

Posudek oponenta bakalářské práce

Autor/autorka práce: **Martin Brožek**

Název práce: **Maven plugin for identification of incompatible changes in product source code**

Obsah práce

Práce řeší téma třicetného porovnání implementace softwarového produktu na úrovni souborů a jejich závislostí, pro účely snazší kontroly potenciálně problémových (nekompatibilních) změn během údržby upravených verzí produktu. Text práce je psán v anglickém jazyce, implementace nástroje je v jazyce Java bez použití specifických knihoven.

Kvalita řešení a dosažených výsledků

Text práce se výrazně soustředí na obsah související s účelem a vytvořením cílového nástroje, který pokrývá vcelku kvalifikovaně včetně popisů postupů a technických detailů příslušného firemního prostředí. Slabá je ovšem analýza možných přístupů k řešení – viz dále bod splnění zadání. V textu se objevují věcné nepřesnosti a chyby -- např. rozdíl „produkt“ vs „projekt“ (oddíl 2.3), definice DSL (oddíl 3.2; oboje významné pro práci), popis ProMo-DSL (str. 19) je spíše marketingový.

V rámci analýzy řešení nejsou podrobně vysvětlena použitá DSL vč. schematu na obr. 3.1, rozdíl produktových verzí a projektu, ani uveden komplexnější příklad jejich porovnávání, což znesnadňuje porozumění popisu implementace a získání odhadu pracnosti práce. Očekával jsem také detailnější popis systému Maven, zejména způsobu vytváření rozšiřujících modulů (pluginů) pro něj.

Popis realizace samotný je koncipovaný správně ale není příliš srozumitelný v detailech; například:

- není zřejmé, proč `JavaFile` není v obr. 7.1 odděleno od `AbstractParser`;
- věta „The program loads single files using the Java stack class to avoid recursion“ (7.2.3) je matoucí, nicméně příslušná část implementace je jasná;
- pro porovnání souborů je dle textu 7.2.2 použito pouze primitivní `FileUtils::contentEquals()` z Apache Commons balíčků, ale pak je prováděna (7.3.1) jejich analýza pomocí regulárních výrazů;
- vyhledává se výskyt rodičovských tříd a podobných prvků, ale vypisuje „changed classes and file paths affected by the change“ (není jasné, o jaké změny by mělo jít);
- shrnující diagram 7.2 nevysvětluje krok „Continue Mojo execution“.

V textu chybí ukázka detailního výstupu nástroje, tj. alespoň části logového souboru (je nutné nahlédnout do elektronických příloh). Není diskutováno, proč nebyla vytvořena prezentace výsledků přístupnější formou, ve vazbě na oddíl 6.4 v analytické části. Podle oddílu 8.3 autor řešil základní analýzu použitelnosti, ale není vysvětleno, jakým způsobem byla organizována a v jakém rozsahu.

Implementace je poměrně přehledná a dobře strukturovaná, využívá jen minimum knihoven, což je v daném případě výhodné pro udržovatelnost nástroje. Některé použité konstrukce by bylo vhodné vyčistit, např. použití konkrétních typů nebo opakované alokace proměnných uvnitř cyklů. Překlad a

testy proběhly při ověření bez problémů, použitelnost realizačního výstupu byla ověřena zadavatelem. Důležitým výsledkem práce je dotažení nástroje do prakticky používané podoby.

Formální úroveň

Rozsah práce odpovídá očekávání. Text je logicky strukturovaný a na slušné jazykové úrovni. Občas se vyskytují zkratky nerozvedené v místě prvního výskytu, překlepy a (naštěstí řídké) nesprávné anglické formulace; celkově tím srozumitelnost textu příliš netrpí. Obrázky jsou čitelné, až na několik výjimek vč. nevhodně použitého černého pozadí u snímků obrazovky.

Zdrojové kódy nejsou vůbec komentované, neobsahují ani základní metadata (jméno autora, rok vytvoření), nicméně samotný kód je dobře čitelný a srozumitelný.

Práce s literaturou

Seznam referencí obsahuje 60 (!) záznamů, korektně odkazovaných v textu, z nichž ale méně než deset jsou významnější technické či odborné zdroje. Zcela nevhodně je odkazován každý jednotlivý obrázek z interní firemní wiki. Citace jsou špatně formátovány (uvádění autorů, zkratky).

Splnění zadání

Bakalářská práce Martina Brožka zjevně vyřešila potřebu, pro kterou byla zadána, použitelným a udržitelným způsobem. Slušně zpracovány jsou části zadání související s konkrétním řešením.

Významným problémem práce však je, že body 1 a 3 zadání jsou v textu práce řešeny velmi okrajově, pokud vůbec. Kapitola 4 „Version control“ a některé části kapitoly 5 zřejmě zčásti souvisí s bodem 1, ale není to explicitně uvedeno, a tak jediným textem explicitně popisujícím přizpůsobování produktu v zákaznických projektech je odstavec v oddílu 2.3. Ohledně bodu 3 postrádám aspoň základní informace o způsobech analýzy kompatibility programů, je zahrnuto pouze stručné pojednání (1/2 strany, oddíl 6.1) o technologiích pro generické porovnávání obsahu souborů.

Závěr

Celkový cíl práce byl dosažen a výsledek je na slušné úrovni, ale za diskutabilní považuji splnění dvou bodů zadání. Navrhuji proto hodnocení známkou **dobře** a práci doporučuji k obhajobě, při které bude žádoucí uvedené nedostatky vysvětlit.

V Plzni 28.7.2022

doc. Ing. Přemysl Brada, MSc. Ph.D.

Dotazy k práci

- Pro parsování souborů (hledání rodičovských tříd atd.) je použit jednoduchý přístup využívající regulární výrazy. Bylo by pro analýzu možné použít gramatiky a syntaktické analyzátoři, a jaké výhody / nevýhody by to mělo?
- Jaký je specifický důvod použití konkrétního typu *LinkedList* (nikoli abstraktního kontejneru) u proměnných a návratových typů funkcí v implementaci “mojo” tříd?