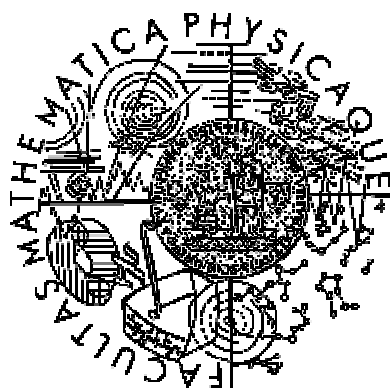


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Jiří Täuber

## **Webové rozhraní pro vedení softwarových projektů s důrazem na správu lidských zdrojů**

Katedra softwarového inženýrství  
Vedoucí bakalářské práce: RNDr. Tomáš Bureš, Ph.D.  
Studijní program: Informatika, Programování

**2008**

## **Poděkování**

Na tomto místě bych chtěl poděkovat vedoucímu mé bakalářské práce panu RNDr. Tomášovi Burešovi, Ph.D. za pomoc při tvorbě této práce.

Dále bych chtěl poděkovat panu doc. Ing. Petrovi Tůmovi, Dr. za pomoc s uspořádáním myšlenek obsažených v tomto textu a všem, kdo se podíleli na odstranění chyb.

## **Prohlášení**

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 7. 8. 2008

Jiří Täuber

# Obsah

<b>1</b>	<b><u>ÚVOD</u></b>	<b>6</b>
	1.1. Správa projektů .....	6
	1.2. Přehled dalších kapitol .....	7
<b>2</b>	<b><u>CÍLE PROJEKTU</u></b>	<b>8</b>
	2.1. Motivace.....	8
	2.2. Projektový manažer.....	8
	2.3. Funkční požadavky.....	9
<b>3</b>	<b><u>ANALÝZA ŘEŠENÍ</u></b>	<b>11</b>
	3.1. Webhosting.....	11
	Vlastní server .....	11
	Webhosting .....	11
	3.2. Existující programy .....	13
	SourceForge.net .....	13
	Trac .....	13
	Mantis .....	14
	dotProject .....	14
	Best Practical Solutions .....	15
	Srovnávací tabulka.....	15
<b>4</b>	<b><u>NÁVRH</u></b>	<b>17</b>
	4.1. Moduly .....	17
	Přihlašování.....	17
	Uživatelské účty.....	17
	Projekty .....	17
	Týmy .....	18
	Fórum.....	18
	Wiki.....	18
	SVN.....	19
	Pravidelně spouštěné úlohy.....	19
	Osobní zprávy .....	19
	Burza práce .....	20
	4.2. Prostředí .....	20

4.3. Rozdělení tříd .....	20
phpMVC .....	20
Modulové pojetí .....	21
<b>5 IMPLEMENTACE</b> .....	<b>22</b>
5.1. phpMVC .....	22
5.2. Moduly .....	22
Wiki.....	23
Fórum .....	23
SVN.....	24
Osobní zprávy a burza práce.....	24
5.3. Hlavní problémy .....	24
Fórum .....	25
SVN Klient.....	26
5.4. Výsledek.....	26
<b>6 PŘEHLED PROJEKTU</b> .....	<b>28</b>
6.1. Jak se vyznat v kódu.....	28
Nastavení.....	28
Action class .....	28
Action form .....	29
Bussiness class .....	29
Templates .....	29
6.2. Jak najít, co hledám.....	30
Action class podle URL adresy.....	30
Template podle URL adresy .....	30
Chyby v programu .....	31
<b>7 VYHODNOCENÍ</b> .....	<b>32</b>
7.1. Funkčnost .....	32
7.2. Srovnání s existujícími projekty .....	33
<b>8 ZÁVĚR</b> .....	<b>35</b>
8.1. Shrnutí .....	35
8.2. Možnosti dalšího vývoje .....	35
<b>9 LITERATURA</b> .....	<b>37</b>
<b>A OBSAH PŘILOŽENÉHO CD</b> .....	<b>39</b>

**Název práce:** Webové rozhraní pro vedení softwarových projektů s důrazem na správu lidských zdrojů

**Autor:** Jiří Täuber

**Katedra (ústav):** Katedra softwarového inženýrství

**Vedoucí diplomové práce:** RNDr. Tomáš Bureš, Ph.D.

**e-mail vedoucího:** Tomas.Bures@mff.cuni.cz

**Abstrakt:** V dnešní době existuje mnoho různých programů, které pomáhají se správou projektů. Programy pro správu projektů se výrazně liší svými možnostmi, schopnostmi a především svým zaměřením. Jsou mezi nimi i programy zaměřené na koordinaci činností v týmu lidí. Také zde máme mnoho webových stránek, které dávají uživatelům možnost nabídky i poptávky pracovních sil. Je překvapující, že nikoho nenapadlo tyto dvě myšlenky spojit. Zvláště pro vývoj software to nabízí značnou flexibilitu. Pracovní týmy tak mohou snadno vznikat a zanikat podle potřeby.

Cílem tohoto projektu je vytvoření jednoduché webové aplikace, která poskytne dynamické prostředí pro snadnou spolupráci mnoha uživatelů na mnoha projektech. Součástí aplikace má být velmi jednoduchý projektový manažer zaměřený na správu lidských zdrojů a zároveň místo pro zveřejnění nabídky a poptávky po práci.

**Klíčová slova:** projektový manažer, webové rozhraní, burza práce

**Title:** Web-based interface for software project management with emphasis to manpower administration

**Author:** Jiří Täuber

**Department:** Department of Software Engineering

**Supervisor:** RNDr. Tomáš Bureš, Ph.D.

**Supervisor's e-mail address:** Tomas.Bures@mff.cuni.cz

**Abstract:** There are many different applications which help with project management nowadays. Project management applications differ considerably in their options, capabilities and most of all in their area of focus. Applications focused on coordinating work of group of people are among them. There are also many webpages that give their users the means to offer or request workforce. It is surprising that nobody has thought about bringing those two ideas together. It offers significant flexibility, especially in the area of software development. Workgroups can be easily created and disbanded as needed.

The goal of this project is to create a simple web application which will provide dynamic environment for easy collaboration of many users on many projects. Part of the project will be a very simple project management software focused on manpower administration as well as a place to publish workforce supply and demand advertisements.

**Keywords:** project manager, web-based, job market

# Kapitola 1

## Úvod

### 1.1. Správa projektů

„Řízení projektů je proces plánování, koordinování a řízení úloh a zdrojů pro dosažení definovaného cíle, obvykle za daný čas, s definovanými zdroji a s omezenou cenou.“ [1]□[1]

Tímto je definována poměrně rozsáhlá disciplína, jejímž obsahem je mimo jiné rozdělování zodpovědnosti a koordinace týmu lidí pracujících na splnění společného cíle, neboli dokončení projektu. U rozsáhlejších projektů se často řízení projektu realizuje na několika úrovních. Jednotlivé týmy mají své dílčí cíle a netriviální bývá i jejich spojení do funkčního celku.

V dnešní době existuje mnoho různých programů, které pomáhají se správou projektů. Tyto programy se výrazně liší svými možnostmi, schopnostmi a především svým zaměřením. V dalším textu se zabývám především aplikacemi, které se ve větší míře zaměřují na koordinaci týmů nebo jednotlivců spolupracujících na projektu. Aplikace, které se soustředí na efektivitu práce, využití času, finančních a materiálních zdrojů vynechám, protože jsou příliš vzdálené od předmětu práce. I mezi aplikacemi soustředícími se na správu týmů a lidských zdrojů však existuje mnoho rozdílů.

Jednou z hlavních charakteristik je způsob práce online a možnost koordinace přes internet. Existují varianty od plně distribuovaných aplikací bez centrálního prvku přes distribuované aplikace s centrální databází až k plně centralizovaným řešením, kde uživatelé přistupují k aplikaci pomocí webového prohlížeče. Všechny tyto varianty mají svá využití, výhody a nevýhody. Distribuované systémy s centrální databází i bez ní často nabízí možnost práce offline, tím se ale uživatelé vystavují riziku neaktuálnosti informací. Na druhou stranu centralizované systémy jsou vždy aktuální, ale pro jejich využívání je třeba být k nim připojen.

Další rozdíly mezi jednotlivými aplikacemi jsou hodně závislé na způsobu použití a konkrétním zaměření aplikace. Tyto charakteristiky se dají shrnout do pojmu „nabízené funkčnosti“. Správa projektů je bohužel tak rozsáhlá disciplína,

že se těžko dá hovořit i o nějakém společném základu pro všechny aplikace, i když, jak uvidíme dále, existují oblasti, které pokrývá většina aplikací. Předmětem této práce je systém, který se zabývá softwarovými projekty a při takovém omezení už je jednoduché stanovit požadavky, podle kterých budeme hodnotit.

