

Využitie statickej analýzy kódu pri vývoji softwaru

Lukáš Častven

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xcastven@stuba.sk

5. november 2021

Abstrakt

Statická analýza je proces, pri ktorom je počítačový kód zanalyzovaný bez samotného spúšťania kódu. Po tejto procedúre, sú programátorovi prezentované nájdené chyby, ich možný spôsob opravy a aj varovania o menej závažných nedostatkoch a ich riešenia. Pomocou tejto metódy dokážeme v celom analyzovanom projekte zlepšiť kvalitu kódu a udržať konzistentný štýl, ktorý taktiež spĺňa osvedčené postupy pri vývoji softvéru. Veľkou výhodou je tiež urýchlenie hľadania chýb a softvérových defektov v porovnaní s manuálnou kontrolou. V tomto článku pochopíme, prečo developeri používajú nástroje statickej analýzy, ako ich používajú na opravu a zlepšenie kódu a ako ich implementujú do ich pracovného prostredia.

1 Úvod

S rastom komplexity vyvíjaného softvéru, rastie aj jeho minimálna požadovaná kvalita. Chyby a defekty softvéru dokážu firmám spôsobiť nemalé finančné straty. Preto existujú viaceré metódy ako kvalitu softvérových riešení zlepšiť a udržať počas vývoja, napríklad podrobné testovanie alebo revízie nového kódu iným developerom. V tomto článku si bližšie priblížime metódu statickej analýzy, ktorá je používaná na odhalenie chýb a defektov a na udržiavanie kvality kódu na požadovanej úrovni. [1, 2]

O statickej analýze samotnej sa dozvieme v časti 2, presnejšie, pochopíme čo to je (č.2.1) a ako funguje (č.2.2). Ako sa nástroje statickej analýzy používajú v praxi je uvedené v časti 3, kde zistíme ich benefity (č.3.1) a aj nedostatky (č.3.2). Implementácia v pracovnom prostredí je podrobnejšie vysvetlená v časti 4, aké formy sa využívajú (č.4.1) a ako vlastne vyzerá práca za použitia statickej analýzy z pohľadu developera (č.4.2).

2 Princípy statickej analýzy

2.1 Čo je statická analýza

Statická analýza je proces, pri ktorom je kód posudzovaný bez spustenia samotného programu a bez vstupu. Preto sa nazýva statická. Analyzuje sa zdrojový kód a v niektorých prípadoch aj objektový kód¹. Po úspešnom ohodnotení kódu, sú developerovi zobrazené nájdené chyby a defekty. Taktiež sú poskytnuté vysvetlenia, prečo sú nájdené chyby realnymi chybami a ako ich je možné opraviť. [3]

2.2 Čo statická analýza robí

Existuje veľa typov nástrojov statickej analýzy, ale nie všetky poskytujú rovnake funkcionality ako iné. Našťastie hlavný postup je jasne definovaný. Skladá sa len z dvoch krokov:

1. Analýza zdrojového (objektového) kódu.
2. Informovanie o chybách, defektoch a narušení osvedčených postupov.

Aby sme lepšie pochopili čo statická analýza robí, pozrieme sa na konkrétny nástroj FindBugs [4]. Tento nástroj sa používa na odhalenie chýb v programovacom jazyku Java.

Po kroku číslo (1), zanalyzovanie kódu, FindBugs rozdelí defekty do kategórií podľa ich závažnosti. Tie sú: vysoká, stredná, nízka. Nasledovne, ako súčasť kroku (2), poskytne developerovi aj niekoľko možných rýchlo návrhov opravy². [1]

3 Využitie pri vývoji softvéru

V praxi nie je využitie statickej analýzy bez problémové. Samozrejme, benefity tejto metódy sú jasne viditeľné, avšak developerov viac odrádzajú ťažkosti a nedostatky ako pozitíva, ktoré táto metóda poskytuje [1].

3.1 Benefity

3.2 Nedostatky

4 Implementácia nástrojov statickej analýzy

4.1 Formy

4.2 Z pohľadu developera

5 Najdôležitejšie funkcionality nástrojov statickej analýzy

6 Záver

¹Väčšinou ide o súbori s rozšírením názvu „.o”.

²ang. Quick fixes

Literatúra

- [1] B. Johnson, Y. Song, E. Murphy-Hill, and R. Bowdidge, “Why don't software developers use static analysis tools to find bugs?,” IEEE, may 2013.
- [2] L. N. Q. Do, J. Wright, and K. Ali, “Why do software developers use static analysis tools? a user-centered study of developer needs and motivations,” 2020.
- [3] Wikipedia, “Static program analysis — Wikipedia, the free encyclopedia.” <http://en.wikipedia.org/w/index.php?title=Static%20program%20analysis&oldid=1050273451>, 2021. [Online; accessed 02-November-2021].
- [4] “Findbugs.” <http://findbugs.sourceforge.net/>. Online; accessed 02-November-2021.