

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SROVNÁNÍ SOUČASNÝCH OPEN SOURCE REDAKČNÍCH SYSTÉMŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ MARTINÁK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SROVNÁNÍ SOUČASNÝCH OPEN SOURCE REDAKČNÍCH SYSTÉMŮ

COMPARE OF CONTEMPORARY OPEN SOURCE CONTENT MANAGEMENT SYSTEMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

LUKÁŠ MARTINÁK

Ing. PETR JAŠA

BRNO 2009

Abstrakt

Tato práce se zabývá srovnáním open source redakčních systémů. V úvodních kapitolách se nachází seznámení s redakčním systémem jako takovým a jsou uvedeny vybrané systémy – Drupal, Joomla! a WordPress. Ve všech je implementován webový projekt obsahující různé úrovně uživatelů, blogy, diskusní fórum a e-shop. Systémy jsou porovnávány v mnoha fázích vývoje webových aplikací – od seznámení s názvoslovím, přes instalaci, správu uživatelů, vytváření struktury webu, uživatelské rozhraní, změnu vzhledu, rozšiřitelnost, zálohování, až po aktualizace nebo lokalizaci. Je také probrána problematika SEO a testování webových aplikací.

Abstract

This thesis is about comparing open source content management systems. There is a description of the content management system and there are three chosen systems to compare – Drupal, Joomla! and WordPress. The same web project containing multiple users, blogs, discussion forum and e-shop is consequently implemented in each of them. Systems are compared in many phases of development cycle of web applications – comparison includes introduction, installation, managing users, creating web structure, changing design, extendability, user interface, managing backups, actualization, localization and more. There is also a chapter discussing SEO and web application testing.

Klíčová slova

redakční systém, systém pro správu obsahu, CMS, svobodný software, open source, PHP, MySQL, Drupal, Joomla!, Wordpress, srovnání

Keywords

Content Management System, CMS, free software, open source, PHP, MySQL, Drupal, Joomla!, Wordpress, comparison

Citace

Lukáš Martinák: Srovnání současných open source redakčních systémů, bakalářská práce, Brno, FIT VUT v Brně, 2009

Srovnání současných open source redakčních systémů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Jaši.

.....
Lukáš Martinák
18. května 2009

Poděkování

Děkuji svému vedoucímu za hodnotné připomínky a přístup v průběhu tvorby celé bakalářské práce.

© Lukáš Martinák, 2009.

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

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Seznámení s redakčním systémem | 4 |
| 2.1 | Open source, svobodný software, licence | 4 |
| 2.2 | Open source CMS Awards | 5 |
| 2.3 | Drupal | 5 |
| 2.4 | Joomla! | 6 |
| 2.5 | WordPress | 7 |
| 3 | Používané technologie | 9 |
| 3.1 | PHP | 9 |
| 3.2 | MySQL | 9 |
| 3.3 | Apache | 9 |
| 3.4 | (X)HTML, CSS | 9 |
| 3.5 | Javascript, AJAX, jQuery | 10 |
| 4 | Systémové požadavky | 11 |
| 4.1 | Souhrn požadavků | 11 |
| 4.2 | Webhosting | 11 |
| 5 | Vytvoření testovacího webu | 13 |
| 5.1 | Struktura webu, kategorizace obsahu | 13 |
| 5.2 | Design layout | 14 |
| 5.3 | Uživatelské role | 15 |
| 6 | Srovnání CMS | 17 |
| 6.1 | Architektura a názvosloví | 17 |
| 6.2 | Instalace | 19 |
| 6.3 | Rozšiřitelnost | 21 |
| 6.4 | Správa uživatelů, rolí a oprávnění | 23 |
| 6.5 | Design – šablony, motivy vzhledu | 24 |
| 6.6 | Vytváření struktury webu | 25 |
| 6.7 | Zálohování a obnova dat | 26 |
| 6.8 | Aktualizace systému a rozšíření | 27 |
| 6.9 | Lokalizace, podpora češtiny | 27 |
| 6.10 | Uživatelské rozhraní | 29 |
| 6.11 | Optimalizace pro internetové vyhledávače | 29 |
| 6.12 | Testování a ladění webových aplikací | 30 |

| | |
|--|-----------|
| 6.13 Rozšíření systémů pro dosažení funkčnosti | 34 |
| 7 Závěr | 36 |
| A CD s instalačním manuálem, zálohami projektů a screenshoty. | 41 |

Kapitola 1

Úvod

V dnešní době používá internet čím dál více uživatelů, ať už se jedná o práci nebo zábavu. Lidé si mohou vést svůj osobní i pracovní blog, navzájem si předávat zkušenosti v diskusních fórech, komentovat téměř každý článek, nakupovat a prodávat zboží i služby, natáčet, sdílet a sledovat videoklipy, připojovat se k ostatním lidem do nejrůznějších sociálních sítí, kde mohou sdílet prakticky jakýkoli obsah.

Tato práce je zaměřena na srovnání redakčních systémů, které umožňují právě pohodlné publikování nejrůznějšího obsahu. I když jde o vzájemné srovnání, nelze předpokládat, že na konci budeme mít vítěze a poražené. Spíše se zaměříme na to, jak se od sebe tyto systémy liší a co mají společné. K tomu poslouží testovací webový portál, který je navržený tak, aby prověřil systémy v různých situacích – obsahuje kalendář akcí, články, blogy, diskusní fórum a internetový obchod.

V první části se seznámíme s redakčním systémem jako takovým a představíme si tři srovnávané systémy, kterými se staly Drupal, Joomla! a WordPress. Ty byly vybrány z neřeberného množství všech redakčních systémů na základě obecné uživatelské oblíbenosti, osobních zkušeností a pozitivních referencí. Při rozhodování pomohly i výsledky ocenění CMS Awards z posledních ročníků. Také se podíváme na definici a licencování svobodného softwaru.

Ve druhé části si uvedeme technologie těmito systémy používané. Nebudeme se jimi ovšem zabývat více, než je nutné. Třetí část je věnována systémovým požadavkům jednotlivých systémů a webhostingu. Ve čtvrté části je proveden návrh webového portálu, který je poté implementován ve všech systémech. K samotnému srovnání se dostaneme v páté části, která nás provází implementací krok za krokem od instalace až po zálohování a aktualizace, přičemž v každé podkapitole je dílčí závěr.

Hlavním cílem práce je seznámení čtenáře s dostupnými redakčními systémy a jejich vlastnostmi z několika pohledů (např. web designer, web developer, editor, zákazník) při jejich použití pro různé účely, přičemž dochází k jejich okamžitému porovnávání. Kromě subjektivního srovnání určitých vlastností systémů se zaměříme i na objektivní metriky (např. velikost instalačního balíčku, počet souborů výsledného webového projektu, velikost databáze, počet použitých rozšíření a bude také provedeno měření výkonnosti nástrojem JMeter).

Kapitola 2

Seznámení s redakčním systémem

Redakční systém (*CMS – Content Management System*) neboli „systém pro správu obsahu“ v poslední době nabývá stále většího významu. Umožňuje mnohem rychlejší vývoj webových stránek a hlavně přináší jejich lepší správu. Už dávno nemusí platit, že se o obsah stránek stará jeden člověk, který je navrhnul, implementoval a je odborník na tvorbu webu. Při používání CMS tvoří obsah editoři, kteří nemusí vědět vůbec nic o tom, jak se webové stránky vytváří. Intuitivně a pohodlně přidávají obsah (fotogalerie, videa, články, novinky, produkty, atp.). Ať už se jedná o osobní stránky, firemní weby nebo rozsáhlé komunitní portály, CMS se uplatní všude tam, kde se obsah často mění, přidává ho více lidí nebo je požadována jeho pohodlná správa (je možné rozlišit uživatelské role – články mohou podléhat schvalovacímu procesu, atd.). Web tak může být neustále aktualizovaný a udržovaný.

Z pohledu webového vývojáře je CMS výhodný v tom, že poskytuje flexibilní a snadno rozšiřitelný systém, který už sám o sobě poskytuje spoustu možností. Samotný vývoj je s použitím šablon rychlejší a jednodušší, než kdyby se měl každý web vyvíjet „z ničeho“. Další výhodou je, že výrobci CMS drží krok s vývojem nových technologií a nabízí čím dál více rozšíření, která jsou snadno použitelná. Významnou roli hraje také bezpečnost, přičemž se celá komunita uživatelů snaží, aby jejich aplikace byla proti nejrůznějším druhům útoků co nejvíce odolná.

2.1 Open source, svobodný software, licence

Open source je označení softwaru s otevřeným kódem – to znamená, že je dostupný každému, kdo o něj má zájem. Samotná dostupnost zdrojového kódu však není jedinou podmínkou pro to, aby mohl být software takhle označen. Těch podmínek je celkem deset a jsou spravovány organizací Open Source Initiative [1].

Aby mohl být softwarový program označen za svobodný software (*free software*), musí uživateli zaručit svobodu spouštění, upravování a redistribuování dalším lidem, ať už je to za poplatek nebo zdarma [2]. Za tyto svobody se nemusí nikomu nic platit, není nutné o ně žádat a pochopitelně je nelze nikomu upírat. Nezbytnou podmínkou pro svobodný software je samozřejmě dostupnost zdrojových kódů. Je nutné vidět rozdíl v termínech *svobodný software* a *software zdarma*, které jsou zaměňovány především kvůli rychlému rozšíření nesprávného překladu. Svobodný software musí být dostupný i pro komerční využití.

Licencí pro svobodný software je více. Nejznámější a nejpoužívanější je zřejmě *GNU General Public Licence*, zkráceně *GNU GPL* nebo jen *GPL* [3], kterou používají všechny tři vybrané systémy – Drupal, Joomla! i WordPress. V praxi to znamená, že můžeme získat

zdrojové kódy, ty můžeme upravovat a dále šířit (klidně i za peníze), ale nikdy nemůžeme změnit licenci – ta musí zůstat GPL.

2.2 Open source CMS Awards

Packt Publishing Ltd. je společnost publikující knihy a články ze světa IT profesionálů. Od roku 2004 vydala stovky knih a s neustále se rozšiřující komunitou se stala opravdu významným informačním zdrojem. Od roku 2006 uděluje ceny *CMS Awards* [4], které při vybírání redakčního systému mohou pomoci.

2.2.1 The 2006 Open Source CMS Award

V prvním ročníku ještě nebyly kategorie, pouze celkové umístění:

1. Joomla!
2. Drupal
3. Plone

2.2.2 The 2007 Open Source CMS Award

Kategorie celkový vítěz:

1. Drupal
2. Joomla!
3. CMS Made Simple

Kategorie nejlepší PHP Open Source CMS:

1. Joomla!
2. Drupal
3. e107

Kategorie nejlepší Open Source CMS pro sociální síť:

1. WordPress
2. Drupal, elgg (dělené druhé místo)

2.2.3 The 2008 Open Source CMS Award

Kategorie celkový vítěz:

1. Drupal
2. Joomla!
3. DotNetDuke

Kategorie nejlepší PHP Open Source CMS:

1. Drupal
2. Joomla!, CMS Made Simple (dělené druhé místo)

Kategorie nejlepší Open Source CMS pro sociální síť:

1. WordPress
2. Drupal, elgg (dělené druhé místo)

2.3 Drupal

Drupal je svobodný software, který umožňuje uživatelům jednoduše publikovat, spravovat a organizovat širokou škálu obsahu internetových stránek. Desítky tisíc lidí a organizací používají Drupal jako základní kámen nejrůznějších webů [5], [6].

2.3.1 Historie

V roce 2000 vytvořil Dries Buytaert stránku, kde si s kamarády ze školy mohli zapisovat poznámky a sdílet svoje zážitky. Jakmile dokončili školu, chtěli zůstat v kontaktu i nadále a rozhodli se vystavit stránku na internet. Dries hledal vhodné jméno a rozhodl se pro *dorp.org* (*dorp* znamená v holandštině *vesnice – malá komunita*), ale udělal překlep a dal tak vzniknout stránce *drop.org*. Přidávalo se čím dál více uživatelů, kteří diskutovali o webových technologiích a softwaru.

V lednu roku 2001 se Dries rozhodl vydat software pod názvem Drupal (z holandského *druppel*, což znamená anglické *drop*). Hlavním účelem bylo umožnit ostatním uživatelům používat a rozšiřovat systém, a tím objevovat nové cesty vývoje.

Dnes má komunita Drupalu téměř půl milionu členů a neustále se rozšiřuje, počet stažení Drupalu se za poslední rok blíží k 1,5 milionu [7].

2.3.2 Mise a hodnoty

Drupal si stanovil svoje mise a hodnoty takto [8]:

- Flexibilita, jednoduchost, užitečnost
- Týmová spolupráce, inovace, otevřenost komunity
- Modularita, rozšiřitelnost, udržovatelnost kódu

2.3.3 Drupal v praxi

Z webových stránek používajících Drupal [9] mohou uvést:

- <http://www.mtv.co.uk> – Oficiální web britské hudební stanice MTV.
- <http://gadgets.zive.cz> – Tento a mnoho dalších serverů z dílny Computer Press.
- <http://theworld.org/> – Informační server se zprávami z celého světa

2.4 Joomla!

Joomla! je více než jenom software, jsou to lidé – komunita Joomla! zahrnuje vývojáře, designéry, systémové administrátory, překladatele, tvůrce obsahu, a ty nejdůležitější – koncové uživatele [10].