## 1.2. Přehled dalších kapitol

Ve druhé kapitole jsou popsány cíle projektu a především funkční požadavky definované před samotnou implementací. Z těch jsem vycházel při návrhu a implementaci Projektového manažeru.

Třetí kapitola rozebírá dostupné technologie vzhledem k funkčním požadavkům definovaným dříve. Nejprve se zabývá webhostingem a možnostmi nasazení, pak stručně popisuje a srovnává vybrané aplikace, které z větší části nebo úplně postihují zadání.

Ve čtvrté kapitole je popsán modulový návrh řešení včetně popisu zodpovědností jednotlivých modulů a rozdělení tříd na několik základních kategorií podle úhlu pohledu.

Postup implementace a problémy, na které jsem narazil v průběhu psaní aplikace jsou popsány v páté kapitole.

V šesté kapitole je stručně popsána struktura zdrojového kódu projektu. Tyto informace přijdou vhod při čtení a případných úpravách zdrojového kódu. To se týká i nejčastěji očekávaných modifikací – úprava vzhledu a překlad do jiných jazyků.

Sedmá kapitola hodnotí práci po jejím dokončení. Vrací se k funkčním požadavkům definovaným na začátku práce a také srovnává výsledný program s již existujícími aplikacemi uvedenými ve třetí kapitole.

Na závěr je shrnut a zhodnocen dosažený stav a také jsou zde zmíněny možnosti dalšího vývoje.

Úplný závěr práce patří přehledu použitých zdrojů a popisu obsahu příloženého CD.

## Kapitola 2

### Cíle projektu

#### 2.1. Motivace

Aplikace zabývající se správou projektů lze těžko srovnávat kvalitativně. Jediným rozumným způsobem jak vybrat vhodné řešení, je stanovit funkční požadavky, které od aplikace očekáváme.

Podrobný seznam funkčních požadavků je popsán ve zvláštní kapitole. Jsou k nim přidány i podmínky, které nesouvisí přímo s funkcemi, ale jsou důležitou charakteristikou programu. Jako hlavní body mezi požadavky figurují: cena (bez poplatků a omezení na používání), dostupnost (z jakéhokoliv počítače s přístupem na internet), jednoduché nasazení a používání (snadno instalovatelný i pro jednotlivce bez firemního zázemí) a důraz na správu lidských zdrojů (přirazování zodpovědností a průhledná struktura týmu).

Nad rámec správy projektů by systém ještě měl umožnit snadnější hledání a výběr spolupracovníků. Uživatel registrovaný v systému by měl mít možnost definovat své schopnosti a nabídnout své služby lidem, kteří zakládají nové pracovní týmy. Pro daný projekt je pak jednodušší najít realizační tým. Naopak pokud projekt potřebuje nové pracovníky na určité pozice, měla by v systému být možnost zveřejnit tento požadavek a případné podmínky pro zájemce.

Systém, který by integroval takovou burzu práce přímo s prostředím, kde se práce organizuje, zatím neexistuje. Proto jsem se rozhodl takovou aplikaci vytvořit.

#### 2.2. Projektový manažer

Hlavním cílem je vytvořit jednoduché rozhraní pro organizaci práce. Toto rozhraní umožní definovat jednotlivé atributy projektu jako název, cíl, autora a realizační tým. Stejně tak bude možné určit role jednotlivých řešitelů v rámci projektu a jejich pravomoci. Další nástroje zprostředkují komunikaci mezi uživateli v rámci projektu a ukládání projektové dokumentace tak, aby vždy byla k dispozici její aktuální verze.



Navíc bude k tomuto rozhraní připojen modul burzy práce, který umožní každému registrovanému uživateli specifikovat jeho schopnosti a nabídnout spolupráci na nově zakládaných projektech. Tyto nabídky budou veřejně přístupné a bude možné v nich jednoduše vyhledávat. Pro účely kontaktování uživatele bude v systému možno poslat krátkou zprávu uživateli jako odpověď na jeho nabídku.

Celý systém bude k dispozici zdarma co největšímu počtu uživatelů. Z tohoto důvodu se bude jednat o webovou aplikaci přístupnou pomocí nejběžnějších webových prohlížečů. Navíc musí být systém použitelný na co největším množství webových serverů, aby mohl být snadno šířen.

### 2.3. Funkční požadavky

Pro přehlednější vyjádření jednotlivých požadavků jsem je definoval jako seznam bodů, které by měl výsledný systém splňovat. Na začátku seznamu se nacházejí požadavky na konkrétní funkce systému, ke konci požadavky přechází k technickým vlastnostem programu.

- uživatelské účty
  - registrace uživatele
  - editace nastavení
  - zasílání zapomenutých hesel
  - ověření zadaných emailových adres
- projekty
  - vytváření projektů
  - editace atributů
  - možnost propojení se Subversion [2]
  - rušení projektů
- pracovní týmy
  - přiřazení uživatelů k projektům
  - změny v tomto přiřazování
  - určení funkce a práv člena týmu vzhledem k projektu
- systém pro správu a údržbu dokumentace
  - jednoduchý způsob editace dokumentace
  - možnost úpravy dokumentace několika uživateli z různých počítačů
- komunikace mezi uživateli
  - fórum pro členy projektu (nepřístupné ostatním)
  - fórum určené pro obecnou diskuzi k projektu
  - globální fórum nezávislé na projektu
  - osobní zprávy (private messages)
- burza práce
  - nabídka pracovní síly
  - poptávka po pracovnících
  - vyhledávání v nabídce i poptávce

- Jednoduché použití
  - přímočaré funkce bez vedlejších závislostí
  - funkce systému může uživatel využívat volně
  - funkce systému uživatel nemusí používat
- dostupnost pomocí internetového prohlížeče
- žádné poplatky za používání nebo instalaci
- snadná instalace na většinu webových serverů
  - často používané technologie
  - nízké nároky na prostor a výpočetní výkon
  - kompaktnost

# Kapitola 3

## Analýza řešení

### 3.1. Webhosting

V požadavcích je definována snadná instalace na většinu webových serverů. Abychom mohli tento požadavek splnit, musíme nejprve určit, jakou konfiguraci má většina webových serverů.

#### Vlastní server

Jedním ze způsobů, jak vůbec získat webový server je provozování vlastního serveru. Z našeho pohledu nemusíme rozlišovat, zda jde o server celý (dedikovaný) nebo jeho část, kterou máme ve správě, případně o virtuální server. Ve všech těchto případech je konfigurace většinou proprietární a dá se snadno přizpůsobit specifickým požadavkům aplikace. Navíc pronájem serveru nebo připojení vlastního serveru do sítě internet je vždy placená služba a tak nepředpokládám použití naší aplikace v tomto prostředí.

#### Webhosting

Alternativou k vlastnímu serveru je takzvaný *webhosting*. Obvykle tato služba zahrnuje poskytnutí úložného prostoru a aplikací potřebných pro prezentaci obsahu na webu. Především by to měl být webový server, dále interprety skriptovacích jazyků, databázový server a další. Pokud chceme pokročilejší možnosti, pak se obvykle musíme připravit na to, že za tuto službu budeme platit. Existují však i varianty webhostingu zdarma, ty pak označujeme jako *freehosting*.

Na internetu existuje až nepřehledné množství poskytovatelů webhostingových a freehostingových služeb. Využil jsem databázi společnosti [hostingy.cz](http://hostingy.cz) [3], která mapuje nabídku webhostingu a souvisejících služeb na českém trhu. Tabulka 1 byla vytvořena na základě informací získaných z jejich serveru. Jsou v ní uvedeny služby nabízené českými společnostmi v rámci jejich freehostingových programů.

V tabulce 1 je vždy název poskytovatele, zda k freehostingu nabízí možnost spouštět skripty v jazyce PHP, ASP (ASP.NET), jaké nabízí SQL databáze a případně jaké jsou další možnosti spuštění skriptů na straně serveru.

