

Posudek oponenta bakalářské práce

Autor práce: **Lukáš Tancer**

Název práce: **TestRunner – softwarový prostředek pro správu testů**

Obsah práce:

Tato bakalářská práce byla zadána oddělením kontroly kvality software firmy Kerio Technologies, s.r.o. Cílem práce je úprava aplikace TestRunner, která umožňuje testerům společnosti vytvářet testovací scénáře, zaznamenávat výsledky testů prováděných podle těchto scénářů, a tyto výsledky archivovat.

Na začátku práce (po zadání, prohlášení, abstraktu, obsahu a úvodu) popisuje autor problematiku testování software, přechází k agilní metodice vývoje software Scrum a pokračuje popisem role testování ve Scrum. Následně se věnuje testování software podrobněji a zmiňuje a vysvětluje všechna důležitá rozdělení (manuální vs. automatické, black- vs. white- vs. gray- box) a typy testů (unit, funkční, akceptační, atd.). Popsány jsou vývojové fáze software (pre-alfa, alfa, beta, RC). V závěru teoretické části autor popisuje atributy chyb (závažnost, priorita) a životní cyklus chyby.

Implementační část začíná stručným popisem aplikace TestRunner a pokračuje zjednodušeným dokumentem specifikace požadavků, následuje analýza problému, výčet chyb určených k opravení, výčet a popis funkcí určených k odebrání/redukci. Na konec je uveden výčet a popis funkcí, které mají být implementovány.

Následuje kapitola o testování aplikace TestRunner, ve které autor popisuje, jakým způsobem zkontroloval bezchybnost jím provedených změn. Následuje závěr, přehled zkratk, seznam obrázků, seznam použité literatury, a uživatelský manuál aplikace TestRunner. Přiložené DVD obsahuje zdrojové soubory práce v LaTeX, konverzní skripty pro databázi aplikace TestRunner, zdrojové kódy aplikace TestRunner, a soubor README.txt, ve kterém je popsáno, co a kde se na DVD nachází.

Kvalita řešení a dosažených výsledků:

Práce je do jednotlivých kapitol dobře logicky členěna. Jen v kapitole 2 *Problematika testování* se přechází od teorie testování software k agilním metodikám a zpět k testování, do nějž se mísí popis fází vývoje software. I přesto je ale text pochopitelný a čtenář je ve stručnosti seznámen s teoretickým rámcem nutným pro pochopení procesu testování software.

Kvalita použité angličtiny v abstraktu je, dle mého názoru a mé úrovně jazykových znalostí, nižší, než si abstrakt bakalářské práce zaslouží. Obsah je zpracován dobře a drží se typografických konvencí. Úvod je po slohové stránce podprůměrný, nicméně svou úlohu plní s tou výjimkou, že z něj není zřejmé, k čemu je aplikace TestRunner dobrá.

Kapitola 3 *TestRunner* je věnována popisu funkčnosti aplikace, jejíž úprava byla cílem implementační části hodnocené práce. Stručný popis funkčnosti aplikace, motivace k provedeným úpravám i samotné úpravy jsou ve stručnosti popsány. Škoda, že místo odstavce popisujícího funkčnost aplikace TestRunner odkazuje autor na přečtení 29stránkové uživatelské dokumentace. Nechybí specifikace požadavků a popis aktuálního stavu aplikace včetně běhového prostředí. Stručnost hodnotím v tomto případě kladně. Analýza problému je slabší. Popis datové vrstvy je nahrazen ERA modelem databáze, ve kterém chybí popisy vazeb a jejich kardinality. V aplikační vrstvě je mnoho tříd, jejichž název

obsahuje suffix `_form`. Nejen z názvů, ale i popisů těchto tříd je zřejmé, že patří do prezentační vrstvy. Ta není zmíněna vůbec. Seznam opravených chyb je dostatečně popsán. Následující popis nově přidaných funkcí považuji za zdařilý. Ke každé funkci uvádí autor stručnou analýzu problému a popisuje, jakým způsobem byla funkčnost implementována. U těch z nich, které jsou viditelné přímo v GUI aplikace, jsou rozdíly ukázány na obrázcích porovnávající původní a nové – autorem implementované – řešení. Se způsobem otestování výsledné verze aplikace TestRunner, popsaném ve čtvrté kapitole, jsem spokojen. Závěr práce velmi dobře shrnuje odvedenou práci a plyne z něj, že všechny body zadání práce byly splněny.

