



Posudek oponenta bakalářské práce

Petr Tománek

Efektivita technic pro implementaci byznys pravidel v informačních systémech

Práce si bere za cíl řešit reprezentace byznys logiky v podnikových systémech, implementaci případové studie, porovnávací přístupy a dále shrnutí výhod jednotlivých postupů.

Již z úvodní kapitoly je patrný následný obsah práce, který lze označit za chaotický, neorganizovaný, vágní, a mnohdy nepřesný či nepodložený. Ku příkladu kapitola říká, že se práce bude zabývat dynamickou webovou aplikací, což ovšem není obsahem zadání ani obsahu. Nebo dále, že webové aplikace jsou nesmírně složité, velmi rozsáhlé, apod.

Kapitola specifikace požadavků práce chybí a lze konstatovat, že zpracování požadavků je nedostačené. Následuje kapitola analýza, která opět prezentuje nepodložená tvrzení, což ukazuje například první odstavec. Práce zmiňuje techniku „neoptimalizovaného programování“ a dává ji v porovnání s objektovým programováním, avšak odkaz na zdroj takového programování chybí. Mám tedy první otázku (1) co to je neoptimalizované programování a odkud to pochází, je možné citovat zdroj? Úvahové skoky popisované v kapitole, které říkají např. že problematika je řešena v JSR 316, jsou nesouvisející a postrádají logiku. Dále se zmiňuje že Beans Validation validuje prvky, avšak není řečeno nic více či jaké prvky, následuje skok na CDI, a nešťastné řešení JSR 303, čtenář z textu jednotlivé návaznosti nemůže vyvodit. Následuje podkapitola řešení v prezentační vrstvě, avšak práce nezmiňuje co to je vrstva ani jak souvisí s analýzou a problémem. Analýza dále prezentuje požadavky a databázi obojí však zjevně nepatří do kontextu byznys logiky a zadání. Požadavky, i pokud by byly k tématu, nelze zachytit jen use-case diagramem. Databáze nepatří do analýzy, ale do návrhu. Dále kapitola zmiňuje, že pro vývoj web aplikací se používají dvě architektury a to 3-tier a MVC. Avšak druhá zmíněná je návrhových vzor pro tvorbu uživatelského rozhraní a interakci s modelem, dále pak je tvrzení milné, jelikož existují desítky dalších architektur. Opět téma nepatří do analýzy.

Kapitola řešerše se snaží říci, co je to dědičnost, abstrakce a další, avšak chybně, což lze hodnotit jako nedostatečnou znalost předmětu algoritmizace a úvodu do programování. Obsahově hodnotím kapitolu nedostatečně.

Kapitola „řešení pravidel v service“ říká že „ve světě programování se nalézá mnoho technik, jak problém vyřešit a naprogramovat“. Dále kapitola prezentuje jednotlivé autorem definované příklady řešení jakéhosi kódu třídy Comment. (2) O co přesně se jedná a kde je třída definována? První příklad neoptimalizovaného programování říká že je to jednoduché na tvorbu a dobře čitelné, ovšem pokud je to dobře čitelné mělo by to mít o dobrou kohezi a nízký coupling jinak je to v rozporu, (3) proč to tedy máme vylepšit? Diskutované

metaprogramování nepracuje na principu Proxy a Decorator, jak uvádí práce. Pomocí metaprogramování se Proxy vytváří a metaprogramování využívá vzoru reflexe. Zmiňování ukázka není na metaprogramování, ale na JSR 318 a tedy Java EE Interceptor. V této kapitole bych čekal nějaké vysvětlení proč, co, kdy, jak.. ale není tu. Práce se omezuje jen na platformu Java, jiné platformy problém neřeší, co třeba C#, MDA, a jiné?

Následující případová studie je vytvořena autorem nebo vedoucím práce (4)? Dle datového modelu je patrné, že je převzata z příkladu uvedeném ve vědeckém článku odeslaném na FedCSIS 2014 konferenci (Enterprise Information Systems: Comparison of Aspect-driven and MVC-like Approaches). Práce v textu popisuje, že „pro tuto práci je třeba vytvořit příkladovou studii“, což je neurčité zda to udělal autor nebo někdo jiný? V textu je opět demonstrována neznalost z předmětu Softwarové Inženýrství, když je tvrzeno že Kontrolér z MVC odpovídá Business vrstvě ve třívrstevném modelu, což je nepravdivé, řada dalších nedostatků zasahuje do analýzy a návrhu.

Z měření pak vyplívá, že objektové programování je plně dostačující a není třeba žádný interoceptor nebo aspektově orientované programování (5)? Co je vlastně závěrem a přínosem práce pro budoucí rozvoj a na základě čeho je to podloženo (6)?

Z pohledu na celou práci, nejsou řádně vysvětleny zmiňované termíny, není použito zkratk, práce nedostatečně cituje, vlastní prezentace práce je nedostatečná, není patrné jaké jsou požadavky, odkud přicházejí některé názvy či postupy, rešerše je mimo kontext zadání. Dle toho usuzuji, že není dostatečně splněné zadání a navíc práce obsahuje řadu nepravd.

Z výše uvedeného doporučuji znovu absolvovat předměty programování, tak aby došlo vyjasnění termínu dědičnost, především ale předmětu softwarové inženýrství, jelikož práce demonstruje významné nedostatky ať už z pohledu architektury, analýzy nebo návrhu.

Dle výše uvedeného musím navrhnout hodnocení:

F (neprospěl)

Dále doporučuji zadání nové práce s odlišným tématem



Ing. Tomáš Černý, M.S.
Oponent práce

V Praze 7. 6. 2015