<b>firma</b>	<b>PHP</b>	<b>ASP</b>	<b>MySQL</b>	<b>další SQL</b>	<b>další skriptování</b>
3NET	+	-	+	PostgreSQL	-
ASP2.cz	-	+	-	MSSQL	-
ASPone.cz	-	+	-	MSSQL, Access	AJAX, ADO
ASPWEB.cz	-	+	-	MSSQL, Access	-
AtlasWeb Free	+	-	-	-	-
BonusWeb.cz	+	-	+	-	-
CZonline.eu	+	-	+	-	-
domeny.as	+	-	-	-	-
DreamNET	+	-	+	-	-
FBI	+	-	+	-	-
Gringo vizitka	+	-	-	-	-
HostingZdarma	+	-	+	-	cron
HOSTUJU.CZ	+	-	+	-	-
HY.cz	+	-	+	-	cron
Internet Centrum	+	-	+	PostgreSQL, SQLite	cron
KGB	+	-	+	-	-
LTweb	+	-	+	-	cron
ObchodOnLine	+	-	+	-	-
PHP5.cz	+	-	+	-	-
Pipni.cz	+	+	+	PostgreSQL, FireBird	JSP, cron
Profitux	+	-	+	-	-
QSH Freehosting	+	+	-	MSSQL, Access	-
Tenzor	+	-	+	-	-
uzasne.cz	+	+	+	-	-
webstranka.info	+	-	+	-	-
Webzdarma	+	-	+	-	-
YX.CZ	+	-	+	-	CGI, mod_perl, mod_python, mod_rewrite

**tabulka 1: Přehled nabídky freehostingových služeb na českém internetu podle [3]**

Jak je vidět, mezi programovacími jazyky jednoznačně převládá PHP [4]. Z databázových systémů je viditelně nejpoužívanější MySQL [5]. Alternativou k MySQL by bylo použití univerzálnějšího rozhraní ODBC (Open Database Connectivity), které bylo navrženo jako univerzální rozhraní pro databáze. PHP podporuje obě možnosti, jak nativní přístup k MySQL tak i ODBC. Vzhledem k jednoduššímu použití a rozšířenosti MySQL jsem jako základní řešení zvolil tuto databázi a nativní přístup k ní.

## 3.2. Existující programy

Před začátkem práce jsem se poohlédl po již existujících řešeních a srovnal je s požadavky, které jsem definoval ve druhé kapitole. V následujících odstavcích je stručně popsáno několik velice dobře zpracovaných řešení, která jsou často používána pro správu softwarových projektů. Vybral jsem je tak, aby aplikace postihovaly různé způsoby řešení tohoto problému. Další aplikace jsou vzhledem k našim funkčním požadavkům často velice podobné následujícím příkladům.

### SourceForge.net

SourceForge.net [6] je velice známé a robustní řešení zaměřené na vývoj volně šiřitelných programů. Vyhovuje téměř celému seznamu funkčních požadavků z předchozí kapitoly a nabízí mnoho funkcí navíc. Zaměřuje se na otevřené softwarové projekty a poskytuje mnoho funkcí pro vývojáře software, jako například systémy pro správu verzí, hlášení chyb, žádosti na zlepšení, publikování záplat, a další.

Aplikace je na rozdíl od ostatních provozována zdarma společností SourceForge, Inc. na internetové adrese <http://sourceforge.net/>. Používání této centrální verze nevyžaduje žádnou instalaci a je zdarma. Vzhledem k velkému počtu uživatelů i projektů zde však existují jistá omezení. Vytváření nových projektů prochází cenzurou a mazání projektů, které publikovaly alespoň jednu verzi programu nebo záplatu není povoleno.

Aby nedocházelo k omezením při vytváření a mazání projektů, je zde také možnost celý systém stáhnout [7] a nainstalovat. Zde se projeví hlavní nevýhoda SourceForge.net. Velikost instalačního souboru je více než 900MB a celá aplikace požaduje specifické podmínky na hostitelském serveru.

### Trac

Systém Trac [8] jde téměř opačným směrem než SourceForge. Je založen na rozšířené wiki s podporou sledování otázek spojených s vývojem projektu. Prosazuje minimalistický přístup, takže se snaží vývojářům pomoci, aniž by jim stál v cestě. V základní verzi poskytuje zhruba polovinu funkcí popsaných v kapitole 2.3., pro zbytek je třeba stáhnout nebo naprogramovat další zásuvné moduly. V plné verzi se ale stává velmi silným nástrojem.

Trac není provozován nikde centrálně, takže před použitím je třeba ho nainstalovat na vlastní server. Samotná instalace je jednoduchá, ale vyžaduje poměrně

velké množství samostatných aplikací a knihoven jako prerekvizity. Těmi hlavními jsou interpret jazyka python, SQL databáze (SQLite, PostgreSQL nebo experimentálně MySQL ) a několik specifických knihoven jazyka python. Pokud chceme dosáhnout požadované funkčnosti, tak musíme k aplikaci navíc nainstalovat dodatečné zásuvné moduly. Ač to na první pohled vypadá na poměrně jednoduché požadavky, tak z kapitoly 3.1 už víme, že použít tuto aplikaci na .veřejném freehostingovém serveru je prakticky nemožné.

## **Mantis**

Autoři aplikace Mantis [9] kladli důraz především na sledování chyb a úkolů v rámci projektu. Je zde poměrně propracovaný systém tiketů pro přidělování práce, ale ostatní funkce jsou dost omezené nebo úplně chybí. Narozdíl od systému Trac zde ani není systém zásuvných modulů, který by rozšiřoval možnosti aplikace.

Nespornou výhodou tohoto systému je nenáročnost jeho instalace. Ta je možná prakticky kdekoliv. Jedinými prerekvizitami jsou interpret jazyka PHP a jedna z podporovaných SQL databází (mimo jiné i MySQL). Toto řešení je obzvláště vhodné pro menší softwarové projekty, kde jsou účastníci v jedné lokalitě. Tým programátorů tak může těžit z jednoduchého a přitom dobrého způsobu rozdělování práce, ale nebude omezen nedostatkem funkcí pro komunikaci.

## **dotProject**

dotProject [10] je jeden ze systémů, který není zaměřen čistě na softwarové projekty. Má velmi dobře zpracované nástroje pro správu projektů, úkolů, jejich rozvrhu, sdílení zdrojů a komunikaci v týmu. Cenou za takto dobře propracovanou správu projektů je velké množství administrativy navíc, která často nejde obejít. Uživatel je tak nucen využívat všechny vlastnosti systému i když by je jinak nepotřeboval.

Aplikace je opět nenáročná na instalaci. Provozovat jí lze na webovém serveru, kde je k dispozici interpret jazyka PHP s knihovnou GD pro vykreslování grafů. Podporovanou databází je v současné době jen MySQL a to navíc jen v případě, že v databázi máme povoleno vytvářet dočasné tabulky.

## **Best Practical Solutions**

Na rozdíl od předchozích aplikací, je řešení od společnosti Best Practical Solutions [11] založeno na několika oddělených aplikacích, z nichž každá řeší část problému správy projektů. Tento přístup dovoluje přizpůsobit každou aplikaci specifickým potřebám jednotlivých částí této disciplíny. Pro řízení projektů může uživatel použít jen ty aplikace, které skutečně potřebuje a případně je doplnit dalšími aplikacemi jiných vydavatelů.

Balíček Best Practical Solutions obsahuje 3 aplikace. Hlavní z nich, RT - Request Tracker, je zaměřena na správu požadavků. Poskytuje sofistikovaný mechanismus, kde se pomocí tiketů dají naplánovat jednotlivé činnosti v rámci projektu. Bohužel zde úplně chybí nástroje použitelné na správu týmů. Aplikace je psaná v jazyce Perl a dokáže využít 3 různé SQL databáze. Nicméně instalace má některé značně netriviální předpoklady, jako například práva uživatele root. Navíc návody na instalaci pro každý operační systém zvlášť nedávají naději na jednoduchou proceduru.

Další je distribuovaná aplikace integrovatelná s několika systémy pro správu verzí, která rozšiřuje možnosti těchto systémů o práci offline. Její instalace je ještě komplikovanější než instalace systému RT, protože je vyžadována na každém počítači zvlášť. Zdá se, že výhody této aplikace nestojí za komplikace při její instalaci, zde se projeví výhoda oddělených aplikací.

Dokumentační technologie je v tomto řešení poskytována neobvyklým způsobem založeným opět na tiketovacím systému. Způsob práce i rozhraní je stejné jako u RT. Zadavatel jen nepopisuje problém, který potřebuje řešení, ale naopak popisuje řešení k problému nebo spíše odpovídi na otázky. Dokumentace je pak rozdělena na jednotlivé články, ve kterých jde vyhledávat podle jednotlivých témat. Každý článek má navíc uloženou celou svou historii.

Způsob instalace je velice podobný první části. Pokud však již máme RT nainstalované, pak by instalace a nastavení údajně měla být otázkou minut.

## **Srovnávací tabulka**

V tabulce 2 jsou uvedeny výše zmíněné aplikace a jejich hodnocení vzhledem k jednotlivým kategoriím našich funkčních požadavků. Pokud je v poli uvedeno jen znaménko +, pak podle mého názoru aplikace vyhovuje podmínkám.

