

Implementirano \neq testirano \neq ispravno

Nenad Ajvaz, Stefan Kapunac,
Filip Jovanović, Aleksandra Radosavljević

Univerzitet u Beogradu, Matematički fakultet

19.5.2019.

Pregled

- 1 Uvod
- 2 Verifikacija
- 3 Modeli i metrike
- 4 Budućnost
- 5 Zaključak
- 6 Literatura

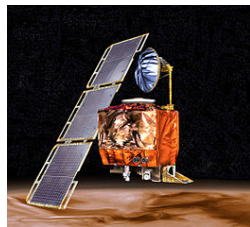
Primena softvera u svakodnevnom životu

Softver se danas koristi doslovno svuda

- Administracija
- Edukacija
- Komunikacija
- Industrija
- Saobraćaj
- Ekonomija
- Zdravstvo
- Nauka
- Inženjerstvo
- ...

Primeri neispravnog softvera

- Therac-25, 1985.
- Marsov orbiter za proučavanje klime, 1999.
- Letovi u Los Angelesu, 2004.
- Boeing 737 MAX, 2019.



Testiranje u razvoju softvera



„Program testing can be used to show the presence of bugs, but never to show their absence!“

Edsger W. Dijkstra

Verifikacija softvera

Definicija

- *verification = Are we building the product right?*
- *validation = Are we building the right product?*

Podela

- 1 Dinamička verifikacija
 - testiranje crne kutije (funkcionalno testiranje)
 - testiranje bele kutije (strukturno testiranje)
- 2 Statička verifikacija
 - simboličko izvršavanje
 - aptraktna interpretacija

Alati za verifikaciju softvera

- **Alati za automatsko testiranje**
Obezbeđuju automatsko sprovođenje testova
 - Selenium
 - ...
- **Formalni dokazivači ispravnosti**
Omogućavaju interaktivnu formulaciju matematičkog dokaza korektnosti
 - Isabelle/HOL
 - Coq
 - ...

Modeli i metrike pouzdanosti

- ① Deterministički modeli
 - Holstedova metrika
 - Mek-Kejbova ciklomatična složenost
 - ...
- ② Probabilistički modeli
 - Modeli stope neuspeha
 - Modeli rasta pouzdanosti
 - ...

Holstedova metrika

- Meri se kompleksnost programa
- Broj operatora i operanada se dovodi u vezu sa pojavom bagova



Mek-Kejbova ciklometrična složenost

$$C = E - V + 2P$$

E = broj grana grafa G

V = broj čvorova grafa G

P = broj povezanih

komponenti grafa G

G = graf kontrole toka

programa

Sample code

```
statement 1;  
If ( condition 1 ) {  
    statement 2;  
} else {  
    statement 3;  
}  
statement 4;  
for( condition 2 ) {  
    statement 5;  
}  
statement 6;
```

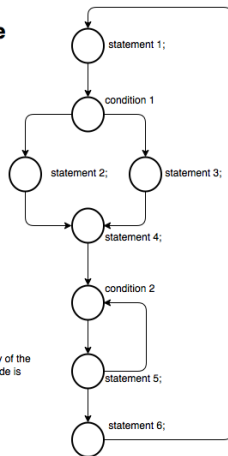
Complexity

The cyclomatic complexity of the graph representing the code is

$$v(G) = E - N + 2$$

$$= 9 - 8 + 2$$

$$= 3$$



Probabilistički modeli

- Greška kao verovatnosni događaj
- Razne metode iz oblasti statistike

Podela

- 1 Modeli stope neuspeha
 - prikazuju stopu otkazivanja programa po pojavi greške
 - nekoliko varijacija ovih modela
 - ...
- 2 Modeli rasta pouzdanosti
 - predviđaju da li dolazi do poboljšanja pouzdanosti kroz testiranje softvera
 - dva podmodela
 - ...

Budućnost softvera

- Razni alati već danas automatizuju mnoge faze razvoja
 - Generisanje koda
 - Optimizacija
 - Debugovanje
- Sa napretkom veštačke inteligencije i mašinskog učenja, proces razvoja može da se ubrza eksponencijalno
- Već se radi na sistemima koji automatski generišu kod (Bayou)
- Verujemo da će u budućnosti kod da pišu mašine, a zadatak programera će biti samo da ih kontroliše i usmerava

Zaključak

- 1 Greške su neizbežne
- 2 Testiranje je važno, ali ne uvek i dovoljno
- 3 Formalna verifikacija je ponekad neophodna
- 4 Mašine će zavladatai svetom?

Literatura

-  M. J. Quinn, *Ethics for the Information Age*.
Pearson, 7th ed., 2016.
-  J. Laski and W. Stanley, *Software Verification and Analysis*.
London: Springer-Verlag, 2009.
-  H. Pham, *System Software Reliability (Springer Series in Reliability Engineering)*.
Berlin, Heidelberg: Springer-Verlag, 2005.
-  Department of Computer Science, Rice University, “Bayou.”
On-line at: <http://askbayou.com//>.

01001000 01110110 01100001
01101100 01100001 00100001
(Hvala!)

Pitanja?