



Posudek oponenta závěrečné práce

Student: Ondřej Máca
Oponent práce: Ing. Pavel Štěpán
Název práce: Systém pro zpracování firemní listové pošty
Obor: Softwarové inženýrství

Datum vytvoření: 27. 1. 2019

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – následující škálou 1 až 4:</i>
1. Splnění zadání	1=zadání splněno, 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno
<i>Popis kritéria:</i> Posuďte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posuďte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.	

Komentář:

Úkolem bylo znovuvytvořit systém pro zpracování firemní listové pošty dle dosavadní aplikace použitím moderních technologií tak, aby se zlepšila udržitelnost a rozšiřitelnost systému. To autor v zásadě udělal.

Za hlavní problém považuji to, že se autor nechal fascinovat vidinou automatického převodu s použitím analyzátorů (parserů) PHP kódu. Nakonec ale i on sám připouští (v kapitole Realizace), že přínos automatického převodu nebyl příliš velký. Výsledky automatického převodu (který byl – dle práce – použit v podstatě na převod dvou switch na controllery) se stejně musely manuálně upravovat, byly problémy s některými identifikátory a samozřejmě bylo třeba vytvořit skripty pro použití analyzátoru. Přitom většinu kódu musel převést manuálně – stránky PHP s HTML na šablony Angular, příkazy TSQL na DQL atd.

Při tvorbě programu se samozřejmě může stát, že autor použije nějakou technologii nebo postup, který se ukáže nevhodný. Ale zde se domnívám, že bylo možné se od začátku této v podstatě slepé uličky vyhnout. Hlavním problémem programů, které se mají modernizovat, je obvykle jejich nevhodná struktura, kterou je třeba změnit. Takový úkol ale v podstatě znamená napsat program znovu s tím, že některé části (funkce, objekty, ...) lze použít i v novém systému. Zde však těžko může pomoci nějaký analyzátor. Ten je jistě schopen rozebrat program s danou syntaxí a zobrazit jej jiným způsobem (možná vhodnějším pro zpracování). Ovšem takto nelze řešit zásadnější restrukturalizaci programu. Např.: autor uvádí, že dosavadní program obsahuje mnoho duplicit kódu. S těmi ovšem analyzátor těžko pomůže – ty „duplicity“ budou asi stěží identické – může být jiné pořadí příkazů, použity jiné operátory a jiné identifikátory, takže analyzátor zřejmě nepozná, že jde o totéž. „Duplicity“ je nutno vyhledat manuálně a vhodně začlenit (po případných úpravách) do nového programu. Podobné problémy budou vznikat i při dalších operacích pro refaktorizaci programu.

Pokud by se tedy autor vyhnul snaze o automatizaci, mohl získat čas na provedení úprav, které považuji za podstatně důležitější. Jde např. o to, že se snažil (co nejdělečněji) o zachování současného kódu, jak jen to bylo možné. To vyplývá již z toho, že autor popisuje v úvodu, že např. adresář s CSS soubory ponechá beze změny. Dále se – až úpěnlivě (navíc pomocí automatické konverze) - snaží zachovat controllery (a šablony na straně klienta), odpovídající stránkám (resp. URL adresám) původní aplikace. Podobně na straně servera přepis SQL na DML – ve většině případů opět snaha o převod 1:1. Tedy žádná snaha o úpravu vzhledu aplikace (stejně CSS), stejné „stránky“ (controllery a šablony) místo snahy o změnu struktury, funkčnosti a počtu stránek (či vizuálních interfaces) či úprava serverových algoritmů jinak, než pouhým převodem na nové technologie apod. Samozřejmě pro uživatele je příjemné, když se aplikace vizuálně a ovládáním nemění. Ale vzhledem k tomu, že se jedná o starší aplikaci, jistě si již vyžaduje určitou (třeba ne revoluční) změnu jak v klientské části, tak v serverové. K tomu ale nedošlo.

Dále je poněkud zarážející fakt, že autor uvádí, že použitá databáze má poněkud problematickou strukturu. Její úprava by tedy měla být jedním z prvních kroků při přechodu na nový systém, který má být dobře udržovaný a rozšiřovaný. K tomu však autor pouze konstatuje, že „to nebylo cílem jeho práce“.

A k databázi se váže i otázka, zda bylo vhodné použití ORM knihovny Doctrine, která nepodporuje tak běžné databázové konstrukce jako views a stored procedures. Na druhé straně je třeba říci, že si autor s tímto problémem poradil, i když poněkud pracným způsobem.

Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

2. Písemná část práce

75 (C)

Popis kritéria:

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti. Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

Komentář:

Autor velmi stručně popsal svoji práci – zhruba na dolní hranici přijatelnosti. Jako příklad: relativně hodně místa věnuje automatizaci převodu programu pomocí analyzátoru, ale jeho činnost ukazuje na jediném příkladu – analýze příkazu echo 'Hi', 'World'; (i když textu je zde poměrně hodně – různé tabulky funkčních a nefunkčních požadavků, controllerů atd., což však k objasnění vlastní práce příliš nepřispívá).

Navíc zde zcela chybí vysvětlení, zda tato práce pouze předvádí na příkladě existující aplikace možnosti přechodu na nové technologie, nebo zda jde opravdu o reálnou aplikaci, která se používá (resp. bude používat) ve skutečném provozu. A pokud jde o druhý případ, jaké jsou zkušenosti z praktického provozu.

Po jazykové stránce: student se (ve srovnání s průměrnou úrovní prací studentů FIT) vyjadřuje poměrně dobře. Práce však obsahuje menší chyby typu překlepů a přehlédnutí a bohužel i určitý počet hrubek (tvrdé/měkké y/i – „vsjo ravno“).

Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

3. Nepísemná část, přílohy

75 (C)

Popis kritéria:

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů

Komentář:

Zhodnocení práce je v uvedeno v bodě 1

Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

4. Hodnocení výsledků, jejich využitelnost

85 (B)

Popis kritéria:

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

Komentář:

Hodnocení výsledků je dosti obtížné, protože jde o refaktoring specializovaného řešení, které neznám. Přitom to, zda bylo dosaženo cíle (zejména kritéria udržitelnosti a rozšiřitelnosti systému) se projeví až v praxi při požadavcích na změny a úpravy systému (kterým by tato práce měla vycházet vstříc). Autor se však snažil tyto možnosti podpořit tím, že oddělil kód programu od HTML, postavil aplikaci na využití frameworků Symfony a Angular, aplikaci strukturoval jako SPA apod.

Na druhé straně se zdá, že se příliš držel stávající aplikace a formálních stránek převodu, takže do určité míry propásł možnost skutečné modernizace dosavadního řešení.

Hodnotící kritérium:

Způsob hodnocení – nehodnotí se

5. Otázky k obhajobě

Popis kritéria:

Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

Otázky:

- 1) Popište použití analyzátoru ve Vašich skriptech
- 2) Proč jste se rozhodl pro dané technologie
- 3) Co považujete za největší přínos při tvorbě práce
- 4) Jaké jsou největší rozdíly mezi starým a novým systémem

Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

6. Celkové hodnocení

75 (C)

Popis kritéria:

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.

Text hodnocení:

K práci je možno mít dost výhrad – viz bod 1. Na druhé straně se ale autor samostatně seznámil s mnoha moderními web technologiemi a provedl převod dosavadního systému, což bylo jistě poměrně náročné. Tím se mu podařilo dosáhnout značné technologické modernizace stávající aplikace.

Podpis oponenta práce: