

Využitie statickej analýzy kódu pri vývoji softvéru

Metódy inžinierskej práce 2020/2021

Lukáš Častven

Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave

28. november 2021

Motivácia

- Spôsob udržania kvality kódu
- Zautomatizované hľadanie chýb a defektov
- „Hoci čo, čo zautomatizuje nudnú prácu je skvele.”¹

¹B. Johnson, Why don't software developers use static analysis tools to find bugs?, "IEEE, may 2013.

Prehľad

- 1 Principy statickej analýzy
- 2 Využitie pri vývoji softveru
- 3 Implementácia nástrojov statickej analýzy
- 4 Využitie v praxi

Principy statickej analýzy

- Analyzovanie kódu bez spušťanie
- Informovanie o chybách a defektoch
- Príklad chybného kódu ²

```
public String founType() {
    return this.foundType();
}
```

²N. Ayewah, Using static analysis to find bugs

Využitie pri vývoji softveru

<i>Benefity</i>	<i>Nedostatky</i>
Automatické hľadanie chýb	Falošné pozitíva
Urdžanie kvality kódu	Prerušenie pracovného priebehu
Predintegrácia	Nejasnosť
Urdžanie tímových praktík	Nedostatočná podpora tímovej práce
Nastaviteľnosť	Netriviálnosť

Implementácia nástrojov statickej analýzy

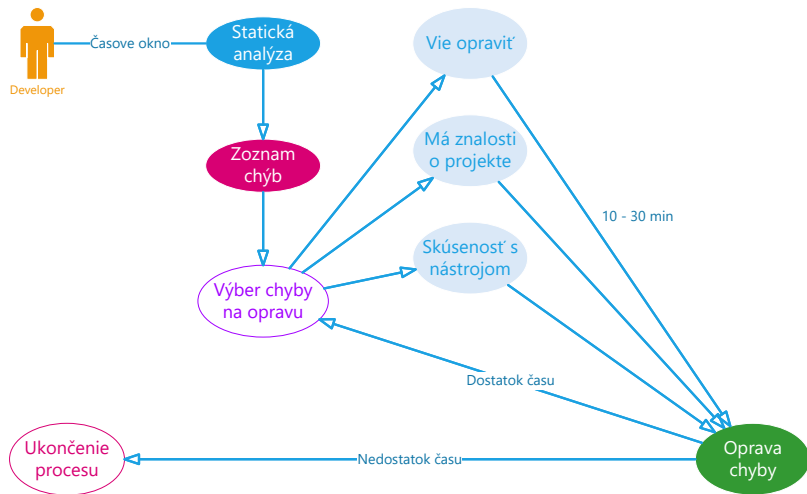
- Integrované v IDE
- Integrované v kompilátoroch ³

```
$ gcc -c -fanalyzer double-free-1.c
double-free-1.c: In function 'test':
double-free-1.c:6:3: warning: double-'free' of 'ptr' [CWE-415] [-Wanalyzer-double-free]
   6 |   free(ptr);
     |   ~~~~~^
'test': events 1-2
|
|   5 |   free(ptr);
|     |   ~~~~~^
|     |   |
|     |   | (1) first 'free' here
|   6 |   free(ptr);
|     |   ~~~~~^
|     |   |
|     |   | (2) second 'free' here; first 'free' was at (1)
```

- Rigorózne analyzátory

³ https://developers.redhat.com/blog/2020/03/26/static-analysis-in-gcc-10#diagnostic_paths

Využitie v praxi



Zhodnotenie a ďalšia práca

- Každá prezentácia musí byť nejakou uzavretá
- Ale vždy je čo robiť ďalej. . .