Posudek oponenta diplomové práce

Název: Survey of Code Review Tools
Autor: Martin Žember
Oponent: Pavel Jančík

Cílem práce bylo vytvořit přehled nástrojů určených pro code review se zaměřením na hledání zranitelností v kódu. Práce má analyzovat jejich schopnosti a ověřit jejich použitelnost na netriviálních programech. Dalším cílem práce je vybrat z množiny testovaných nástrojů malou podmnožinu, která by dokázala nejvíce zlepšit kvalitu vyvíjeného software.

Autor použil 3 různé sady testů (vlastní jednoduché testy, podmnožina SAMATE Reference Dataset (SRD), reálné programy – balíčkovací systém RPM, Firefox), které postupně aplikoval na vybrané nástroje. Jednotlivé sady testů považuji za adekvátní, a dobře plnící svůj účel. U reálných programů chválím výběr a jeho zdůvodnění. Pouze u sady vlastních jednoduchých testů bych, kromě popsání zranitelnosti, očekával i odůvodnění jejich výběru.

Hlavní část práce je rozdělena do dvou kapitol. V první z nich (kapitola č. 3) autor porovnává schopnosti vybraných nástrojů na uvedených testech. Druhá (kapitola č. 4) obsahuje seznam 34 nástrojů „určených pro code review“.

V kapitole č. 3 se autor omezil na nástroje pro jazyky C/C++, v podstatě porovnává schopnosti 5 nástrojů. Nástroje RSM (3.3) a Source Navigatoru (3.4) neslouží pro automatické prohledání zranitelností kódu a tematicky nepatří do dané kapitoly, proto nechápu jejich zařazení do této kapitoly. Autor také, bez bližšího vysvětlení, testoval na jednotlivých nástrojích jinou podmnožinu testů z SRD (tabulky 3.1, 3.2, 3.4). Dále u výsledků testů nezmiňuje význam sloupce „Result“. U testů, které neobsahují chybu (Good Code), je možná dvojí interpretace (YES - program nalezl chybu, či test dopadl správně a tedy chyba nebyla reportována). Pozitivně hodnotím analýzu výsledků testů a srovnání schopností nástrojů v rámci této kapitoly.

Kapitola č. 4 obsahuje abecední seznam nástrojů. Jejich popis je velmi stručný, obvykle 1-2 odstavce. Seznam obsahuje velmi rozdílné druhy programů od kvalitních nástrojů, určených pro statickou analýzu kódu jako „Fortify 360“ či „CodeSurfer“, jejichž autoři neposkytli pro testování licenci, přes nástroje pro procházení v kódu (Source Insight), až po nástroje, které s code review (téměř) nesouvisí, jako je například CheckStyle a Jacobe Code Beautifier pro formátování kódu či Esc/Java2 pro testování JML anotací. Přehlednosti druhé části by daleko více prospělo setřídění nástrojů podle jejich účelu namísto abecedního pořadí a větší hloubka u klíčových nástrojů.

K vyhodnocení schopností jednotlivých nástrojů nemám žádné výhrady.

Doporučuji, aby práce byla přijata jako diplomová a připuštěna k obhajobě.

16. 8. 2011 Pavel Jančík