	SourceForge	Trac	Mantis	dotProject	Best Practical
<b>uživatelé</b>	+	+	+	+	+
<b>projekty</b>	cenzura	vůbec	různé fronty	bez SVN	různé fronty
<b>týmy</b>	+	vůbec	vůbec	+	slabě
<b>dokumentace</b>	+	+	+	+	+
<b>komunikace</b>	+	tikety,email	RSS, email	+	tikety
<b>burza práce</b>	+	vůbec	vůbec	vůbec	vůbec
<b>jednoduchost</b>	?	+	+	—	—
<b>dostupnost</b>	+	+	+	+	+
<b>instalace</b>	nestandardní	náročná	+	+	náročná
<b>použitý jazyk</b>	?	Python	PHP	PHP	Python
<b>databáze</b>	?	+	MySQL	MySQL	+

**tabulka 2: Srovnání aplikací zabývajících se správou projektů**



# Kapitola 4

## Návrh

### 4.1. Moduly

Základní úrovní dekompozice celé aplikace je rozdělení na *moduly*. Modul je uzavřená část aplikace, která se zabývá konkrétním problémem. Moduly, ze kterých se skládá Projektový manažer, jsou popsány níže. Tvoří nezávislé celky a jediné informace, které se celkem přirozeně sdílí napříč moduly, jsou uživatelské nastavení, práva a informace o aktuálně zvoleném projektu. Moduly jsou voleny tak, aby sdružovaly podobné funkčnosti a zároveň pokrývaly všechny požadavky.

#### **Přihlašování**

Přihlašovací modul má samozřejmě na starosti přihlašování a odhlašování uživatelů. Méně zřejmou zodpovědností je ověřování platnosti přihlášení a nahrávání údajů o uživateli v pravidelných intervalech během jeho návštěvy. Pokud se totiž uživateli například změní práva v průběhu jeho přihlášení, pak by se tato změna měla promítnout do jeho možností co nejdříve a ne až při dalším přihlášení.

#### **Uživatelské účty**

Modul spravuje vytváření, mazání a změny nastavení uživatelských účtů. To zahrnuje také zasílání ověřovacích emailů na zadanou adresu při vytváření účtu nebo při změně emailové adresy v nastavení. Zadaná adresa je už při registraci ověřena zasláním registračního emailu. Trochu neočekávanou vlastností tohoto modulu je generování nového hesla v případě, že své staré uživatel zapomene. Tato funkce je zařazena k účtům namísto přihlašování proto, že je velice podobná zasílání ověřovacích emailů při vytváření účtu nebo změně emailové adresy.

#### **Projekty**

Správa projektů zahrnuje jejich vytváření, mazání, změny základních atributů a výpisy. Projekty je možné třídit jen abecedně a z hlediska omezení výběru jsou rozlišovány pouze projekty, na kterých se uživatel podílí. Podle příslušnosti k projektu jsou také zobrazovány funkce v jiných modulech.

## **Týmy**

Týmy účastníků jednotlivých projektů byly odděleny od projektu proto, že jejich správa je v Projektovém manažeru stejně rozsáhlá jako jakýkoliv další modul. Uživatel může být do projektu přijat, může být vyhozen, může odejít sám a samozřejmě má jistá nastavení týkající se projektu. Hlavními vlastnostmi jsou jeho oprávnění a označení pozice (role), kterou v projektu zaujímá. Role majitele projektu má speciální status. Majitel projektu je vždy ten, kdo ho vytvořil, nemůže být změněna a vždy má absolutní práva k projektu.

## **Fórum**

Fórum je dnes již standardem co se týče komunikace na internetu. V mém systému fóra představují základní formu komunikace. Máme zde právě 3 úrovně sdružování příspěvků. Nejnižší úrovní jsou vlákna, ta jsou sdružena do témat a pro každý projekt existují 3 kategorie témat. Jsou to témata určená veřejnosti, kde každý může číst i přispívat. Dále jsou to témata určená pro jednosměrnou komunikaci správců projektu s okolním světem, zde mohou všichni číst, ale přispívat mohou jen někteří členové projektu. Poslední kategorií jsou témata pro projektovou potřebu, číst i přispívat zde mohou všichni členové projektu. Přispívání do fór může být zakázáno nastavením práv.

Dnes existuje mnoho kvalitních systémů, které jsou napsané v různých programovacích jazycích. V době návrhu se předpokládá použití některé z těchto aplikací. Hlavním výběrovým kritériem je programovací jazyk PHP a dostatečná kompatibilita se zbytkem systému. Případné malé úpravy se předpokládají.

## **Wiki**

Stejně jako fóra jsou standardem pro komunikaci, tak wiki systém je standardem pro prezentaci informací a dokumentaci. Každý projekt má proto vlastní wiki stránky, které mohou být použity pro dokumentaci, plánování a nebo uveřejňování dalších informací týkajících se projektu. Všechny tyto informace zůstávají na serveru včetně historie změn a tak není problém jejich editace z různých počítačů nebo náprava chyb.

Wiki je založena na otevřenosti a možnosti všech uživatelů přispívat svými znalostmi, ale pokud jde o plánování projektu, tak rozhodně nechceme dát každému možnost měnit informace podle libosti. Proto i Wiki má 3 oddělení s různými právy

přístupu podobně jako fóra. Je pak jen na členech projektu, kam se rozhodnou umístit které informace. Podobně jako u fór, i zde se předpokládá použití jedné z již existujících aplikací.

## SVN

*Subversion* [2] nebo-li SVN je mezi programátory velice známý nástroj pro správu verzí. Málokterý programátor si dovede představit práci bez SVN a práce v týmu je bez tohoto nebo podobného systému téměř nemožná. S vědomím důležitosti tohoto nástroje jsem začlenil do Projektového manažeru i modul pro jeho podporu. Bohužel žádný poskytovatel freehostingu nenabízí možnosti používání SVN úložišť. V některých ojedinělých případech se však dá zařídit možnost spouštění SVN klienta, Z těchto důvodů považuji tento modul spíše za experimentální. Nabízí pouze možnost procházení<sup>1</sup> libovolného vzdáleného<sup>2</sup> SVN úložiště a to i v případě, že je požadováno jméno a heslo.

Pokud webserver neumožňuje přístup k SVN klientskému programu, pak stačí jednoduchým zásahem do souboru se záhlavím stránek smazat odkaz na tento modul. Tím se stane nedostupný a nemůže tedy nastat chyba při jeho běhu.

## Pravidelně spouštěné úlohy

Tyto jsou zde opět jako možnost. Bez zásahu do programu je tento modul spouštěn vždy, když je ověřena platnost přihlášení nějakého uživatele<sup>3</sup>. Tento přístup je značně neefektivní, ale funguje vždy. Pokud má uživatel k dispozici pravidelné spouštění úloh například pomocí cronu, pak musí toto spouštění stejně nastavit sám a alespoň nahlédnutí do zdrojového kódu je nutné. V tomto modulu jsou schovány úlohy pro čištění databáze od nepotvrzených registrací a nepotvrzených změn emailových adres.

## Osobní zprávy

Samozřejmou schopností tohoto modulu je posílání a čtení osobních zpráv. Další možností bude preposílání zpráv nebo upozornění na email. Vyhledávání ve zprávách je dnes již samozřejmostí a tak nemůže chybět.

---

<sup>1</sup> Pouze výpisy adresářů bez dalších informací a výpis souboru (příkazy list a cat)

<sup>2</sup> Vzdálené úložiště musí být vytvořeno a umístěno na serveru třetích stran. Modul si neklade za cíl vytvářet vlastní úložiště pro projekty.

<sup>3</sup> Volání této metody je v konstruktoru třídy AbstractBaseAction, o kterém ještě bude řeč později

## Burza práce

Tato komponenta bude nabízet všechny funkce požadované od burzy práce. Uživatel by měl mít možnost zde definovat své schopnosti, připsat komentář a nabídnout tak své služby. Jako druhá část se zde dají inzerovat nabídky pracovních příležitostí. Vyhledávání je i zde samozřejmostí a mělo by zde být umožněno především podle nabízených nebo požadovaných schopností uživatele. To vytváří tlak na standardizování označení programátorských schopností.

Pokud se podaří najít nějaký standard, pak do budoucna může ještě přibýt modul pro hodnocení schopností programátorů. To však již není součástí této práce.

## 4.2. Prostředí

Programovat takto rozsáhlý program bez použití knihoven, frameworků nebo již hotových aplikací je zbytečně vynaložené úsilí. Na internetu se dá nalézt mnoho volně šiřitelných a použitelných knihoven nebo programů, které se dají použít. Z druhé kapitoly vyplývá, že žádné z existujících řešení bohužel není vhodné ani jako základ pro Projektový manažer.

Jak vyplývá z tabulky 1, použitým programovacím prostředím bude PHP a databáze MySQL. Častým zlovykem programátorů v PHP je míchání výkonného kódu mezi zobrazované části nebo naopak. Proto jsem jako kostru celého programu zvolil framework phpMVC [12], který pracuje s architekturou *Model-view-controller* [13]. Tato architektura rozděluje program na 3 téměř nezávislé části a to programovou logiku (model), uživatelské rozhraní (view) a na controller, který tyto dvě části propojuje a zpracovává vstupy od uživatele. PhpMVC obsahuje prakticky kompletní část controlleru, takže zbývá doplnit programovou logiku a pohledovou část aplikace.

