování cílů. U zbylých dvou možných stavů záleží na konkrétní situaci, zda bude proces pokračovat nebo budou muset být přeformulovány některé cíle projektu.

Výstupem SWOT analýzy jsou identifikované slabé a silné stránky, příležitosti a hrozby projektu a také určení převládajících charakteristik projektu. Více informací o SWOT analýze lze získat z (CIPD, 2008).

3.6. Fáze Analýza uživatelských požadavků

3.6.1. Vývojový proces

Třetí fází procesu vývoje webové aplikace je analýza uživatelských požadavků. V rámci této velmi důležité fáze je nutné identifikovat a charakterizovat třídy uživatelů, které budou pracovat s webovou aplikací a provést analýzu jejich požadavků. Výsledky analýzy požadavků představují podklady pro následující fáze datového návrhu, návrhu navigace i návrhu uživatelského rozhraní. Proces fáze *Analýza uživatelských požadavků* je zobrazen na obrázku 3.4.



Obrázek 3.4: Vývojový proces fáze Analýza uživatelských požadavků

3.6.2. Identifikace tříd a skupin uživatelů

Prvním krokem je identifikace tříd uživatelů webové aplikace. Třída uživatelů představuje množinu uživatelů, kteří mají stejné informační požadavky a požadavky na funkcionalitu. Uživatelské třídy definované v tomto kroku musí pokrývat všechny možné typy uživatelů, kteří mohou s webovou aplikací pracovat. Tím se tento krok odlišuje od identifikace cílových uživatelů provedené v úvodní fázi procesu.

Podle (De Troyer, Casteleyn, 2001a) je vhodné jednotlivé identifikované třídy uživatelů uspořádat do hierarchické struktury. Na vrcholu hierarchie se nachází obecná uživatelská třída *Uživatel*, která představuje průnik všech charakteristik a požadavků všech identifikovaných tříd. V rámci hierarchie poté každá podtřída přebírá všechny požadavky definované svoji nadtřídou a přidává vlastní požadavky.

V rámci jednotlivých tříd ovšem mohou navíc existovat rozdílné požadavky na uživatelské rozhraní. Na základě těchto požadavků je možné dále dělit třídy na jednotlivé skupiny. Tyto skupiny mohou být charakterizovány například pomocí:

- úrovně zkušeností s prací na internetu,
- frekvence využívání internetu,
- přítomnosti handicapů ztěžujících práci s webovou aplikací,
- jazykových znalostí,
- věku,
- úrovně vzdělání,
- životního stylu,
- využívaného zobrazovacího zařízení.

Výstupem tohoto kroku je vytvořená hierarchie uživatelských tříd a skupin společně s jejich charakteristikou.

3.6.3. Specifikace uživatelských požadavků

Specifikace uživatelských požadavků je velmi úzce spojena s identifikací uživatelských tříd, jelikož pro každou uživatelskou třídu je nutné určit požadavky na informace, funkce a také na uživatelské rozhraní. Pokud je to možné, ideální způsobem pro získání uživatelských požadavků jsou osobní interview se zástupci jednotlivých uživatelských tříd. V ostatních případech lze využít např. dotazování pomocí emailu, dotazníků či expertní odhady.

Cílem tohoto kroku je pouze základní identifikace informačních a funkčních uživatelských požadavků. Požadavky na informace a funkce lze zaznamenat v tabulkové formě ve struktuře *uživatelská třída – název požadavku – krátký popis požadavku* nebo pomocí UML *Use Case diagramů* (diagramy případů užití). Podrobné informace o Use Case diagramech lze získat z (OMG, 2007).

Pro záznam požadavků na uživatelské rozhraní lze využít tabulku se sloupci *uživatelská třída – uživatelská skupina – název požadavku – popis požadavku*. Požadavky na uživatelské rozhraní představují podklad pro návrh a implementaci uživatelského rozhraní. Při specifikaci požadavků je nutné klást velký důraz na použitelnost a přístupnost webových stránek. Základním zdrojem informací o přístupnosti webových stránek je dokument shrnující závěry

získané v rámci projektu *Přístupnost webových stránek orgánů státní správy* (Špínar, et al., 2007). Základní informace o použitelnosti lze získat z (Hejl, 2007).

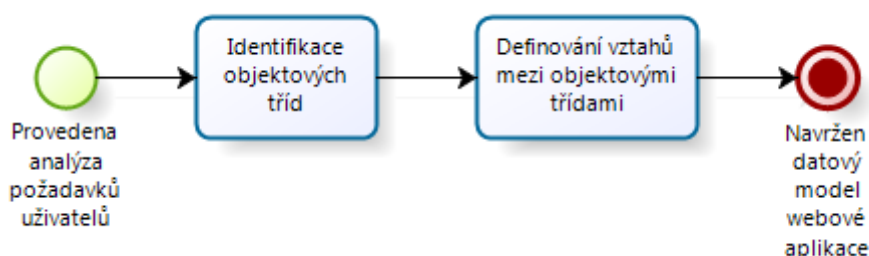
3.6.4. Tvorba scénářů použití

Posledním krokem analýzy uživatelských požadavků je vytvoření scénářů použití. Pomocí scénářů jsou rozpracovávány informační a funkční požadavky do podoby procesů. Pro záznam lze využít tabulky ve struktuře používané metodikou SOHDM (viz tabulka 2.4). Vypracované scénáře pro všechny uživatelské třídy a všechny informační a funkční požadavky představují základní podklady pro navazující návrhové fáze.

3.7. Fáze Datový návrh

3.7.1. Vývojový proces

Čtvrtou fází vývojového procesu je datový návrh. Cílem datového návrhu je identifikace objektových tříd, které budou použity ve webové aplikaci a vytvoření vztahů mezi těmito třídami na základě výstupů z analýzy uživatelských požadavků. Objektové třídy představují zapouzdřená data a funkce pracující s těmito daty. Vývojový proces fáze *Datový návrh* je zobrazen na obrázku 3.5.



Obrázek 3.5: Vývojový proces fáze Datový návrh

3.7.2. Identifikace objektových tříd

Prvním krokem datového návrhu je samotná identifikace objektových tříd. Všechny třídy by měly vycházet z vytvořených scénářů použití. Ke každé třídě je navíc nutné přiřadit *atributy*, které vyjadřují vlastnosti třídy a umožňují jednoznačnou identifikaci různých výskytů třídy, a také *zodpovědnosti* třídy. Zatímco atributy jsou datového charakteru, zodpovědnosti lze považovat za funkcionalitu, kterou může třída vykonávat.

Při identifikaci objektových tříd lze využít následujících doporučení:

- každá uživatelská třída představuje jednu objektovou třídu,
- hierarchie uživatelských tříd se transformuje na hierarchii odpovídajících objektových tříd,
- všechna podstatná jména užitá při tvorbě scénářů použití představují potenciální kandidáty na objektové třídy,
- informace potřebné ke spuštění aktivity definované při tvorbě scénářů použití představují potenciální kandidáty na objektové třídy,
- nelze-li objektovou třídu pojmenovat méně než třemi slovy, nejedná se s největší pravděpodobností o samostatnou třídu, ale spíše o zodpovědnost jiné třídy nebo o určitou vazbu mezi třídami (Rubin, 1998).

Výstupem tohoto kroku je seznam objektových tříd s definovanými atributy a zodpovědnostmi.

3.7.3. Definování vztahů mezi objektovými třídami

Druhým krokem datového návrhu je definování vztahů mezi jednotlivými identifikovanými objektovými třídami. V této práci jsou pro účely zachycení vztahů mezi objektovými třídami použity *CRC karty* a *Class Structure Diagram* využívané metodikou SOHDM. Alternativním způsobem vyjádření vazeb mezi jednotlivými třídami je *UML Class Diagram* (diagram tříd). Podrobné informace o UML diagramech tříd lze získat z (OMG, 2007).

CRC karty zachycují informace o názvech tříd, jejich attributech a zodpovědnostech a také o jejich vztahu k ostatním objektovým třídám. Při datovém návrhu pomocí CRC karet lze využít čtyř druhů vztahů:

- *nadtřída/podtřída* – vyjadřuje vztah typu generalizace/specializace,
- *kolaborace* – vyjadřuje vztah, kdy jedna třída potřebuje získat či upravit data, která patří jiné třídě,
- *komponenta* – vyjadřuje vztah, kdy se jedna třída skládá z jiných tříd,

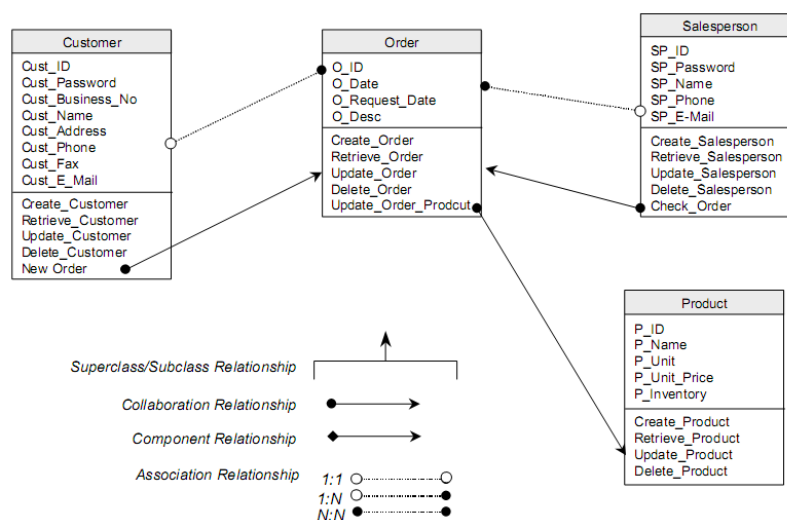
- *asociace* – vyjadřuje sémantický vztah mezi dvěma třídami, který má stanovenou svou kardinalitu²⁸(1..1, 1..N, N..N).

Struktura CRC karty využívaná v metodice SOHDM (Lee, Lee, 2000) je vyjádřena v tabulce 3.5.

Třída: název třídy	Nadtřída: název nadtřídy
Atributy: seznam názvů atributů třídy	Podtřídy: seznam názvů podtříd
	Asociace: seznam názvů asociovaných tříd společně s uvedenou kardinalitou
	Komponenty: seznam názvů tříd tvořících komponenty třídy
Zodpovědnosti: seznam zodpovědností třídy případně doplněný o názvy kolaborujících tříd	

Tabulka 3.5: Struktura CRC karty (Lee, Lee, 2000)

Na základě vytvořených CRC karet pro jednotlivé objektové třídy je následně vhodné zkonstruovat Class Structure Diagram (CSD), který umožňuje zobrazit vazby mezi třídami v grafické podobě. Příklad CSD i s legendou pocházející z (Lee, Lee, 2000) lze nalézt na obrázku 3.6.



Obrázek 3.6: Příklad Class Structure Diagramu s legendou (Lee, Lee, 2000)

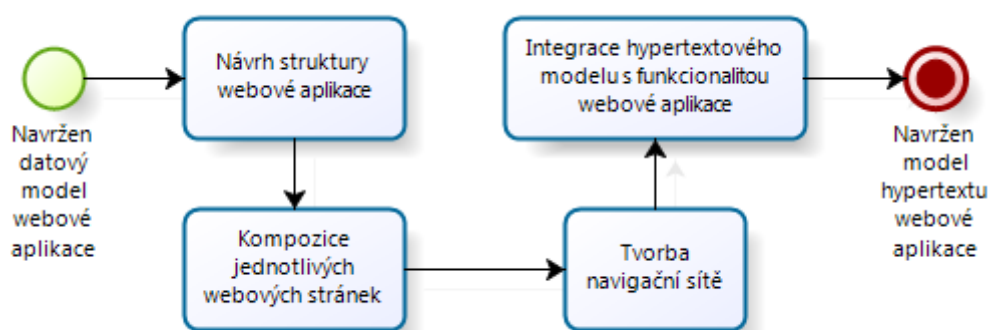
²⁸ Představuje omezení v počtu výskytů druhé objektové třídy, které mají vztah s jakýmkoliv výskytem první objektové třídy.

Výstupem tohoto kroku jsou CRC karty pro všechny identifikované objektové třídy a také CSD zachycující vztahy mezi třídami. Více informací o CRC kartách lze získat z (Rubin, 1998).

3.8. Fáze *Hypertextový návrh*

3.8.1. Vývojový proces

Pátou fází procesu vývoje webové aplikace je hypertextový návrh. Cílem hypertextového návrhu je navržení struktury webové aplikace s ohledem na identifikované třídy uživatelů, specifikování navigace mezi jednotlivými částmi aplikace a specifikace prvků, ze kterých se skládají jednotlivé webové stránky. Vývojový proces fáze *Hypertextový návrh* je znázorněn na obrázku 3.7.



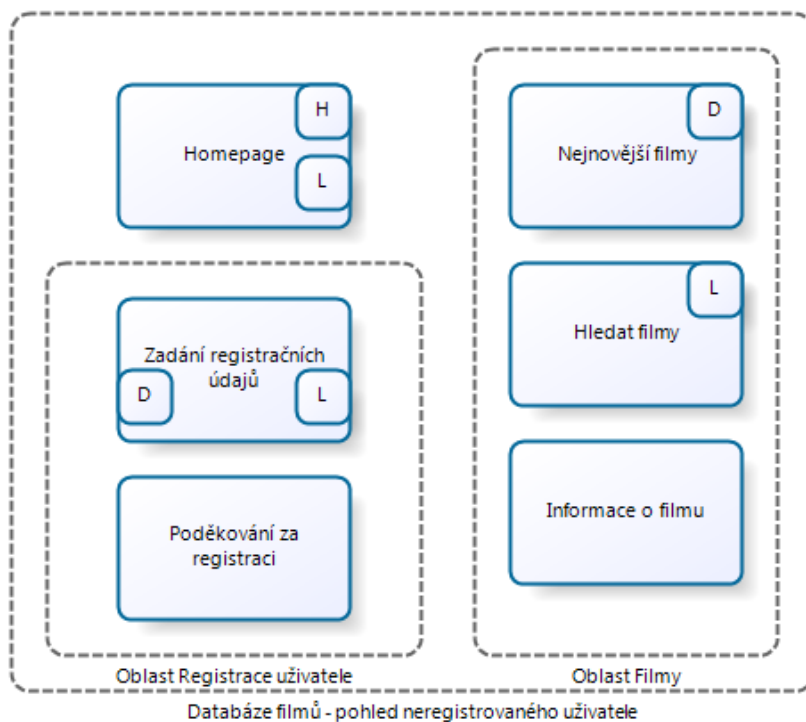
Obrázek 3.7: Vývojový proces fáze *Hypertextový návrh*

Obsah jednotlivých kroků této fáze se shoduje s obsahem fáze *Hypertextový návrh* z metodiky WebML. Informace uvedené v následujících čtyřech podkapitolách vychází z (Brambilla, 2007) a (Politecnico di Milano, 2008).

3.8.2. Návrh struktury webové aplikace

Prvním krokem hypertextového návrhu je navržení celkové struktury celé webové aplikace. Vzhledem k existenci různých tříd uživatelů, které mají různé požadavky na informace a na funkce, nemůže existovat pouze jeden typ navigační struktury. Je nutné pro každou uživatelskou třídu vytvořit tzv. *pohled*, který představuje takovou strukturu webové aplikace, která umožňuje uspokojit informační a funkční požadavky dané třídy uživatelů.

Jednotlivé pohledy se skládají z tzv. *oblastí*, které lze chápat jako abstraktní sdružení jednotlivých webových stránek²⁹ nebo rekurzivně i dalších oblastí. Výjimkou je domovská stránka (homepage), která není součástí žádné oblasti a představuje první stránku, na kterou se uživatel dostane po vstupu do webové aplikace. Pro záznam pohledů se nejčastěji využívá grafické formy (viz obrázek 3.8).



Obrázek 3.8: Příklad pohledu na webovou aplikaci

Na obrázku 3.8 se kromě oblastí a stránek objevují také tři symboly. Tyto symboly náležejí k jednotlivým stránkám a oblastem a vyjadřují jejich důležitost v rámci celého pohledu. Jejich význam je následující:

- **Homepage** – unikátní stránka v rámci celého pohledu, která se uživateli zobrazuje jako první při vstupu do webové aplikace,
- **Default page** – unikátní stránka nebo podoblast v rámci oblasti, která se zobrazí jako první při vstupu uživatele do dané oblasti,
- **Landmark page** – stránka či oblast, která je dostupná ze všech ostatních stránek v rámci daného pohledu.

²⁹ Dle metodiky WebML mohou jednotlivé webové stránky být navíc složeny z dalších podstránek, které mohou představovat stavební bloky, z nichž je webová stránka poskládána.

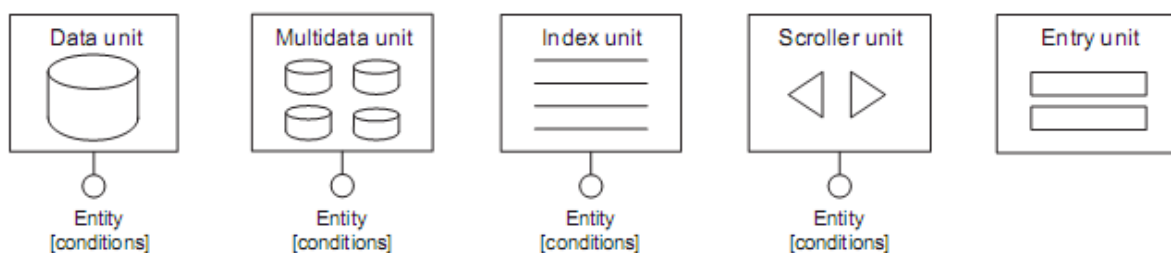
Výstupem jsou navržené pohledy na webovou aplikaci pro jednotlivé uživatelské třídy.

