

# Debagovanje

Problemi, metodologije, tehnike i saveti ispravljanja defekata

G. Vinterhalter

Matematički fakultet, Beogradski univerzitet  
(Razvoj softvera 2)

Prezentacija radova, 2016

# Šta je to debagovanje?

- 1 Debagovanje je proces identifikacije *pravog* problema i njegovo ispravljanje.
- 2 Definicija: Debagovanje je duplo teže od kodiranja.  
Posledica: Ako napišete kod na najlukaviji (odnosno najkomplikovaniji) način, po definiciji niste dovoljno pametni da ga debugujete.  
Brian w. Keringhan

# Terminologija

- 1 U širem kontekstu bag, greška, defekt ili propust se odnosi na bilo koju vrstu problema u bilo kojoj fazi procesa razvoja.
- 2 U užem smislu termin bag ili defekt se odnose na greške u kodu. (Naša današnja tema)

# Defekti

Defekti nisu uvek loši za pojedinca. Uz pozitivan stav uviđamo prednosti:

- Poboljšano razumevanje šta i kako program radi.
- Učimo o sebi, kakve greške najčešće pravimo.
- Stičemo uvid o čitljivosti i kvalitetu sopstvenog koda.
- Šta nas ne ubije čini nas efektivnijim debagerom.
  - lakše traženje defekata
  - lakše ispravljanje defekata

# Naučni metod

Da li nauka rešava sve probleme?

- ➊ Stabilizacija greške
- ➋ Lociranje greške
  - ➊ Beleženje svih okolnosti koje dovode do greške
  - ➋ na osnovu prikupljenog napraviti pretpostavku o grešci
  - ➌ Smisliti način testiranja predpostvaljene greške
  - ➍ Testirati pretpostavku o grešci
- ➌ Ispravka greške
- ➍ Testiranje ispravke
- ➎ Traženje sličnih grešaka

# Stabilizacija greške

- 1 Nije samo pronalaženja testa koji reprodukuje grešku. Potrebno je maksimalno minimizovanje kompleksnosti testa.
- 2 Koristiti naučni metod

# Lociranje greške

- 1 Naučni metod.

# Saveti za lociranje greške

- ❶ Pri pravljenju pretpostavke uzeti u obzir što više informacija.
- ❷ Reprodukција greške na više načina
- ❸ (engl. *Refine the test case that produces the error*)
- ❹ Sužavanje prosta pretrage, podeli pa vladaj (print, trace)
- ❺ Učiti iz pogrešnih pretpostavki (sužavanje prostora pretpostavki)
- ❻ Zapišite listu mogućih problema (potencijalnih pravaca)
- ❼ Proveriti uobičajene greške
- ❽ (engl. *Brainstorming*)
- ❾ Proveriti delove koda koju su i pre bili problematični (imali greške)
- ❿ Proveriti delove koji su skoro menjani
- ⓫ Inkrementalna integracija je vaš prijatelj
- ⓫ Unit test
- ⓫ Alati
- ⓫ Kad baš neće, konsultujte se sa nekim drugim
- ⓫ Odmorite malo



# Sintaksne greške

- ❶ Ne verujte (uvek) broju linije koju prijavljuje kompajler
- ❷ ne verujte (uvek) prijavljenoj grešci od strane kompajlera
- ❸ Skoro absolutno nikad ne verujte drugoj prijavljenoj grešci
- ❹ Podeli pa vladaj
- ❺ zaboravljeni komentari ili navodnici? `/*"*/`

# Ispravljanje defekta

Uh konačno... Našli smo defekt.

Sad ide lakši deo posla?

- 1 Razumite problem
- 2 Razumite program ne samo problem
- 3 Budite apsolutno sigurni u dijagnozu (potvrdite da to jeste problem)
- 4 Rešite problem ne simptom
- 5 Ne grupisati ispravke, uvek jedna po jedna
- 6 Ispravke nisu pretpostavko, ispravljamo samo ako smo sigurni
- 7 Ne žurite sa objavom ispravke
- 8 Uvek detaljno istestirajte ispravku, uverite se da je rešila 'pravi' problem
- 9 Nakon ispravke, dodati unit test proveru za defekt.
- 10 Nakon ispravke i poboljšanog razumevanja domena potažiti slične defekte.
- 11 Uvek imati originalni kod sačuvan (version sistemi)

# Psihološki aspekti debagovanja

Prevazilaženje ljudskog sindroma.

- ① Čovek je sačijen od 365 delova i kroz njega teku 12 meridijana koje sprovode či. (U čoveku se ogleda okolina koja je poznata)
- ② Očekujemo ponašanje na koje smo navikli. (slepi smo, (engl. *Debugging blindness*))
- ③ Psihološka udaljenost. Ili mera lakoće razlikovanja sličnih stvari.
  - ① Imena promenljivih, funkcija, klasa
  - ② Potpisi funkcija
  - ③ Slično ponašanje

Zaključak?

- ① Čitljiv i razumljiv kod i ostale dobre programerske prakse
- ② Dobar programer lako mentalno izoluje delove koda koji su relevantni (podeli pa vladaj tehnikom). Mogućnost greške je uvek prisutna!

# Alati za debagovanje

- 1 Alati poređenja izvornog koda (Diff)
- 2 Upozorenja kompilatora!!! (standardizovati kompilacijske zastavice)
- 3 Profajleri kao debug alati?
- 4 Debageri

Hvala!