## 4.3. Rozdělení tříd

### phpMVC

phpMVC používá třídy jen v části modelu a controlleru. Uživatelské rozhraní je z pravidla tvořeno HTML kódem, do kterého jsou vloženy PHP příkazy a tak nejde mluvit o rozdělení do tříd. Controller je součástí frameworku, tak je všechen výkonný

kód koncentrován v části označované jako model. Třídy modelu jsou rozděleny na 3 základní kategorie.

První třída, která obvykle přijde do styku s daty získanými ze vstupu je takzvaná *action form*. Podle doporučení autorů phpMVC se zde ověří platnost vstupů (často z webového formuláře) a data se vyčistí od případných pokusů o nabourání. Ne při každé akci je vstupních dat tolik, aby bylo potřeba je validovat, proto je tento krok někdy přeskočen.

Jako druhá v pořadí dostane slovo takzvaná *action class*. V případě, že si přejeme ověření vstupů přeskočit nebo zařadit jinam než na začátek zpracování, pak je tato třída samozřejmě první. Ověření platnosti vstupů v tomto případě musí být vyvoláno explicitně. Hlavním úkolem *action class* je zpracovat vstupní data a podle nich provést příslušné akce, jako například modifikaci databáze, provedení výpočtu, načtení dat, zpracování a uložení případných chyb, a další. Nakonec controlleru předá informaci o tom, jaký pohled (která stránka) bude zobrazen uživateli.

Z několika důvodů není rozumné aby všechny části výkonného kódu byly soustředěny v *action class*. Takový program by byl těžko modifikovatelný a hrozilo by slučování nesourodých činností do jedné třídy. Proto *action class* ve skutečnosti data jen mírně modifikuje a volá metody tříd specializovaných na práci s nimi. Tyto třídy phpMVC označuje jako *bussiness class*.

První dva druhy tříd (*action form* a *action class*) jsou uloženy v adresáři *classes*. Poslední druh, tedy *bussiness class*, jsou vzhledem ke své rozdílné povaze a nezávislosti na phpMVC uloženy v adresáři *lib*.

## **Modulové pojetí**

Modulové pojetí celého projektu rozděluje třídy trochu jiným způsobem. Každý modul je zde zastoupen jednou *action class* a případně příslušnou *action form*. Poslední skupina tříd je sdílená přes všechny moduly a jednotlivé *action class* je využívají podle potřeby. V tomto je rozdělení stejné jako v pojetí phpMVC.

# Kapitola 5

## Implementace

### 5.1. phpMVC

Nejprve jsem se musel seznámit s prostředím, ve kterém je projekt vytvořen. Framework obsahuje controller, což je centrální část zvolené architektury. Znalost nastavení controlleru a jeho možností je základem pro úspěšnou práci s phpMVC. Bohužel tento framework není zrovna dobře dokumentován, a tak jsem se spoléhal především na funkce uvedené v příkladu použití a metody pokusů a omylů.

I z manuálu se dá vyčíst, že nejdůležitějším prvkem celého phpMVC je konfigurační XML soubor. V něm jsem nastavil rozdělení aplikace na jednotlivé moduly a přiřadil akce příslušným třídám. Do controlleru poskytnutého frameworkem jsem pak naprogramoval vlastní *ActionDispatcher*. To je objekt, který prakticky zpracovává sekci pohledů. Hlavní činností je nalezení správného HTML template a jeho zaslání na klientský počítač.

### 5.2. Moduly

Dále jsem začal pracovat na jednotlivých modulech. Společné metody všech action form a action class jsou definovány v příslušných *AbstractBase...* třídách. Pro formulářové třídy jsou to například metody na validaci dat různých typů. *AbstractBaseAction* zase obsahuje inicializační rutinu a metody na zpracovávání chyb. Konkrétní instance modulových tříd jsou odvozeny vždy od těchto dvou.

*AbstractRunAction* je speciální třída, která nezapadá do modulového systému. Je odvozena od *AbstractBaseAction* a slouží jako mezistupeň pro moduly, které jsou nahrazeny vnějším programem<sup>4</sup>. Action class příslušného modulu má v těchto případech za úkol jen předpřipravit data pro vnější program, ten spustit a pak zpracovat jeho výstup. Už z tohoto popisu je jasné, že se zde vyskytuje několik činností společných pro všechny externí aplikace využitě v Projektovém manažeru. Právě tyto funkce jsou obsaženy v *AbstractRunAction*.

---

<sup>4</sup> Po neúspěšných pokusech s aplikacemi pro fóra a SVN využívá externí aplikaci jen modul wiki

## Wiki

Modul wiki slouží k uchovávání dokumentace a případně k prezentaci dalších informací souvisejících s projektem. Tyto informace mohou být určeny jak pro veřejnost tak pouze pro potřebu uvnitř týmu. K tomu je potřeba alespoň dvou různých druhů wiki stránek a rozlišení přístupu k nim.

Jako první jsem prováděl pokusy s *PWP wiki procesor* [14]. Je mimořádně snadné ho začlenit do většího projektu, protože kód je malý, jednoduchý a neobsahuje mnoho nepříjemností popsanych v kapitole 5.2 nahoře. Navíc stránky uchovává ve formě souborů a je plně závislý na vnějších informacích o aktuálním uživateli. Obě tyto vlastnosti jsou z hlediska integrace žádoucí. Bohužel se ukázalo, že odstranění i těch několika málo problémů a přeložení do češtiny tvoří příliš velkou zátěž. PWP wiki procesor jsem tedy odložil a nahradil ho sofistikovanější *DokuWiki* [15].

DokuWiki je aplikace přímo určená k vytváření dokumentace. Nabízí více možností jak pro uživatele tak pro programátora, který jí chce začlenit do většího celku. Data také ukládá ve formě souborů a je jednoduché doprogramovat vlastní způsob ověřování uživatelů.

Rozdělení wiki prostoru na jednotlivé projekty je provedeno pomocí takzvaných jmenných prostorů, což jsou jakési adresáře. Stejným způsobem je prostor každého projektu rozdělen ještě na 3 části. Hierarchie jmenných prostorů může být téměř neomezeně hluboká, takže v tomto přehledu hvězdička znázorňuje buď stránku nebo jmenný prostor.

- `nazev_projektu::private::*` - stránky viditelné jen pro členy projektu
- `nazev_projektu::public::*` - stránky, kde mají všichni právo číst i psát
- `nazev_projektu::*` - stránky viditelné pro všechny, ale zapisovat do nich mohou jen členové projektu

Z hlediska DokuWiki rozdělujeme uživatele na dvě skupiny: členové projektu a ostatní. Každá z těchto skupin uživatelů má při vytvoření projektu nastavena příslušná práva k výše zmíněným jmenným prostorům. Úkolem `RunClass` je pak určit, do které ze skupin uživatel patří vzhledem k právě navštěvované stránce a předat tuto informaci DokuWiki, která se postará o zbytek práce.

## Fórum

Tento modul mi způsobil největší problémy. Po několika neúspěšných pokusech o zapojení existujících aplikací do phpMVC jsem byl nucen naprogramovat

alespoň kostru nějakého fóra sám, což znamenalo mnoho práce navíc a tedy i velké zdržení.

Vzhledem k omezenému času jsem se snažil redukovat vlastnosti fór na ty nejnütnější tak, aby byly stále použitelné pro základní komunikaci mezi uživateli, ale i tak tento modul tvoří téměř polovinu veškerého výkonného kódu.

## **SVN**

Speciálně pro vedení softwarových projektů je zde experimentálně zaveden modul SVN klienta. Tento modul umí číst z libovolného SVN úložiště a to i tehdy, pokud je chráněno heslem. Vzhledem k jeho experimentálnímu charakteru nepodporuje jiné operace než výpis adresářů a souborů. Pro další operace by bylo potřeba definovat podstatně sofistikovanější systém práv uživatelů i systému, hrozí zde totiž zneužití.

## **Osobní zprávy a burza práce**

Neplánované zdržení programováním kostry komunikace v Projektovém manažeru si vyžádalo tolik času navíc, že na tyto dva moduly již žádný nezbyl. Mohl jsem sice zvolit jiné pořadí a fóru se věnovat až na konec, ale jsem přesvědčen, že fórum je obecně větším přínosem než osobní zprávy, protože může sloužit k řešení opakovaných problémů. Osobní zprávy stejně bývají často nahrazeny nějakým chatem nebo instantním messengerem.