3.8.3. Kompozice jednotlivých webových stránek

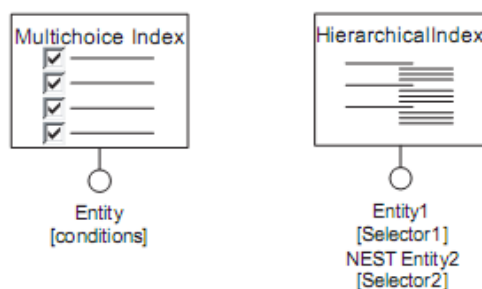
Poté, co je navržena základní struktura webové aplikace, je nutné definovat obsah jednotlivých webových stránek. Základními stavebními kameny každé webové stránky dle WebML jsou tzv. *obsahové jednotky*. Existuje pět základních typů obsahových jednotek a dvě odvozené:

- *Data Unit* – zobrazuje informace o jednom výskytu objektové třídy,
- *Multidata Unit* – zobrazuje informace o více výskytech objektové třídy,
- *Index Unit* – zobrazuje seznam výskytů objektové třídy s možností výběru,
- *Scroller Unit* – umožňuje procházení uspořádané sady výskytů objektové třídy,
- *Entry Unit* – formulář pro vstup uživatelských údajů,
- *Multichoice Index Unit* – zobrazuje seznam výskytů objektové třídy s možností několikanásobného výběru,
- *Hierarchical Index Unit* – zobrazuje seznam výskytů objektových tříd organizovaných do hierarchické struktury s možností výběru.

Na obrázku 3.9 resp. 3.10 jsou zobrazeny grafické reprezentace jednotlivých základních resp. odvozených obsahových jednotek pocházející z (Brambilla, 2007).



Obrázek 3.9: Grafická reprezentace základních obsahových jednotek (Brambilla, 2007)



Obrázek 3.10: Grafická reprezentace odvozených obsahových jednotek (Brambilla, 2007)

Jednotlivé obsahové jednotky mají čtyři základní charakteristiky:

- *vstupní parametry* – hodnoty potřebné pro výpočet výběrových podmínek,
- *výstupní parametry* – hodnoty předávané obsahovou jednotkou dále,
- *specifikace objektových tříd* – objektové třídy, které představují zdroj zobrazovaných dat,
- *výběrové podmínky* – podmínky, které ovlivňují výběr jednotlivých výskytů objektových tříd (na obrázcích 3.9 a 3.10 vyjádřeny jako *[conditions]*).

Výstupem tohoto kroku je specifikace obsahových jednotek užitých v rámci jednotlivých stránek.

3.8.4. Tvorba navigační sítě

Třetím krokem hypertextového návrhu je vytvoření navigační sítě mezi různými prvky jednotlivých pohledů. Navigační síť webové aplikace je tvořena hypertextovými odkazy. WebML rozlišuje dva základní druhy odkazů:

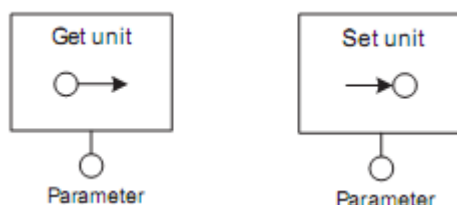
- *kontextové* – odkazy mezi jednotlivými obsahovými jednotkami v rámci pohledu, které přenášejí kontextové informace,
- *nekontextové* – odkazy mezi jednotlivými webovými stránkami v rámci pohledu, které nepřenášejí žádnou informaci.

Informace přenášené kontextovými odkazy jsou přenášeny ve formě parametrů. Tyto parametry mohou být stanoveny buď explicitně, nebo může být využit tzv. *defaultní kontext*³⁰, který vyplývá z datového návrhu a z kompozice dané webové stránky. Defaultního kontextu využívají dva podtypy kontextových odkazů, které umožňují automatické přesunutí defaultního kontextu ze zdrojové do cílové obsahové jednotky okamžitě po zobrazení zdrojové jednotky:

- *automatické* – uživatel může změnit přenášený kontext kliknutím na jiný odkaz,
- *transportní* – uživatel nemůže změnit přenášený kontext kliknutím na jiný odkaz.

³⁰ Např. u odkazu ze seznamu filmů na stránku s detailem filmu není nutné explicitně uvádět přenášený parametr, jelikož ze situace vyplývá, že se bude přenášet jedinečný identifikátor filmu.

Ne vždy jsou ovšem parametry přenášeny pomocí kontextových odkazů dostačující. Kvůli tomu v metodice WebML existují i tzv. *globální parametry*, které jsou dostupné ve všech webových stránkách v rámci jednoho pohledu. Pro práci s globálními parametry lze využít jednotky pro uložení (*Set Unit*) a získání (*Get Unit*) hodnoty parametru zobrazené na obrázku 3.11 (Brambilla, 2007).



Obrázek 3.11: Grafická reprezentace jednotek pro práci s globálními parametry (Brambilla, 2007)

Výstupem tohoto kroku je návrh navigační sítě mezi jednotlivými obsahovými jednotkami a také mezi různými stránkami pro všechny definované pohledy.

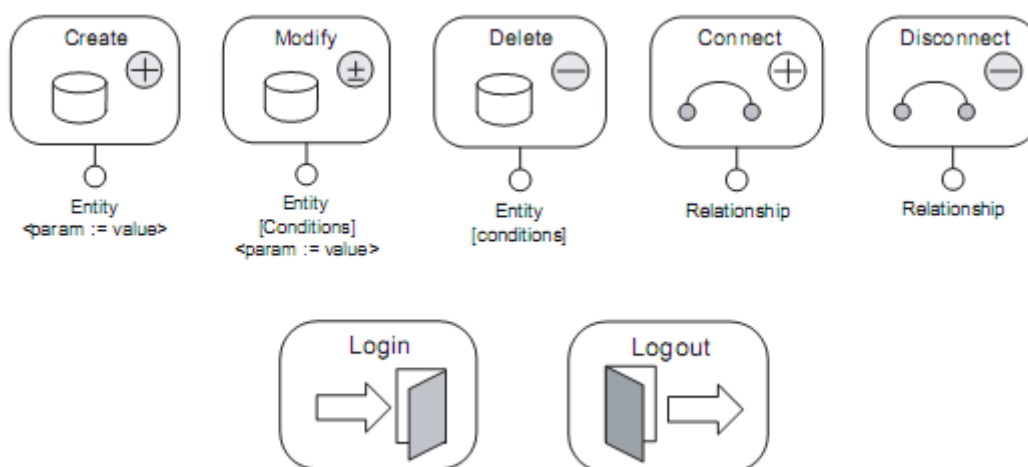
3.8.5. Integrace hypertextového modelu s funkcionalitou webové aplikace

V současné době existuje na internetu mizivé procento webů, které nabízí čistě informační obsah ve formě webových stránek propojených odkazy bez využití jakékoliv dodatečné funkcionality. Za tuto funkcionality lze považovat např. registraci a přihlašování uživatelů, správu obsahu, práci s nákupním košíkem a širokou škálu dalších složitějších úkonů. Metodika WebML zavádí pro zachycení této funkcionality koncept tzv. *operací*.

Operace jsou vyjádřeny sedmi základními tzv. *operačními jednotkami*, které však mohou být doplněny dalšími uživatelskými operačními jednotkami (např. odeslání emailu). Mezi základní operační jednotky patří:

- *Create Unit* – vytvoření nového výskytu objektové třídy,
- *Modify Unit* – úprava stávajícího výskytu objektové třídy,
- *Delete Unit* – vymazání stávajícího výskytu objektové třídy,
- *Connect Unit* – vytvoření vztahu mezi dvěma výskyty objektových tříd,
- *Disconnect Unit* – vymazání vztahu mezi dvěma výskyty objektových tříd,
- *Login Unit* – přihlášení uživatele,
- *Logout Unit* – odhlášení uživatele.

Grafická reprezentace základních operačních jednotek pocházející z (Brambilla, 2007) je zachycena na obrázku 3.12.

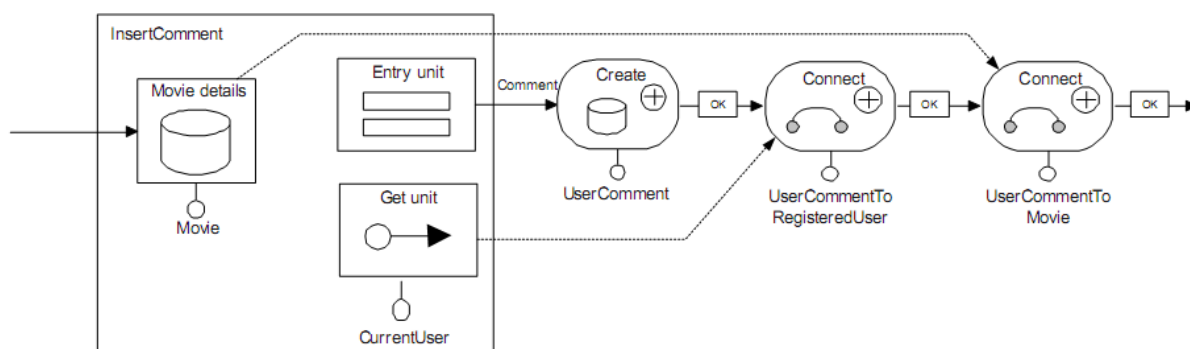


Obrázek 3.12: Grafická reprezentace základních operačních jednotek (Brambilla, 2007)

Jednotlivé operační jednotky mohou být mezi sebou propojeny stejně jako obsahové jednotky a tvořit tak posloupnost operací. V souvislosti s posloupností operací je nutné zmínit, že každá operační jednotka má dva typy výstupních odkazů:

- *OK odkaz* – je následován, pokud operace proběhne celá v pořádku,
- *KO odkaz* – je následován, pokud dojde při operaci k chybě.

Příklad zřetězení operačních jednotek pocházející z (Brambilla, 2007) týkající se přidání komentáře k filmu lze nalézt na obrázku 3.13.



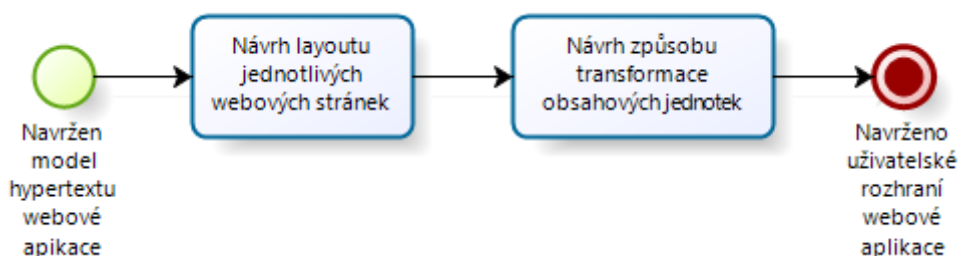
Obrázek 3.13: Příklad zřetězení operačních jednotek (Brambilla, 2007)

Při hypertextovém návrhu s využitím zde prezentované metodiky WebML je možné využít nástroj WebRatio. Výstupem je dříve navržená navigační síť obohacená o propojení s funkcionalitou webové aplikace.

3.9. Fáze Návrh uživatelského rozhraní

3.9.1. Vývojový proces

Šestou fází procesu vývoje webu je návrh uživatelského rozhraní. Cílem této fáze je transformace jednotek definovaných v předchozí fázi na prvky uživatelského rozhraní. Zároveň je nutné dodržet požadavky na uživatelské rozhraní identifikované pro jednotlivé skupiny uživatelů v rámci analýzy uživatelských požadavků. Vývojový proces fáze *Návrh uživatelského rozhraní* je znázorněn na obrázku 3.14.



Obrázek 3.14: Vývojový proces fáze Návrh uživatelského rozhraní

3.9.2. Návrh layoutu jednotlivých webových stránek

Prvním krokem návrhu uživatelského rozhraní je návrh základního rozložení (layoutu) jednotlivých stránek pro všechny uživatelské skupiny. Jak již bylo uvedeno dříve, uživatelé webové aplikace se člení do tříd na základě společných informačních a funkčních požadavků a jednotlivé třídy se mohou dále dělit na uživatelské skupiny na základě společných požadavků na uživatelské rozhraní. Podle typu uživatelské skupiny může poté být kladen větší důraz např. na podrobnější navigaci či přehlednější obsah.

Při návrhu layoutu je vhodné postupovat ve dvou krocích. Prvním krokem by mělo být rozvržení webové stránky na větší významové bloky. Ve druhém kroku by tyto bloky měly být naplněny obsahovými jednotkami identifikovanými ve fázi hypertextového návrhu. Mezi nejčastěji užívané typy významových bloků lze zařadit:

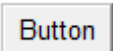
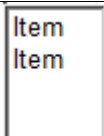
- *hlavička stránky* – blok nacházející se na začátku webové stránky a obsahující nejčastěji identifikační prvky webové aplikace (název, logo),

- *navigační menu* – horizontální nebo vertikální blok sloužící k navigaci mezi jednotlivými částmi webové aplikace,
- *obsahový blok* – blok sloužící k prezentaci samotného obsahu webové aplikace,
- *patička stránky* – blok nacházející se na konci webové stránky a obsahující nejčastěji informace o tvůrcích webové aplikace a kontaktní informace.

Výstupem toho kroku jsou navržené layouty všech webových stránek pro všechny identifikované uživatelské třídy a skupiny.

3.9.3. Návrh způsobu transformace obsahových jednotek

Poté, co je navržen základní layout jednotlivých webových stránek a rozmístěny obsahové jednotky v rámci tohoto layoutu, je nutné určit způsob, jakým budou obsahové jednotky transformovány na prvky uživatelského rozhraní. Nejčastěji využívané prvky uživatelského rozhraní lze nalézt v tabulce 3.6.

Anglický název	Český název	Možná grafická reprezentace
Label	Textový popis	Label
Textbox	Textové pole	
Button	Tlačítko	
Link	Odkaz	HyperLink
Combobox	Rozbalovací menu	
Listbox	Seznam s možností výběru	
Checkbox	Zaškrtačací pole	☒ Checkbox
Radiobutton	Přepínací tlačítko	☐ Radiobutton
Image	Obrázek	

Bulleted List	Odrážkový seznam	• Item • Item				
Table	Tabulka	<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>				

Tabulka 3.6: Nejčastěji využívané prvky uživatelského rozhraní

Možné namapování jednotlivých typů obsahových jednotek na prvky uživatelského rozhraní je uvedeno v tabulce 3.7.

Obsahová jednotka	Možné prvky uživatelského rozhraní
Data Unit	Sada textových popisků a obrázků
Multidata Unit	Sada textových popisků a obrázků uspořádaná do tabulky
Index Unit	Rozbalovací menu; sada přepínacích tlačítek; odrážkový seznam skládající se z odkazů nebo tlačítek
Scroller Unit	Sada odkazů nebo tlačítek
Entry Unit	Sada textových polí
Multichoice Index Unit	Sada zaškrťovacích polí; seznam s možností výběru
Hierarchical Index Unit	Vnořené odrážkové seznamy skládající se z odkazů nebo tlačítek

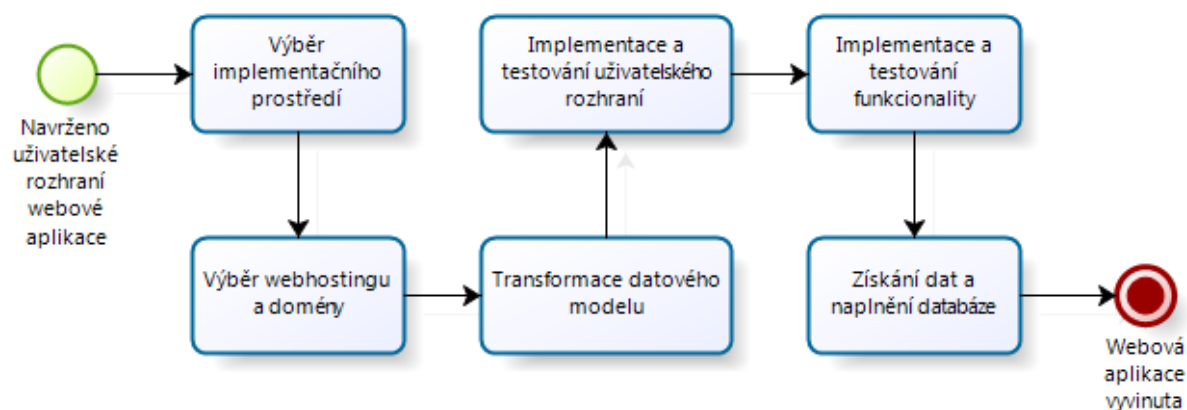
Tabulka 3.7: Možné mapování obsahových jednotek na prvky uživatelského rozhraní

Za výstup lze považovat navržené mapovací schéma mezi obsahovými jednotkami a prvky uživatelského rozhraní.

3.10. Fáze Implementace

3.10.1. Vývojový proces

Předposlední, sedmou, fází procesu vývoje webu je implementace webové aplikace. Mezi hlavní cíle této fáze patří výběr implementačního prostředí a poté i samotná implementace webové aplikace vycházející z výstupů vytvořených v předchozích analytických a návrhových fázích. Vývojový proces fáze *Implementace* je znázorněn na obrázku 3.15.



Obrázek 3.15: Vývojový proces fáze Implementace

3.10.2. Výběr implementačního prostředí

Prvním krokem implementační fáze je výběr implementačního prostředí pro webovou aplikaci. V rámci tohoto kroku je nutné vybrat programovací jazyk či framework, který bude použit pro vývoj webové aplikace a také je potřeba vybrat způsob, jakým budou ukládána data používaná webovou aplikací.

Mezi nejčastěji využívané programovací jazyky pro vývoj webových aplikací lze na základě (DedaSys LLC, 2009) a (Kolb, 2007) zařadit (řazeno dle abecedy):

- *ASP.NET* – www.asp.net,
- *C#* – msdn.microsoft.com/en-us/vcsharp/default.aspx,
- *Flash* – www.adobe.com/products/flash/,
- *Java* – java.sun.com,
- *Javascript* – developer.mozilla.org/en/JavaScript,
- *Perl* – www.perl.org,
- *PHP* – www.php.net,
- *Python* – www.python.org,
- *Ruby* – www.ruby-lang.org,
- *VB.NET* – msdn.microsoft.com/en-us/vbasic/default.aspx.

K významným webovým frameworkům je možné zařadit:

- *Django (Python)* – www.djangoproject.com,
- *Mason (Perl)* – www.masonhq.com,
- *Ruby on Rails (Ruby)* – rubyonrails.org,
- *Spring (Java)* – www.springsource.org,
- *Zend (PHP)* – www.zend.com.