2.4.1 Historie

Název *Joomla!* je fonetickým přepisem svahilského slova *Jumla*, které znamená „všichni společně“ nebo „jako celek“. Tento název byl z tisíců návrhů vybrán komunitou a marketingovými profesionály. Historie sahá až do roku 2000, kdy byl založen projekt *Mambo* konkurující tehdy převládajícímu *phpNuke*. Po letech vývoje a několika oceněních za nejlepší open source projekt se v roce 2005 kvůli vnitřním rozporům oddělila skupina vývojářů, kteří založili organizaci *Open Source Matters* a nedlouho po tom vydali první verzi Joomla! 1.0 [11]. Oba projekty si teď jdou svou vlastní cestou.

2.4.2 Mise a hodnoty

Naše mise je zajistit flexibilní platformu pro digitální publikování a spolupráci [12].

- Svoboda (*freedom*) – v duchu svobodného softwaru a GPL
- Rovnocennost (*equality*) – stejný přístup ke všem, nezávisle na národnosti, pohlaví. . .
- Důvěra (*trust*) – možnost spolehnout se na produkt, důvěra mezi vývojáři
- Komunita (*community*) – bez komunity bychom nebyli schopni dosáhnout ničeho
- Spolupráce (*collaboration*) – spolupráce s lidmi mimo komunitu
- Použitelnost (*usability*) – lepší použitelnost by mohla komunitu dále rozšířit

2.4.3 Joomla! v praxi

Existuje spousta webů, které shromažďují stránky vytvořené v CMS Joomla!, např. [13], [14]. Některé ze zajímavých stránek:

- <http://gsas.harvard.edu/> – Harvard University
- <http://www.quizilla.com/> – Sociální síť
- <http://alkohol-tester.cz/> – Internetový obchod
- Citibank (intranet)

2.5 WordPress

WordPress je vyspělý a stabilní produkt, jehož kořeny sahají až do roku 2001. V roce 2003 byly napsány první kusy kódu vylepšující každodenní psaní a uživatelů bylo asi tolik, že by se dali spočítat na prstech jedné ruky. Časem však dorostl do nejrozšířenějšího blogovacího nástroje na světě, používaného na stovkách tisíc webových stránek, které jsou shlédnuty desítkami milionů lidí denně [15].

2.5.1 Historie

WordPress byl vytvořen za účelem elegantního, dobře navrženého publikačního systému, zaměřeného na uživatele a webové standardy. Je to oficiální následník *b2/cafelog*. Postupem času se neustále zvyšovala jeho oblíbenost – v roce 2005 byl stažen 900 000 krát, v roce 2006 to bylo 1,5 milionu stažení a v roce 2007 skoro čtyři miliony! Zajímavostí je, že vývojáři jádra WordPressu jsou fanoušky jazzu, a tak všechny hlavní verze jsou pojmenovány podle jazzových hudebníků [15].

2.5.2 Mise a hodnoty

WordPress nemá explicitně vyjmenované svoje mise a hodnoty, nicméně z jeho filozofie se dá vyčíst [16]:

- Svoboda používání
- Zaměření na uživatele

2.5.3 WordPress v praxi

WordPress byl původně vytvořen jako blogovací nástroj, který sloužil především pro snadné blogování a až v posledních verzích se z něj stává plnohodnotný redakční systém. Na většině webů je tedy stále hlavním prvkem blog, ale objevují se i jiné [17]:

- <http://wordpress.com/> – blogovací portál
- <http://www.cultura.gov.br/site/> – brazilské ministerstvo kultury
- <http://autoshow.ford.com/> – automobilka Ford

Za zmínku ještě stojí WordPress MU (Multi-User WordPress) [18] a BuddyPress [19], což jsou rozšířené varianty WordPressu zaměřené na víceuživatelské blogy a celé komunity uživatelů.

Kapitola 3

Používané technologie

Uvedeme si nejpoužívanější technologie open source redakčních systémů. Pro samotné používání CMS jejich koncovými uživateli/editory zas tak důležité nejsou – nepotřebují vědět, jak a kam se ukládají data, ale pro webové vývojáře je to nezbytně nutné. Sice je možné vytvořit celou webovou prezentaci pomocí CMS bez jakýchkoli znalostí PHP, MySQL, XHTML, CSS či Javascriptu, ale ne vždycky všechno vyhovuje natolik, aby se nemuselo zasáhnout do kódu. Úroveň požadovaných znalostí musí odpovídat tomu, co s tím kterým systémem chceme provádět.

3.1 PHP

PHP (*PHP: Hypertext Preprocessor*) je skriptovací jazyk používaný především pro psaní serverové části webových aplikací (*server-side scripting*), kdy generuje (X)HTML kód. Dá se použít pro psaní konzolových i desktopových aplikací, ale jeho hlavní pole působnosti je na webu. Oblíbený je především díky své dostupnosti, multiplatformovosti a v neposlední řadě rozšířenosti. Aktuální verze je PHP 5.2.9 [20], [21].

3.2 MySQL

MySQL je nejoblíbenější variantou relačního databázového systému, zvláště při použití jazyka PHP. Stejně jako PHP se jedná o open source software a je dostupný zdarma. MySQL server zpracovává dotazy jazyka MySQL patřícího do rodiny dotazovacích jazyků SQL (*Structured Query Language*). Aktuální verze je MySQL 5.1.34 [20], [22].

3.3 Apache

Apache je nejrozšířenější webový server, který je možné použít na různých platformách. Podle [23] je použitý na téměř polovině serverů. Aktuální verze je Apache 2.2.11.

3.4 (X)HTML, CSS

Webové stránky bývají vytvořeny značkovacím jazykem HTML (*HyperText Markup Language*) [24] nebo jeho modernější podobou XHTML (*Extensible HyperText Markup Language*) [25], která používá strukturu XML (*Extensible Markup Language*) [26]. Všechny tyto jazyky

patří do rodiny SGML (*Standard Generalized Markup Language*). Současné nejpoužívanější verze jsou nejspíš HTML 4.01 a XHTML 1.0. Pro vizuální podobu stránek se používá kaskádových stylů (*CSS – Cascading Style Sheets*) [27].

3.5 Javascript, AJAX, jQuery

Javascript je multiplatformní, objektově orientovaný skriptovací jazyk, který je v oblasti webu používán ke skriptování na straně klienta (kontrola formulářů před odesláním dat na server, změna vzhledu dokumentů přepínáním jeho stylů, atd.) [28].

AJAX (*Asynchronous Javascript and XML*) je označení technologií, které slouží k tomu, aby se na webových stránkách mohl dynamicky zobrazovat obsah (komunikace klient-server) bez nutnosti znovunačtení stránky. Umožňuje tak vytváření interaktivnějších webových aplikací [29].

jQuery je zdarma dostupná javascriptová knihovna, která usnadňuje programování javascriptových částí webových aplikací. V redakčních systémech je hojně používána, ať už se jedná o zobrazování kalendářů, kontrolu formulářů, hodnocení obsahu nebo zlepšení použitelnosti uživatelských rozhraní [30].

Kapitola 4

Systemové požadavky

Základním požadavkem pro každý systém je webový server s PHP a databáze. Pro vybrané systémy se doporučuje *Apache*, ale nastavit se dá i *Microsoft IIS (Internet Information Services)* či *Litespeed*, i když to třeba není oficiálně podporováno. Doporučovaná databáze je *MySQL*, Drupal si ovšem poradí i s *PostgreSQL*. Někdy si systém nebo jeho modul klade určité nároky na nastavení serveru – například limity paměti, velikosti nahrávaných souborů, vyžaduje povolené knihovny, ať už je to knihovna PHP `gd` pro práci s obrázky, anebo Apache modul `mod_rewrite` pro „čisté URL“. Upřesňující informace o nastavení serverů jsou k dispozici na webových portálech jednotlivých systémů [31], [32], [33], popř. diskutovány na komunitních fórech.

4.1 Souhrn požadavků

Základní souhrn požadavků na provoz systému zachycuje následující tabulka:

| | Drupal 6.11 | | Joomla! 1.5.10 | | WordPress 2.7.1 | |
|--------|-------------|-----------|----------------|-----------|-----------------|-----------|
| | Doporučené | Minimální | Doporučené | Minimální | Doporučené | Minimální |
| Apache | 2.x+ | 1.3 | 2.x+ | 1.3 | 2.x+ | 1.3 |
| PHP | 5.2+ | 4.3.5 | 5.2+ | 4.2 | 4.3+ | 4.3 |
| MySQL | 5.0+ | 4.1 | 4.1.x.+ | 3.23 | 4.0+ | 4.0 |

Tabulka 4.1: Porovnání systémových požadavků

- Drupal 7 bude vyžadovat MySQL 5.0+
- Joomla! 1.6 bude vyžadovat PHP 5.2+ a MySQL 4.1+
- WordPress neudává žádné upřesňující informace k nastavení serveru. Těžko říct, zda nejsou nutné, nebo tím jenom nechťejí zatěžovat uživatele – spoléhají na webhostera.

4.2 Webhosting

Hostingové společnosti většinou ví, zda je možné ten či onen systém použít, nebo dokonce přímo nabízí instalaci „na jedno kliknutí“ z administračního rozhraní domény. Na sdíleném hostingu však může nastat problém s nastavením některých konfiguračních hodnot

(*direktiv*) PHP – nejčastěji se jedná o `safe_mode` [34] a `register_globals` [35]. Safe mode bývá zapnutý z důvodu bezpečnosti – zamezuje například použití funkcí pro přístup k souborům. Register globals se zachovává pouze z důvodů zpětné kompatibility – před verzí PHP 4.2.0 se k proměnným globálních polí (například `$_POST['klic']`, `$_GET['klic']` nebo `$_SERVER['klic']`) dalo přistupovat přímo přes `$klic`, což není zrovna nejbezpečnější a doporučuje se tento způsob nepoužívat. Proto je vždy nejlepší se předem zeptat, jak je to s těmito nastaveními a zda je systém na serveru v daném hostingovém programu podporován.

Ideální hosting by byl zadarmo, s neomezeným prostorem, garantovanou rychlostí a spolehlivostí, pravidelnými zálohami projektu i databáze, rychlou a ochotnou zákaznickou podporou, bez reklam. Takový ovšem neexistuje. Je potřeba si na začátku ujasnit, co je pro nás hlavním prioritou, z čeho slevit nehodláme a co naopak budeme tolerovat. Jestli jsme ochotni na webu umístit reklamní proužek, uskromnit se v prostoru na disku, obejít se bez zákaznické podpory, provádět si zálohy sami, přetrpět občasné výpadky, čekat dlouhé chvíle, než se nám naše webová prezentace nahraje na server, atp. Ne vždy je ovšem placený hosting lepším řešením než neplacený.

Ještě je třeba dodat, že redakční systémy mívají na svých stránkách seznam doporučovaných a také nedoporučovaných hostingů. Stejně jako mají lepší hostingsy seznam redakčních systémů, které doporučují a na které jsou připraveni. Momentálně se cena dobrého hostingu, aniž bych dělal reklamu nějakému konkrétnímu, pohybuje okolo 100 Kč/měs.

Kapitola 5

Vytvoření testovacího webu

Pro srovnání systémů je potřeba vytvořit web, který by v ideálním případě vypadal ve všech systémech stejně a měl naprosto stejnou funkčnost. Toho se sice kvůli odlišnému konceptu systémů nepodaří dosáhnout nikdy, přesto je tento web základem subjektivního hodnocení toho kterého redakčního systému. Pro vývoj byl zvolen web zaměřený na redakční systémy s potenciálně vyšší komunitou uživatelů, protože je tento typ webových stránek v posledních letech na ohromném vzestupu a také proto, že umožňuje pojmout široké spektrum použitých prvků, které pomohou úplnějšímu hodnocení. Tyto testovací weby nejspíše skončí „v šuplíku“, ale při návrhu a implementaci pokusíme postupovat tak, jako by měly být reálně nasazeny.

5.1 Struktura webu, kategorizace obsahu

Při návrhu webu je nutné si na začátku říct, komu jsou stránky určeny, co bude jejich obsahem a jak budou rozděleny. Jinými slovy jde o vytváření logické struktury webových stránek. Stránky tedy mají určitou hierarchii a pohybujeme se mezi nimi pomocí navigace (seznam odkazů na jednotlivé stránky). Nejčastěji je zobrazena formou menu. Navigaci rozlišujeme na globální (dosažitelná ze všech stránek) a lokální (přístupná pouze v dané sekci), případně je možno použít drobečkovou navigaci (*breadcrumbs*) umístěnou většinou nad hlavním nadpisem stránky v dané sekci. Dalším druhem navigace může být také mapa stránek (*site map*), což je speciální stránka obsahující přehledný a strukturovaný seznam všech stránek na webu.