Burza práce jde zase relativně snadno a úspěšně provozovat na moderovaném fóru, kde existuje jedno téma na nabídku pracovních sil a druhé na poptávku. Takové fórum lze v tomto systému vytvořit a proto jsem považoval jeho dokončení za důležité.

## **5.3. Hlavní problémy**

Použití phpMVC je výhodou také proto, že aplikace má jediný vstupní bod. Mimo jiné to má výhody v tom, že můžeme snadno vytvářet odkazy na jednotlivé části systému a případný neplatný nebo nefungující odkaz zavede uživatele vždy na stejnou (platnou) stránku. Mechanismus `AbstractRunAction` umožňuje zachovat tuto charakteristiku i ve chvíli, kdy používáme externí aplikace, ale tím je na ně dáno několik omezení.



Například taková aplikace nesmí neočekávaně skončit a přerušit běh skriptu. Aplikace by tak sice skončila, ale tím by ukončila i metodu modelu a controlleru, ve kterých je spuštěna. Uživatel by tak nedostal žádný výstup, protože controller by nedostal příležitost dokončit svou práci a vydat nějaký pohled. Tento problém ale lze obejít.

Ukázalo se, že paradoxně téměř neřešitelné omezení je zákaz automatického přesměrování<sup>5</sup>, kterého některé aplikace využívají pro zjednodušení navigace. Při odeslání formuláře aplikaci jsou data zpracována a podle výsledku zpracování je uživatel automaticky přesměrován buď na patřičnou stránku nebo zpět na formulář (např. pokud se v něm vyskytly chyby). Adresy takových přesměrování bývají v aplikacích pevně zakódovány na mnoha místech a neexistuje způsob, jak je změnit bez přímé editace všech míst, kde se vyskytují.

Kvůli těmto a několika dalším omezením se jako největší problém ukázala nekompatibilita hotových specializovaných systémů, které jsem chtěl začlenit do projektu. Problémy s kompatibilitou wiki systému jsem popsal výše, ale hlavně se jednalo o mnohokrát vyřešená fóra, kterých je na internetu k nalezení mnoho různých variant, přesto žádná z nich nebyla vhodná pro použití uvnitř jiné aplikace.

## Fórum

Mezi nejznámější a nejpoužívanější patří zřejmě *phpBB* [16]. Jeho nespornou výhodou je robustnost, univerzálnost a lokalizace snad do všech světových jazyků. Bohužel trpí oběma výše zmíněnými vadami. Navíc je příliš univerzální, poskytuje funkce, které nejsou pro Projektový manažer potřeba a vyžaduje pro to zbytečně velké množství tabulek v databázi. Další nevýhodou je obtížná změna vzhledu a vlastní databázová tabulka pro uchovávání informací o uživateli.

Další sofistikovaná fóra jsou na tom téměř stejně jako *phpBB*, jen jsou možná o něco menší. Nicméně vždy se v nich vyskytuje automatické přesměrování a téměř vždy mají vlastní databázi uživatelů, kterou by bylo potřeba udržovat aktualizovanou paralelně s hlavní databází, případně by bylo potřeba tyto dvě databáze komplikovaně kombinovat do jedné.

Méně pokročilá fóra, která mají jen základní funkce byla z hlediska kompatibility o něco lépe koncipovaná. Autoři těchto aplikací však nepočítali

---

<sup>5</sup> automatickým přesměrováním je myšleno vygenerování HTML hlavičky „location“, které nelze z vnějšku aplikace nijak ovlivnit narozdíl od přesměrování pomocí skriptu umístěného na stránce.

s širokým nasazením a proto se v nich vyskytují chyby, výstup programu je promíchán s výkonným kódem, což významně ztěžuje změnu vzhledu, o překladu uživatelského rozhraní ani nemluvě.

Po neúspěších s hotovými řešeními jsem byl donucen naprogramovat alespoň základní funkce fůr sám. To se bohužel podepsalo na časovém rozvrhu a celkové kvalitě aplikace.

### **SVN klient**

Podobné problémy, jaké byly u fóra nastaly i u SVN klientských aplikací. Zde je navíc výběr omezen ještě různými funkcemi navíc. Více obvykle znamená méně, protože pokročilé funkce pro práci s SVN archivem obvykle vyžadují nestandardní práva a operace na webovém serveru.

Jako nejzdařilejší se na první pohled jeví systém *webSVN* [17], který je publikovaný na webu tvůrců samotného systému Subversion. Při bližším prozkoumání však zjistíme, že v tomto rozhraní jsou chyby a jeho vývoj skončil v roce 2004. Navíc využívá přesměrování, nepodporuje připojení do úložišť chráněných heslem a nedá se prakticky rozšířit o další funkce.

V tomto případě jsem našel knihovnu ještě lepší než jakékoliv hotové řešení. *VersionControl\_SVN* [18] dokáže zprostředkovat spuštění SVN klienta a zachycení jeho výstupu. Tato knihovna patří do rodiny balíčků *Pear*, které se dají snadno integrovat do většiny PHP projektů.

## **5.4. Výsledek**

Vzhledem k nečekaným zdržením způsobeným problémy s fóry se nepodařilo naplnit všechny funkční požadavky. Celá burza práce a část komunikace mezi uživateli, konkrétně osobní zprávy, nebyla naimplementována. Ostatní funkční požadavky jsou splněny. Aplikace má dokonce vlastní instalátor, který se pokusí ověřit, zda jsou splněny všechny podmínky pro instalaci a nainstaluje aplikaci v základní verzi. I přes to je doporučeno ještě před samotnou instalací nastavit některé parametry aplikace.

Pochybnosti o splnění funkčních požadavků by mohly nastat pouze v případě snadné instalace na většinu webových serverů. Pro plnou funkčnost vyžaduje Projektový manažer webový server s podporou PHP verze 5.0 nebo vyšší, MySQL

databází, možností serveru zapisovat do některých adresářů webového stromu a možnost spouštět SVN klienta. Zvláště posledním dvěma požadavkům většina freehostingových poskytovatelů nerada vyhovuje.

Pokud se však obejdeme bez experimentálního modulu SVN, pak stačí mírný zásah do zdrojového kódu, kterým odstraníme odkaz na SVN modul a ten se tak stane nedostupný. K provozování zbytku aplikace stačí PHP verze 4.1 nebo vyšší, MySQL databáze a možnost serveru zapisovat do některých adresářů webového stromu. Poslední rekvizitu zvládne zkušený uživatel zařídit tím, že adresářům DokuWiki nastaví webserver jako majitele.

Celá aplikace je nyní ve fázi alfa testování, které by mělo probíhat alespoň půl roku. Za tu dobu se odstraní všechny nedostatky, které se v aplikaci jistě usídlily během vývoje.

# Kapitola 6

## Přehled projektu

### 6.1. Jak se vyznat v kódu

V aplikaci je použito několik různorodých technologií a celý systém obsahuje téměř 2 000 souborů se zdrojovým kódem. Vyznat se v takové změti informací je pro neznalého člověka téměř nemožné, proto zde přikládám stručný návod, který by měl zjednodušit základní orientaci ve zdrojovém kódu. Cílem není popsat jednotlivé třídy, natožpak metody. Zájemce o tyto informace musím odkázat přímo do zdrojového kódu. Zde jen uvedu informace, které každému zjednoduší orientaci v záplavě souborů, definic a příkazů.

#### Nastavení

Hned v hlavním adresáři projektu vidíme vstupní bod celé aplikace (**index.php**) a instalační skript (**install.php**). Ještě před prvním spuštěním instalace bychom však měli projít jednotlivá nastavení a zkontrolovat, že odpovídají našim představám. Soubor **WEB-INF/constants.php** obsahuje nastavení všech parametrů Projektového manažeru. Při instalaci ještě vznikne ve stejném adresáři soubor **config.php**, který obsahuje informace o databázovém připojení.

Wiki má jako externí aplikace také vlastní nastavení, nicméně jeho změny nejsou doporučované, pokud to není nezbytně nutné. Některé funkce DokuWiki jsou záměrně omezené a jejich povolení nebo změny nastavení mohou mít nechtěné důsledky (například dvojí registraci nebo přihlašování uživatelů).

#### Action class

Data na každou stránku jsou připraveny právě jednou action class z adresáře **WEB-INF/classes**. Framework vždy předpřipraví vstupy a spustí metodu `execute` v příslušném objektu. Při výrobě action class objektu hraje významnou roli třída *AbstractBaseAction*, která je předkem. V konstruktoru tohoto objektu je zřejmě nejdůležitějším momentem inicializace databázového připojení.

Metoda `execute` obvykle obsahuje jeden velký `switch`, který rozhoduje, jaká akce bude provedena v rámci daného modulu. Vždy se v jednotlivých větvích tohoto rozhodování ověří vstupní data, práva a případné další požadavky.