Zaměříme-li se na ukládání dat, je možné využít dva základní způsoby uložení:

- *souborový systém* – data jsou ukládána do samostatných souborů v rámci adresářové struktury,
- *databázový systém* – data jsou ukládána do databáze a spravována systémem pro řízení báze dat (SŘBD).

V této práci bude dále prezentován pouze databázový způsob ukládání a přístupu k datům, konkrétně *relační databáze*. Při výběru konkrétního relačního SŘBD lze vybírat ze široké nabídky, přičemž mezi nejvýznamnější zástupce je možné zařadit:

- IBM DB2,
- Microsoft SQL Server,
- MySQL,
- Oracle.

Výstupem tohoto kroku je identifikované programovací prostředí a databázová platforma.

3.10.3. Výběr webhostingu a domény

Druhým krokem implementace je výběr webhostingu a domény. Pod pojmem *webhosting* se skrývá datové úložiště, které je přístupné prostřednictvím internetu a je využíváno pro ukládání webových stránek a datových souborů. Pro účely webhostingu není nutné využívat pouze služeb externích dodavatelů webhostingových služeb, ale lze využít i vlastních hardwarových a softwarových prostředků společně s rychlým a permanentním připojením k internetu. Rozhodneme-li se však pro nákup webhostingových služeb, můžeme vybírat z tisíců nabídek od stovek různých společností.

Mezi celosvětově nejlépe hodnocené webhostingové společnosti dle (Web Hosting Choice, 2009) je možné zařadit:

- *Just Host* – www.justhost.com,
- *Host Monster* – www.hostmonster.com,
- *FatCow* – www.fatcow.com,
- *InMotion Hosting* – www.inmotionhosting.com,
- *BlueHost.com* – www.bluehost.com,
- *Start Logic* – www.startlogic.com,
- *HostGator* – www.hostgator.com,
- *1&1 Internet* – www.1and1.com,
- *Go Daddy* – www.godaddy.com,
- *Yahoo!* – smallbusiness.yahoo.com/webhosting/.

Zaměříme-li se na české webhostingové společnosti, pak mezi pět nejvýznamnějších společností na základě počtu spravovaných webhostingů v doméně .cz ke dni 21. 3. 2009 patří dle (Internet Info, 2009):

- *FORPSI* – www.forpsi.com (82550 domén),
- *ACTIVE 24* – www.active24.cz (69212 domén),
- *IGNUM* – www.domena.cz (43972 domén),
- *ZONER software* – www.czechia.com (26762 domén),
- *PIPNI* – www.pipni.cz (15721 domén).

Při výběru konkrétního webhostingového programu je nutné dle (Janovský, 2009) sledovat kromě reputace společnosti a počtu spravovaných domén také další charakteristiky:

- cena,
- velikost nabízeného prostoru,
- možnosti nahrávání souborů na server,
- limit přenesených dat,
- použitý operační systém,
- použitý webový server,
- podporované programovací jazyky a frameworky,

- dostupné databáze,
- garantovaná dostupnost,
- kvalita technické podpory.

Na základě určeného programovacího prostředí a databázového systému z předchozího kroku je možné zúžit okruh webhostingových nabídek. Poté je vhodné stanovit prioritu zbylých charakteristik a vybrat odpovídající webhosting.

S webhostingem jsou úzce spojeny i *domény*. Zatímco webhosting představuje datové úložiště přístupné z internetu, doména figuruje jako identifikace tohoto úložiště, např. www.google.cz. Téměř všechny webhostingové společnosti nabízejí současně nákup webhostingových služeb i registraci domény. Webhosting a doménu lze však zakoupit také samostatně. Více o doménách a výběru doménového jména bude uvedeno v rámci kapitoly 4 věnované problematice optimalizace pro vyhledávače.

Výstupem tohoto kroku je konkrétní vybraný webhostingový program a vybraná doména pro provoz webové aplikace.

3.10.4. Transformace datového modelu

Třetím krokem implementace webové aplikace je transformace navrženého objektového datového modelu do podoby tabulek a vztahů mezi tabulkami ve vybrané relační databázi. Při modelování databází se využívá tzv. *principu tří architektur*, který je založen na využívání různé míry abstrakce při návrhu informačních systémů. Míra abstrakce se odráží ve třech úrovních:

- *konceptuální úroveň* – určuje, **co** má být obsahem databáze; výstupem je konceptuální datový model,
- *technologická úroveň* – určuje, **jak** bude daný obsah v databázi realizován; výstupem je logický datový model,
- *implementační úroveň* – určuje, **čím** bude dané technologické řešení realizováno; výstupem je fyzický datový model.

Při konceptuálním modelování se nejčastěji využívají tzv. *Entity-Relationship Diagramy (ERD)*. Pro využití ERD je nutné pochopit jeho základní prvky:

- *entita* – významný objekt v rámci modelované oblasti,
- *atribut* – vlastnost entity,
- *vztah* – vzájemný vztah mezi entitami,
- *kardinalita vztahu* – počet možných výskytů entit, které jsou účastníky vztahu (1:1, 1:N nebo M:N),
- *parcialita vztahu* – určuje povinnou resp. volitelnou existenci vztahu.

V rámci zde prezentovaného procesu vývoje webu je možné při tvorbě ERD vycházet z CRC karet nebo UML diagramů tříd vytvořených ve fázi datového návrhu. Identifikované objektové třídy a jejich atributy je možné transformovat na entity a odpovídající atributy, přičemž je nutné určit tzv. *primární klíč*, který slouží k jednoznačné identifikaci výskytu entity. Asociace mezi objektovými třídami lze podobně převést na vztahy mezi entitami s uvedením odpovídající kardinality a parciality. U jednotlivých vztahů je nutné poté specifikovat jejich název a případné atributy tohoto vztahu. Podrobnější informace o ERD lze získat z (Russell, 2004).

Navazující částí je transformace konceptuálního datového modelu na logický datový model. Logický datový model je v rámci této práce postaven na relačním přístupu k modelování. Postup při logickém modelování je možné shrnout do následujících kroků:

1. *Určení implementačně nezávislých datových typů jednotlivých atributů* – celé číslo, desetinné číslo, datum, čas, text s pevně daným počtem znaků, text se stanoveným maximálním počtem znaků, logická hodnota.
2. *Dekompozice vztahů obsahujících atributy* – např. pro vztah mezi entitou Pracovník a Oddělení s atributem datum_nástupu je nutné vytvořit novou vazební entitu Nástup pracovníka, která bude mezi atributy obsahovat primární klíč pracovníka, primární klíč oddělení a také datum nástupu.
3. *Transformace vztahů M:N na vztahy 1:N* – např. při vztahu M:N mezi entitou Pracovník a Projekt je nutné vytvořit novou entitu Přiřazení k projektu, která bude ve vztahu 1:N s oběma entitami a bude vyjadřovat stejnou skutečnost jako vztah M:N.

4. *Určení cizích klíčů pro vyjádření vztahů 1:N* – např. ve vztahu 1:N mezi entitami Pracovník a Oddělení je nutné vytvořit nový atribut id_oddělení, který bude obsahovat hodnotu primárního klíče výskytu entity Oddělení, ve kterém pracovník pracuje.
5. *Normalizace* – slouží k odstranění redundance dat a omezení složitosti modelu, více viz (Vebloud, 2007).
6. *Kontrola a doplnění integritních omezení (IO)* – entitní IO spočívá v určení primárního klíče pro každou entitu, doménové IO spočívá v určení datových typů a rozsahů hodnot jednotlivých atributů, referenční IO je založena na stanovení cizích klíčů pro vyjádření vztahů mezi entitami (Vebloud, 2006).

Logický datový model vytvořený na základě výše uvedených kroků je následně nutné transformovat do podoby fyzického datového modelu, který je na rozdíl od předcházejících modelů již závislý na vybrané databázové platformě. Je tedy především nutné:

- transformovat entity na databázové tabulky,
- transformovat atributy na sloupce v databázových tabulkách a konkretizovat datové typy jednotlivých atributů,
- realizovat vztahy mezi tabulkami v daném databázovém prostředí,
- realizovat integritní omezení v daném databázovém prostředí,
- realizovat další požadavky, např. denormalizace tabulek kvůli výkonu.

Fyzický datový model je možné vyjádřit ve formě schématu i ve formě SQL skriptu, který po svém spuštění vytvoří definované tabulky a vztahy v konkrétním databázovém prostředí. Pro modelování databází lze využít širokého spektra softwarových nástrojů. Přehled významných nástrojů lze nalézt v (DatabaseAnswers, 2002).

Mezi hlavní výstupy patří:

- konceptuální, logický a fyzický datový model,
- SQL skripty pro vytvoření databáze.

3.10.5. Implementace a testování uživatelského rozhraní

Dalším krokem implementační fáze je implementace a testování uživatelského rozhraní webové aplikace. V rámci tohoto kroku je nejdříve nutné implementovat návrhy layoutů jednotlivých webových stránek a také transformovat jednotlivé obsahové jednotky na dříve

definované prvky uživatelského rozhraní. Možným implementačním prostředím je jazyk HTML, popř. technologie Flash. Při implementaci uživatelského rozhraní v jazyce HTML je vhodné využít služeb některého z HTML editorů. Přehled a porovnání často užívaných HTML editorů lze nalézt v (Wikipedia, 2006)

Společně se samotnou implementací uživatelského rozhraní je nutné jej testovat vůči požadavkům jednotlivých uživatelských skupin identifikovaným ve fázi analýzy uživatelských požadavků a případně provést odpovídající změny. Pro testování použitelnosti uživatelského rozhraní webových stránek lze využít některé metody prezentované v (Hejl, 2007). Společně s testováním použitelnosti je vhodné také testovat přístupnost webové aplikace. Více o způsobech testování přístupnosti lze nalézt v (Špinar, 2008).

Výstupem tohoto kroku je implementované a otestované uživatelské rozhraní webové aplikace.

3.10.6. Implementace a testování funkcionality

Následujícím krokem je implementace a testování funkcionality webové aplikace vycházející z identifikovaných uživatelských požadavků rozpracovaných v jednotlivých scénářích použití a z návrhu integrace mezi funkcionalitou webové aplikace a hypertextovým modelem. V závislosti na vybraném programovacím jazyce je vhodné využít možností široké škály integrovaných vývojových prostředí, viz (Wikipedia, 2007a). Mezi často využívaná vývojová prostředí pro vývoj webových aplikací lze zařadit:

- *ActiveState Komodo* – www.activestate.com/komodo/,
- *Eclipse* – www.eclipse.org,
- *Microsoft Visual Studio* - msdn.microsoft.com/vstudio,
- *NetBeans* – www.netbeans.org.

Stejně jako u implementace uživatelského rozhraní i u implementace funkcionality je nutné průběžně testovat vznikající webovou aplikaci a na základě výsledků testů rozhodovat o dalším postupu implementace. Naimplementovaná webová aplikace musí odpovídat identifikovaným uživatelským funkčním požadavkům. Více o testování softwarů se lze dozvědět z (Wikipedia, 2004). Hlavním výstupem je naimplementovaná a otestovaná webová aplikace.

3.10.7. Získání dat a naplnění databáze

Závěrečným krokem implementační fáze je získání dat, která budou využívána implementovanou webovou aplikací a naplnění databáze (vzniklé spuštěním vytvořených SQL skriptů) těmito daty. Data mohou být získána dvěma základními způsoby:

- převzetím z existujícího zdroje,
- vytvořením nových dat.

Převzetím dat z existujícího zdroje zde není myšleno kopírování textů z jiných webových stránek, ale např. využití dat dostupných ve vlastních interních informačních systémech (informace o produktech a službách) nebo využití textů použitých ve vlastních propagačních materiálech. Častějším případem je ovšem potřeba tvorby nových dat. Může se jednat např. o tvorbu marketingových textů, odborných článků, popisků k nabízeným produktům či titulků webových stránek. Oba způsoby je samozřejmě možné kombinovat, např. sepsání propagačního článku založeného na dostupných informacích o produktu.

Předtím než získaná data mohou být vložena do databáze, je nutné některá data upravit, aby odpovídala navržené struktuře databáze. Samotné vložení dat do databáze může být provedeno třemi, vzájemně se doplňujícími, způsoby:

- *pomocí SQL skriptů* – automatizovaný způsob pro získání dat ze zdrojového datového úložiště, transformaci dat a uložení dat do cílové databáze,
- *přímým vkládáním v databázovém prostředí* – vyžaduje od uživatele výbornou znalost navržené struktury databáze,
- *pomocí aplikačního rozhraní* – vkládání dat pomocí uživatelsky přívětivého rozhraní v rámci webové aplikace.

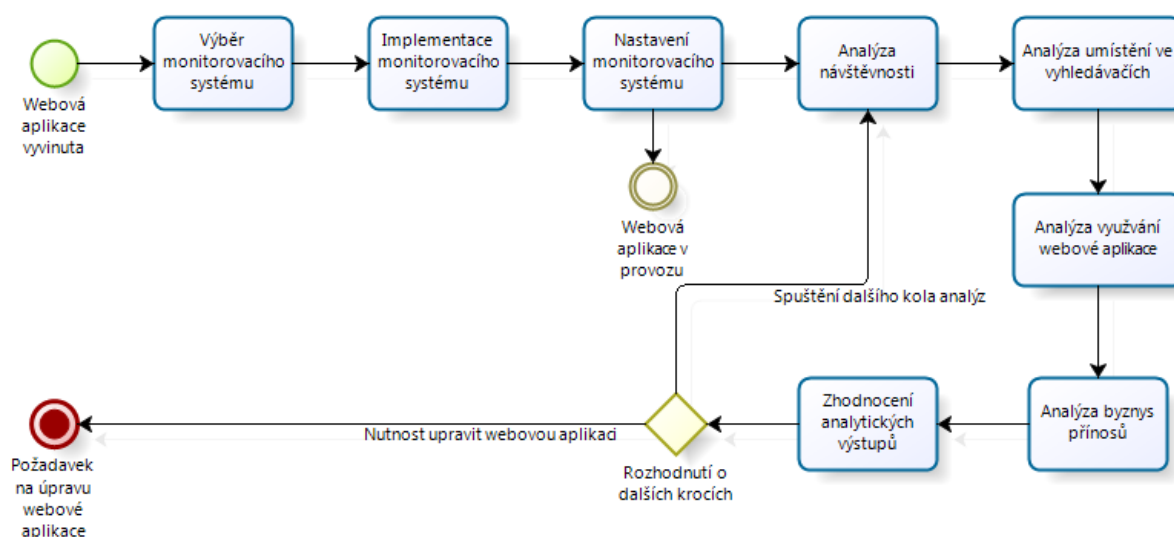
Mezi výstupy tohoto kroku lze zařadit:

- SQL skripty pro automatizovanou transformaci a přenos existujících dat do databáze,
- nově vytvořená data.

3.11. Fáze Provoz

3.11.1. Vývojový proces

Poslední fází celého vývojového procesu tvorby webu je samotný provoz naimplementované webové aplikace. V rámci provozu je kladen největší důraz na monitoring a analýzu návštěvnosti webu a využívání webové aplikace. Vývojový proces fáze *Provoz* je znázorněn na obrázku 3.16. Velmi důležitým prvkem procesu je vyhodnocení analytických výstupů a následné rozhodnutí o dalších krocích. Odpovídají-li výsledky analýz plánovaným cílům, je možné spustit nové kolo analýz, v opačném případě je nutné provést úpravy webové aplikace. Při úpravách webové aplikace se začíná opět od úvodní studie projektu a prochází se jednotlivé kroky a hledají se možnosti, jak webovou aplikaci vylepšit na základě výstupů z provozních analýz. Závěrem snahy o vylepšení aplikace může být i změna definovaných cílů či úplné ukončení prací na projektu a ukončení provozu webové aplikace.



Obrázek 3.16: Vývojový proces fáze Provoz

3.11.2. Výběr monitorovacího systému

Prvním krokem provozní fáze je výběr systému, pomocí kterého bude monitorován provoz webové aplikace. Tento systém by měl umožňovat analýzu zdrojů návštěvnosti webu, analýzu využívání webové aplikace a v neposlední řadě také analýzu byznys přínosů webové aplikace. Na trhu lze nalézt velké množství monitorovacích systémů, které se liší cenou, nabízenými možnostmi monitorování i samotným způsobem monitorování provozu.

Monitorovací systémy lze klasifikovat především na základě dvou kritérií (Omega Digital Media, 2008):

- způsob sběru dat,
- způsob dodání služby.

Podle způsobu sběru dat lze monitorovací systémy dělit následovně:

- *sběr dat na straně klienta* – založeno na vložení monitorovacího kódu do těla jednotlivých webových stránek,
- *sběr dat na straně serveru* – založeno na analýze logovacích souborů uložených na serveru,
- *hybridní způsob sběru dat* – založeno na analýze logovacích souborů společně s využitím monitorovacího kódu.

Zaměříme-li se na způsob dodání monitorovací služby, lze identifikovat dva základní způsoby:

- *softwarový balík* – software nakoupený od dodavatele, který je instalován a spouštěn na serveru, kde běží implementovaná webová aplikace.
- *hostovaná monitorovací služba* – služba dostupná prostřednictvím webového rozhraní běžící na serveru dodavatele služby.

Mezi nejznámější monitorovací systémy lze zařadit:

- *Awstats* – www.awstats.org,
- *Lyris ClickTracks* – www.clicktracks.com,
- *Google Analytics* – www.google.com/analytics,
- *Omniure SiteCatalyst* – www.omniure.com/en/products/online_analytics,
- *WebTrends Analytics* – www.webtrends.com/Products/WebTrendsAnalytics8.aspx.

Nejobsáhlejší seznam monitorovacích systémů lze nalézt v (Webanalyticsbook, 2007). Aktuální přehled nejvýznamnějších monitorovacích systémů tříděný dle způsobu monitorování poskytuje (CMS Watch, 2009). Výstupem tohoto kroku je vybraný monitorovací systém.

3.11.3. Implementace monitorovacího systému

Po výběru vhodného monitorovacího systému je nutné provést implementaci tohoto systému. Způsob implementace je závislý na způsobu sběru dat a způsobu dodání monitorovací služby:

- *softwarový balík* – nainstalování softwarového balíku,
- *hostovaná monitorovací služba* – založení účtu a vytvoření profilu pro danou webovou aplikaci,
- *sběr dat na straně klienta* – vložení monitorovacího kódu do všech webových stránek,
- *sběr dat na straně serveru* – nastavení vytváření logovacích souborů.

Výstupem je implementovaný monitorovací systém.

3.11.4. Nastavení monitorovacího systému

Poté, co je monitorovací systém implementován, je potřeba provést základní nastavení systému. V rámci nastavení systému je nejčastěji nutné:

- *specifikovat uživatele systému a nastavit jejich přístupová práva* – např. administrátorská práva či práva pouze k prohlížení výstupů,
- *nastavit filtry a pravidla* – např. pravidla pro shlukování webových stránek či filtry pro vyřazení určitých IP adres z monitorování,
- *nastavit cíle, cílové stránky a hodnoty cílů* – např. jedná-li se o e-shop, je možné sledovat proces objednávky, počet dokončených objednávek či vygenerované tržby.

Výstupem tohoto kroku je nastavený monitorovací systém. Více informací o monitorovacích systémech a jejich nastavení lze nalézt v (Cerman, 2007).

3.11.5. Analýza návštěvnosti

Po úspěšném dokončení nastavení monitorovacího systému je možné přistoupit k analýze návštěvnosti webových stránek. V rámci analýzy návštěvnosti je vhodné zaměřit se především na:

- *celkový počet návštěvníků a zobrazených stránek* – analýza hodnot návštěvnosti a počtu zobrazených stránek v průběhu času,

- *zdroje návštěvnosti* – identifikace převažujícího zdroje návštěvnosti (pomocí odkazu z jiných stránek, pomocí vyhledávače, přímým zadáním URL adresy), poměr mezi jednotlivými zdroji, identifikace používaných klíčových slov,
- *typologii návštěvníků* – klasifikace uživatelů podle geografického umístění, jazyka či loajality,
- *technické zázemí návštěvníků* – využívaný operační systém, webový prohlížeč, rozlišení obrazovky či rychlost připojení.

Výstupy toho kroku se skládají ze závěrů zjištěných v rámci jednotlivých analýz dle výše uvedených oblastí.

3.11.6. Analýza umístění ve vyhledávačích

Společně s analýzou návštěvnosti webu je nutné provést analýzu umístění webových stránek ve vyhledávačích. Podklady pro tuto analýzu tvoří seznam relevantních vyhledávačů a seznam klíčových slov identifikovaných v rámci fáze *Analýza konkurence*. Cílem tohoto kroku je určení umístění webových stránek při vyhledávání identifikovaných klíčových slov v relevantních vyhledávačích. Ačkoliv je možné kontrolovat pozice manuálním vyhledáváním, vhodnější je využití automatizovaného nástroje pro sledování pozic. Tyto nástroje je možné dělit podle dvou základních kritérií:

- možnosti vyhledávání,
- způsob implementace.

Dle možností vyhledávání je možné jednotlivé nástroje zařadit do jednoho z kvadrantů podle tabulky 3.8.

		Počet využívaných vyhledávačů	
		Jeden	Více
Počet vyhledávaných klíčových slov	Jedno	I	II
	Více	III	IV

Tabulka 3.8: Dělení automatizovaných nástrojů dle možností vyhledávání

Podle způsobu implementace lze automatizované nástroje pro zjišťování pozic ve vyhledávacích klasifikovat do tří základních skupin:

- *webové aplikace* – např. iWEBTOOL Search Engine Position Tool (kvadrant I),
- *doplňky do webových prohlížečů* – např. Firefox Rank Checker (kvadrant IV),
- *desktopové aplikace* – např. Free Monitor for Google (kvadrant III).

Mezi výstupy tohoto kroku lze zařadit:

- vybraný automatizovaný nástroj pro zjišťování pozic ve vyhledávacích,
- přehled umístění webových stránek na jednotlivá klíčová slova v relevantních vyhledávacích.

3.11.7. Analýza využívání webové aplikace

Dalším krokem v rámci provozní fáze je analýza samotného využívání webové aplikace. Cílem analýzy je zjištění způsobu využívání webové aplikace uživateli a identifikování případných slabých míst aplikace. Analýza využívání webové aplikace by se měla zaměřovat především na:

- *sledování návštěvnosti jednotlivých webových stránek* – určení nejvíce a nejméně navštěvovaných stránek v rámci aplikace, sledování trendů v návštěvnosti stránek v průběhu času, sledování využití aplikace v průběhu dne,
- *identifikace konečných webových stránek* – určení webových stránek, ze kterých návštěvníci nejčastěji opouštějí aplikaci,
- *analýzu využívání odkazů na jednotlivých webových stránkách* – určení, které odkazy na jednotlivých stránkách jsou nejvíce a nejméně využívány,
- *určení častých navigačních cest* – identifikace často se opakujících posloupností navštěvovaných webových stránek.

Výstupy z tohoto kroku se skládají ze závěrů dílčích analýz dle výše uvedených bodů.

3.11.8. Analýza byznys přínosů

Následujícím analytickým krokem je analýza byznys přínosů webové aplikace. Tento krok je možné vynechat u webových aplikací, které nejsou primárně zaměřené na prodej produktů a

služeb na internetu. Při analýze byznys přínosů se vychází z cílů a hodnot cílů definovaných při nastavování monitorovacího systému. V rámci této analýzy je vhodné sledovat:

- *počet dosažených cílů* – množství zobrazených nastavených cílových stránek v průběhu času,
- *hodnoty dosažených cílů* – celková hodnota vygenerovaná pro provozovatele na základě dosažených cílů,
- *cesty vedoucí k dosažení cílů* – nejčastější posloupnosti zobrazovaných stránek vedoucích k zobrazení nastavených cílových stránek

Mezi výstupy je možné zařadit:

- vypočítanou celkovou hodnotu generovanou webovou aplikací,
- seznam nejčastějších navigačních cest vedoucích ke splnění cíle.

3.11.9. Zhodnocení analytických výstupů

Posledním krokem provozní fáze je vyhodnocení analytických výstupů z předchozích kroků. Cílem tohoto zhodnocení je určení dalšího postupu při vývoji a provozu webové aplikace. V rámci tohoto kroku je nutné porovnat dosažené výsledky zjištěné jednotlivými analýzami s cíli stanovenými v úvodní fázi procesu. Porovnávání mohou být například:

- počty návštěvníků webových stránek,
- počty dosažených cílů,
- hodnoty generované dosažením cílů.

Na základě těchto porovnání je poté nutné rozhodnout o dalších krocích. Odpovídají-li zjištěné hodnoty plánovaných hodnotám nebo není-li prozatím dostatek údajů pro formulování závěrů, je vhodné po čase opět spustit další kolo provozních analýz. Jsou-li naopak naměřené hodnoty v rozporu s plánem, je vhodné přistoupit k úpravě webové aplikace. Úpravu webové aplikace je nutné začít od samotného začátku od úvodní studie projektu.

Výstupem tohoto kroku je rozhodnutí o dalším postupu při provozu a vývoji webové aplikace.

3.12. Shrnutí kapitoly

Hlavním cílem této kapitoly bylo navrzení procesního modelu pro vývoj webové aplikace s využitím informací získaných při analýze metodik pro vývoj webových aplikací. Před samotným návrhem bylo nutné identifikovat nedostatky současných webových metodik. Za hlavní nedostatek těchto metodik lze považovat primární zaměření na návrh a vývoj aplikace a opomenutí aspektů týkajících se analýzy konkurence a implementace a provozu aplikace.

Navržený proces pro vývoj webu se skládá z osmi základních fází:

- *úvodní analýza projektu* – určení předmětu projektu, stanovení cílů, provedení časové, zdrojové a nákladové analýzy,
- *analýza konkurenčního prostředí* – identifikace relevantních konkurenčních segmentů, analýza konkurence, provedení SWOT analýzy,
- *analýza uživatelských požadavků* – identifikace uživatelských tříd a skupin, specifikace uživatelských požadavků, tvorba scénářů použití,
- *datový návrh* – identifikace objektových tříd a vzájemných vztahů mezi těmito třídami,
- *hypertextový návrh* – návrh navigační struktury webové aplikace, navržení kompozice webových stránek a integrace s funkcionalitou webové aplikace,
- *návrh uživatelského rozhraní* – navržení layoutu webových stránek, návrh způsobu transformace definovaných obsahových jednotek,
- *implementace* – výběr implementačního prostředí, transformace datového modelu, implementace a testování uživatelského rozhraní a funkcionality, naplnění databáze daty,
- *provoz* – výběr a nastavení monitorovacího systému, analýza návštěvnosti a využívání webové aplikace, zhodnocení analytických výstupů.

4. Implementace principů SEO do procesního modelu

Cílem této kapitoly je krátké uvedení do problematiky optimalizace pro vyhledávače (SEO) a především implementace principů SEO v rámci navrženého procesního modelu. Implementace principů SEO bude mít charakter doporučení sdružených do větších tematických celků. Všechna doporučení vychází z (Planet Ocean, 2009), pokud není uvedeno jinak.

4.1. Úvod do problematiky SEO

4.1.1. Definice SEO

Před tím, než se budeme věnovat samotné implementaci principů SEO, je nutné definovat, co se pod zkratkou SEO skrývá. SEO je akronymem pro *Search Engine Optimization*, což je do češtiny nejčastěji překládáno jako *optimalizace pro vyhledávače*. Z možných definic tohoto pojmu je možné vybrat například následující:

„Proces výběru cílových klíčových slov vztahujících se k webu a zajištění, že se web bude umísťovat dobře ve výsledcích vyhledávání těchto klíčových slov.“

(Media Generation Nine, 2008)

„Proces zvyšování množství a kvality návštěvníků webu přicházejících z vyhledávačů pomocí přirozených (organických, algoritmických) výsledků vyhledávání.“ (Wikipedia, 2004)

Nejobsáhlejší přehled definic od nejvýznamnějších osobností zabývajících se problematikou SEO a internetového marketingu lze nalézt v (Schoemaker, 2008). Na základě různých definic pojmu SEO je možné vysledovat určité společné znaky, které mohou SEO charakterizovat:

- SEO je opakující se proces, ne jednorázová činnost,
- cílem SEO je zvýšení množství a kvality návštěvníků webu,
- prostředkem pro zvýšení návštěvnosti je zlepšení umístění ve výsledcích vyhledávání relevantních klíčových slov.

4.1.2. SEO jako marketingový nástroj

Budeme-li pohlížet na SEO z marketingového hlediska, lze jej považovat za jeden z nástrojů internetového marketingu, konkrétněji se jedná o jednu z forem marketingu ve vyhledávačích. Pojem marketing ve vyhledávačích (*Search Engine Marketing, SEM*) zahrnuje takové marketingové postupy, které jsou využívány pro propagaci webových stránek v rámci vyhledávačů. Jednotlivé formy SEM jsou představeny v tabulce 4.1.

Forma SEM	Anglický termín	Zkratka	Český termín
Zápis do katalogu	Pay For Inclusion	PFI	Platba za zařazení
Placené umístění ve výsledcích vyhledávání	Paid Placement	PP	Platba za umístění
Kontextová reklama	Pay Per Click	PPC	Platba za proklik
Optimalizace pro vyhledávače	Search Engine Optimization	SEO	Optimalizace pro vyhledávače

Tabulka 4.1: Formy marketingu ve vyhledávačích

Formy SEM lze dělit na základě dvou kritérií (viz tabulka 4.2).

		Využívaná služba vyhledávače	
		Fulltextový vyhledávač	Katalog webových stránek
Nutnost platby vyhledávači	Placené	PP, PPC	PFI
	Neplacené	SEO	PFI

Tabulka 4.2: Dělení forem SEM

S používáním optimalizace pro vyhledávače jsou spojeny určité výhody i nevýhody oproti ostatním formám marketingu ve vyhledávačích (Krutiš, 2005). Mezi výhody lze zařadit:

- nízké náklady na dlouhodobou propagaci webu,
- dobrá optimalizace je účinná ve více vyhledávačích a přiláká více návštěvníků,
- dlouhodobý účinek.

K nevýhodám SEO patří:

- nutnost reagovat na často se měnící algoritmy vyhledávačů,
- nutnost tvorby stále nového obsahu webu u vysoce konkurenčních oborů,
- není zaručen výsledek SEO tj. není jisté, že po provedení optimalizace bude web zobrazen vysoko ve výsledcích vyhledávání.

4.1.3. Optimalizační proces

Jak již bylo výše uvedeno, optimalizace pro vyhledávače není jednorázovou činností ale opakujícím se procesem. Dle (Krutiš, 2005) se optimalizační proces skládá z pěti opakujících se kroků:

- *určení cílových skupin webu* – identifikace typických skupin uživatelů, které by měly využívat webovou aplikaci,
- *analýza klíčových slov* – identifikování relevantních klíčových slov a analýza konkurence,
- *optimalizace on-page faktorů* – faktory, které jsou přímo spojené s kódem a obsahem webových stránek a s navigační strukturou celé webové aplikace,
- *optimalizace off-page faktorů* – faktory, které souvisí s internetovým okolím webové aplikace,
- *monitoring výsledků* – monitoring a analýza důsledků provedených optimalizačních postupů.

Frekvence opakování jednotlivých kroků se ovšem liší. Zatímco kroky optimalizace on-page a off-page faktorů a monitorování výsledků probíhají téměř neustále, určení cílových skupin webu a analýza klíčových slov se provádí vždy při startu optimalizačního procesu a pak již jen zřídka.

Většinu kroků optimalizačního procesu lze namapovat na činnosti definované v rámci procesního modelu pro vývoj webové aplikace. Tabulka 4.3 obsahuje přehled mapování optimalizačních kroků na fáze a činnosti navrženého procesu a také informace o tom, zda je obsah těchto kroků již dostatečně zachycen v rámci popisu procesního modelu.

Krok optimalizačního procesu	Fáze navrženého procesního modelu	Činnost v rámci fáze navrženého procesního modelu	Pokryto
Určení cílových skupin webu	Úvodní analýza projektu	Identifikace cílových uživatelů	Ano
Analýza klíčových slov	Analýza konkurenčního prostředí	Celá fáze	Ano
Optimalizace on-page faktorů	Implementace	Výběr webhostingu a domény a Získání dat a naplnění databáze	Ne

Optimalizace off-page faktorů	Implementace	Neodpovídá žádné činnosti ³¹	Ne
Monitoring výsledků	Provoz	Celá fáze	Ano

Tabulka 4.3: Mapování optimalizačních kroků na fáze a činnosti navrženého procesního modelu

Jak z tabulky 4.3 vyplývá, obsah optimalizačních kroků *Určení cílových skupin webu*, *Analýza klíčových slov* a *Monitoring výsledků* byl již pokryt v rámci popisu odpovídajících činností při návrhu procesního modelu. V následujících kapitolách tudíž budou podrobněji popsána pouze doporučení a postupy týkající se optimalizace on-page a off-page faktorů. Oba typy optimalizace probíhají v rámci fáze *Implementace* navrženého procesního modelu.

4.2. Výběr doménového jména

První sada doporučení se týká výběru doménového jména. Doménové jméno se skládá z několika částí oddělených tečkou, např. *analytics.google.com*. Jednotlivé části doménového jména je možné číslovat zprava doleva, tj. *com* představuje doménu první úrovně (neboli top level doménu, TLD), *google.com* představuje doménu druhé úrovně a *analytics.google.com* je doména třetí úrovně. Při registraci doménového jména je ovšem důležitá pouze doména druhé úrovně.

Existují tři základní přístupy k výběru primárních doménových jmen:

- doménové jméno se shoduje s názvem podniku, organizace či se jménem osoby,
- doménové jméno se skládá z relevantních klíčových slov,
- krátké a zapamatovatelné doménové jméno, které lze dobře propagovat.

Jednotlivé přístupy lze ovšem i vzájemně kombinovat, tj. například zaregistrovat doménové jméno shodující se s názvem podniku a spolu s ním zaregistrovat také doménové jméno, které se bude skládat s klíčovými slovy a bude sloužit k propagaci specifického výrobku. Doména první úrovně pro daná primární doménová jména by měla vždy odpovídat hlavnímu trhu, na který je web cílen. Společně s TLD hlavního trhu by mělo být zaregistrováno i doménové jméno s TLD *com*.

³¹ Optimalizace off-page faktorů nemůže být namapována na žádnou činnost navrženého procesního modelu pro vývoj webové aplikace, jelikož off-page faktory nesouvisí se samotnou aplikací a jejím vývojem, ale vyjadřují vztah webového okolí k implementované webové aplikaci.

Kromě primárních doménových jmen je vhodné zaregistrovat i několik sekundárních doménových jmen. Tato doménová jména mohou být tvořena následovně:

- přidáním či odebráním pomlček z primárního doménového jména,
- jako nejčastěji překlady anebo chybné zápisy primárního doménového jména,
- změnou TLD primárního doménového jména.

Při volbě TLD je vhodné vyvarovat se domény *info*, jelikož tyto domény jsou často zneužívány spammery (díky jejich nízké pořizovací ceně), a proto se netěší důvěře návštěvníků. Podle některých názorů, např. (TzuVelli, 2008), i samotné vyhledávače těmto doménám přisuzují nižší důvěryhodnost.

Při výběru doménových jmen lze často narazit na situaci, že doménové jméno má již zaregistrován jiný subjekt. Pro zjištění, zda je doménové jméno již použito a případně kdo ho má zaregistrováno, lze využít služeb tzv. *Whois serverů*, např. *whois.domaintools.com*. Je-li požadované doménové jméno zaregistrováno jiným subjektem, existují stále určité možnosti, jak lze doménové jméno získat:

- vyčkání do konce období, po které má stávající vlastník registrovanou doménu, s očekáváním, že stávající vlastník zapomene prodloužit registraci,
- přímé kontaktování vlastníka doménového jména ohledně možností prodeje,
- využití internetových služeb zajišťujících nákup a prodej domén,
- využití internetových služeb nabízejících aukce domén.

Mezi významné zástupce internetových služeb nabízejících nákup, prodej a aukce domén lze zařadit především:

- *BuyDomains* – www.buydomains.com,
- *Go Daddy Auctions* – www.tdnam.com,
- *Sedo* – www.sedo.com,
- *SnapNames* – www.snapnames.com.