Na testovacím webu budou následující sekce:

- Domů – hlavní stránka obsahující poslední články, blog posty a kalendář akcí
- Články – články editorů s možností komentářů
- Blogy – seznam blogů správců webu, poslední blog posty, komentáře
- Fórum – základní diskusní fórum pro uživatele
- Akce – Seznam pořádaných akcí (semináře, školení) – napojený na kalendář
- E-shop – internetový obchod s reklamními předměty – trička, čepice, odznaky
- Uživatelé – seznam uživatelů

5.1.1 Blog

Blog je ve své podstatě takový internetový deníček. Je to jeden ze způsobů, jak se na internetu prezentovat a sdílet svoje názory s ostatními. Využívají ho miliony lidí na celém světě a nejen pro osobní účely – vytvoření blogu s vhodným obsahem a zacílením může být dobrým marketingovým tahem. *Blog post* je označení pro příspěvek do blogu (řekněme zápis do deníčku). V souvislosti s blogováním se můžeme setkat ještě s několika dalšími pojmy, jako jsou např. *Linkback*, *Pingback*, *Refbck*, *Trackback* a tyto pojmy se snad ani nepřekládají [36]. O co se vlastně jedná?

Linkback je souhrnné označení pro metody (*Trackback*, *Pingback*, *Refbck*), kterými můžeme dát vědět autorovi jiných stránek, že se odkazujeme na jeho příspěvek – on pak na svých stránkách může zobrazit seznam stránek (příspěvků), které se na jeho příspěvek odkazují. Výhodou je tedy provázanost souvisejících příspěvků, ať už z pohledu SEO (čím více se na náš web bude odkazovat, tím více bude vidět, bude mít vyšší návštěvnost, atd.), nebo z pohledu čtenáře, kterého daná problematika příspěvku zajímá a rád by si o ní přečetl něco více.

Trackback funguje na následujícím principu: pokud si *blogger* přeje, aby se na jeho příspěvek mohli ostatní odkazovat, uveřejní u příspěvku svoji *Trackback URL* (odkaz na příspěvek – díky CMS je tento proces automatizován). Odkazující tuto adresu použije při vytváření svého příspěvku – vloží ji do políčka určeného k tomuto účelu a odešle požadavek.

Pingback je novějším způsobem této notifikace. Není potřeba vkládat adresu do speciálního pole, stačí ji použít v textu příspěvku. Při ukládání se text projde a odešle pingback na všechny adresy v textu použité (pokud ještě nebyly odeslány). *Pingback* je svoji strukturou jednodušší – obsahuje pouze zdrojovou a cílovou adresu, zatímco *trackback* může obsahovat i název stránky, název příspěvku a jeho krátký úryvek. *Pingback* je méně náchylný na SPAM (nevyžádané zprávy) a obecně platí zásada, že pokud systém podporuje pingback, neměly by se na něj posílat trackbacky, resp. oboje najednou. Mnohdy totiž platí, že uveřejnění na stránkách původního autora podléhá schvalovacímu procesu – tzn., že původní autor musí zobrazení na svých stránkách schválit, a když bude muset schvalovat dvakrát stejnou adresu, mohl by nás označit jako SPAM.

Refbck je nejméně používanou metodou, která je založena na zjišťování, odkud návštěvník přišel (na jaké stránce kliknul na odkaz).

5.1.2 Tag, tag cloud

Setkáváme se s tím čím dál častěji – box vyplněný slovy, která mají různou velikost písma. V poslední době se jedná o jeden z nejpoužívanějších postupů kategorizace obsahu. Každému článku/příspěvku se přiřadí štítky (*tags*) – klíčová slova, která jej charakterizují. Ty se potom zobrazují v obláčku/boxu štítků (*tag cloud*), přičemž velikost písma, kterou se daný štítek zobrazí, je určena jeho četností použití. Umožňuje tak uživateli na první pohled zjistit, čím se stránka zabývá nejvíce a kliknutím na štítek mu zobrazí veškeré články/příspěvky, které byli tímto štítkem označeny.

5.2 Design layout

Když už máme strukturu webu, můžeme se zaměřit na to, jak budou stránky vypadat. Designem myslíme grafickou podobu webových stránek, co se týče obrázků, písem a dekozací. *Layout* znamená rozložení prvků na stránce. Dnes se využívá beztabulkových layoutů,

především kvůli odstranění nevýhod těch tabulkových, kdy se např. stránka zobrazila až v okamžiku, kdy byla úplně načtená a do té doby viděl uživatel jen prázdnou obrazovku. Tabulky by měly být používány pouze pro zobrazování tabulárních dat – problematika sémantického webu a SEO je mnohem hlubší, nám teď stačí, že budeme používat beztabulkový layout a veškerý design bude definován pomocí kaskádových stylů (*CSS*). Layout dále můžeme rozdělit podle počtu sloupců, které obsahuje (např. dvousloupcový layout s navigací vlevo a obsahem vpravo), nebo podle toho, zda je definována jeho šířka absolutně (fixní) či relativně (plovoucí), resp. zda se šířka mění podle velikosti okna.

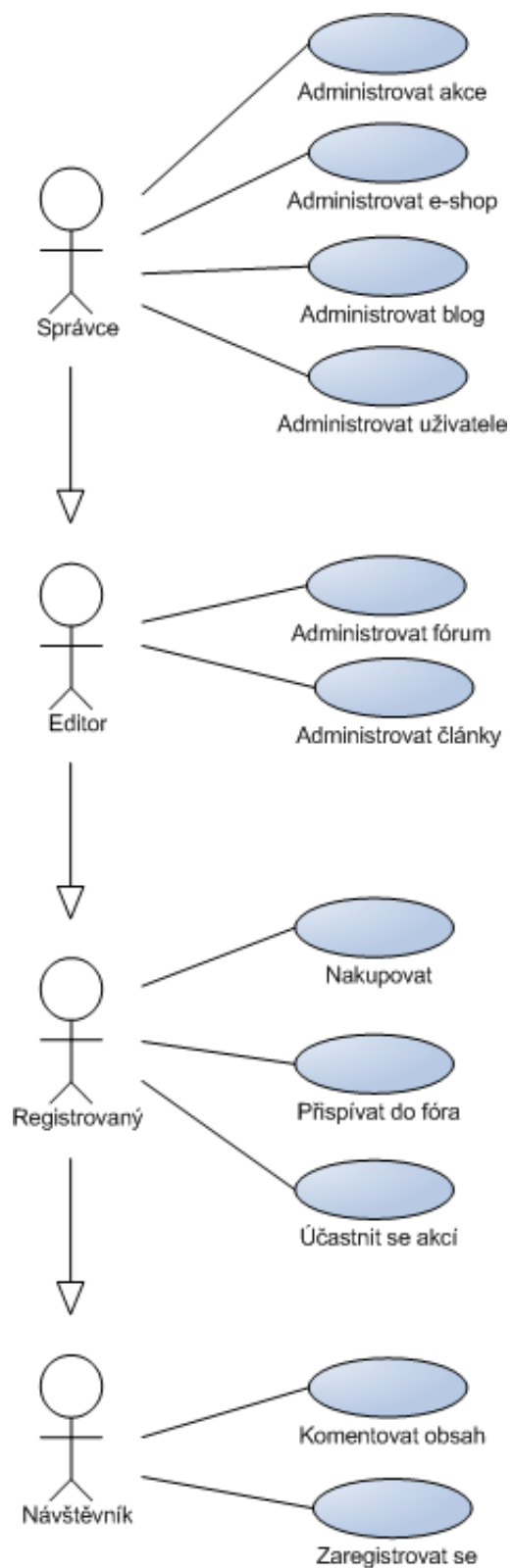
Návrh pro testovací web:

- Záhlaví a hlavní logo (header) obsahuje loga všech CMS, je ovšem vyznačeno, který systém je na webu použit a zdůrazněno, že se jedná o stránky pro testovací účely
- Horizontální menu a vyhledávací políčko
- Třísloupcový layout:
 - Levý sloupec: navigace, kalendář akcí, anketa
 - Prostřední sloupec: drobečková navigace, vlastní obsah
 - Pravý sloupec: přihlašovací formulář, štítky, nákupní košík
- Patička s informacemi o redakčním systému

5.3 Uživatelské role

Nadefinování uživatelských rolí je pro systém CMS jednou z nejdůležitějších záležitostí, protože na tom myšlenka redakčního systému stojí. Je potřeba označit, do jaké role uživatel patří, jaké akce může provádět, a ke kterým nastavením a datům má přístup.

V našem případě byly role navrženy takto: správci portálu mají svůj blog, editoři píšou články, registrovaní uživatelé mohou přispívat do fóra a nakupovat, nepřihlášení uživatelé mohou psát pouze komentáře. Registrovat se může kdokoli. Platí základní hierarchie, že nadřazená role může dělat všechno, co všechny role pod ní (i správce webu může napsat komentář).



Obrázek 5.1: Use-case diagram

Kapitola 6

Srovnání CMS

V této kapitole se dostáváme k implementaci navrženého webu a srovnání vybraných systémů. Je rozdělena do několika podkapitol tak, aby bylo vidět, jak si proti sobě jednotlivé systémy stojí v situacích, které bude dříve či později snad každý uživatel/vývojář řešit.

6.1 Architektura a názvosloví

Je dobré si na začátku zjistit, z jakých částí se systém skládá, protože když je potřeba něco upravit (a to je skoro vždy), musíme vědět, kde se to dělá. Nebo když potřebujeme s něčím pomoci a ptáme se na diskusním fóru, musíme umět formulovat otázku správně, aby i ostatní věděli, o čem to vlastně mluvíme. V každém systému totiž stejný termín může znamenat něco jiného, a když vás lidé nepochopí, nemají chuť a také ani nemůžou pomoci.

Systémy jsou založeny na třívrstvé architektuře – rozdělení vrstev aplikace na datovou, aplikační a prezentační (*data, business and presentation layer*). V té datové je napojení na databázi, v aplikační se nachází jádro systému (*core*) a přídatných modulů, prezentační vrstvu tvoří šablony.

6.1.1 Drupal

Drupal je vysoce modulární systém, přičemž i vlastní **jádro** je na moduly rozděleno. Systém lze rozšířit pomocí **modulů** (*modules*). Moduly fungují tak, že jsou v nich implementovány tzv. **háky** (*hooks*) [37], což jsou funkce, které jsou volány jádrem, nastane-li určitá situace a je-li modul povolený. **Motivy vzhledu** (*themes*) jsou určeny k úpravě výstupu (HTML, XML, atp.) a jsou tvořeny **šablonami** (*templates*). Základním typem obsahu je **uzel** (*node*).

Z toho jsou potom odvozeny všechny ostatní typy obsahu, ať už se jedná o článek, blog, nebo vlastní nadefinovaný typ. Výhoda je především v tom, že je možné využívat jejich společných vlastností – např. u každého uzlu je možné nastavit, zda je publikovaný, jestli se zobrazí na hlavní stránce, atd. **Bloky** (*blocks*) jsou prvky webu, které obsahují nejrůznější informace (např. navigaci, ankety, odkazy na nejnovější články, štítky, atp.) a jsou zobrazeny na místě k tomu určeném (v **regionech** definovaných šablonou).

Prvotní zmatení uživatele mohou způsobit (a většinou se tak stane) nešťastně pojmenované předdefinované typy obsahu **Stránka** (*Page*) a **Článek** (*Story*). Uživatel si většinou řekne, že si nejspíš bude muset vytvořit stránku a na ní pak bude přidávat články, ale to se mu nepodaří. Stránka a článek jsou v tomto případě dva v podstatě stejné (rovnocenné) typy obsahu a slouží jenom k jejich odlišení při dalším nastavování, kterým se dá zajis-

tit, aby se například stránky zobrazovaly v menu (budou sloužit k zobrazení statických informací), zatímco články se tam zobrazovat nebudou.

Kategorizace obsahu se provádí pomocí modulu *Taxonomy*, který poté spravuje jednotlivé **slovníky** (*vocabularies*) obsahující **termíny** (*terms*) [38].

6.1.2 Joomla!

Joomla! má vlastní **framework**, který lze rozšířit pomocí **zásuvných modulů** (*plugins*) a **knihoven** (*libraries*). Na tomto frameworku je pak postavená aplikace, kterou lze dále rozšířit pomocí **rozšíření** (*extensions*), což jsou **komponenty** (*components*), **moduly** (*modules*) a **šablony** (*templates*).

Knihovny jsou užívány samotným frameworkem nebo nástroji třetích stran. Zásuvné moduly rozšiřují základní chování frameworku. Šablony jsou určeny ke změnám vzhledu výsledných stránek, komponenty jsou rozšíření (např. blog, e-shop, fórum) a moduly slouží k zobrazování informací (např. navigace, ankety, ...) na určitém místě stránky definovaném šablonou. Po srovnání s Drupalem zjistíme, že šablony odpovídají motivům vzhledu, komponenty odpovídají modulům a moduly odpovídají blokům [39]. Pro nováčka pěkný zmatek.

V názvosloví obsahu je to mnohem jednodušší – hlavní kategorizace se provádí pomocí **sekcí** (*sections*), které pak obsahují **kategorie** (*categories*) a do nich se zařazují jednotlivé **články** (*articles*).