V celé práci má autor tendenci používat nepřesná vyjádření: „velké množství chyb“, „pozdější fáze vývoje“, „u mě na počítači pravidelně více než 10s“, atd. Nepřesná vyjádření někdy vedou k nepravdivým tvrzením, ačkoliv jsem přesvědčen, že na přímý dotaz by s uváděnými tvrzeními autor nesouhlasil: „Předtím, než může produkt postoupit do alpha verze, je třeba ověřit, zdali jsou opraveny všechny nahlášené chyby.“, „Jakákoliv další vylepšení se už nedělají“ vztaheno k produktu, který podstupuje akceptační testování.

Formální úroveň:

Sazbu práce považuji za vydařenou s následujícími výhradami: některé obrázky tlačítek vkládané přímo do textu odstavců v uživatelské příručce jsou nevhodně zvětšené a tím pádem rozmazané (např. na straně s kapitolou 6.2 Odeslání výsledku), uživatelské příručky chybí čísla stránek, ačkoliv jsou uvedena v jejím obsahu. Na konci výtisku práce, kterou jsem měl k dispozici, je 5 prázdných listů. V kapitole 3.9.3 je odkázáno na obrázek bez uvedení jeho čísla. Kvalita tisku Obr. 3.1 je podprůměrná.

Jazykovou úroveň považuji za slabší. Počtem chyb práce nevystupuje z průměru.

Práce s literaturou:

Použitá literatura obsahuje zejména příručky a návody k technologiím/aplikacím/knihovnám/protokolům, které autor použil při vypracování implementační části práce, což je vzhledem k jejímu charakteru pochopitelné. Krom toho je práce opřena o další jednu knihu, jeden odborný článek a jednu externí přednášku uskutečněnou v rámci předmětu KIV/ZSWI. Ocenil bych více literatury z oblasti teorie testování software, o něž by se opírala teoretická část. Veškerá použitá literatura pochází od důvěryhodných autorit/organizací/portálů a nelze ji tedy jakkoliv zpochybnit.

Splnění zadání:

Splněno bez výhrad.

Dotazy k práci:

- Objevil jste v návrhu aplikace TestRunner nějaké chyby, které Vám komplikovaly implementaci změn, které jste měl dle zadání provést?
- Odpovídá zdrojový kód aplikace TestRunner teorii, která je přednášena na KIV/FAV/ZČU (konvence pojmenování proměnných a metod/funkcí, komentáře v kódu, jednotný styl psaní kódu, odsazování, ošetřování chybových stavů, úsporný kód, atp.)?
- V kapitole 2.3.2 *Automatické testování* uvádíte „Bohužel ani automatické testy nemohou testovat všechno, a tak je nelze používat jako jedinou metodu testování.“. Co tedy nemůže být pokryto automatickými testy a proč?

- V podkapitole *Zátěžové testy* kapitoly 2.5.4 *Systémové testy* uvádíte, že stabilitu programu lze testovat „Spouštěním velkého množství kopií programu.“. Co přesně je myšleno „kopií programu“? Co budete při takovém testu sledovat? Uveďte příklad chyby, kterou takovým testem můžete odhalit.

Závěr:

Autor splnil všechny body zadání bakalářské práce pocházející z komerční sféry, čímž prokázal svou schopnost samostatné práce v komerčním prostředí. Z práce je zřejmé, že tvorba dokumentů/dokumentací nepatří mezi autorovy silné stránky. Dokázal ale správně porozumět cizímu zdrojovému kódu, provést v něm požadované změny, tyto změny otestovat, zmodernizovat běhové prostředí aplikace a výslednou aplikaci v tomto prostředí nasadit a toto řešení prezentovat a předat zadavateli.

Navrhuji hodnocení známkou **velmi dobře** a práci doporučuji k obhajobě.

V Plzni 19. 8. 2014

Ing. Tomáš Řondík