4.3. Vytvoření webu přívětivého k vyhledávacím robotům

Druhá sada doporučení pro on-page optimalizaci se týká přívětivosti webu k vyhledávacím robotům. Vyhledávací roboty jsou programy vytvořené programátory jednotlivých vyhledávačů, které postupně procházejí internet a sbírají informace o jednotlivých webových stránkách. Na základě těchto informací vyhledávače poté zařazují dané stránky do své databáze (provádí indexaci stránek) a tyto stránky mohou být následně prostřednictvím vyhledávače nalezeny uživateli. Je tedy nutné vyhledávacím robotům co nejvíce usnadnit práci, neboť jakákoliv překážka může zkomplikovat indexaci a následnou vyhledatelnost webových stránek.

4.3.1. Struktura URL adres

První oblastí, kterou je nutné se zabývat, je struktura URL adres používaných v rámci webové aplikace. URL lze v zásadě dělit na dva typy:

- *statické URL adresy* – adresy neobsahující žádné parametry, např. `www.eshop.cz/boty`,
- *dynamické URL adresy* – adresy využívající URL parametrů, např. `www.eshop.cz/index.php?zbozi=boty&barva=cerna`.

Ačkoliv vyhledávače zlepšují svoje vyhledávací algoritmy a některé z nich již zvládají na slušné úrovni pracovat s dynamickými URL adresami, doporučuje se přímo používat krátké a výstižné statické adresy nebo provádět transformace dynamických adres na statické např. pomocí *Apache mod_rewrite*³² nebo *ISAPI*³³.

Doporučuje se také nepoužívat v rámci URL žádné identifikátory návštěvníka, např. *session ID* pro sledování jeho pohybu v prostředí webové aplikace. Statické URL adresy by navíc neměly obsahovat takové části, které nepřinášejí žádnou dodatečnou informaci pro návštěvníky ani pro vyhledávače např. zbytečné užívání *index.html* v rámci URL.

V rámci URL adres jednotlivých webových stránek by měla být použita klíčová slova, která jsou relevantní k obsahu dané stránky. Ačkoliv prozatím nebyl prokázán významný vliv

³² http://httpd.apache.org/docs/2.0/mod/mod_rewrite.html

³³ <http://en.wikipedia.org/wiki/ISAPI>

přítomnosti klíčových slov v URL adrese na zlepšení umístění ve výsledcích, je vhodné je využívat, jelikož jsou tato klíčová slova při vyhledávání zvýrazněna a mohou zvýšit počet kliknutí na daný odkaz.

4.3.2. Sitemapy (mapy webu)

Pro zvýšení přívětivosti webové aplikace k vyhledávacím robotům, je vhodné využít tzv. *mapy webu (sitemapy)*. V současnosti lze využít dva typy map webu:

- XML sitemapy,
- HTML sitemapy.

XML sitemapy vznikly ze společné iniciativy společností Google, Yahoo a Microsoft, aby jednak usnadnily vyhledávačům nalézání nových webových stránek a naopak také daly webmasterům možnost informovat vyhledávače o tom, že vznikla nebo byla upravena webová stránka. XML sitemapy poskytují vyhledávačům informace o stránkách, ze kterých se webová aplikace skládá. Mezi poskytované informace patří:

- URL adresa webové stránky,
- datum poslední úpravy webové stránky,
- frekvence změn webové stránky,
- relativní priorita webové stránky vůči ostatním stránkám.

XML mapa webu by měla být uložena v kořenovém adresáři webu, tj. měla by být dostupná na adrese *www.domena.cz/sitemap.xml*. Je-li umístěna jinde, měl by být na ni uveden odkaz ze souboru *robots.txt* (více informací o robots.txt viz následující kapitola). Podrobné informace o XML sitemapách lze nalézt na oficiálním webu *www.sitemaps.org*.

Zatímco XML sitemapy slouží pouze vyhledávačům, HTML sitemapy slouží navíc i návštěvníkům webu. Pro vyhledávače mají oba typy map webu stejný význam – umožňují jim zjistit strukturu webu a snadněji tak zaindexovat jednotlivé webové stránky. Pro správnou funkci mapy webu jako rozcestníku pro vyhledávače je vhodné mít na stránce maximálně 100 odkazů a velikost HTML kódu stránky by neměla překročit 101 KB. Je-li web obsáhlejší, je možné vytvořit více HTML sitemap, které budou buď vzájemně propojené, nebo budou dostupné přes odkaz ze stránky, která bude obsahovat odkazy na všechny dílčí sitemapy. HTML sitemapa by měla být pomocí odkazu dostupná nejlépe z každé webové stránky.

4.3.3. Soubor robots.txt

Soubor robots.txt představuje mechanismus, jak říci vyhledávacím robotům, které stránky nemá procházet a tudíž je nemá odpovídající vyhledávač indexovat. Pokud mají být indexovány všechny stránky, není nutné soubor robots.txt používat. Tento soubor také neslouží pro zabezpečení citlivých informací v rámci webové aplikace, jelikož na jeho základě pouze vyhledávací roboti neprocházejí stanovené stránky a naopak pro živého návštěvníka mohou být blokovány stránky uvedené v tomto souboru branou k zajímavým informacím.

Základní syntaxe souboru robots.txt se skládá ze dvou direktiv:

- *User-Agent* – určuje jméno vyhledávacího robota, pro kterého platí dále uvedená omezení,
- *Disallow* – určuje relativní cestu k souboru či adresáři, které nemají být procházeny.

Společnosti Google, Yahoo a Microsoft se v červnu 2008 dohodly na rozšíření specifikace a vznikl *Robots Exclusion Protocols (REP)*, který navíc obsahuje (Search Tools Consulting, 2008):

- *direktivu Allow* – specifikuje relativní cestu, kterou by měl robot procházet,
- *podporu zástupného znaku ** – zastupuje libovolný počet znaků v rámci specifikované relativní cesty,
- *direktivu Sitemap* – určuje cestu k XML sitemap.

Více informací o souboru robots.txt lze získat z (Search Tools Consulting, 2008) a také ze stránek www.robotstxt.org.

4.3.4. Omezení využívání špatně indexovatelných technologií

Dle (Planet Ocean, 2009) existují tři hlavní technologie používané při tvorbě webových stránek, které zhoršují indexovatelnost jednotlivých webových stránek. Jedná se o:

- rámce (frames),
- JavaScript,
- Flash.

Z hlediska SEO se doporučuje tyto technologie nepoužívat. Je-li je však nutné z určitých důvodů používat, je potřeba držet se následujících doporučení:

- při použití rámců je nutné přidat do HTML kódu sekci `<noframes>`, která bude obsahovat obsah určený pro vyhledávače, který ovšem musí být relevantní k obsahu nabízenému návštěvníkovi pomocí rámců,
- při použití technologie Flash je nutné přidat do HTML kódu sekci `<noembed>`, která bude obsahovat obsah určený pro vyhledávače a návštěvníky bez podpory této technologie,
- Flash by měl být použit pouze pro část stránky, přičemž zbytek by měl být v HTML a obsahovat klíčová slova relevantní k obsahu Flash prezentace,
- JavaScript by měl být umístěn do externího souboru, čímž se sníží velikost webové stránky a zvýší se také znovupoužitelnost daného skriptu.

4.4. Klíčová slova a jejich umístění

4.4.1. Titulek webové stránky

Titulek je jedním z nejdůležitějších prvků každé webové stránky, jelikož poskytuje návštěvníkům i vyhledávačům základní informace o tom, co je obsahem dané stránky. Z hlediska optimalizace pro vyhledávače má titulek klíčový význam. V HTML kódu je titulek stránky vyjádřen jako element `<title>` v rámci hlavičky webové stránky.

Při tvorbě titulků webových stránek je vhodné držet se následujících doporučení vycházejících z (Planet Ocean, 2009) a (Bobnar, Mahaney, 2009):

- *titulek by měl mít maximálně 60 znaků* – tato délka zajistí zobrazení celého titulku ve výsledcích vyhledávání ve většině vyhledávačů,
- *neměla by se používat slova s nulovou hodnotou*³⁴ – např. domovská stránka, homepage, web page, untitled page, nová stránka,
- *mělo by být minimalizováno použití stopových slov* – např. členy v angličtině, předložky či spojky, avšak nesmí to být na úkor čitelnosti titulku,

³⁴ Přes 7 milionů webových stránek má v titulku „untitled page“ a přes 80 milionů webových stránek obsahuje v titulku „welcome to our homepage“.

- *každá webová stránka by měla mít unikátní titulek* – každá stránka by měla být cílena na jiná klíčová slova,
- *nejdůležitější klíčová slova by měla být co nejbližší začátku titulku* – vyhledávače přikládají důležitost pozici klíčových slov v rámci titulku,
- *titulek by měl obsahovat dvě varianty nejdůležitějšího klíčového slova* – např. jednotná a množná čísla nebo prohozené pořadí slov,
- *titulek by neměl být přecpaný opakujícími se klíčovými slovy* – takový titulek je penalizován vyhledávači,
- *titulek by neměl obsahovat klíčová slova, která nesouvisí s obsahem stránky* – jedná se o nedodržení samotného významu titulku jako zdroje informací o obsahu stránky.

4.4.2. Metatagy description a keywords

V rámci hlavičky HTML stránky je možné specifikovat také tzv. *metatagy*. Metatagy slouží k uvedení popisných informací (metadat) o dané webové stránce. Mezi nejčastěji používané metatagy patří:

- *meta description* – obsahuje krátký popis webové stránky,
- *meta keywords* – obsahuje seznam klíčových slov vztahujících se k webové stránce,
- *meta author* – obsahuje přehled autorů webové stránky,
- *meta Content-type* – umožňuje specifikovat MIME³⁵ typ a použité kódování webové stránky.

Z hlediska optimalizace pro vyhledávače mají dle (SEOMoz, 2007) alespoň minimální význam pouze první dva metatagy. Metatag keywords dle (Sullivan, 2007) nemá žádný význam pro vyhledávače Google a Microsoft Live a vyhledávače Yahoo a Ask mu přikládají určitý význam při vyhledávání. Metatag description nemá přímý dopad na zlepšení umístění ve vyhledávání, ale umožňuje zvýšit počet kliknutí na odkaz z výsledků vyhledávání, jelikož obsah metatagu description bývá zobrazen společně s titulkem webové stránky v rámci těchto výsledků.

³⁵ Multipurpose Internet Mail Extensions – internetový standard původně určený pro popis typu obsahu pro přílohy emailových zpráv, později používaný pro popis jakéhokoliv elektronického obsahu.

Při tvorbě obsahu metatagů description a keywords je vhodné řídit se následujícími doporučeními:

- vždy používat metatag description,
- maximální velikost textu v rámci metatagů by neměla překročit 250 znaků,
- nejvýznamnější klíčová slova by se měla vyskytovat co nejbližší začátku metatagů.

Speciálně pro metatagy keywords platí:

- neopakovat stejná klíčová slova,
- používat pouze klíčová slova, která jsou relevantní k dané webové stránce,
- vyvarovat se používání chráněných obchodních značek,
- zahrnout běžné překlady klíčových slov.

4.4.3. Tělo webové stránky

Zatímco předchozí dvě kapitoly se věnovaly doporučením pro klíčová slova umístěná v hlavičce stránky (titulek a metatagy), tato kapitola je zaměřena na klíčová slova umístěná v rámci těla webové stránky tj. v hlavní oblasti zobrazované ve webovém prohlížeči. Dále uvedená doporučení je vhodné vzít v potaz při tvorbě obsahu jednotlivých webových stránek. Kromě čistého textu by klíčová slova měla být umístována především v rámci:

- *nadpisů (elementy <h1> až <h6>)* – představují druhé nejvýznamnější umístění klíčových slov po titulku stránky, přičemž význam pro SEO klesá od nadpisu první úrovně k nadpisu šesté úrovně,
- *významových elementů , <i>, , , * – vyhledávače přikládají význam klíčovým slovům umístěným v rámci těchto elementů, jelikož tyto elementy slouží ke zdůraznění části obsahu,
- *textu a titulku odkazů* – vyhledávače přikládají význam klíčovým slovům v textu odkazů a titulek odkazu jim umožňuje získat další informace o cíli odkazu,
- *alternativního textu a titulku obrázků* – dávají obrázkům význam pro vyhledávače a umožňují sdělit informaci o obrázku pro návštěvníky, kteří mají vypnuté zobrazování obrázků.

Pro zesílení účinku optimalizace je možné kombinovat některá výše uvedená doporučení, např. umístit klíčová slova do elementu `<b>`, z něj učinit text odkazu a vše označit jako nadpis, tj. `<h1><a href="stranka.html"><b>klíčová slova</b></a></h1>`. Nejdůležitější klíčová slova by se měla nacházet co nejbližně začátku stránky, z hlediska umístění v HTML kódu.

Při tvorbě obsahu je také nutné uvažovat počet výskytů jednotlivých klíčových slov v poměru k celkovému počtu slov na stránce³⁶ tj. *hustotu klíčového slova*. Při nadměrné hustotě bude s největší pravděpodobností webová stránka penalizována vyhledávači. Problém ovšem nastává při snaze určit optimální hustotu, jelikož neexistuje žádná přesně definovaná hodnota, jejíž překročení by vedlo k penalizaci stránek a nedosažení by naopak nemělo žádný pozitivní vliv na umístění ve vyhledávačích. Na základě různých experimentů však bývá nejčastěji uváděno, že hustota klíčových slov by se měla pohybovat od dvou do sedmi procent.

Tvorba textů, která akcentuje kromě samotného obsahu také potřebu uvážlivého umístění klíčových slov v rámci textu a hustotu klíčových slov v rámci webové stránky, se nazývá *copywriting*.

4.5. Link building (získávání odkazů)

4.5.1. Základní pojmy

Zatímco předchozí tři kapitoly byly věnovány on-page optimalizaci, tato kapitola je zaměřena na off-page optimalizaci, konkrétně na problematiku tzv. *link buildingu* neboli získávání odkazů z webových stránek mimo optimalizovaný web. Na začátku je nutné představit některé základní pojmy využívané při link buildingu.

Máme-li dvě webové stránky A a B a jeden odkaz vedoucí mezi nimi (viz obrázek 4.1), pak lze tvrdit, že:

- A odkazuje na B,
- A má odchozí odkaz na B,

³⁶ Od celkového počtu slov je nutné odečíst počet tzv. stopových slov jako např. předložky či členy.

- B má příchozí odkaz z A,
- existuje jednosměrný odkaz z A do B.



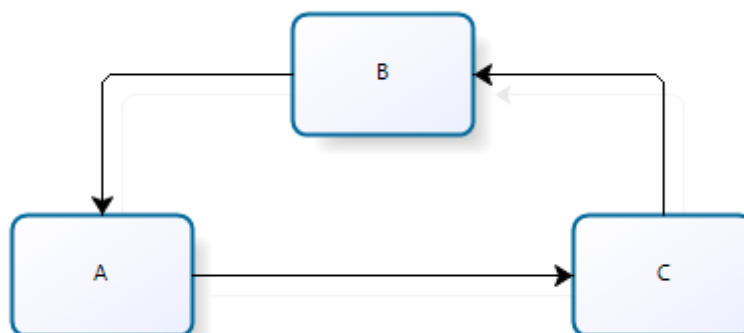
Obrázek 4.1: Jednosměrný odkaz

Máme-li stejné dvě webové stránky A a B, které ovšem na sebe vzájemně odkazují (viz obrázek 4.2), nazývají se tyto odkazy jako *reciproční odkazy*.



Obrázek 4.2: Reciproční odkazy

Reciproční odkazy mají obecně pro vyhledávače nižší hodnotu než jednosměrné odkazy, jelikož vznik recipročních odkazů nevypadá pro vyhledávače tak přirozeně jako vznik jednosměrných odkazů. K obelstění vyhledávačů jsou proto často používána složitější odkazová schémata, do kterých je zapojeno více webových stránek. Typickým případem jsou *triangulární odkazová schémata* (viz obrázek 4.3). Vyhledávače jsou ovšem v současnosti schopné triangulární odkazová schémata odhalit.



Obrázek 4.3: Triangulární odkazové schéma

4.5.2. Výměna recipročních odkazů

Ačkoliv je možné provádět výměnu recipročních odkazů mezi weby, které patří stejnému vlastníkovi, ve většině případů počet vlastněných webů nebude dostačující pro získání dostatečného množství kvalitních odkazů a bude nutné získávat odkazy i z jiných webových stránek. Dle (Planet Ocean, 2009) se proces navázání partnerství pro výměnu recipročních odkazů skládá z následujících kroků:

1. Vyhledání prvních 30 webových stránek umístujících se na cílené klíčové slovo.
2. Vyřazení konkurenčních webových stránek.
3. Výběr webových stránek nabízejících komplementární produkty, služby či informace.
4. Vytvoření a odeslání emailů s žádostí o přidání odkazu osobám odpovědným za správu vybraných webových stránek.

Email s žádostí o přidání odkazu by měl být adresovaný nejlépe administrátorovi webových stránek popř. osobě zodpovědné za internetový marketing a měl by být psán osobně, aby se předešlo podezření, že byl email odeslán nějakým automatizovaným skriptem. V rámci obsahu emailu by mělo být uvedeno především:

- výhody plynoucí potenciálnímu partnerovi z výměny odkazů,
- požadovaná cílová URL adresa, text odkazu a titulek odkazu, ze kterých by se měl skládat vložený odkaz,
- webová stránka, ze které je již nyní veden odkaz na webové stránky partnera.

V emailu je též možné specifikovat kvalitativní požadavky na umístění odkazu, které je poté vhodné pravidelně kontrolovat:

- *množství návštěvníků proudících na webovou stránku s umístěným odkazem* – odkaz se může stát zdrojem návštěvnosti,
- *umístění odkazu v rámci layoutu stránky* – ideální umístění je nad zlomem stránky³⁷, např. v hlavním menu, v bočním menu nebo na začátku článku,
- *frekvence indexování stránky* – frekvence jak často prochází vyhledávací roboti danou webovou stránku,

³⁷ Prostor nad zlomem stránky je ta část obsahu, která je viditelná pro návštěvníka, aniž by musel stránkou posunovat.

- *množství odchozích odkazů ze stránky* – čím méně je odchozích odkazů, tím lépe,
- *zdroje příchozích a cíle odchozích odkazů stránky* – důležité je, aby ze stránky nebylo odkazováno na škodlivé webové stránky, resp. aby škodlivé stránky neodkazovaly na danou stránku, což by stránku začleňovalo do tzv. *špatného okolí* (*bad neighbourhood*),
- *typ odkazu* – řazeno dle významu pro SEO – klasický odkaz, odkaz pomocí JavaScriptu, odkaz pomocí přesměrování, odkaz s uvedeným atributem *nofollow*.