6.1.3 WordPress

WordPress byl původně určen pouze pro blogy a z toho vychází spousta jeho vlastností. Uživatelé může zmást, že při instalaci (nebo v nastavení) si má zvolit „název blogu“, přitom tam třeba blog ani mít nechce. V české verzi už se to překládá jako „název webu“, aby to bylo méně matoucí.

WordPress se dá rozšiřovat pomocí **pluginů/zásuvných modulů** (*plugins*), o úpravy vzhledu stránky se starají **motivы vzhledu** (*themes*), které obsahují **šablony** (*templates*). Pluginy mohou obsahovat **widgety** (*widgets*), což je obdoba bloků Drupalu (zobrazují blok informací – ankety, navigace, atp.). Šablony musí být na použití widgetů připraveny a většinou jsou. Také lze implementovat **háky** (*hooks*), které nesou stejný význam jako v Drupalu – když nastane nějaká událost, jsou volány všechny funkce, které daný hák implementují. WordPress umožňuje konfiguraci pluginů pomocí **Shortcode API** (*shortcodes*), což jsou krátké úseky textu ohraničené hranatými závorkami, které se vkládají do příspěvků nebo stránek. Při zobrazení příspěvku se potom tento shortcode nahradí skutečným obsahem, který vygeneruje daný plugin.

WordPress má dva základní typy obsahu – **stránky** (*pages*) a **příspěvky** (*posts*). Stejně jako u Drupalu jsou stránky myšleny ve smyslu statického obsahu (např. o webu, kontaktní informace, atd.), ale s rozdílem, že nemohou být zařazeny do **rubrik/kategorií** (*categories*) a označeny **štítky** (*tags*). Kategorie a štítky slouží ke kategorizaci příspěvků (v původní myšlence blog postů) [40].

6.1.4 Závěr

Z předchozího textu je vidět, že pro uživatele může být názvosloví dosti matoucí, zvlášť když má některé termíny zažité z jednoho systému a přechází na druhý nebo vidí všechny

systemy poprvé. Také jsme se seznámili s pojmy, které budou používány v následujících kapitolách.

6.2 Instalace

Pokud splňujeme základní požadavky uvedené v kapitole 4, můžeme přejít k samotné instalaci, která se vždy skládá z několika kroků. Obecně lze říci, že je potřeba nahrát data na webový server, vytvořit na databázovém serveru databázi a tyto dvě věci nějakým způsobem propojit. Zároveň se při instalaci ověřují různé systémové požadavky (např. práva na přístup ke složkám a souborům), vytváří administrátorský účet a může se objevit i další nastavení – např. jazyk instalace nebo přístupové údaje k FTP serveru. Všechny systémy nabízí možnost zvolit předponu tabulek (*table prefixes*), díky které lze instalovat do jedné databáze více systémů – jsou potom navzájem rozlišitelné. Když pomineme délku kopírování souborů na webový server, což je silně závislé na rychlosti internetového připojení, vytíženosti vzdáleného serveru a počtu přenášených souborů, instalace netrvá déle než pár minut.

6.2.1 Instalace na localhost

Pokud budeme chtít vytvářet složitější webové stránky, je vždycky lepší si je nejdříve odladit na lokálním počítači. Odezva stránek je rychlejší, nemusíme trpět pomalé kopírování souborů na FTP, můžeme ovlivnit nastavení serveru, nejsme závislí na nikom jiném, než sami na sobě. Jak tedy co nejjednodušším způsobem zprovoznit web na svém počítači?

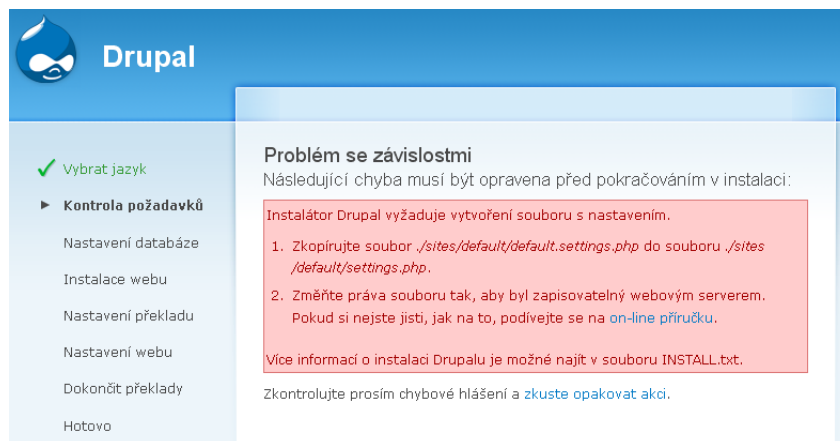
Jak říkají požadavky systémů, potřebujeme Apache, MySQL, PHP a ke správě DB se hodí např. phpMyAdmin. Buď můžeme stahovat, instalovat a nastavovat jednotlivé komponenty zvlášť, nebo můžeme využít speciálních nástrojů pro jejich pohodlnou správu. Možností je více, doporučit lze například:

- XAMPP [41] – zdarma dostupný pro Linux, Windows, Mac OS X i Solaris.
- WAMP [42] – zdarma dostupný pro Windows, umožňuje spravovat více verzí Apache, MySQL i PHP najednou.

6.2.2 Drupal

Celý instalační proces je popsán na [43], zjednodušeně:

1. V adresáři `~/sites/default/` se nachází soubor `default.settings.php`, který je nutné zkopírovat do `settings.php` (nestačí pouhé přejmenování, protože instalační proces potřebuje oba soubory) a tomuto souboru musíme po čas instalace změnit nastavení oprávnění pro zápis.
2. Spuštění instalace se provede přístupem na hlavní stránku (např. `http://drupal`)
3. Necháme se vést instalačním procesem, který je dostatečně přehledný i pro nové uživatele. Popisky k jednotlivým nastavením jsou dostatečné, případné problémy jsou vysvětleny včetně návodu na jejich řešení.

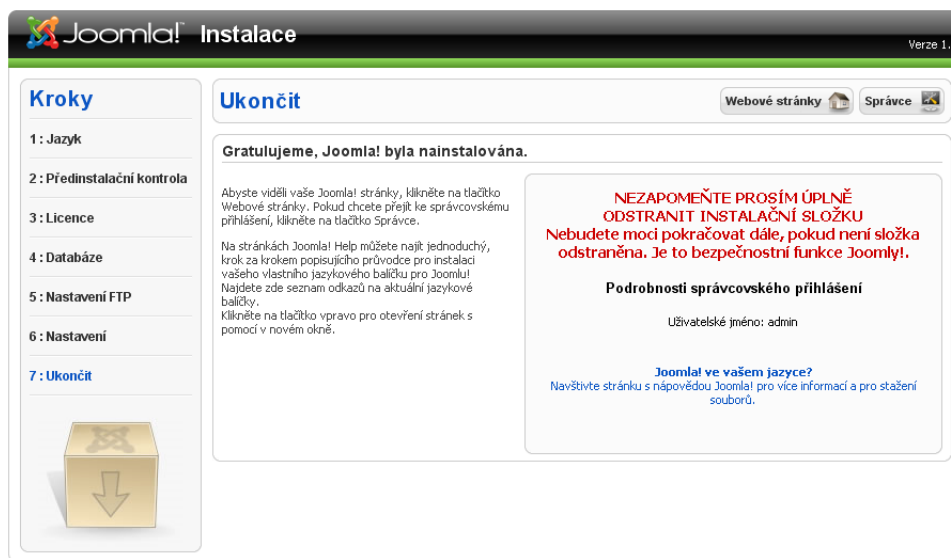


Obrázek 6.1: Instalace Drupalu

6.2.3 Joomla!

Podrobný instalační manuál je dostupný na [44], stejně jako návod jak začít vytvářet stránky. Opět můžeme využít automatické instalace.

1. Pro automatickou instalaci je nezbytně nutná složka `~/installation/`, ve které jsou potřebné instalační skripty a po instalaci se musí odstranit.
2. Instalace se spouští zobrazením titulní stránky (např. `http://joomla`)
3. Během instalace je možné zvolit nahrání testovacích dat, což bych začátečníkům opravdu doporučoval.



Joomla! je svobodný software šířený pod licenci GNU/GPL verze 2.0.

Obrázek 6.2: Instalce Joomla!

6.2.4 WordPress

WordPress se pyšní svojí jednoduchou a rychlou instalací [45].

1. Pro instalaci je potřebný soubor `wp-config.php` (jako vzor může posloužit soubor `wp-config-sample.php`)
2. Instalace se spouští přístupem na hlavní stránku (např. `http://wordpress`)
3. Pokud byl soubor `wp-config.php` vytvořen a správně vyplněn manuálně už před instalací, stačí zadat název webu a e-mail administrátora. Pokud ovšem uživatel soubor nevytvoří, WordPress ho umí vytvořit sám a uživateli umožní vyplnit přístupové údaje k databázi ve formuláři instalace (samozřejmě musí mít příslušná oprávnění pro zápis do kořenového adresáře, což je po instalaci vhodné zrušit).



Obrázek 6.3: Instalce WordPress

6.2.5 Závěr

Instalace vybraných CMS není vůbec složitá, je ale potřeba, aby se uživatel řídil pokyny, zvláště pokud nemá žádné zkušenosti. Každý systém má svoje specifické postupy, kterými zjišťuje, v jakém stavu se nachází – zda je server správně nastaven, jestli je systém již nainstalován, zda má potřebná data k provozu. Přes všechny rozdíly je náročnost prakticky stejná a zcela intuitivním způsobem lze za několik málo minut dosáhnout výsledku. Velikosti instalačních balíčků jsou uvedeny v tabulce 6.3 kapitoly 6.13, kde je vidět rozdíl mezi čistou instalací a zálohou projektu.

6.3 Rozšiřitelnost

Po čisté instalaci lze provádět základní operace a nastavení, přidávat nějaký obsah, ale pro vytvoření finální podoby webových stránek zřejmě dostačovat nebude. Bez přidavných

rozšíření se prostě neobejdeme a hlavně ani nechceme. Díky nim se dá ten či onen systém vyšperkovat a stává se použitelným pro nejrůznější účely. Za vybranými systémy stojí rozsáhlá komunita uživatelů a vývojářů, kteří se zabývají právě těmito rozšířeními. Existují rozšíření dostupná zdarma, ale na druhé straně také i naprosto komerční řešení – to je totiž jedna z věcí, na kterých se dá opravdu vydělat.

Při shánění rozšíření je potřeba dávat pozor na jejich verzi – všechna rozšíření uvádějí, pro jakou verzi systému jsou určena. Pokud rozšíření vyhovuje, je nutné ho nějakým způsobem zkopírovat do určeného umístění na serveru a aktivovat. Právě na to, jaké máme možnosti, se podíváme dále.

6.3.1 Drupal

Modulů pro Drupal existuje několik tisíc, ale na jeho domovských stránkách [46] je vyhledávání poměrně složité a je těžké najít požadované rozšíření, pokud neznáme jeho jméno. Pro vyhledávání modulů tak mohu doporučit stránku [47], která je mnohem přehlednější, umožňuje pohodlné filtrování a vyhledávání, obsahuje hodnocení uživatelů, atd.

Stažený modul je potřeba rozbalit a zkopírovat do složky `modules`. Tato složka se sice nachází i v kořenovém adresáři, ale ten doporučuji nevyužívat, protože by bylo složitější zálohování. Mnohem lepší je použít `~/sites/all/modules/`. Pokud bychom měli více webů na jednom serveru a modul se měl využít jenom pro jeden konkrétní, pak by se měl zkopírovat do `~/sites/<nazev-webu>/modules/`. Potom jej stačí povolit v seznamu modulů. Často se stává, že jsou na sobě různé moduly závislé a jeden bez druhého nebude funkční. Tyto závislosti jsou přehledně zobrazené v seznamu modulů – u každého modulu je seznam, které moduly tento vyžadují a na kterých je sám závislý.

Ještě bych se měl zmínit o modulu *Plugin Manager*, který by se měl postarat o téměř automatickou instalaci modulů (viz [48]) a napodobit tak WordPress, ale ne vždy je tento modul použitelný.

6.3.2 Joomla!

Pro Joomla! je také k dispozici několik tisíc rozšíření, na stránkách [49] jsou přehledně řazeny do kategorií a dá se v nich poměrně dobře orientovat. Každé rozšíření se může skládat z komponent, modulů a pluginů, přičemž pro každý typ rozšíření je na serveru vytvořena složka – tedy `~/components`, `~/modules/` a `~/plugins`.

Instalace se provádí z administračního rozhraní, kde máme možnost instalovat rozšíření zadáním URL adresy jeho instalačního balíčku, pomocí nahrání staženého instalačního balíčku, nebo zadáním složky, ve které je dané rozšíření rozbaleno. Poté je nutné dané rozšíření povolit ve Správci rozšíření.

6.3.3 WordPress

I když tomu tak dříve nebylo, WordPress rychle dohnal ostatní redakční systémy a také se může pyšnit několika tisíci pluginů. Navíc nabízí vyhledávání a instalaci pluginů přímo v pohodlí svého administračního rozhraní. Samozřejmě je možné nahrát instalační balíček pomocí formuláře nebo manuálně nakopírovat rozbalený balíček do `~/wp-content/plugins/`, ale pohodlnost nikdy neomrzí.