### **Action form**

Jak již bylo řečeno, obvykle data přijdou do styku nejprve s formulářovou třídou. Ta má za úkol ověřit platnost zadaných dat a případně je vyčistit od podezřelých znaků.

Action form objekt je vytvořen ještě před action class. Pokud je framework nastaven tak, aby data ověřil sám, pak hned po vytvoření a inicializaci objektu spustí jeho metodu `validate`. Když tato metoda skončí s chybou, pak action class není ani vytvořena, což není žádoucí chování. Proto je validační metoda volána z patřičných míst action class podle potřeby.

### **Bussiness class**

Poslední druh objektů, které tvoří logiku programu jsou takzvané *Bussiness class*. Jsou uloženy v adresáři **WEB-INF/lib**. Většina výkonného kódu a veškerý kód pracující s databází je soustředěn do těchto tříd. Action class v tomto ohledu tvoří jen jakési rozcestníky a správce chybových hlášení, skutečnou práci dělají právě bussiness class.

### **Templates**

Velice zajímavým adresářem jsou **templates**. Zde je soustředěn veškerý výstup aplikace. Soubory nejsou programovány v žádném template prostředí, jde o čistý HTML kód s vloženými PHP příkazy. Jedinou výjimkou jsou template soubory zasílaných emailů, které jsou jen textem a nastavují proměnné (především *\$subject*).

Podadresář **entities** sdružuje malé kusy kódu používané napříč celou aplikací. Nejdůležitějšími entitami jsou hlavička a patička. Hlavička je vždy vytištěna před libovolným dalším template a patička vždy nakonec. O tuto funkci se trochu skrytě stará `ActionDispatcher` (umístěný ve **WEB-INF/classes**).

Trochu stranou od věcí, které byly právě popsány stojí ještě template systém DokuWiki uložený v podadresáři **wiki**. Zde se musíme řídit nápovědou k DokuWiki a přitom myslet na to, že výstup bude ještě parsován pomocí *WikiActionClass*. Přitom stále platí, že společná hlavička a patička budou také vytištěny.

Zcela speciálně je třeba přistupovat k souboru **MyAppResources.properties**, který obsahuje všechna chybová hlášení aplikace.

Pokud někdo bude chtít změnit vzhled nebo přeložit Projektový manažer do svého jazyka, pak v adresáři templates je vše, co potřebuje upravit.

## 6.2. Jak najít, co hledám

### Action class podle URL adresy

Pokud známe jen adresu, na které došlo k problému a potřebujeme najít action class, která chybu způsobuje, pak musíme nejprve nahlédnout do souboru s konfigurací phpMVC a to je **phpmvc-config.xml**. Tento soubor definuje vztahy mezi adresou URL (konkrétně parametrem „pm\_section“) a jednotlivými action class případně action form.

Pokud budeme hledat například odkaz `<index.php?pm_section=login>`, pak pro nás bude důležitá následující část konfiguračního souboru:

```
<!-- Login -->
<action path="login"
        type="LoginAction"
        name="LoginForm"
        validate="true"
        input="login.tpl">
  <forward name="login-form" path="login.tpl"/>
  <forward name="login-success" path="welcome.tpl"/>
  <forward name="logout" path="welcome.tpl"/>
</action>
```

Odtud můžeme vyčíst, že action class se jmenuje *LoginAction* a máme zde nadefinovanou i třídu action form jménem *LoginForm*.

### Template podle URL adresy

Tato otázka je o něco běžnější, ale o dost zapeklitější. Nejprve totiž musíme zjistit, která action class zpracovává požadavek. V této třídě pak musíme určit výsledek vyhodnocení a zjistit návratovou hodnotu funkce `execute`. Pak se můžeme vrátit zpět ke XML souboru, kde podle třídy a návratové hodnoty zjistíme template soubor.

V předchozím příkladě je validace vstupních dat zapnutá, tudíž formulář bude zkontrolován ještě před vytvořením instance *LoginAction*. Pokud vstupní data nebudou vyhovovat požadavkům, pak se zobrazí rovnou stránka uvedená jako

hodnota atributu *input*. Elementy *forward* určují stránky použité při různých výsledcích metody *execute*.

### **Chyby v programu**

Program je teprve v testovací fázi, proto jsou chyby očekávané. Obsahuje také nástroje na jejich zaznamenávání a hledání. Hlavním nástrojem je zaznamenávání pohybu uživatelů na stránkách. K tomu je ještě testerům dán k dispozici formulář na nahlášení chyby na konkrétní stránce.

Sledování historie uživatele se dá provádět na 3 úrovních. Buď vidíme jen navštívené stránky, nebo k nim je přidán ještě výpis průběhu programu (trace). Ten přidá informace o tom, jaké funkce byly spouštěny a jak skončily. Pokud by ani tato informace nestačila k odhalení příčiny problému, pak je zde ještě možnost výpisu tzv. debug informací. To obsahuje všechny SQL příkazy včetně výsledků, data předávaná podprocedurám a mnoho dalších informací. Tato varianta ale velice rychle zabírá diskový prostor. Třetí případ je také jediný, který odhalí uživatelská data zadaná do formulářů na stránkách.

Verze programu přiložená na CD má nastavené pouze zaznamenávání navštívených stránek. Toto nastavení se dá upravit změnou inicializace logů v konstruktoru *AbstractBaseClass*. Rozlišení uživatelů je provedeno tak, že pro každou IP adresu je vytvořen nový logovací soubor.

Pokud uživatel nalezne chybu a vyplní příslušný formulář, pak se jeho poznámka objeví v jeho logu u naposledy navštívené stránky. Není tedy velký problém chybu reprodukovat, protože známe historii navštívených stránek. Problém nastává v případech, kdy dva uživatelé se stejnou IP adresou přistupují na různé části systému zároveň, nebo pokud je chyba závislá na datech v nějakém formuláři. Tyto informace by nám uživatel musel sdělit.

# Kapitola 7

## Vyhodnocení

Webové rozhraní pro vedení softwarových projektů s důrazem na správu lidských zdrojů se ukázalo být projektem o rozsahu několika bakalářských prací. Původním předpokladem bylo, že se podaří zakomponovat do mé práce nějaký z existujících wiki systémů a jedno z mnoha fór. Za těchto okolností nebyly požadavky nijak přehnané ani příliš skromné.

První předpoklad byl splněn i když ani to nebylo bez problémů. Druhý předpoklad bohužel selhal zcela., což se podepsalo na výsledku mé práce. Vytvořit fórum je podle mého názoru téma na samostatnou práci.

### 7.1. Funkčnost

Aplikace je stále ve fázi testování, takže se v ní dají očekávat chyby. Nicméně vzhledem k původním požadavkům podporuje následující funkce:

- uživatelské účty
  - registrace uživatele
  - editace nastavení
  - zasílání zapomenutých hesel
  - ověření zadaných emailových adres
- projekty
  - vytváření projektů
  - editace atributů
  - možnost propojení na Subversion
  - rušení projektů
- pracovní týmy
  - přiřazení uživatelů k projektům
  - změny v tomto přiřazování
  - určení funkce a práv člena týmu vzhledem k projektu
- systém pro správu a údržbu dokumentace
  - jednoduchý způsob editace dokumentace
  - možnost úpravy dokumentace několika uživateli z různých počítačů
- komunikace mezi uživateli
  - fórum pro členy projektu (nepřístupné ostatním)
  - fórum určené pro obecnou diskuzi k projektu
  - globální fórum nezávislé na projektu
- Jednoduché použití
  - přímočaré funkce bez vedlejších závislostí
  - funkce systému může uživatel využívat volně



- funkce systému uživatel nemusí používat
- dostupnost pomocí internetového prohlížeče
- žádné poplatky za používání nebo instalaci
- snadná instalace na většinu webových serverů
  - často používané technologie
  - nízké nároky na prostor a výpočetní výkon
  - kompaktnost

Naopak nejsou explicitně naimplementovány následující funkce. Některé z nich však jdou provozovat na podporovaných funkcích, jak bylo popsáno např. u burzy práce v kapitole 5.2

- komunikace mezi uživateli
  - osobní zprávy (private messages)
- burza práce
  - nabídka pracovní síly
  - poptávka po pracovnících
  - vyhledávání v nabídce i poptávce

## 7.2. Srovnání s existujícími projekty

Tabulka 3 hodnotí tentokrát 6 aplikací určených pro správu projektů. Počet kladných nebo záporných znamének určuje, jak moc aplikace v dané oblasti požadavků uspěla podle mého názoru.