Mezi další doporučení týkající se recipročních odkazů patří:

- *vyvarovat se tzv. side-wide odkazů* – odkazy vedoucí ze všech webových stránek partnerského webu nevypadají pro vyhledávače jako přirozeně vzniklé, a proto jim přiřkládají nižší váhu nebo je přímo penalizují,
- *zachovat konzistenci URL adresy, kterou se odkazuje* – nestřídat varianty s a bez počátečního *www*, s a bez závěrečného lomítka, s a bez uvedení *index.html*,
- *nepoužívat stále stejný text v příchozích odkazech* – odkazy s různým textem vypadají pro vyhledávače přirozeněji,
- *všechny odkazy nesměrovat pouze na domovskou stránku* – přirozeněji pro vyhledávače vypadá proměnlivý cíl příchozích odkazů

4.5.3. Pokročilé techniky link buildingu

Zatímco výměna recipročních odkazů je v současnosti běžnou praxí, existují i pokročilejší a méně využívané způsoby získávání odkazů na vlastní webové stránky, které mohou při správném použití přinést mnohem lepší výsledky než výměna recipročních odkazů.

Prvním způsobem, který je možné využít, je *tvorba vlastních článků*, které budou nabízet takový obsah, že na něj budou autoři jiných webových stránek automaticky odkazovat, a jejich umístění na vlastní web. Může se jednat například o:

- články týkající se daného oboru,
- návody a tutoriály,
- řešení problémů, které trápí větší okruh návštěvníků,
- rozhovory s významnými osobnostmi.

Kromě umístění článků na vlastní web je možné tyto články také *zasílat do katalogů článků*³⁸ nebo je *nabízet k publikaci jiným webům* pod podmínkou, že v rámci článku bude uveden odkaz na umístění článku na vlastním webu anebo odkaz na domovskou stránku. Vhodné je také nenabízet celé články, ale pouze jejich části s uvedením odkazu na plnou verzi článku na vlastním webu, což dává čtenářům informaci, že právě náš web je zdrojem unikátních článků a informací a příště čtenáři budou chodit právě na náš web a navíc nevznikají problémy s duplicitním obsahem dostupným na různých místech internetu. Při nabízení obsahu jiným webům je nutné pravidelně kontrolovat, zda tyto weby dodržují dohodnuté podmínky a také, zda některé weby neoprávněně nekopírují naše články. Ke zjištění webů, které publikují námi vytvořené články lze využít například služeb Google Alerts³⁹ nebo Copyscape⁴⁰.

Kromě tvorby a publikování článků nabízejících unikátní obsah je vhodné také publikovat texty, které slouží k propagaci podniku, organizace či jednotlivce. Takovéto texty je možné publikovat ve formě:

- příspěvků na blogu,
- tiskových prohlášení a PR článků.

Další pokročilou technikou pro získávání odkazů je využívání virálního marketingu a sociálních sítí. Může se jednat například o:

- *vyvinutí unikátní aplikace nebo služby* – uživatelé budou spontánně odkazovat na umístění, kde si lze aplikaci stáhnout nebo začít využívat službu,
- *vytvoření profilu a přispívání v rámci sociálních sítí* – např. Facebook⁴¹, Twitter⁴², MySpace⁴³, LinkedIn⁴⁴.

³⁸ Seznam padesáti nejvýznamnějších katalogů článků lze nalézt na <http://www.vretoolbar.com/articles/directories.php>.

³⁹ <http://www.google.com/alerts>

⁴⁰ <http://www.copyscape.com>

⁴¹ <http://www.facebook.com>

⁴² <http://www.twitter.com>

⁴³ <http://www.myspace.com>

⁴⁴ <http://www.linkedin.com>

Mimo výše uvedené způsoby je možné mezi pokročilé techniky zařadit také:

- *nákup opuštěných webů* – jedná se o nalezení⁴⁵ a vytvoření nabídky pro koupi webů, které jsou delší dobu nevyužívané, zacílené na relevantní klíčová slova a které mají za dobu své existence již nashromážděno určité množství příchozích odkazů,
- *přispívání do diskuzí a fór* – je nutné často a hodnotně přispívat s využitím odkazu v patičce svého příspěvku a získat si dobrou reputaci v rámci diskuzních fór,
- *nákup reklam v newsletterech rozesílaných vhodné cílové skupině* – zaprvé se jedná o dobře cílenou reklamní kampaň a zadruhé jsou často tyto newslettery poté archivovány na webu, což zaručuje dlouhodobě hodnotné odkazy,
- *získávání odkazů z TLD domén gov a edu* – tyto odkazy mají vyšší váhu než odkazy z jiných TLD; získat takovéto odkazy je možné například pomocí nabízení různých akcí studentům, jestliže umístí odkaz směřující na náš web na své osobní stránky hostované na univerzitní edu doméně,
- *vytvoření affiliate⁴⁶ programů* – každý affiliate partner automaticky vytváří odkazy směřující na web tvůrce affiliate programu,

Dle (Planet Ocean, 2009) je při získávání odkazů vhodné dodržovat *pravidlo 80/20* pro strukturu příchozích odkazů:

- 80 % odkazů z tematických stránek, 20 % z jiných,
- 80 % odkazů na domovskou stránku, 20 % jinam,
- 80 % odkazů s klíčovým slovem v odkazu, 20 % např. s využitím doménového jména nebo „*click here*“,
- 80 % jednosměrných odkazů, 20 % recipročních odkazů.

Tato struktura příchozích odkazů se jeví vyhledávačům jako nejvíce přirozená a lze díky tomu dosáhnout dobrých pozic ve výsledcích vyhledávání.

⁴⁵ Webové stránky, které lze nalézt ve výsledcích vyhledávání dotazu „*copyright 2003*“ + „*klíčové-slovo*“. Za rok 2003 je možné dosadit jakýkoliv jiný dostatečně vzdálený rok, který vypovídá o neaktualizovanosti webu.

⁴⁶ Provizní systém založený na propagaci produktů či služeb prostřednictvím webových stránek affiliate partnerů.

4.6. *Shrnutí kapitoly*

Cílem této kapitoly bylo představení pojmu SEO a implementace principů SEO do navrženého procesního modelu. Pojem SEO (optimalizace pro vyhledávače) lze chápat jako opakující se proces, jehož cílem je zvýšení množství a kvality návštěvníků webu pomocí zlepšení umístění ve výsledcích vyhledávání relevantních klíčových slov. SEO z hlediska marketingového představuje jednu z forem internetového marketingu, konkrétně jednu z forem marketingu ve vyhledávačích.

Samotný proces optimalizace pro vyhledávače sestává z pěti kroků, z nichž tři již byly popsány v rámci návrhu procesního modelu. Zbývající dva kroky, optimalizace on-page a off-page faktorů, které lze z hlediska navrženého procesního modelu zařadit do fáze *Implementace*, byly popsány formou doporučení sdružených do větších tematických celků v rámci jednotlivých podkapitol této kapitoly.

5. Specifikace zadání aplikace

Hlavním cílem této kapitoly je specifikace uživatelských požadavků na aplikaci sloužící k podpoře vývoje webu s důrazem na SEO. Vedlejším cílem je určení míst v rámci procesního modelu, které by bylo vhodné touto aplikací podpořit. Jednotlivé požadavky jsou zde z důvodu rozsahu specifikovány na obecnější úrovni. Specifikované zadání bude sloužit jako podklad pro následný návrh a implementaci aplikace.

5.1. Identifikace činností vhodných pro aplikační podporu

Dříve než budou specifikovány požadavky na funkcionalitu a uživatelské rozhraní, je nutné identifikovat taková místa v rámci procesního modelu, která je vhodné podpořit novou aplikací z hlediska automatizace a usnadnění práce při procesu optimalizace pro vyhledávače. Nejdůležitějším podkladem pro identifikaci těchto míst je tabulka 4.3, která poskytuje informace o mapování jednotlivých kroků optimalizačního procesu na fáze a činnosti navrženého procesního modelu. Na základě této tabulky byla vytvořena tabulka 5.1, která nabízí informace o tom, zda je vhodné, aby jednotlivé fáze a činnosti procesního modelu spojené s optimalizačním procesem byly novou aplikací podporovány či nikoliv.

Fáze navrženého procesního modelu	Činnost v rámci fáze navrženého procesního modelu	Aplikační podpora
Úvodní analýza projektu	Identifikace cílových uživatelů	Ne
Analýza konkurenčního prostředí	Identifikace konkurenčních segmentů	Ne
	Identifikace relevantních klíčových slov	Ano
	Identifikace konkurentů	Ano
	Analýza konkurentů	Ano
	SWOT analýza	Ne
Implementace	Výběr webhostingu a domény	Ano
	Získání dat a naplnění databáze	Ne
	Optimalizace on-page faktorů (přidáno v rámci optimalizačního procesu)	Ano
	Optimalizace off-page faktorů (přidáno v rámci optimalizačního procesu)	Ano
Provoz	Výběr monitorovacího systému	Ne
	Implementace monitorovacího systému	Ne
	Nastavení monitorovacího systému	Ne
	Analýza návštěvnosti	Ne

	Analýza umístění ve vyhledávačích	Ano
	Analýza využívání webové aplikace	Ne
	Analýza byznys přínosů	Ne
	Zhodnocení analytických přínosů	Ne

Tabulka 5.1: Identifikace činností procesního modelu vhodných pro aplikační podporu

Z tabulky vyplývá, že podpora optimalizačního procesu novou aplikací je vhodná pro přibližně třetinu relevantních činností. Zbylou část tvoří činnosti, pro které platí jedno z následujících tvrzení:

- *činnost je vhodnější aplikačně podpořit již existujícím softwarovým řešením* – využití monitorovacích systémů typu Google Analytics pro monitoring a analýzu návštěvnosti,
- *automatizace činnosti by pouze vytvořila prostředí pro záznam a úpravu tabulkových dat* – činnosti Identifikace cílových uživatelů, Identifikace konkurenčních segmentů a SWOT analýza,
- *činnost se překrývá s jinou činností* - činnost Získání dat a naplnění databáze, jejíž obsah je z větší části zakomponován v rámci činnosti Optimalizace on-page faktorů.

5.2. Definice uživatelských požadavků

5.2.1. Identifikace relevantních klíčových slov

Identifikace relevantních klíčových slov spočívá v transformaci jednotlivých konkurenčních segmentů na seznam klíčových slov, které tento segment charakterizují. Aplikace by měla podporovat navržený proces tvorby a úpravy seznamu klíčových slov. Ve spojitosti s tvorbou klíčových slov by aplikace měla umožňovat získávat z dostupných zdrojů informace o příbuzných klíčových slovech.

5.2.2. Identifikace konkurentů

Mezi výstupy identifikace konkurentů patří:

- seznam relevantních vyhledávačů,
- přehled konkurenčních webových stránek pro jednotlivá klíčová slova,
- vyhodnocení konkurenčnosti jednotlivých klíčových slov.

Aplikace by měla umožnit automatizaci vzniku těchto výstupů. Prvním krokem je identifikace relevantních vyhledávačů na základě definovaného geografického cílení. Při vybraném geografickém cílení na Českou republiku by měly být použity vyhledávače Seznam a Google.cz, při jiném geografickém cílení by měly být použity lokální verze vyhledávačů Google, Yahoo a Live. Druhým krokem je automatické vyhledání a zobrazení konkurenčních webových stránek podle jednotlivých konkurenčních segmentů a odpovídajících klíčových slov. Ke každému klíčovému slovu by měla aplikace zjistit celkový počet konkurenčních webových stránek.

5.2.3. Analýza konkurentů

Analýza konkurentů vychází ze seznamu identifikovaných konkurentů vytvořeného v předchozím kroku. Aplikace by měla umožňovat zaznamenávat a upravovat informace o jednotlivých konkurenčních webových stránkách. V tabulkové formě by měly být dostupné informace o:

- URL adrese,
- zaměření webu,
- titulku webové stránky,
- obsahu metatagů webové stránky

Pro každou konkurenční webovou stránku by aplikace měla být schopná zjistit počet příchozích odkazů.

5.2.4. Výběr webhostingu a domény

Hlavní výstupem této činnosti je vybraný poskytovatel webhostingu a webhostingový program a především vybrané domény sloužící k provozování webové aplikace. Aplikace by měla umožňovat:

- zjištění dostupnosti doménového jména,
- získání informací z Whois serverů,

5.2.5. Optimalizace on-page faktorů

V rámci on-page optimalizace by měla aplikace podporovat dvě skupiny funkcí. První skupina funkcí se týká tvorby webu, který je přívětivý k vyhledávacím robotům. V rámci této

oblasti by aplikace měla umožňovat vytváření XML sitemap a souborů robots.txt a také testování platnosti nastavených pravidel uvedených v souboru robots.txt.

Druhou skupinou funkcí jsou funkce týkající se obsahu HTML stránky. Zaměříme-li se na hlavičku webové stránky, aplikace by měla usnadňovat tvorbu titulků a metatagů. Z hlediska těla stránky by aplikace měla být schopná určit celkový počet výskytů zadaných klíčových slov, počet výskytů klíčových slov v rámci významných typů elementů a také by měla zvládat výpočet hustoty klíčových slov.

5.2.6. Optimalizace off-page faktorů

Stejně jako u on-page optimalizace i u off-page optimalizace by měla aplikace podporovat dvě základní skupiny funkcí. První skupina funkcí by měla být zaměřena na podporu navrženého procesu navázání a udržování partnerství pro výměnu recipročních odkazů. Aplikace by měla usnadňovat především:

- vyhledání potenciálních partnerů pro výměnu odkazů,
- navázání kontaktu s vybranými partnery,
- kontrola umístění a kvality smluvených odkazů.

Druhá skupina funkcí by se měla týkat podpory některých pokročilých technik link buildingu:

- kontrola duplicitního obsahu a neoprávněného kopírování vytvořeného obsahu,
- hledání opuštěných webů.

5.2.7. Analýza umístění ve vyhledávačích

Poslední oblastí, nikoliv však podle významu, kterou by měla vytvořená aplikace podporovat, je analýza umístění webových stránek ve vyhledávačích. Na základě identifikovaných relevantních vyhledávačů a seznamů klíčových slov by aplikace měla umožňovat provádět pravidelnou kontrolu umístění webových stránek ve výsledcích vyhledávání.

5.3. Shrnutí kapitoly

Cílem této kapitoly byla specifikace uživatelských požadavků na aplikaci, která by měla sloužit k podpoře vývoje webu s důrazem na SEO. Nejprve byly identifikovány jednotlivé činnosti z procesního modelu, které by bylo vhodné podpořit danou aplikací. Následně bylo

pro jednotlivé činnosti specifikováno, jakým způsobem by je aplikace měla podporovat, neboli byla definována rámcová funkcionalita aplikace

6. Presentace implementované aplikace

Cílem této kapitoly je prezentace aplikace implementované na základě požadavků specifikovaných v předchozí kapitole. V rámci prezentace implementované aplikace budou představeny především základní datové modely užitě při tvorbě aplikace, využitě nástroje a také bude představeno samotné prostředí aplikace. Aplikaci společně se zdrojovými kódy a navrženými modely lze stáhnout na adrese <http://zdenek-hejl.com/seo-diplomka.zip>.

6.1. Charakteristiky implementace

6.1.1. Implementační prostředí

Jedním z prvních kroků při vývoji aplikace pro podporu vývoje webu s důrazem na SEO bylo rozhodnutí o implementačním prostředí aplikace. V rámci tohoto kroku bylo nutné určit:

- zda se bude jednat o webovou nebo desktopovou aplikaci,
- programovací jazyk,
- databázovou platformu.

Výsledkem tohoto rozhodnutí byl vývoj desktopové aplikace v jazyce C# s využitím databázového prostředí MS Access 2007. Desktopová aplikace byla vybrána proto, že tento typ aplikace nabízí více možností při tvorbě uživatelského rozhraní. Nejpřístupnější databázovou platformou pro využití společně s desktopovou aplikací je právě MS Access, jelikož nevyžaduje žádnou dodatečnou instalaci pro svůj běh. Jazyk C# byl vybrán vzhledem k mým předchozím zkušenostem s programováním v tomto jazyce.

6.1.2. Použité nástroje

Při návrhu a implementaci aplikace byly použity tři nástroje:

- *Sybase Powerdesigner 12.5* – www.sybase.com/powerdesigner,
- *Microsoft Access 2007* - office.microsoft.com/access,
- *Microsoft Visual C# 2008 Express Edition* – www.microsoft.com/express/vcsharp.

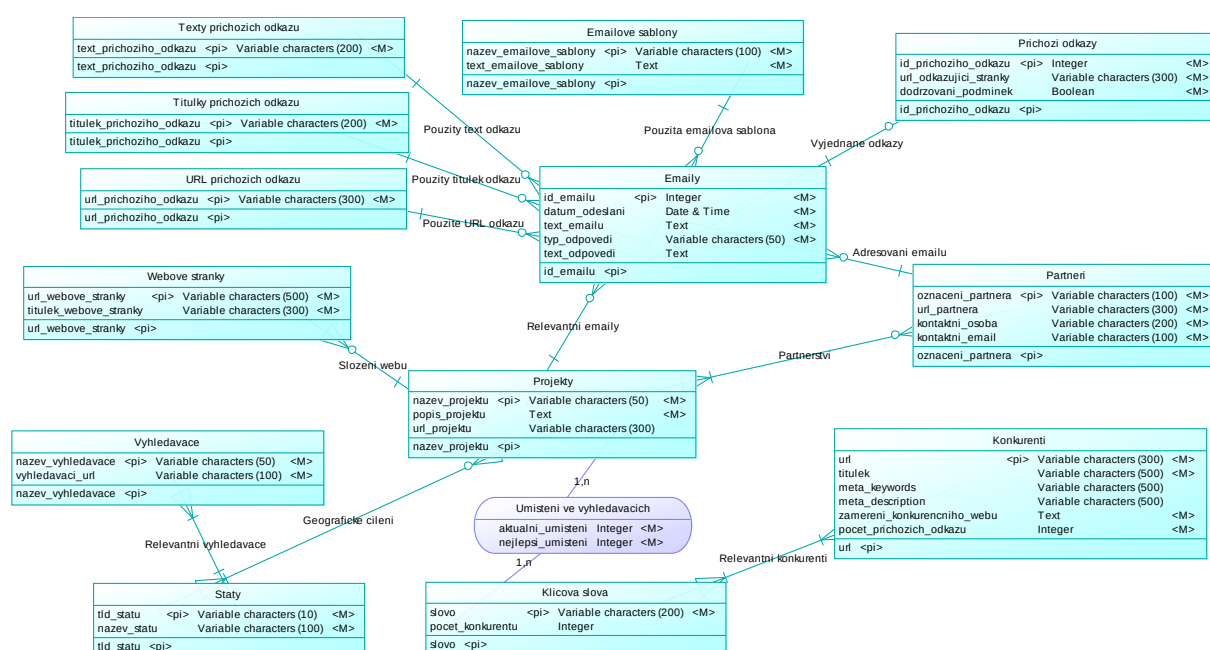
Powerdesigner byl využit při návrhu logického a fyzického datového modelu. Access 2007 byl využit pro implementaci navržené struktury databáze a základní naplnění databáze daty. Visual C# sloužil jako vývojové prostředí pro samotné programování aplikace.

6.2. Základní modely

V rámci návrhu aplikace byly vytvořeny některé vývojové diagramy a modely, které posléze sloužily jako podklad pro samotnou implementaci aplikace. V této kapitole budou zobrazeny tři datové modely:

- *logický datový model* – vytvořeno v Sybase Powerdesigner 12.5,
- *fyzický datový model* – vytvořeno v Sybase Powerdesigner 12.5,
- *relační datový model* – vytvořeno v MS Access 2007,

Logický datový model vyjadřuje takový model dat, který je nezávislý na konkrétní implementační platformě. Logický datový model je znázorněn na obrázku 6.1.



Obrázek 6.1: Logický datový model vyjádřený v prostředí Sybase Powerdesigner 12.5

Fyzický datový model je na rozdíl od logického již přizpůsoben konkrétnímu implementačnímu prostředí, zde MS Access 2007. Fyzický datový model vycházející z logického datového modelu (viz obrázek 6.1) a vyjádřený v prostředí Sybase Powerdesigner 12.5 je znázorněn na obrázku 6.2.

6.3. Aplikační prostředí

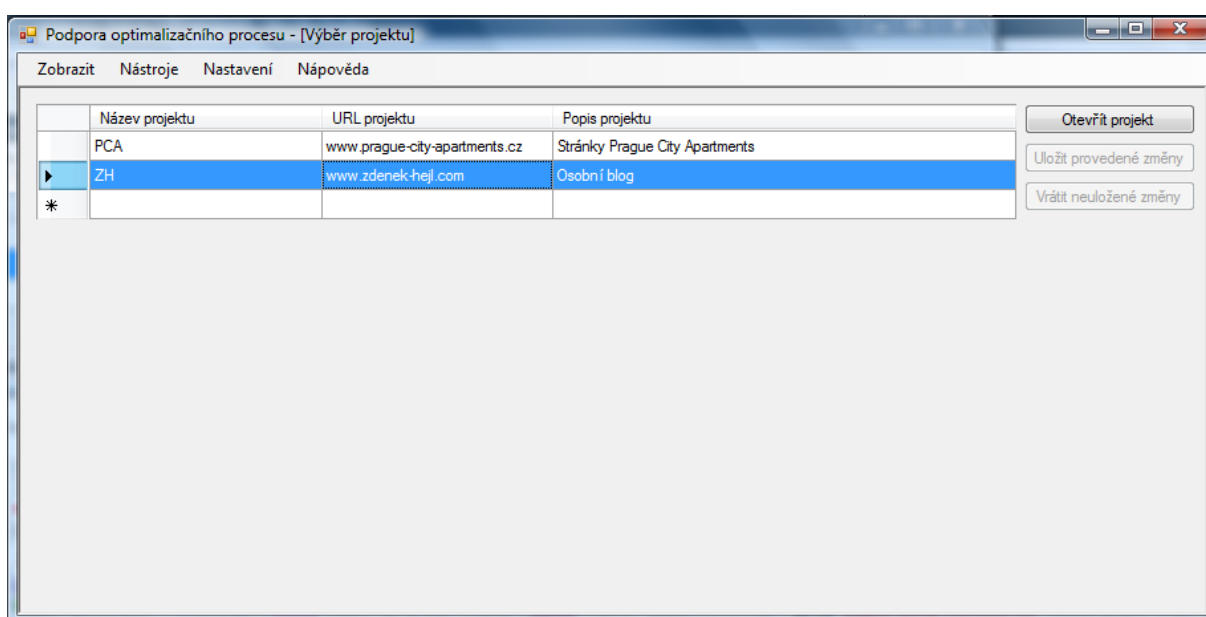
Závěrečným bodem prezentace implementované aplikace je představení samotného aplikačního prostředí. Okno aplikace se skládá ze dvou částí:

- menu aplikace,
- pracovní plocha aplikace.

Menu aplikace sestává ze čtyř položek:

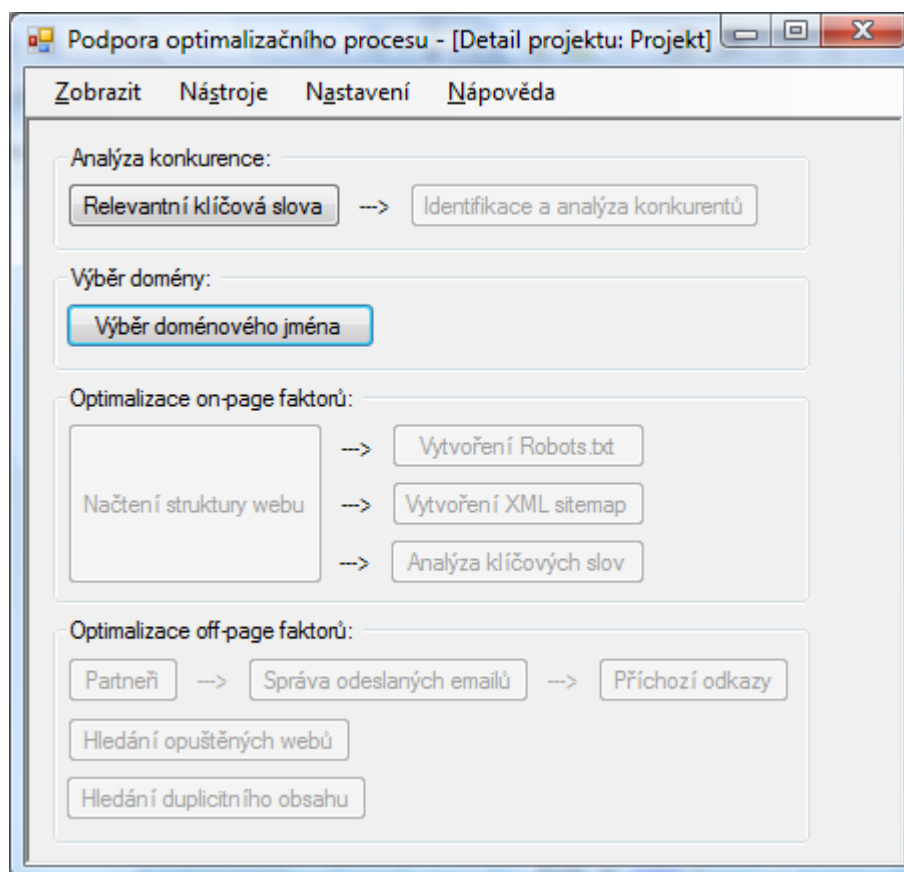
- *menu Zobrazit* – umožňuje navigaci v rámci aplikace, konkrétně zobrazení obrazovky s možností výběru projektu a obrazovky s detailem aktuálního projektu,
- *menu Nástroje* – nabízí přístup k nástrojům, které nejsou přímo vázány na konkrétní projekt,
- *menu Nastavení* – poskytuje přístup k nastavením platným pro všechny projekty,
- *menu nápověda* – umožňuje přístup k nápovědě aplikace a k informacím o aplikaci.

V rámci pracovní plochy jsou poté dostupné obrazovky s ovládacími prvky sloužící pro podporu optimalizačního procesu. Aplikace obsahuje dvě základní obrazovky – *Výběr projektu* a *Detail projektu*. *Výběr projektu* je úvodní obrazovka aplikace (viz obrázek 6.4), která umožňuje správu projektů (vytváření, mazání, editace) a je z ní možné jednotlivé projekty otevírat.



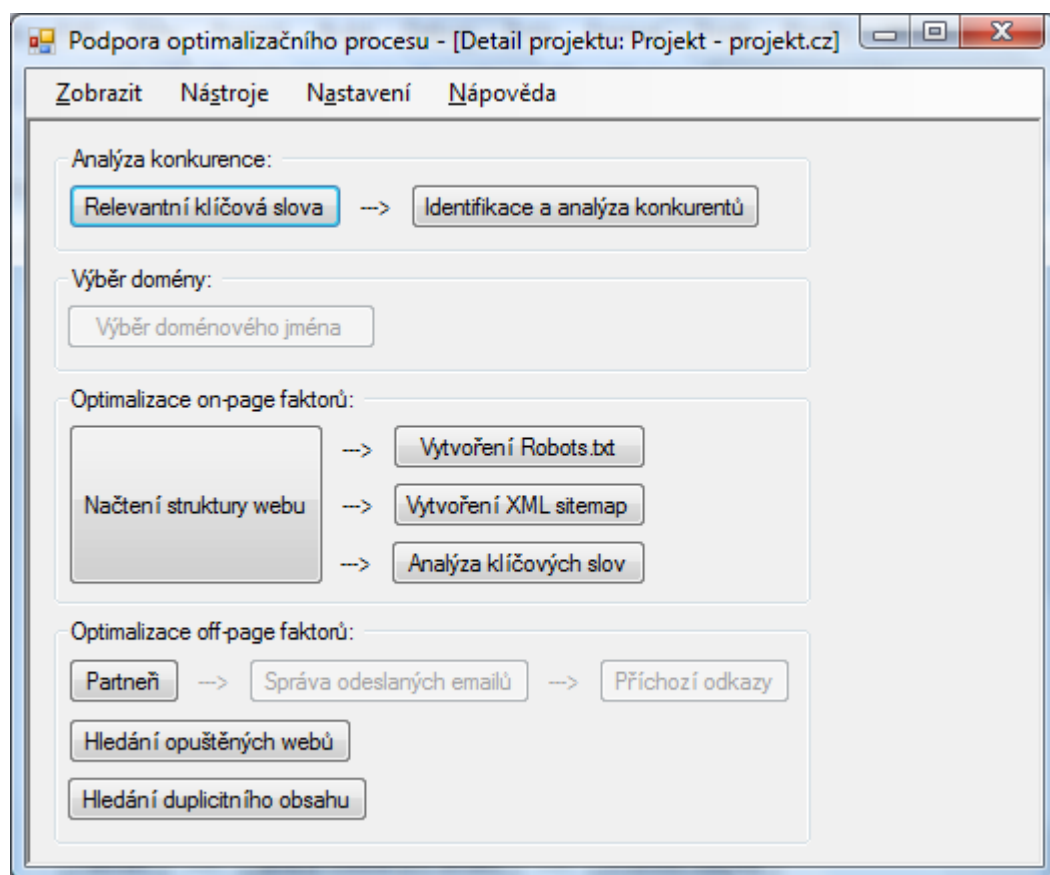
Obrázek 6.4: Obrazovka Výběr projektu

Po otevření projektu se objeví druhá základní obrazovka *Detail projektu* (viz obrázek 6.5), která nabízí funkcionalitu potřebnou pro podporu optimalizačního procesu pro vybraný projekt.



Obrázek 6.5: Obrazovka Detail projektu – začátek projektu

Dostupnost jednotlivých tlačítek, tedy nabízených funkcí, se liší podle aktuálního stavu projektu. Tímto způsobem je zachycena procesní podstata aplikace, která vychází z navrženého procesního modelu a jeho propojení s optimalizačním procesem. Na obrázku 6.5 je zachycen projekt v počáteční fázi, kdy ještě nebyla provedena analýza konkurence, a nebylo vybráno doménové jméno. Naopak na obrázku 6.6 je zachycen projekt s identifikovanými klíčovými slovy, vybraným doménovým jménem, načtenou strukturou webu, avšak doposud bez identifikovaných potenciálních partnerů pro výměnu odkazů.



Obrázek 6.6: Obrazovka Detail projektu – průběh projektu

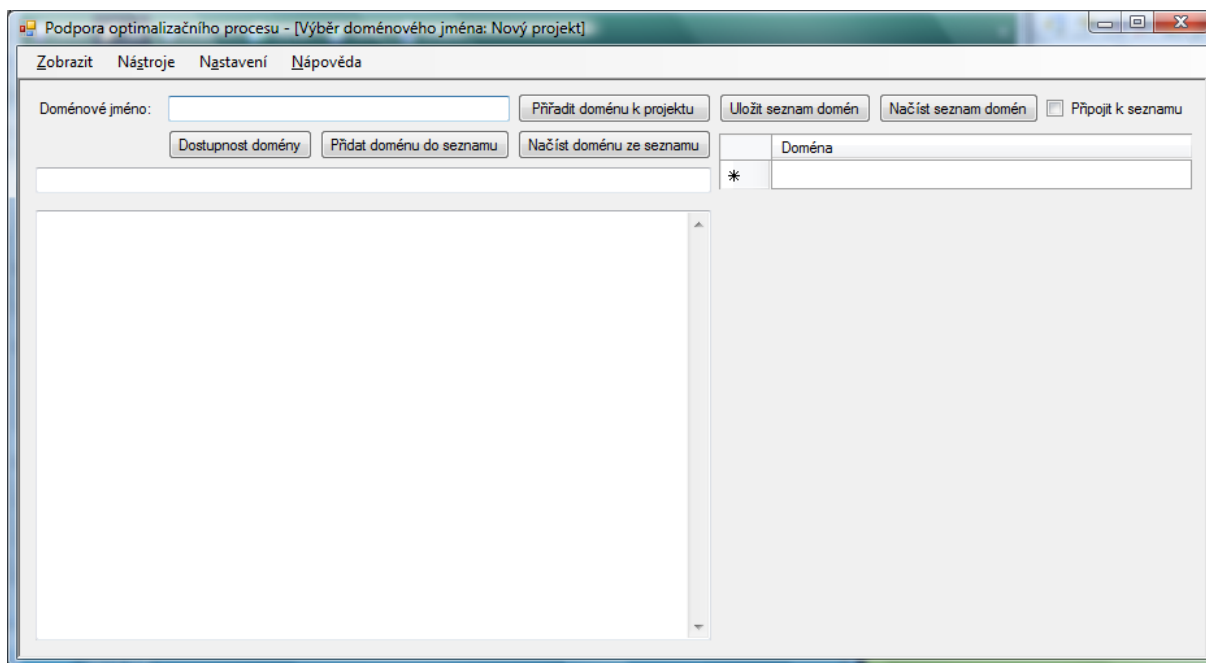
Pravidla pro nastavení dostupnosti jednotlivých tlačítek vychází z navrženého procesního modelu a optimalizačního procesu a jsou vyjádřeny v tabulce 6.1.

Název tlačítka	Podmínky dostupnosti tlačítka
Relevantní klíčová slova	Dostupné vždy
Identifikace a analýza konkurentů	Musí být vybráno alespoň jedno klíčové slovo
Výběr doménového jména	Není-li doposud vybrané doménové jméno
Načtení struktury webu	Musí být vybráno doménové jméno
Vytvoření Robots.txt	Musí být načtena struktura webu
Vytvoření XML sitemap	Musí být načtena struktura webu
Analýza klíčových slov	Musí být načtena struktura webu
Partneři	Musí být vybráno doménové jméno
Správa odeslaných emailů	Musí být přiřazen alespoň jeden partner
Příchozí odkazy	Musí být odeslán alespoň jeden email
Hledání opuštěných webů	Musí být vybráno doménové jméno
Hledání duplicitního obsahu	Musí být vybráno doménové jméno

Tabulka 6.1: Podmínky dostupnosti jednotlivých tlačítek

Poslední zde představenou obrazovkou je *Výběr doménového jména*. Tato obrazovka (viz obrázek 6.7) představuje důležitý krok v rámci celého optimalizačního procesu, jelikož na

výběr doménového jména je navázaná další funkcionalita aplikace. V rámci této obrazovky je možné testovat dostupnost doménových jmen, spravovat seznam možných doménových jmen, ukládat a načítat tento seznam a také přiřazovat vybrané doménové jméno k aktuálnímu projektu.



Obrázek 6.7: Obrazovka Výběr doménového jména

6.4. Shrnutí kapitoly

V rámci této kapitoly bylo nejdříve představeno implementační prostředí aplikace, jímž je desktopová aplikace naprogramovaná v jazyce C# s využitím databáze MS Access 2007. V souladu s cíli kapitoly byly poté představeny základní datové modely, které vznikly během návrhu aplikace. Jedná se o logický a fyzický datový model vyjádřený v prostředí Sybase Powerdesigner 12.5 a relační datový model vyjádřený v prostředí MS Access 2007. Poslední částí prezentace implementované aplikace bylo představení samotného prostředí aplikace.

7. Závěr

Internet je v současnosti nejrychleji se rozrůstajícím médiem, který představuje zároveň konkurenci i komplement ke klasickým masmédiím – tisk, rozhlas, televize. Konkurenční funkci plní díky tomu, že dokáže emulovat všechna tři uvedená masmédia – tisk pomocí internetových informačních portálů, rozhlas pomocí streamovaného internetového rozhlasového vysílání a televizi díky možnosti streamovaného přenosu videa. Zároveň však představuje doplněk k těmto klasickým médiím, jelikož nabízí možnosti, které tato média nenabízí, např. možnost okamžité odezvy, interaktivitu a nelineárnost příjmu sdělení či propojení více typů médií dohromady.