6.3.4 Závěr

Už z popisu jednotlivých částí kapitoly se dá vycítit, že z porovnávaných systémů je na tom z hlediska uživatele a instalace rozšíření WordPress nejlépe, zatímco Drupal nejhůře. Co se týče programování vlastních rozšíření nebo jejich pochopení, zvláště méně zkušení programátoři ocení dokumentační stránky WordPressu, které jsou plné tutoriálů a dokonce i API je popsáno do detailu s trpělivostí a polopatickým vysvětlením. Na druhou stranu je i dokumentace systémů Drupal a Joomla! na velice dobré úrovni a především zkušenějším programátorům by mohla vyhovovat více. Závěrem této kapitoly je také zjištění, že už dávno neplatí tvrzení, že v Drupalu lze díky jedinečnému modulárnímu systému udělat jakýkoli web, narozdíl od WordPressu, který se hodí jenom na blog. Není sporu o tom, že lze Drupal rozšířit, ale není zdaleka jediným systémem s takovým počtem a kvalitou rozšíření.

6.4 Správa uživatelů, rolí a oprávnění

Protože je redakční systém od počátku považován za řešení problému víceuživatelské správy obsahu, je nutné mít pohodlnou správu uživatelských účtů a možnost specifikovat, co který uživatel může a nemůže v systému dělat. Podíváme se tedy, jaké možnosti nám vybrané systémy nabízí.

6.4.1 Drupal

Drupal nabízí možnost definování vlastních uživatelských rolí, do nichž mohou být uživatelé přiřazeni. Každý uživatel může patřit do několika rolí, do jedné je přiřazen automaticky – podle toho, zda je přihlášený (*authenticated*) nebo nepřihlášený (*anonymous*). Pro každý modul se dá označit, které akce (např. mazání příspěvků v blogu) mohou dané role vykonávat.

6.4.2 Joomla!

Joomla! rozděluje oprávnění do uživatelských skupin (*user groups*), přičemž odlišuje administracní rozhraní (*back-end*) od živých webových stránek (*front-end*) – je tedy možné uživateli povolit přidávání příspěvků do blogu rovnou z hlavní webové stránky a zároveň mu odeprít přístup do administracní sekce. Pro *front-end* jsou dostupné skupiny *Registered*, *Author*, *Editor* a *Publisher*, pro *back-end* potom *Manager*, *Administrator* a *Super Administrator* (seřazeny od nejméně oprávněných).

Podrobný popis lze nalézt na [50]. Důležité je, že není možné přidávat a odebírat vlastní role, dokonce ani měnit jejich oprávnění. To sice řeší nějaké moduly, ale většinou nejsou dostupné zdarma. Nová verze Joomla! 1.6 by měla obsahovat právě lepší správu uživatelských skupin a oprávnění.

6.4.3 WordPress

Ve WordPressu je pět základních uživatelských rolí (*roles*) – *Administrator*, *Editor*, *Author*, *Contributor* a *Subscriber*, které se liší svojí způsobilostí (*capabilities*). Každá role má přiřazené konkrétní způsobilosti (od instalace pluginů a změn vzhledu až po čtení obsahu), přičemž jsou zařazené do různých úrovní (*level* 1-10). Pokud by nevyhovovala původní sada těchto oprávnění, jejichž kompletní popis lze najít na [51], existují pluginy, které umožní definovat vlastní role a jejich způsobilosti, např. *RoleManager* nebo *CapabilityManager*.

6.4.4 Závěr

Joomla! v této oblasti za WordPressem a Drupalem hodně zaostala, i když by to následující verze měla řešit. V Drupalu je možné úroveň oprávnění vyladit do nejmenších detailů, akorát se po instalování nového modulu nesmí zapomenout na to, že se ta oprávnění musí nastavit. Velice mě pobavila hláška Drupalu „*Chcete-li roli smazat, zvolte upravit*“, ale to už je jiná kapitola. WordPress nabízí poměrně dobrý způsob řešení uživatelských oprávnění a pluginy na něm postavené slouží velice dobře.

6.5 Design – šablony, motivy vzhledu

Jakmile se se systémem seznámíme trochu blíže, vyzkoušíme si publikovat první článek a uvidíme první výsledky snažení, bude nás okamžitě zajímat, jak je možné změnit vzhled a rozmístění prvků. K tomu slouží motivy vzhledu (*themes*) a/nebo šablony (*templates*). Podíváme se tedy na to, jak se vytvářejí a používají v jednotlivých systémech. Vždy platí pravidlo, že jako začátečník je lepší začít úpravami stávajících motivů a šablon, než vytvářet svoje vlastní úplně od začátku.

6.5.1 Drupal

Motivy vzhledu se v Drupalu skládají z několika šablonových souborů, přičemž jejich název odpovídá jeho účelu – např. `page.tpl.php` nebo `comment.tpl.php`. Samozřejmě nesmí chybět hlavní stylový soubor `style.css` a informace o motivu v souboru `<nazev>.info`, popř. jeho obrázkový náhled. Celá složka s motivem vzhledu se kopíruje do `~/sites/all/themes` a poté se dá v administraci povolit a nastavit.

6.5.2 Joomla!

Šablony se do Joomla! instalují přes administračním rozhraní. Balíček obsahuje složky s obrázky a stylovými předpisy, popis ve formátu XML a náhled výsledného vzhledu, soubor `index.php`, ve kterém je hlavní kód (směsice (X)HTML, PHP a speciálních výrazů `<jdoc:include />` [52]). Je také možné používat parametry v souboru `params.ini`, například pro definování šířky nebo barev. Vše se potom dá upravit přímo v administračním rozhraní a každý uživatel může používat jinou šablonu, nebo její nastavení. Pomocí šablony je možné upravit i zobrazení jednotlivých modulů a komponent.

6.5.3 WordPress

Většina uživatelů WordPressu si motiv stáhne, aktivuje a nemá potřebu jej ručně upravovat. Narozdíl od možnosti automatického vyhledávání a instalace pluginů přímo v administračním rozhraní, je potřeba si zvolený motiv stáhnout a ručně nakopírovat na server (do `~/wp-content/themes`). V administračním rozhraní si lze z dostupných motivů vybrat a také jej přímo editovat.

Každý motiv se skládá z několika šablonových souborů (např. `page.php`, `index.php`, atd.), souboru `readme.txt` a stylového předpisu `style.css`. Výsledný kód se generuje na základě hierarchie jednotlivých souborů – nejdříve se zjistí, zda je požadována například domovská stránka, kategorie příspěvků nebo konkrétní příspěvek a poté je procházena struktura šablon od těch nejspecializovanějších po ty nejobecnější. Podrobně je to popsáno zde: [53].

6.5.4 Závěr

Vytváření šablon a motivů vzhledu není složité, ale ne každý je to schopen zvládnout. Nemělo by se zapomínat na lokalizaci, validitu kódu a použitelnost ve všech prohlížečích. Navíc není jednoduché zaujmout barvami, nabídnout možnosti úprav a v neposlední řadě být připraven na použití bloků/modulů/widgetů. Kvalitní motiv vzhledu se tak hledá těžko a mnohdy je lepší si takový zakoupit od profesionálů, kteří se tím opravdu zabývají, než ztratit spoustu času úpravami nevyhovujícího či vývojem vlastního. Když přijde na srovnávání šablon Drupalu a Joomla, většinou má druhý jmenovaný navrch.

6.6 Vytváření struktury webu

První, co si zákazník (resp. ten, pro něhož se webové stránky vytvářejí) musí rozhodnout, je struktura webu. To znamená, že musí vědět, jaké sekce na stránkách chce mít a co budou obsahovat. Je to dáno do jisté míry tím, pro koho jsou stránky určeny – jaká cílová skupina lidí by měla tyto stránky navštěvovat. Struktura webu je vždy jeho základem, ať už je jeho vzhled jakýkoli.

6.6.1 Drupal

Drupal nabízí po základní instalaci tři nabídky (menu). Jsou to hlavní odkazy (*primary links*), vedlejší odkazy (*secondary links*) a navigace (*navigation*). Hlavní odkazy většinou bývají používány šablonou pro zobrazení hlavního menu (jednotlivých sekcí webu) a někdy jsou využity i vedlejší odkazy (například pro zobrazení odkazů v patičce). Navigace je používána především pro přihlášené uživatele a umožňuje jim základní správu webu. V Drupalu je možné vytvořit libovolný počet nabídek, a tak jsme jako vývojáři omezeni pouze vlastní fantazií. Do menu lze přidávat odkazy na stránky, kategorie i štítky.

6.6.2 Joomla!

V Joomla lze vytvářet také neomezený počet nabídek, ale samotný proces je možná uživatelsky přívětivější než v Drupalu – je možné si vybrat, jakého typu daná položka bude, kdo k ní má přístup a zda se má zobrazit. Při vytváření obsahu je nutné, aby byly předem nadefinovány sekce a kategorie, do nichž obsah patří. Nelze je totiž vytvářet současně s obsahem a je potřeba na to pamatovat.

6.6.3 WordPress

WordPress po základní instalaci nenabízí žádnou možnost vytvoření menu, aniž by se nezasáhlo do šablony nebo nepoužilo nějakého pluginu. Zřejmě to pramení z jeho minulosti a zaměření především na blogy, které jsou organizovány podle data a zařazovány do jednotlivých kategorií, případně označeny štítky. Statické stránky lze strukturovat, stejně jako lze zanořovat i kategorie, ale o jejich zobrazení se musíme postarat v šabloně, což není ani přes podrobné a polopatické vysvětlení uživatelsky nejpřívětivější.

6.6.4 Závěr

Přirozenou a pohodlnou správu nabídek, sekcí a kategorií nabízí Joomla!, přičemž Drupal je v téhle oblasti také velice flexibilní. Při vytváření složitějších nabídek je ve WordPressu

nutné použít nějaké rozšíření a/nebo zasáhnout do kódu šablon, což by mohlo některé uživatele odradit.

6.7 Zálohování a obnova dat

Velice důležitou součástí provozu webových stránek je jejich zálohování. Ztráta dat je vždy nepříjemná záležitost a v některých případech by mohla být i kritická. Je zřejmé, že u jednoduchých statických stránek, které se prakticky nemění, není potřeba zálohovat tolik, jako v případě elektronických obchodů a dynamicky se tvořících webů, což je právě náš případ. Protože máme oddělená data (v databázi, často na jiném serveru) od aplikační části, je potřeba zálohovat obojí. Sice je pravdou, že se o pravidelné zálohování postará každý lepší webhosting, ale stejně bychom to v některých situacích mohli využít. Srovnávané systémy takové zálohování řeší až pomocí rozšíření, kterých má každý systém několik, z nichž se dá vybrat.

6.7.1 Drupal

Uvedu například modul *Backup and Migrate*, který umožní výběr tabulek databáze, které mají být zálohovány, zda se mají zálohovat i s daty (například u *cache* to rozhodně není nutné), po kolika hodinách se má zálohovat a kolik posledních záloh se má uchovávat. Z těchto záloh je možné databázi obnovit. Co se týče zálohování fyzických souborů, je potřeba zálohovat adresář `~/sites/`, kde jsou uloženy rozšiřující moduly, motivy vzhledu a další soubory, které nejsou součástí jádra.

6.7.2 Joomla!

Z komponent, které zálohování Joomla dokážou zpříjemnit, jsem nakonec vybral *Joomla-Pack*. Umožní zálohovat nejen databázi, ale také celý projekt – vše zabalí do instalačního balíčku, pomocí kterého lze všechna data pohodlně obnovit. Hodí se to například při vývoji webu na lokální počítači a jeho nasazení na produkční server. Nabízí se také automatické zálohování a různá nastavení toho, co všechno se má do zálohy přidávat, popř. jak se mají data komprimovat.

6.7.3 WordPress

Pro WordPress jsem vybral *WP-DBManager*, který neslouží jen k zálohování a obnově, ale je možné provádět i dotazy nad databází a optimalizovat tabulky. Umožňuje naplánování automatických záloh a je možné je posílat i e-mailem. I když nabízí tolik možností, v praxi se to určitě moc používat nebude. Pokud jde o fyzické soubory, je lepší je zálohovat všechny.

6.7.4 Závěr

Zálohování je vždy potřeba nějak řešit, ale v některých případech (záleží na hostingu) budeme stejně nuceni manuálně kopírovat soubory z webového serveru a exportovat databázi. Tam, kde je možné zálohy pohodlně vytvářet, je *JoomlaPack* zřejmě nejlepším řešením.

6.8 Aktualizace systému a rozšíření

Po analýze, návrhu, implementaci, ladění, testování a nasazení do provozu se dostáváme do neméně důležité fáze životního cyklu naší webové aplikace – udržování jí v aktuálním stavu. Jednak je potřeba občas stránky občerstvovat tak, aby šly s dobou – nové verze obsahují spoustu nových funkcí a možností, ale nedílnou součástí všech aktualizací je opravení nalezených chyb a hlavně těch bezpečnostních.

Musíme si připustit, že každý přidaný modul, každá nová vlastnost, každá úprava kódu obsahuje potenciální bezpečnostní riziko. Aktualizací modulů na jejich nejnovější stabilní verzi si zajišťujeme vyšší bezpečnost. Nikdy nebudeme mít 100% jistotu, že tam žádná díra není, ale budeme mít přehled o tom, které jsou již zaceleny. Pro útočníka není žádný problém si zjistit seznam stránek, které byly vytvořeny starší verzí CMS obsahující bezpečnostní díru a tuto zneužít. Bezpečnostní chyby jsou neustále hlášeny a ve všech projektech opravovány s vyšší prioritou.

Před každou aktualizací je silně doporučováno zálohovat projekt i databázi a také si zjistit, zda budou všechna rozšíření fungovat i pod novou verzí, mohli bychom se totiž dočkat nemilých překvapení (hlavně u Drupalu). Zároveň je nutné pozorně prozkoumat pokyny k její instalaci.

6.8.1 Závěr

Automatická aktualizace systému i jeho rozšíření je prozatím výsadou WordPressu, Drupal i Joomla! se musí aktualizovat ručním přepsáním zdrojových souborů v adresáři projektu. Nemohu se nezmínit o tom, že před aktualizací Drupalu je nutné zakázat všechny uživatelské moduly a po každé aktualizaci přistoupit na adresu `~/update.php` a tam potvrdit aktualizaci tabulek databáze – tyto dvě podmínky aktualizace nejsou vůbec uživatelsky přívětivé, zvláště když si vezmeme v potaz, že použitých modulů jsou mnohdy desítky. Drupal neumí zakázat a znovu povolit uživatelské moduly sám od sebe, je tedy nutné kliknout na políčko u každého modulu, kterým se zakáže, poznamenat si moduly, které byly takto zakázány a po provedení aktualizace opět kliknout na každý modul a znovu jej aktivovat. Můžeme jen doufat, že se tento proces v dalších verzích automatizuje, nebo se vytvoří odpovídající modul.

6.9 Lokalizace, podpora češtiny

Lokalizaci můžeme rozdělit na dvě části – lokalizaci rozhraní CMS (určená pro vývojáře, editory) a lokalizaci webových stránek (určená pro návštěvníky). Vždy záleží na tom, komu jsou webové stránky určeny. Pokud si člověk vede osobní blog s fotogalerií, zřejmě nebude potřebovat mít přeložené stránky do čtyř světových jazyků, ale na druhou stranu by možná uvítal, pokud by bylo rozhraní CMS v jeho rodném jazyce. V případě horského hotelu, kde se očekává především návštěva ze zahraničí, je naopak lokalizace stránek velice důležitá. Na lokalizaci je dobré myslet i tam, kde se počítá se spoluprací lidí z celého světa – zde se hodí například podpora časových pásem (*timezones*).

V oblasti vícejazyčné podpory softwaru se setkáváme se dvěma pojmy – internacionalizace (*internationalization* – *i18n*) a lokalizace (*localization* – *L10n*), přičemž *i18n* znamená podporu softwaru pro cizí jazyky a *L10n* znamená samotné rozšíření internacionalizovaného programu jazykovou verzí podle místních zvyklostí.

Podíváme se tedy na to, jak jsou vybrané systémy použitelné pro vícejazyčné stránky (multilingual web pages), v jakém stavu se nachází český překlad jádra i modulů a jakým způsobem se vlastně lokalizuje. [54].

6.9.1 Drupal

Drupal lze lokalizovat pomocí *.po souborů, což je velice často používaný formát nástroje GNU gettext [55]. Samotná lokalizace se provádí prostým nainportováním lokalizačního souboru přes administrační rozhraní nebo jeho nakopírováním do příslušné složky (moduly mívají složku `translations`, kam se tyto soubory umísťují). Nejdříve je ovšem potřeba mít povolený modul *Locale* a v *Nastavení webu* -> *Jazyky* přidat pro tento jazyk podporu.

Drupal 6 nabízí možnost vytváření jazykových verzí obsahu už přímo v jádře a další rozšiřující moduly jsou k dispozici [56].

6.9.2 Joomla!

Způsobů lokalizace systému Joomla! je více a každá komponenta může tuto problematiku řešit jinak. Instalace jazykového balíčku pro základní uživatelské rozhraní komponenty, pluginy a moduly může být provedena pomocí instalátoru ve správci rozšíření (lokalizaci tvoří textové *.ini soubory), nebo se může jednat o jazykový patch, který přímo přepíše zdrojové soubory dané komponenty a/nebo přidá nové *.php soubory s definovanými konstantami. Joomla! umožňuje mít lokalizované uživatelské rozhraní pro konkrétní uživatele a zjednodušuje tak spolupráci mezinárodních týmů.

Pro překlad obsahu a přepínání jazykových verzí nabízí celou řadu více či méně použitelných rozšíření a po jejich správném nastavení je možné vícejazyčné weby vytvářet.

6.9.3 WordPress

WordPress se dá stejně jako Drupal lokalizovat pomocí gettext souborů *.po, ale využívá také jejich zkompileovaných podob (soubory s příponou *.mo), které umožňují mnohem rychlejší zpracování [57]. Hlavní překlad rozhraní je nutné zkopírovat do složky jazyků `~/wp-content/languages/` a překlady rozšíření se většinou nacházejí přímo v adresáři pluginů ve formě `<nazev pluginu>-<kod jazyka>.mo`, kód jazyka je např. `cs_CZ`.

Co se týče jazykových verzí obsahu, lze toho dosáhnout pomocí pluginů a neměl by být problém vytvořit vícejazyčný web.

6.9.4 Závěr

S lokalizací se potýkají všechny vybrané systémy přibližně na stejné úrovni. Drupal přidal podporu přímo do jádra, ale stejně je nutné použít další rozšíření jako u ostatních systémů. Ve všech systémech je proces lokalizace obsahu poměrně složitý, zejména ze začátku je potřeba tomu věnovat spoustu času. Na druhou stranu není nemožné vytvářet a spravovat vícejazyčné weby (po tom, co si člověk zvykne na ten který přístup).

Pro všechny vybrané systémy je k dispozici český lokalizační balíček a lepší rozšíření také bývají přeložená. Z pohledu tvůrce překladů rozhraní je zřejmě lepším řešením použití nástrojů GNU gettext, překlady Drupalu a WordPressu se vytvářejí mnohem snadněji.

6.10 Uživatelské rozhraní

Uživatelské rozhraní je jednou z nejdůležitějších částí redakčního systému. Pokud se v něm uživatel, editor, správce nebo vývojář nevyzná, nejspíš se poohlédne po systému jiném. Problematika uživatelské přívětivosti (*user friendliness*) a použitelnosti (*usability*) webových aplikací je natolik rozsáhlá, že by vydala na vlastní práci, proto se jí budeme zabývat jen z části a v omezeném rozsahu. Vždy záleží na tom, kdo systém ovládá, jaké má znalosti a zkušenosti, ale existují určité konvence, které by se měly dodržovat. Podíváme se tedy na to, jak k uživatelské přívětivosti vývojáři vybraných systémů přihlížejí a jak to působí na uživatele.

6.10.1 Drupal

Co se týče uživatelského rozhraní, Drupal již od počátku čelí veliké kritice nových uživatelů. Nemá totiž žádné administrační rozhraní a s jeho absencí se někteří uživatelé nechtějí smířit (a většinou do tajů Drupalu nikdy neproniknou). Je dost pravděpodobné, že původní koncepce zůstane i nadále stejná (žádné oddělené administrační rozhraní nebude potřeba), ale na použitelnosti se pracuje – bylo provedeno několik průzkumů i testů použitelnosti (např. [58]) a po vyhodnocení výsledků se pro novou verzi (Drupal 7) připravují znatelné změny.

Prozatím je velice užitečný modul *Administration Menu*, který přidá horní lištu s rozbalovací nabídkou, pomocí které se dá poměrně rychle dostat k požadovanému prvku rozhraní. Je také nutné pochválit možnost *drag-and-drop* umístění bloků do regionů na stránce.

6.10.2 Joomla!

Joomla! si na svém administračním rozhraní zakládá a nutno říci, že je opravdu přehledné. Na první pohled je příjemné barvami, ikonkami, a když si uživatelé stěžují na některé vlastnosti tohoto systému, administrační rozhraní je kritizováno velice zřídka. Určitě má ještě rezervy a dá se vylepšovat, ale je na čem stavět.

6.10.3 WordPress

WordPress má celé uživatelské rozhraní postavené na AJAXu a pro vedení vlastního blogu snad nemůže být nic lepšího (jednoduššího, přehlednějšího, efektivnějšího). Dá se upravit podle vlastních potřeb prostým *drag-and-drop* způsobem – je možné si tedy přeskládat jednotlivé bloky nástěnky (*dashboard*) podle toho, jak je bude chtít uživatel využívat.

6.10.4 Závěr

Zkušenosti uživatelé Drupalu většinou dojdou k názoru, že „*si na omalovánky vlastně moc nepotrpí, dokážou se obejít i bez toho*“ a rychle stočí řeč na výhody, zatímco uživatelé Joomla! se budou i nadále vyžívat v kráse jejich uživatelského rozhraní, přičemž uživatelé WordPressu stojí v povzdálí a mnou si ruce nad jeho jednoduchostí.

6.11 Optimalizace pro internetové vyhledávače

O tom, jak bude webová stránka nakonec (ne)úspěšná, rozhoduje mnoho faktorů. Naším cílem je, aby naše stránky návštěvníci/zákazníci našli, na stránkách zůstali a nakonec se

i vraceli. Jedním z faktorů je tedy pozice ve vyhledávačích – když zákazník hledá nějaké informace, většinou zadá klíčová slova do vyhledávače a klikne na jeden z prvních třech odkazů na vyhledané weby (nebo si otevře všechny tři a začne na stránce, která se načte nejrychleji). Celou první stránku prochází tehdy, když zadal obecnější pojem a první odkazy ho neuspokojí. Na další stránky se pak podívá minimum vyhledávajících – to raději upřesní jejich dotaz nebo použije jiný vyhledávač. Jak se tedy dostat na první místa? Tímto procesem se zabývá mnoho společností, obecně se to nazývá SEO (*search engine optimization*). Buď můžeme nějaké firmě zaplatit nemalé peníze, anebo o SEO zkusíme postarat sami. Výsledkem by měly být stránky, které jsou SEF (*search engine friendly*).

Základní doporučení, co dělat a čemu se vyhnout, nejen od společnosti Google [59], jsou:

- Dodržování standardů W3C (validita kódu (X)HTML)
- Oddělení obsahu od grafiky a designu (použití CSS a beztabulkového layoutu)
- Logické strukturování obsahu, použití výstižných nadpisů
- Jedinečný a smysluplný popis každé stránky (*meta tag description*)
- Smysluplné popisy odkazů a obrázků
- Použití čitelných SEF URL adres (také nazýváno *Clean/Cool/čisté/pěkné URL*)
- Použití mapy stránek – zobrazení ve formátu odkazů na jednotlivé stránky a také ve formátu XML přímo pro vyhledávače (soubor `sitemap.xml`)
- Použití souboru `robots.txt` pro komunikaci s prohledávacími roboty (*crawler*)
- Zaregistrování do různých katalogů pro získání zpětných odkazů (počet zpětných odkazů je jednou z nejdůležitějších částí hodnocení důležitosti webové stránky)

Ze seznamu je vidět, že všechno za nás CMS neudělá, ale může nám významně pomoci. Nejviditelnějším vylepšením bývá použití SEF adres (tyto adresy neobsahují „žádné otazníky“ a je možné z nich vyčíst určitou strukturu – například pro květnové příspěvky do blogu by mohla být adresa `http://jmeno.cz/blog/2009/05/`). Právě proto, aby mohly být takové adresy používány, je většinou nutné mít povolený zápis do souboru `~/htaccess`.

Kromě dodržování obecných zásad je také velice důležité mít kvalitní obsah. Například u internetového obchodu to může být rozhodujícím faktorem, a proto většina společností zabývajících se SEO nabízí také *copywriting* (vytváření textů webových stránek s optimální hustotou klíčových slov pro vyhledávače i zákazníky).

6.11.1 Závěr

Všechny systémy jsou v oblasti SEO na zhruba stejné úrovni. Když ne v základní výbavě, tak to dohání jejich rozšíření, ať už jsou dostupná zdarma nebo za peníze. Vždy záleží na tom, kdo web vytváří a nakolik dodržuje obecné zásady.

6.12 Testování a ladění webových aplikací

Toto téma je také poměrně široké, ale pro srovnání redakčních systémů využijeme jenom malou část, protože většina není na vybraném systému vůbec závislá a záleží na autorovi stránek. Přesto si zde uvedeme, na co bychom se měli při vývoji webových aplikací zaměřit:

- Dodržování dostupných standardů (validní (X)HTML, CSS)
- Dodržování pravidel přístupnosti (accessibility [60], [61], [62])
- Dodržování doporučení pro SEO
- Bezpečnost systému (odolnost vůči *SQL Injection*, *XSS* útokům), přístup k souborům
- Funkčnost všech odkazů
- Testování použitelnosti
- Rychlost načítání stránek, velikost stránek a přenesených dat, počet dotazů na databázi
- Výkonnostní testování (*performance testing*)
- Zátěžové testování (*load testing*)
- Stresové testování (*stress testing*)

Proces testování by se neměl zanedbávat a měl by provázet vývoj od jeho začátku a ne až po jeho dokončení, jak se to bohužel nezdědka stává. Ještě pamatujme na důležitou věc: testovací strategie by měla být vždy řízená riziky – k čemu je bance umístění na prvních pozicích vyhledávačů se skvělou použitelností a přístupností za desetitisíce korun, když kvůli SQL injection může přijít o miliony.

6.12.1 Testování bezpečnosti - SQL Injection, XSS

Úplně nejzákladnějšími úkoly testování bezpečnosti je odhalit možné bezpečnostní díry. Při vývoji by se mělo zajistit, aby se nedaly zneužít uživatelské vstupy (většinou formuláře, parametry adresy) k ničemu jinému, než k čemu jsou určeny. Toto zajišťování není nijak extrémně náročné, avšak v dřívějších dobách se na něj zapomínalo.

SQL Injection [63] je útok na databázi, kdy se útočníkovi podaří vložit do databázového dotazu další dotaz – většinou slouží k odcizení citlivých údajů z databáze nebo ke smazání dat. Bránit se mu dá správným zpracováním vstupních údajů a parametrizováním SQL dotazů.

XSS (*Cross-site scripting*) [64] je útok, kdy útočník podstrčí vlastní kód (nejčastěji Javascript) webové stránce, která jej interpretuje. Může tak dojít k úpravě vzhledu i obsahu stránek, popřípadě jejich přesměrování.

6.12.2 Dodržování standardů a doporučení

Nejen v oblasti webových stránek je dobré se držet nějakých standardů a doporučení. Je to jedna z věcí, která je prospěšná, stojí za to se o ni zajímat a dá se testovat. Pro platné standardy konsorcia W3C existují validátory ([65], [66], které jejich dodržování ověří. Poté je dobré se zaměřit na přístupnost, čili schopnost poskytnout informace i handicapovaným lidem. Můžeme si sice říct, že nemá smysl vynakládat finanční a časové úsilí jenom pro to, aby k nám na web chodili i handicapovaní lidé, ale když si uvědomíme, že za vysoce handicapovaného může být považován i prohledávací robot, problematika získává nový rozměr. Navíc státní instituce a vládní organizace se těmito pravidly řídit musí, dokonce podle zákona (v ČR je to vyhláška č. 64/2008 Sb.). Můžeme se také setkat s označením *bezbariérový web*. V USA se tím zabývají webové stránky [61], v ČR pak například [60]. Existují i automatické validátory [67], ale mnohá pravidla se automaticky otestovat nedají.

6.12.3 Rychlost načítání stránek

Čím rychlejší bude načítání stránek, tím lepší pocit z procházení bude náš návštěvník mít. Kdo v dnešní uspěchané době rád čeká, že? Doba načtení je ovlivněna několika faktory:

- Rychlost internetového připojení serveru (a jeho momentální zatížení)
- Rychlost spojení s databázovým serverem (a jeho momentální zatížení)
- Složitost generujícího kódu (PHP), počet dotazů na databázi
- Počet a velikost externích skriptů (javascript) a obrázků
- Velikost stahované stránky
- Rychlost internetového připojení uživatele (propustnost sítě)

Některé z těchto faktorů ovlivnit nemůžeme, ale například u snižování velikosti používaných obrázků a celkové velikosti stránky můžeme začít. Jejich optimální velikost se dá určit podle cílové skupiny návštěvníků. Pokud se nacházíme v oblasti, kde bude naše stránky navštěvovat 90 % lidí s vysokorychlostním připojením, nebude to pro nás tak důležité, jako v případě vytáčeného připojení (rychlost max. 56 kb/s) nebo omezeného mobilního připojení (kvůli množství přenesených dat). Dříve platilo, že návštěvník vydržel s vytáčeným připojením čekat na načtení stránky i 10-15 sekund a toho se zhruba držíme i dnes. Pokud by měla doba načítání naší stránky při pomalém připojení překročit 15 vteřin, je optimalizace nutná. Rozumná velikost stránky by mohla být například do 20 kB. Podrobný rozbor nabízí například služba [\[68\]](#) nebo [\[69\]](#).

6.12.4 Výkonnostní, zátěžové a stresové testování

Tyto tři pojmy (angl. *performance, load and stress testing*) bývají často zaměňovány, přitom každý z nich označuje něco jiného. Jsou to tři různé metody testování a jsou od nich očekávány jiné typy výsledků.

Výkonnostní testování – performance testing

Cílem výkonnostního testování je otestovat, zda systém vyhovuje základním požadavkům na výkon – například stránka se musí při připojení 1Mbps načíst do 2 vteřin, proces přihlášení nesmí trvat déle než 10 vteřin.

Zátěžové testování – load testing

Zátěžovým testováním se zjišťuje, jak se systém chová při určité zátěži a stanoví se její garantované maximum. Zátěž se během testu průběžně zvyšuje, až se dojde k mezní hodnotě, za kterou už systém nesplňuje naše požadavky. Takovým testem může být například zvyšování počtu konkurenčních připojení na stránku a měření její odezvy nebo zvyšování počtu uživatelů v databázi a zjišťování, kolik jich tam může být, aby jejich vyhledání podle jména netrvalo déle než 10 vteřin. Zátěžové testování by mělo simulovat reálný provoz (např. scénář, kdy se uživatel přihlásí, prohlédne si fórum a zase se odhlásí).

Stresové testování – stress testing

Stresové testování se provádí ke zjištění chování systému v neobvyklých podmínkách. Neobvyklými podmínkami můžeme rozumět například odepření systémových prostředků (procesorový čas, velikost paměti, atp.) nebo simulaci různých výpadků (sítě, databáze, disku). Toto chování se pak může upravovat například upozorněním uživatele, že něco není v pořádku a služba proto trvá déle.

Nástroje pro testování webových aplikací

Nástrojů pro výkonnostní, zátěžové a stresové testování existuje celá řada. Většinou se jedná o poměrně drahou záležitost (v řádech tisíců až desetitisíců amerických dolarů), ale nabízí se také bezplatné varianty [70]. Malý přehled produktů:

- IBM Rational
- Microsoft Visual Studio Team System 2008 Test Edition (+ Load Test Agents, Load Test Controller)
- HP Performance Center (LoadRunner)
- Jakarta JMeter

6.12.5 Použití JMeteru pro srovnání CMS

JMeter je jedním z nejoblíbenějších open source nástrojů pro výkonnostní a zátěžové testování. Jeho použití není složité a umožňuje použití široké škály protokolů. Navíc je možné jej dále rozšiřovat pomocí pluginů. Pro testování webových aplikací se hodí nastavení typu požadavků (GET/POST) například pro přihlašování do systému, možnost zvolení počtu uživatelů, kteří na stránky přistupují. Naměřené výsledky (počet vyřízených požadavků za sekundu (*requests per second* – *RPS*), doba odpovědi (*response time* – *RT*), vytíženost sítě, atp.) mohou být zobrazeny formou tabulek i grafů.

Už z prvních měření bylo bohužel jasné, že není možné jednotlivé systémy mezi sebou porovnávat, resp. není možné je porovnávat bez podrobnějších znalostí optimalizace jejich výkonu a s nedostatkem hardwarových zdrojů. Takové srovnání by bylo nutné provádět za asistence odborníků na optimalizace kódu, databáze, nastavení serverů a nejlépe mít k dispozici reálná data.

V tabulce 6.1 jsou uvedeny některé naměřené hodnoty. Kromě již zmíněných veličin je použito označení SQL pro počet dotazů na databázi. Optimalizací je myšleno použití dostupných nástrojů pro zapnutí cache, případně komprimování obsahu (javascriptu, css), které systémy nebo jejich rozšíření nabízejí. Nastavení výkonu je v Drupalu součástí jádra, u Joomla byl použit zásuvný modul *System-Cache* a pro Wordpress byl vybrán plugin *WP Super Cache*.

Naměřené hodnoty a grafy jsou bohužel zkrácené a prakticky nevypovídající – když jeden počítač generuje zátěž a zároveň slouží jako webový i databázový server, není dost dobře možné sledovat vytížení jednotlivých částí. Navíc je výkonnost silně ovlivněna používanými rozšířeními – i jeden maličký modul dokáže zabrzdit celý server.

| Hlavní stránka (30 kB) | Bez optimalizací | | | S optimalizací | |
|---------------------------|------------------|---------|-----|----------------|---------|
| | RPS | RT [ms] | SQL | RPS | RT [ms] |
| Drupal | 3,75 | 1450 | 170 | 24,60 | 40 |
| Joomla! | 1,35 | 750 | 16 | 3,50 | 300 |
| Wordpress | 0,80 | 1290 | 87 | 142 | 6 |

Tabulka 6.1: Výsledky měření pomocí JMeteru

Závěr

Optimalizace, která je pro koncového uživatele nejdostupnější, je nastavení cacheování. Mělo by se cacheovat, co se jen dá. Tam, kde se obsah příliš často nemění, je to zvláště výhodné. Dalším optimalizačním tipem je komprimování obsahu (především javascriptových souborů). Co se týče výkonu v celosvětovém měřítku, je většinou upřednostňován Drupal před Joomla. Lze totiž poměrně dobře ladit, a když vývojář dokáže omezit počet dotazů na databázi, bývá rychlejší a spolehlivější. Mnoho lidí přechází k Drupalu právě díky jeho výkonu, i když jsem neviděl žádná smysluplná srovnání, která by toto tvrzení podložila. Podobných výsledků se dá určitě dosáhnout i v ostatních systémech – blogy Wordpressu zdatně odolávají stejným (ne-li větším) náporům uživatelů. U Joomla se lidé raději uchylují k placeným řešením. Podrobného srovnání výkonnosti těchto systémů se nejspíš jen tak nedočkáme – jeho testování totiž není nejlevnější záležitostí, vyžaduje spoustu znalostí, zkušeností a času.

6.13 Rozšíření systémů pro dosažení funkčnosti

V tabulce 6.2 jsou uvedena rozšíření, která byla použita pro vytvoření testovacích webů. Kdybychom dali stejné zadání deseti různým vývojářům, pravděpodobně bychom od nich dostali deset různých seznamů rozšíření. Vždy záleží na tom, jaké má vývojář zkušenosti, odkud čerpá reference, jak se mu s danými rozšířeními pracuje a kolik je za ně ochoten zaplatit.

| | Drupal | | Joomla! | | WordPress | |
|-----------|-----------------|---------------|-------------------|----------------------|-----------------------|------------------|
| | Použité | Další | Použité | Další | Použité | Další |
| Anketa | Poll (jádro) | | Polls(jádro) | | WP-Polls | |
| Akce | Event | Calendar | EventList | JEvents, JCal | Events Manager | |
| Blogy | Blog (jádro) | | IDoBlog | SmartBlog, Wordpress | jádro | |
| Články | jádro, Taxonomy | | jádro | | jádro | |
| E-shop | Ubercart | osCommerce | VirtueMart | | WP e-Commerce | |
| Fórum | Advanced forum | Forum (jádro) | Kunena | | WP-Forum | |
| Menu | jádro | | jádro | | WP-MenuManager | Breadcrumb NavXT |
| Uživatelé | jádro, Profile | | Community Builder | | jádro, User Community | |
| Celkem | 34 | | 17 | | 17 | |

Tabulka 6.2: Rozšíření systémů pro dosažení funkčnosti

Číslo uvedené v závorce označuje počet tabulek, instalováno bez testovacích dat.

Existuje několik předpřipravených řešení, které firmy prodávají (většinou se jedná o instalaci, nasazení a následnou podporu). Jedním příkladem za všechny budiž Joomla! + VirtueMart. Je to poměrně ustálené slovní spojení v oblasti e-shopů a dá se využít i u nás. Na druhou stranu spousta e-shopů je postavena na Drupalu a ani WordPress nezůstává pozadu.

| | Drupal | Joomla! | WordPress |
|---|-------------|---------------|-------------|
| Aktuální verze | 6.11 | 1.5.10 | 2.71 |
| Velikost instalačního balíčku (zip) | 1,2 MB | 5,8 MB | 1,8 MB |
| Velikost balíčku finálního projektu (zip) | 11,9 MB | 13,6 MB | 10,1 MB |
| Velikost „čisté“ DB* | 412 kB(46) | 96 kB (36) | 467 kB (10) |
| Velikost DB finálního projektu | 15 MB | 1,3 MB | 1,4 MB |
| Celkový počet složek v projektu | 1262 | 2424 | 245 |
| Celkový počet souborů v projektu | 6098(60 MB) | 16846(135 MB) | 2556(23 MB) |

Tabulka 6.3: Porovnání velikosti instalace

6.13.1 Závěr

Drupal má zřejmě nejširší základnu uživatelských rozšíření, i když jejich počet (4-5 tisíc) rychle dohání Joomla!, která ovšem narozdíl od Drupalu více dodržuje zpětnou kompatibilitu. WordPress si získává sympatie svým pohodlnějším rozhraním pro vyhledávání a instalaci pluginů, ale stává se, že není moc z čeho vybírat.