	Source-Forge	Trac	Mantis	dotProject	Best Practical	Projektový manažer
<b>uživatelé</b>	+	+	+	+	+	+
<b>projekty</b>	— / ++	+	—	+	—	+
<b>týmy</b>	++	—	— —	++	+	+
<b>dokumentace</b>	++	+	+	+	+	+
<b>komunikace</b>	++	+	—	+	—	+
<b>burza práce</b>	+	—	— — —	— — —	— — —	—
<b>jednoduchost</b>	?	+	+	—	— — —	+
<b>dostupnost</b>	+	+	+	+	+	+
<b>instalace</b>	— — —	—	++	+	— — —	++
<b>použitý jazyk</b>	?	—	+	+	—	+
<b>databáze</b>	?	+	+	+	+	+

tabulka 3: Srovnání existujících řešení s implementací Projektového Manažeru

Jak je vidět z tabulky 3, Projektový manažer je nejvíce podobný aplikaci dotProject. Nicméně jak je napsáno již v kapitole 3.2, toto řešení poskytuje velice sofistikované univerzální nástroje a vynucuje jejich používání.

Projektový manažer se naopak snaží nebýt na obtíž a nevynucovat od uživatelů žádné specifické postupy. Je tedy jakousi odlehčenou alternativou určenou především pro softwarové projekty. Softwarovým projektům totiž většinou nebudou chybět pokročilé nástroje a navíc mají v Projektovém manažeru rozhraní pro přístup k SVN úložišti, což dotProjectu chybí.

# Kapitola 8

## Závěr

### 8.1. Shrnutí

Projektový manažer je funkční testovací verze systému použitelného pro vedení softwarových projektů s důrazem na správu lidských zdrojů. V tomto ohledu splnil přesně zadání práce. Poskytuje nástroje použitelné pro plánování a spolupráci více lidí na společném cíli. Přitom nenutí žádná přesná řešení a nechává volnost zavedeným pravidlům jednotlivých týmů. Aplikace má webové rozhraní a je snadno použitelná na většině webových serverů a dokonce i na některých freehostingových serverech. Instalace nevyžaduje žádné pokročilé znalosti administrace webových serverů, databází, ani programování.

Speciálně pro vedení softwarových projektů je zde experimentálně zaveden modul pro SVN klienta. Tento modul umí číst z libovolného SVN úložiště a to i tehdy, pokud je chráněno heslem. Vzhledem k jeho experimentálnímu charakteru nepodporuje jiné operace než výpis adresářů a souborů. Pro další operace by bylo potřeba definovat podstatně sofistikovanější systém práv.

Jako aplikace v testovací fázi má u sebe i nástroje na hlášení chyb a zaznamenání způsobu jejich vzniku. Každý uživatel se tak může podílet na dokončení a zlepšení systému.

Systém bohužel nesplnil očekávání vkládaná do inovativního způsobu hledání spolupracovníků. Modul burzy práce úplně chybí kvůli nedostatku času na jeho implementaci. Z toho si mohu vzít ponaučení pro příště. Důkladně ověřím možnosti propojení externích aplikací a případné zvolení kompatibilního základu celého systému. Zde bylo osudným to, že žádná z databázových aplikací nešla upravit pro potřeby frameworku phpMVC.

### 8.2. Možnosti dalšího vývoje

Základem dalšího vývoje je odladění existující aplikace a doplnění o chybějící moduly. Poté je velice perspektivní rozšíření možností SVN klienta například o pravidelné exporty určité části úložiště.

Pro konkrétní nasazení v nějaké společnosti nebo ve škole bude výhodou i propojení databáze uživatelů s původním systémem. Na naší fakultě by to například mohlo být ověření uživatele ne pomocí zaslání emailu, ale pomocí údajů z centrální databáze LDAP. Pokud by byl systém někdo ochotný dlouhodobě udržovat, pak by jeho použití na fakultě zjednodušilo studentům hledání týmů a témat na softwarové projekty.

Možnosti rozšíření jsou zřejmě omezeny jen lidskou představivostí a schopností věnovat se dalšímu rozvoji.

# Kapitola 9

## Literatura

- [1] Wikipedie: Otevřená encyklopedie: *Řízení projektů* [online].  
© 2008 [cit. 16.7.2008]. Dostupný z WWW:  
<[http://cs.wikipedia.org/w/index.php?title=%C5%98%C3%ADzen%C3%AD\\_  
projekt%C5%AF&oldid=2771537](http://cs.wikipedia.org/w/index.php?title=%C5%98%C3%ADzen%C3%AD_projekt%C5%AF&oldid=2771537)>.
- [2] [subversion.tigris.org](http://subversion.tigris.org) - *Subversion: an open source version control system*  
[online]. © 2006 CollabNet [cit. 1.8.2008] Dostupný z WWW:  
<<http://subversion.tigris.org/>>.
- [3] HOSTINGY.CZ: *Přehled českých free hostingů* [online].  
© 2003 – 2008 hostingy.cz [cit. 1.8.2008] Dostupný z WWW:  
<<http://www.hostingy.cz/webhosting-zdarma/>>.
- [4] PHP: *Hypertext Preprocessor* [online]. © 2001-2008 The PHP Group  
[cit. 1.8.2008] Dostupný z WWW: <<http://www.php.net/>>.
- [5] MySQL :: *The world's most popular open source database* [online].  
© 1995-2008 MySQL AB [cit. 1.8.2008] Dostupný z WWW:  
<<http://www.mysql.com/>>.
- [6] SourceForge.net: *Open Source Software* [online]  
© 1999 – 2008 SourceForge, Inc. [cit. 2.8.2008] Dostupný z WWW:  
<<http://sourceforge.net/>>.
- [7] SourceForge.net: *SourceForge Enterprise Edition Download* [online]  
© 2008 SourceForge, Inc. [cit. 2.8.2008] Dostupný z WWW:  
<<http://sf.net/powerbar/sfee/>>.
- [8] *The Trac Project* – Trac [online] © 2003 – 2008 Edgewall Software  
[cit. 2.8.2008] Dostupný z WWW: <<http://trac.edgewall.org/>>.
- [9] *Mantis Bug Tracker* [online] © 2008 skupina dobrovolníků  
[cit. 2.8.2008] Dostupný z WWW: <<http://www.mantisbt.org/>>.
- [10] dotProject - *Open Source Software :: Open Source Project and Task  
Management Software* [online] © 2008 skupina dobrovolníků  
[cit. 2.8.2008] Dostupný z WWW: <<http://www.dotproject.net/>>.
- [11] Best Practical Solutions, LLC [online] © 2002-2008 Best Practical Solutions  
LLC. [cit. 2.8.2008] Dostupný z WWW: <<http://bestpractical.com/>>.

- [12] *The Model View Controller Framework for PHP Web Applications* [online] © 2008 [cit. 2.8.2008] Dostupný z WWW: <<http://www.phpmvc.net/>>.
- [13] Wikipedie: Otevřená encyklopedie: *Model-view-controller* [online]. © 2008 [cit. 7. 08. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=Model-view-controller&oldid=2893766>>.
- [14] PWP Wiki Processor » *StartPage* [online] © 2008 Lars Ackermann [cit. 3.8.2008] Dostupný z WWW: <<http://www.ackermann-consulting.de/wiki/static>>.
- [15] *wiki:dokuwiki* [DokuWiki] [online] © 2008 [cit. 3.8.2008] Dostupný z WWW: <<http://wiki.splitbrain.org/>>.
- [16] phpBB.cz • *Portál* [online] © 2000, 2002, 2005, 2007 phpBB Group [cit. 3.8.2008] Dostupný z WWW: <<http://www.phpbb.cz/>>.
- [17] [websvn.tigris.org](http://websvn.tigris.org) - *PHP based web interface of Subversion repositories* [online] © 2006 CollabNet [cit. 3.8.2008] Dostupný z WWW: <<http://websvn.tigris.org/>>.
- [18] PEAR :: Package :: *VersionControl\_SVN* [online] © 2001-2008 The PHP Group [cit. 3.8.2008] Dostupný z WWW: <[http://pear.php.net/package/VersionControl\\_SVN/](http://pear.php.net/package/VersionControl_SVN/)>.

# Příloha A

## Obsah příloženého CD

- Adresář **Project\_manager** obsahuje soubory programu včetně instalátoru.
- V souboru **read.me** jsou popsány pokyny k instalaci a prerekvizity potřebné k úspěšnému provozování aplikace.
- Soubor **Bc.doc** je elektronická verze tohoto dokumentu
- Adresář **webserver** obsahuje podadresáře **win** a **linux**. Oba obsahují prerekvizity potřebné pro spuštění aplikace.
  - V adresáři **win** se nachází instalační soubory serveru Apache, interpretu jazyka PHP a MySQL serveru. Pro vyzkoušení všech vlastností programu je doplněn i spustitelný soubor SVN klienta. Postup instalace je opět popsán v souboru **read.me**.
  - Adresář **linux** neobsahuje instalační soubory, ale zdrojové kódy jednotlivých programů. Programy je třeba zkompilovat. Soubor **read.me** zde obsahuje odkazy na stránky s příslušnými postupy.