

POSUDEK OPONENTA DIPLOMOVÉ PRÁCE

Student: Zuzana Ondřej, Bc.

Oponent: Mgr. Tomáš Radil

Studijní program: **Inženýrská informatika**

Studijní obor: **Informační technologie**

Akademický rok: **2016/2017**

Téma diplomové práce: **Vizualizace běhových informací vyvíjených aplikací v prostředí IDE CodeLite**

Hodnocení práce:

Práce je správně strukturována a je rozdělena na teoretickou část sloužící jako úvod do zkoumané problematiky, část sloužící jako rešerše nástrojů pro dynamickou analýzu, a na poslední, nejdůležitější praktickou část.

V teoretickém úvodu oceňuji autorův technicky strohý, ale zároveň velmi srozumitelný styl, který dokáže posloužit jako úvod do zkoumané problematiky.

Ve druhé části oceňuji, že se autor kromě specializovaných opensource nástrojů pro prostředí Linux, které velmi dobře slouží na specializované oblasti statické analýzy zaměřil i na vývojová prostředí (IDE), které integrují všechny potřebné analytické funkce společně s nástroji pro softwarový vývoj do komplexních celků. V komerční praxi se totiž při vývoji rozsáhlejších informačních systémů vyvíjených v týmech setká spíše s jejich použitím.

Autor ovšem pojal tuto část spíše jako výčet možností a zapomněl trochu na smysl rešerše, kterým je nalezení nejlepších řešení. Popisuje zběžně analytické nástroje a IDE, ale nesnaží se analyzovat, v čem je ten který nástroj lepší nebo horší než ostatní. Autor se mohl soustředit na konkrétní programovací jazyk a zkusit najít nejvhodnější řešení z hlediska dynamické analýzy pro vývojáře.

Mohl se např. zaměřit na jazyk Java, který zmiňuje nejčastěji, pro nějž mohl zkusit doporučit vhodnou kombinaci IDE a externích nástrojů. Případá mi to praktičtější než stručný soupis vlastností, protože výběr vhodného implementačního řešení formou podobné rešerše bude často podstupovat, pokud se stane programátorem nebo IT managerem.

Jak sám autor v závěru práce připouští, nepustil se příliš pod povrch problematiky a zájemcům o nástroje dynamické analýzy pouze naznačil směr, kde hledat konkrétní řešení jejich potřeb, i když je potřeba říct, že důkladná analýza této problematiky překračuje rámec jeho diplomové práce. Stručnost popisu vlastností zkoumaných nástrojů naštěstí autor vyvažuje vhodně zvolenými ilustračními obrázky, které jsou názornější než slovní popis.

V praktické části pozitivně hodnotím autorův přístup zvolenou metodou **Waterfall (Vodopád)**.

1) Začal správně **Soupisem požadavků**.

V úvodu praktické části autor správně rozepsal požadavky na vyvíjený analytický plugin pro CodeLite a pokusil se je víceméně úspěšně abstrahovat do scénářů (Use Cases).

2) Pokračoval dalším stupněm Vodopádu **Designem**.

Formou jednoduchých UML diagramů názorně vysvětlil způsob fungování výsledného produktu.

3) Dále se autor zaměřil na **Implementaci**.

Popsal stručně použité nástroje jak pro prostředí Linux, tak i Windows, vytvořil postačující uživatelskou dokumentaci s popisem sestavení dodaných zdrojových kódů, nastavení pluginu.

4) Závěr technické části se stručně zabývá vývojovým stupněm Waterfalů **Verifikace**, kde autor popisuje způsob použití pluginu pro dynamickou analýzu software. Nestanovil si ale, bohužel, žádná verifikační pravidla pro ověření správnosti získaných dat ani funkčnosti pluginu. Tady asi spoléhá na použití knihovny SIGAR, ale zapomíná, že jeho implementace může obsahovat chyby a nepokrývá svůj kód žádným unit testem pro ověření správnosti. Unit testy jsou důležité taky pro budoucí údržbu aplikace.

Mám drobnou výhradu k autorovu přístupu dodat jen samotné zdrojové kódy. Myslím, že měl poskytnout i přeložené binární kódy alespoň pro jedno prostředí (Windows nebo Linux), aby si kdokoliv mohl ihned bez další námahy vyzkoušet výsledek jeho práce. Protože dodal jen zdrojové kódy, očekával bych v instalačním zip balíčku i všechny opensource komponenty v použité verzi (CodeLite, wxWidgets, gcc kompilátor), aby byly zdrojové kódy i s delším odstupem spolehlivě přeložitelné.

Praktická část splňuje, stejně jako předchozí části, požadavky a cíle diplomové práce. Po jazykové stránce je práce vyhovující, pouze občas se vyskytnou nevhodně zvolená slova. Po stránce formy oceňuji vhodné použití obrázků a diagramů. Práce je ovšem místy příliš strohá a mohla by obsahovat vhodně zvolené odstavce, které propojují jednotlivé kapitoly do kompletního vyváženého celku.

Celkové hodnocení práce:

Známku uvede oponent dle svého uvážení dle klasifikační stupnice ECTS:

A – výborně, B – velmi dobře, C – dobře, D – uspokojivě, E – dostatečně, F – nedostatečně.

Stupeň F znamená též „nedoporučuji práci k obhajobě“.

Předloženou diplomovou práci doporučuji k obhajobě a navrhuji hodnocení

B - velmi dobře.

V případě hodnocení stupněm „F – nedostatečně“ uveďte do připomínek a slovního vyjádření hlavní nedostatky práce a důvody tohoto hodnocení.

Datum 23.5.2017

Podpis oponenta diplomové práce