Kapitola 7

Závěr

Pohledů na webové aplikace, redakční systémy a tvorbu webových stránek je samozřejmě více, ale z hlediska rozsahu bakalářské práce není možné se všemi zabývat a už vůbec ne dopodrobna. Tato práce je na první pohled zaměřená teoreticky, nikde nejde extrémně do hloubky – spíše čtenáře seznamuje, nicméně si dovolím tvrdit, že je plná praktických informací a zkušeností, ze kterých mohou čtenáři čerpat. Při vývoji a údržbě webových stránek pomocí redakčních systémů se dá zmíněným oblastem jen těžko vyhnout – dříve či později se jimi bude muset každý vývojář zabývat.

Je vždycky těžké rozhodování, jaký systém zvolit. Na vyzkoušení všech zřejmě není čas, i když by to bylo nejlepší řešení, a tak nezbyvá, než dát na doporučení ostatních. Po světě koluje spousta fám a kolikrát nezdůvodněných tvrzení, že ten či onen systém je zaručeně nejlepší nebo nejhorší, proto je nutné takto extrémní názory přijímat s rezervou, a přitom se na ně dívat očima jejich původce, abychom dokázali pochopit jeho situaci. Co je pro jednoho nepřekonatelný problém, může být pro druhého naprosto banální záležitost. Každý ocení něco jiného. Tak i já přidávám svou troškou do mlýna a pokusím se podat, jak na mě jednotlivé systémy působí.

Nejprve z pohledu uživatele, který vytváří web sám pro sebe a bude se o něj starat. Pokud je začátečník, chce si vytvořit osobní stránky a začít co nejrychleji, volí WordPress. Není problém si na stránky přidat malý e-shop nebo diskusní fórum. Když má zájem se učit novým věcem (např. vlastnoručně upravovat vzhled stránek), existuje pro něj celá řada polopatických návodů, jak se co dělá. Drupal i Joomla! by pro něj byly zřejmě zbytečně složité.

Dále tu máme pohled vývojáře, který má své zkušenosti s programováním, rád by měl vše pod kontrolou a jednou by se třeba mohl vývojem webů i živit. Vybírá si Drupal díky jeho vysoké modularitě a flexibilitě. Je si vědom toho, že by mohl jednodušeji postavit web i na Wordpressu nebo v Joomla, ale také ví, že když překoná počáteční zmatky v názvosloví a prostuduje základy, otevřou se mu dveře nekonečných možností a neměl by mít problémy s vytvořením jakéhokoli webu.

Nakonec je tu případ profesionálního vývojáře, který se webem již živí, nebo se jím hodlá živit. Zřejmě sáhne po Joomla, bude dodávat připravená řešení na míru, vyvíjet komponenty, šablony nebo poskytovat podporu a konkurovat tak ostatním. Samozřejmě ví, že by mohl to samé dělat i po zvolení Drupalu, ale Joomla! mu vyhovuje víc a díky přívětivějšímu uživatelskému rozhraní se i lépe prodává zákazníkům.

Literatura

- [1] Open Source Initiative [online]. <http://opensource.org/about>, 2009 [cit. 2009-05-10].
- [2] Definice svobodného software – GNU Projekt – Nadace pro svobodný software [online]. <http://www.gnu.org/philosophy/free-sw.cs.html>, 2008-04-26 [cit. 2009-05-10].
- [3] GNU Operating System [online]. <http://www.gnu.org/licenses/licenses.html>, 2009-03-24 [cit. 2009-05-10].
- [4] Open Source CMS Award Previous Winners [online]. <http://www.packtpub.com/open-source-cms-award-previous-winners>, 2009 [cit. 2009-01-12].
- [5] About Drupal [online]. <http://drupal.org/about>, 2009 [cit. 2009-02-25].
- [6] Suchý, J.: O systému Drupal [online]. <http://www.drupal.cz/o-systemu-drupal>, 2009 [cit. 2009-02-25].
- [7] About the Drupal Project [online]. <http://drupal.org/History-mission-and-community>, 2009-02-09 [cit. 2009-02-25].
- [8] Mission and values [online]. <http://drupal.org/mission>, 2008-11-01 [cit. 2009-02-25].
- [9] Featured drupal sites [online]. <http://www.drupalsites.net/>, 2009 [cit. 2009-05-16].
- [10] Joomla! [online]. <http://opensourcematters.org/joomla.html>, 2009 [cit. 2009-02-25].
- [11] Joomlaportal.cz [online]. <http://www.joomlaportal.cz/content/view/180/2/>, 2006-04-09 [cit. 2009-02-25].
- [12] OpenSourceMatters [online]. <http://opensourcematters.org/mission-vision-values.html>, 2009 [cit. 2009-02-25].
- [13] Joomla! Topsites [online]. <http://www.joomlapowersites.com/>, 2009 [cit. 2009-05-16].
- [14] Joomla! sites and templates [online]. <http://www.joomlapowersites.com/>, 2009 [cit. 2009-05-16].

- [15] WordPress [online]. <http://wordpress.org/about/>, 2009 [cit. 2009-02-25].
- [16] WordPress [online]. <http://wordpress.org/about/philosophy/>, 2009 [cit. 2009-02-25].
- [17] WordPress [online]. <http://wordpress.org/showcase/>, 2009 [cit. 2009-05-16].
- [18] WordPress [online]. <http://mu.wordpress.org/>, 2009 [cit. 2009-05-16].
- [19] BuddyPress [online]. <http://buddypress.org/>, 2009 [cit. 2009-05-16].
- [20] Naramore, E.; Gerner, J.; Scouarnec, Y. L.; aj.: *Vytváříme webové aplikace v PHP5, MySQL a Apache*. Computer Press, 2006, ISBN 80-251-1073-7.
- [21] PHP: Hypertext Preprocessor [online]. <http://www.php.net/>, 2009 [cit. 2009-05-17].
- [22] MySQL: The world's most popular open source database [online]. <http://www.mysql.com/>, 2009 [cit. 2009-05-17].
- [23] Netcraft LTD – Internet Research, Anti/Phishing and PCI Security Services [online]. <http://news.netcraft.com//>, 2009 [cit. 2009-05-17].
- [24] HTML 4.01 Specification [online]. <http://www.w3.org/TR/html401/>, 2009 [cit. 2009-05-17].
- [25] XHTML 1.0: The Extensible HyperText Markup Language [online]. <http://www.w3.org/TR/xhtml1/>, 2009 [cit. 2009-05-17].
- [26] Extensible Markup Language (XML) 1.0 [online]. <http://www.w3.org/TR/xml/>, 2009 [cit. 2009-05-17].
- [27] Cascading Style Sheets [online]. <http://www.w3.org/Style/CSS/>, 2009 [cit. 2009-05-17].
- [28] JavaScript Tutorial [online]. <http://www.w3schools.com/JS/>, 2009 [cit. 2009-05-17].
- [29] AJAX Introduction [online]. http://www.w3schools.com/Ajax/ajax_intro.asp, 2009 [cit. 2009-05-17].
- [30] jQuery: The Write Less, Do More, JavaScript Library [online]. <http://jquery.com/>, 2009 [cit. 2009-05-17].
- [31] Drupal – System requirements [online]. <http://drupal.org/requirements>, 2009 [cit. 2009-02-25].
- [32] Joomla! – Technical requirements [online]. <http://www.joomla.org/about-joomla/technical-requirements.html>, 2009 [cit. 2009-02-25].
- [33] WordPress – About – requirements [online]. <http://wordpress.org/about/requirements/>, 2009 [cit. 2009-02-25].
- [34] PHP: Safe Mode – Manual [online]. <http://cz.php.net/features.safe-mode>, 2009 [cit. 2009-04-18].

- [35] PHP: Predefined Variables – Manual [online].
<http://us.php.net/variables.predefined>, 2009 [cit. 2009-04-18].
- [36] Linkback – Wikipedia [online]. <http://en.wikipedia.org/wiki/Linkback>, 2009 [cit. 2009-05-17].
- [37] Hooks – Drupal [online]. <http://api.drupal.org/api/group/hooks>, 2009 [cit. 2009-05-17].
- [38] VanDyk, J. K.; Buytaert, D.: *Pro Drupal Development*. Apress, druhé vydání, 2008, ISBN 978-1-4302-0989-8.
- [39] Joomla! Official Documentation [online].
<http://api.drupal.org/api/group/hooks>, 2009 [cit. 2009-03-15].
- [40] WordPress Codex [online]. http://codex.wordpress.org/Main_Page, 2009 [cit. 2009-03-15].
- [41] Apache friends – xampp [online]. <http://www.apachefriends.org/en/xampp.html>, 2009 [cit. 2009-05-17].
- [42] WampServer [online]. <http://www.wampserver.com/en/>, 2009 [cit. 2009-05-17].
- [43] Drupal – Instalation guide [online]. <http://drupal.org/getting-started/install>, 2009 [cit. 2009-02-25].
- [44] Getting Started with Joomla! [online].
<http://www.joomla.org/about-joomla/getting-started.html>, 2009 [cit. 2009-02-25].
- [45] Installing WordPress [online].
http://codex.wordpress.org/Installing_WordPress, 2009 [cit. 2009-02-25].
- [46] Drupal – community plumbing [online]. <http://drupal.org/>, 2009 [cit. 2009-01-12].
- [47] Drupal Modules – Search, Rate, and Review Drupal Modules [online].
<http://drupalmodules.com/>, 2009 [cit. 2009-01-12].
- [48] Plugin manager: poloautomatická instalace modulů pro Drupal [online].
[http://www.maxiorel.cz/
plugin-manager-poloautomaticka-instalace-modulu-pro-drupal](http://www.maxiorel.cz/plugin-manager-poloautomaticka-instalace-modulu-pro-drupal), 2009 [cit. 2009-05-17].
- [49] Joomla! Extensions Directory [online]. <http://extensions.joomla.org/>, 2009 [cit. 2009-05-17].
- [50] Joomla! Help Site [online]. <http://help.joomla.org/content/view/476/280/>, 2009 [cit. 2009-05-17].
- [51] WordPress – Roles and Capabilities [online].
http://codex.wordpress.org/Roles_and_Capabilities, 2009 [cit. 2009-05-17].
- [52] Jdoc statements [online]. http://docs.joomla.org/Jdoc_statements, 2009 [cit. 2009-05-17].

- [53] Template Hierarchy – WordPress [online].
http://codex.wordpress.org/Template_Hierarchy, 2009 [cit. 2009-05-17].
- [54] GNU gettext utilities [online].
<http://www.gnu.org/software/gettext/manual/gettext.html#Concepts>, 2009 [cit. 2009-05-16].
- [55] GNU gettext utilities [online].
<http://www.gnu.org/software/gettext/manual/gettext.html>, 2009 [cit. 2009-05-16].
- [56] Drupal – Internationalization [online]. <http://drupal.org/project/i18n>, 2009 [cit. 2009-05-16].
- [57] Translating WordPress [online].
http://codex.wordpress.org/Localizing_WordPress, 2009 [cit. 2009-05-16].
- [58] UB 2009 Usability Testing [online]. <http://www.drupalusability.org/>, 2009 [cit. 2009-05-16].
- [59] Search Engine Optimization Starter Guide [online].
www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf, 2009 [cit. 2009-03-27].
- [60] Pravidla přístupnosti – Seznam pravidel [online].
<http://www.pravidla-pristupnosti.cz/>, 2009 [cit. 2009-03-27].
- [61] Section 508: The Road to Accessibility [online]. <http://www.section508.gov/>, 2009 [cit. 2009-03-27].
- [62] Web Accessibility Initiative [online]. <http://www.w3.org/WAI/>, [cit. 2009-03-27].
- [63] PHP: SQL Injection – Manual [online].
<http://se.php.net/security.database.sql-injection>, 2009 [cit. 2009-03-27].
- [64] Understand the Cross Site Scripting Vulnerability [online].
<http://www.crosssitescripting.com/>, 2009 [cit. 2009-03-27].
- [65] The W3C Markup Validation Service [online]. <http://validator.w3.org/>, 2009 [cit. 2009-03-27].
- [66] The W3C CSS Validation Service [online]. <http://jigsaw.w3.org/css-validator/>, 2009 [cit. 2009-03-27].
- [67] HiSoftware Cynthia Says Portal [online]. <http://www.cynthiasays.com/>, 2009 [cit. 2009-03-27].
- [68] Web Page Analyzer [online].
<http://www.websiteoptimization.com/services/analyze/>, 2009 [cit. 2009-03-27].
- [69] Automatický audit webu [online].
<http://nastroje.dobryweb.cz/automaticky-audit/>, 2009 [cit. 2009-03-27].
- [70] Open source performance testing tools [online].
<http://www.opensourcetesting.org/performance.php>, 2009 [cit. 2009-03-27].

Dodatek A

**CD s instalačním manuálem,
zálohami projektů a screenshoty.**