Vzhledem k významu internetu velké množství podniků, organizací i jednotlivců zaměřilo část svého snažení právě tímto směrem. Vznikají webové prezentace podniků, informační portály anebo se vytváří webové aplikace, jejichž zaměřením je prodej a nákup produktů a služeb. Stejně jako v reálném světě, i ve světě internetu panuje silná konkurence, jelikož prostředí internetu je složeno z miliard webových stránek. Je tedy nutné vlastním webovým stránkám zajistit propagaci, aby se o nich potenciální návštěvníci vůbec mohli dozvědět. Kromě klasických forem propagace (např. v denním tisku, v časopisech, na letácích, na vizitkách) je vhodné v prostředí internetu využít specifické formy propagace – optimalizace pro vyhledávače (SEO).

Tato práce byla zaměřena právě na problematiku optimalizace pro vyhledávače. Nejednalo se ovšem pouze o vysvětlení pojmu SEO a představení používaných optimalizačních technik a postupů, ale především o zasazení optimalizačního procesu do komplexnějšího procesu zaměřeného na vývoj a provoz webové aplikace. Proces vývoje webové aplikace nebyl převzat z některé konkrétní metodiky pro vývoj webu, ale byl vytvořen speciálně pro účely této práce a vychází z analýzy několika webových metodik a z identifikovaných nedostatků těchto metodik. Pro podporu optimalizačního procesu byla navíc navržena a implementována desktopová aplikace.

Mezi hlavní přínosy této práce lze tudíž zařadit:

- navržený procesní model pro vývoj webových aplikací,
- namapování kroků optimalizačního procesu na činnosti navrženého procesního modelu,
- implementace aplikace pro podporu optimalizačního procesu.

Na výstupy této práce je možné navázat detailnějším rozpracováním jednotlivých navržených fází a činností procesního modelu, podrobnějším rozbořem technik a postupů používaných v rámci optimalizačního procesu anebo rozšířením funkcionality navržené a implementované aplikace.

Použité zdroje

BERANOVÁ, Michaela. *Alternativní přístupy k plánování a řízení projektu zavedení IS/ICT systému*. [s.l.], 2008. 75 s. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra ekonometrie. Vedoucí diplomové práce prof. RNDr. Ing. Petr Fiala, MBA, CSc.

BOBNAR, Esoos, MAHANEY, Stephen. *The 7 Essential Title Tag Strategies for Top Ranking WebPages in 2009* [online]. 2009 [cit. 2009-04-02]. Dostupný z WWW: <<http://www.searchengineneeds.com/>> po registraci.

BRAMBILLA, Marco, et al. *Web Engineering : Modelling and Implementing Web Applications*. [s.l.] : Springer, 2007. Human-Computer Interaction Series. Dostupný z WWW: <<http://www.webml.org/webml/upload/ent5/1/Chapter%20%20-%20WebML.pdf>>. ISBN 978-184628922. Designing Web Applications with WebML and WebRatio, s. 221-260.

CERI, Stefano, FRATERNALI, Piero, BONGIO, Aldo. *Web Modeling Language (WebML): a modeling language for designing Web sites* [online]. [2000] [cit. 2009-02-17]. Dostupný z WWW: <<http://www.webml.org/webml/upload/ent5/1/www9.pdf>>. <http://www9.org/w9cdrom/177/177.html>.

CERI, Stefano, FRATERNALI, Piero, MATERA, Maristella. *Conceptual modeling of data-intensive Web applications* [online]. 2002 [cit. 2009-02-17]. Dostupný z WWW: <<http://www.webml.org/webml/upload/ent5/1/IC.pdf>>.

CERMAN, Michal. *Komparace nástrojů pro analýzu webu*. [s.l.], 2008. 90 s. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra informačních technologií. Vedoucí diplomové práce Ing. Libor Gála.

CIPD. *SWOT analysis* [online]. 2008 [cit. 2009-03-05]. Dostupný z WWW: <<http://www.cipd.co.uk/subjects/corpstrtgty/general/swot-analysis.htm>>.

CMS Watch. *Compare Web Analytics Vendors at CMS Watch* [online]. [2009] [cit. 2009-03-25]. Dostupný z WWW: <<http://www.cmswatch.com/Analytics/Vendors/>>.

DatabaseAnswers. *Data Modelling Tools* [online]. c2002 [cit. 2009-03-22]. Dostupný z WWW: <http://www.databaseanswers.org/modelling_tools.htm>.

DE TROYER, Olga, LEUNE, Cees. *WSDM: A User-Centered Design Method for Web Sites* [online]. 1998 [cit. 2009-02-12]. Dostupný z WWW: <<http://wsdm.vub.ac.be/Download/Papers/WISDOM/WWW7.PDF>>.

DE TROYER, Olga, CASTELEYN, Sven. *The Conference Review System with WSDM* [online]. 2001a [cit. 2009-02-13]. Dostupný z WWW: <<http://wsdm.vub.ac.be/Download/Papers/WISDOM/WSDMPaper.PDF>>.

DE TROYER, Olga, CASTELEYN, Sven. *Appendix to Conference Review System with WSDM* [online]. 2001b [cit. 2009-02-13]. Dostupný z WWW: <<http://wsdm.vub.ac.be/Download/Papers/WISDOM/Appendix.PDF>>.

DE TROYER, Olga, CASTELEYN, Sven. *Modeling Complex Processes for Web Applications using WSDM* [online]. 2003 [cit. 2009-02-13]. Dostupný z WWW: <<http://wsdm.vub.ac.be/Download/Papers/WISDOM/IWWOST2003.PDF>>.

DedaSys LLC. *Programming Language Popularity* [online]. c2007-2008 , Last data update: Mar 05 2009 [cit. 2009-03-19]. Dostupný z WWW: <<http://langpop.com/>>.

DEGEYTER, Stoney. *January 2009 Search Engine Market Share* [online]. 2009-02-02 [cit. 2009-03-04]. Dostupný z WWW: <<http://www.searchnewz.com/latestsearch/senews/sn-4-20090202January2009SearchEngineMarketShare.html>>.

DOLOG, Peter, BIELIKOVÁ, Mária. *Hypermedia systems modelling framework*. Computing and informatics [online]. 2002, vol. 21, no. 3 [cit. 2009-02-08], s. 221-239. Dostupný z WWW: <www.cs.aau.dk/~dolog/pub/cai.ps>. ISSN 1335-9150.

GUELL, Natacha, SCHWABE, Daniel, VILIAN, Patricia. *Modeling Interactions and Navigation in Web Applications* [online]. 2000 [cit. 2009-02-11]. Dostupný z WWW: <<http://www.inf.puc-rio.br/~schwabe/papers/WWWCM00Expanded.pdf>>.

HEJL, Zdeněk. *Analýza metod testování použitelnosti webových aplikací*. [s.l.], 2007. 43 s. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra informačních technologií. Vedoucí bakalářské práce Ing. Libor Gála.

Internet Info. *NAVRCHOLU.cz: Google v Česku roste na úkor ostatních vyhledávačů* [online]. 26. 11. 2008 [cit. 2009-03-04]. Dostupný z WWW: <<http://www.iinfo.cz/tiskove-centrum/tiskove-zpravy/navrcholu-google-v-cesku-roste/>>.

Internet Info. *Webhosteři domén .cz* [online]. c1998-2009 , Aktualizace: 21.3.2009 [cit. 2009-03-21]. Dostupný z WWW: <<http://domeny.lupa.cz/webhosteri/>>.

JANOVSKÝ, Dušan. *Webhosting : a co od něj požadovat* [online]. 2009 , Poslední aktualizace 15. února 2009 [cit. 2009-03-21]. Dostupný z WWW: <<http://www.jakpsatweb.cz/webhosting.html>>.

KACEROVSKÝ, Jiří. *Procesní modelování* [online]. 2008 [cit. 2009-02-05]. Dostupný z WWW: <nb.vse.cz/~KACEROVJ/4it215-03a_procesni-modelovani.pdf>.

- KOLB, Jason. *Web Programming Language Popularity Contest* [online]. February 19, 2007 [cit. 2009-03-19]. Dostupný z WWW: <http://www.jasonkolb.com/weblog/2007/02/programming_tre.html>.
- KOPECKÝ, Petr. *Životní cyklus produktu (1.)* [online]. 22.10.2001 [cit. 2009-02-07]. Dostupný z WWW: <http://www.marketingovenoviny.cz/index.php3?Action=View&ARTICLE_ID=80>.
- KRUTIŠ, Michal. *Internetový marketing: Optimalizace pro vyhledávače (SEO)* [online]. 27. 6. 2005 [cit. 2009-03-28]. Dostupný z WWW: <<http://www.lupa.cz/clanky/internetovy-marketing-optimalizace-pro-vyhledavace-seo/>>.
- LBMS. *Řízení rizik na projektu : Popis procesů zaměřených na identifikaci, analýzu a zmírnění rizik projektu* [online]. 2006 [cit. 2009-02-24]. Dostupný z WWW: <<http://www.lbms.cz/prezentace/RPJ-ukazka.pdf>>.
- LEE, Heeseok, et al. *Developing intranet hypermedia system by using scenario-based object-oriented methodology* [online]. 1996 [cit. 2009-02-10]. Dostupný z WWW: <<http://koasas.kaist.ac.kr/bitstream/10203/5398/1/1996-148.pdf>>.
- LEE, Heeseok, SUH, Woojong. *A workflow-based hypermedia development methodology* [online]. 1998 [cit. 2009-02-12]. Dostupný z WWW: <<http://koasas.kaist.ac.kr/bitstream/10203/5127/1/1998-200.pdf>>.
- LEE, Heeseok, et al. *A view-based hypermedia design methodology* [online]. 1999 [cit. 2009-02-12]. Dostupný z WWW: <<http://koasas.kaist.ac.kr/bitstream/10203/5045/1/1999-044.pdf>>.
- LEE, Choongseok, LEE, Heeseok. *Using scenario for building hypermedia systems* [online]. 2000 [cit. 2009-02-10]. Dostupný z WWW: <<http://koasas.kaist.ac.kr/bitstream/10203/4766/1/2000-171.pdf>>.
- LEWALLEN, Raymond. *Software Development Life Cycle Models* [online]. 2005-06-13 [cit. 2006-12-24]. Dostupný z WWW: <<http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx>>.
- LIMA, Fernanda, SCHWABE, Daniel. *Application Modeling for the Semantic Web* [online]. 2003 [cit. 2009-02-11]. Dostupný z WWW: <<http://www.inf.puc-rio.br/~schwabe/papers/LAWEB2003.pdf>>.
- MAREŠ, Miroslav. *Teorie a metody politologie* [online]. 11. 11. 2007 [cit. 2009-02-05]. Dostupný z WWW: <<http://figlar.ic.cz/fss/new08.html>>.
- MARYŠKA, Miloš. *Metriky v informatice*. [s.l.], 2006. 122 s. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra informačních technologií. Vedoucí diplomové práce Ing. Ota Novotný, Ph.D.
- Media Generation Nine. *Terminology* [online]. c2005-2008 [cit. 2009-03-28]. Dostupný z WWW: <<http://www.mg9.com/terminology.php>>.

- MOLHANEC, Martin. *The Object-Oriented Hypermedia Design Model (OOHDM)* [online]. 2001 [cit. 2009-02-11]. Dostupný z WWW: <martin.feld.cvut.cz/~mmm/VaV/files/publik/2001/OOHDM.pdf>.
- Omega Digital Media. *Web Traffic Data Sources & Vendor Comparison* [online]. 2008 , Updated May 2008 [cit. 2009-03-25]. Dostupný z WWW: <<http://www.advanced-web-metrics.com/docs/web-data-sources.pdf>>.
- OMG. *Unified Modeling Language : Superstructure* [online]. 2007 [cit. 2009-03-06]. Dostupný z WWW: <<http://www.omg.org/docs/formal/07-02-03.pdf>>.
- Planet Ocean. *The Unfair Advantage Book on Winning the Search Engine Wars*. [s.l.] : [s.n.], 2009. 199 s.
- PLESSERS, Peter, CASTELEYN, Sven, DE TROYER, Olga. *Semantic Web Development with WSDM* [online]. 2005 [cit. 2009-02-13]. Dostupný z WWW: <[ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-185/semAnnot05-01.pdf](ftp://informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-185/semAnnot05-01.pdf)>.
- Politecnico di Milano. *WebML Hypertext Design* [online]. [2008] [cit. 2009-03-15]. Dostupný z WWW: <<http://home.dei.polimi.it/mbrambil/webml/lesson3/>>.
- RUBIN, David M.. *Introduction to CRC Cards* [online]. 1998 [cit. 2009-03-12]. Dostupný z WWW: <<http://www.softstar-inc.com/Download/Intro%20to%20CRC.pdf>>.
- RUSSELL, Gordon. *Chapter 2 - Database Analysis* [online]. [2004] [cit. 2009-03-21]. Dostupný z WWW: <<http://db.grussell.org/ch2.html>>.
- Search Tools Consulting. *About Robots.txt and Search Indexing Robots* [online]. 2008 , Page Updated 2008-09-19 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.searchtools.com/robots/robots-txt.html>>.
- SEOMoz. *Search Engine Ranking Factors V2* [online]. April 2, 2007 [cit. 2009-04-02]. Dostupný z WWW: <<http://www.seomoz.org/article/search-ranking-factors>>.
- SCHOEMAKER, Jeremy. *What Is The Definition Of SEO?* [online]. 14. 5. 2008 [cit. 2009-03-28]. Dostupný z WWW: <<http://www.shoemoney.com/2008/05/14/what-is-the-definition-of-seo/>>.
- SCHWABE, Daniel, ROSSI, Gustavo, BARBOSA, Simone D. J.. *Systematic Hypermedia Application Design with OOHDM* [online]. 1995 [cit. 2009-02-11]. Dostupný z WWW: <<http://www-di.inf.puc-rio.br/schwabe/HT96WWW/section1.html>>.
- SCHWABE, Daniel, ROSSI, Gustavo. *An Object Oriented Approach to Web-Based Application Design* [online]. 1998 [cit. 2009-02-11]. Dostupný z WWW: <<http://www.inf.puc-rio.br/~schwabe/papers/TAPOSRevised.pdf>>.

SCHWABE, Daniel, et al. *Design and Implementation of Semantic Web Applications* [online]. 2004 [cit. 2009-02-11]. Dostupný z WWW: <<http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-105/WE-SW%20Wkshop%20Schwabe.pdf>>.

SULLIVAN, Danny. *Meta Keywords Tag 101 : How To “Legally” Hide Words On Your Pages For Search Engines* [online]. Sep 5, 2007 [cit. 2009-04-02]. Dostupný z WWW: <<http://searchengineland.com/meta-keywords-tag-101-how-to-legally-hide-words-on-your-pages-for-search-engines-12099>>.

ŠPINAR, David, et al. *Přístupnost webových stránek orgánů státní správy* [online]. 2007 [cit. 2009-03-22]. Dostupný z WWW: <<http://www.pravidla-pristupnosti.cz/att/publikace.pdf>>.

ŠPINAR, David. *Testování přístupnosti* [online]. [2008] [cit. 2009-03-22]. Dostupný z WWW: <<http://pristupnost.nawebu.cz/texty/testovani.php>>.

TzuVelli. *Why Would BANS Sites On INFO Domains Be Deindexed?* [online]. c2008 [cit. 2009-03-31]. Dostupný z WWW: <<http://tzuveli.com/why-would-bans-sites-on-info-domains-be-deindexed>>.

VAN DEN BROEK, Lodewijk. *How to sabotage goals with 47 simple words* [online]. October 23, 2007 [cit. 2009-02-24]. Dostupný z WWW: <<http://blog.lodewijkvdb.com/2007/10/how-to-sabotage-goals-with-47-simple-words.html>>.

Vebloud. *Teorie relačních databází : Integritní omezení* [online]. 10.9. 2006 [cit. 2009-03-22]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=15>>.

Vebloud. *Teorie relačních databází : Normalizace* [online]. 2.8. 2007 [cit. 2009-03-22]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=13>>.

VORŘÍŠEK, Jiří. *Strategické řízení informačního systému a systémová integrace*. 1. 3. dotisk vyd. Praha : Management Press, 2003. 323 s. ISBN 80-85943-40-9.

Web Hosting Choice. *Top 10 Web Hosting Sites* [online]. c2002-2009 , Last Updated 03/21/09 [cit. 2009-03-21]. Dostupný z WWW: <<http://www.webhostingchoice.com/>>.

Webanalyticsbook. *Webanalytics solution finder* [online]. c2007 [cit. 2009-03-25]. Dostupný z WWW: <<http://www.webanalyticsbook.com/webanalytics-vendor/>>.

WHITE, Stephen A.. *Introduction in BPMN* [online]. 2004 [cit. 2009-02-19]. Dostupný z WWW: <<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>>.

Wikipedia. *Search Engine Optimization* [online]. 29 February 2004, last modified on 27 March 2009 [cit. 2009-03-28]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Search_engine_optimization>.

Wikipedia. *Software testing* [online]. 16 April 2004 , last modified on 20 March 2009 [cit. 2009-03-22]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Software_testing>.

Wikipedia. *Comparison of HTML editors* [online]. 14 May 2006 , last modified on 21 March 2009 [cit. 2009-03-22]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Comparison_of_HTML_editors>.

Wikipedia. *Comparison of integrated development environments* [online]. 8 January 2007a , last modified on 21 March 2009 [cit. 2009-03-22]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments>.

Wikipedia. *SMART criteria* [online]. 9 October 2007b , last modified on 18 February 2009 [cit. 2009-02-22]. Dostupný z WWW: <http://en.wikipedia.org/wiki/SMART_criteria>.

Terminologický slovník

Termín	Zkratka	Význam (zdroj)
Proces	/	Účelně naplánovaná a realizovaná posloupnost činností, ve kterých za pomoci odpovídajících zdrojů probíhá transformace vstupů na výstupy. (<i>Kacerovský, 2008</i>)
Hypermediální aplikace	/	Aplikace, jejichž cílem je získávání, zobrazování, manipulace a přístup k informacím v různých formátech (např. texty, obrázky, zvuky či videa). (<i>Dolog, Bieliková, 2002</i>)
Optimalizace pro vyhledávače	SEO	Proces zvyšování množství a kvality návštěvníků webu přicházejících z vyhledávačů pomocí přirozených (organických, algoritmických) výsledků vyhledávání. (<i>Wikipedia, 2004</i>)
On-page faktory	/	Faktory, které jsou přímo spojené s kódem a obsahem webových stránek a s navigační strukturou celé webové aplikace.
Off-page faktory	/	Faktory, které souvisí s internetovým okolím webové aplikace.