



Programų sistemų inžinerija

Įvadas į programinės įrangos testavimą

Lina Vasiliauskienė

Grafinių sistemų katedra

Vilniaus Gedimino Technikos Universitetas
2009-2010

Užduotis...

- Į dialogo langą įvedami trys sveiki skaičiai, kurie reiškia trikampių kraštinių ilgius. Programa išmeta pranešimą, kuriame parašo, ar gautas trikampis yra **lygiašonis**, **lygiakraštis** ar **paprastas**.
- Pasakykite (parašykite) visus įmanomus testavimo atvejus.

L. Vasiliauskienė. PĮ testavimas

Savikontrolė (1)

- Ar turite testą trikampio egzistavimui patikrinti?
 - Pvz. skaičiai [1,2,3] arba [2,5,10] nesudaro trikampio.
- Ar turite testą lygiakraščio trikampio egzistavimui patikrinti?
- Ar turite testą lygiašonio trikampio egzistavimui patikrinti?
 - Pvz. skaičiai [2,2,4] nesudaro trikampio.
- Ar turite min. tris testus lygių kraštinių kombinacijoms patikrinti?
 - Pvz. [3,3,4], [3,4,3] ir [4,3,3]

L. Vasiliauskienė. Pj testavimas

Savikontrolė (2)

- Ar turite testą, skirtą patikrinti, ar kuri nors trikampio kraštinė:
 - nėra lygi nuliui (pvz. [0, 3, 5])
 - nėra neigiamas skaičius (pvz. [1, -4, 6])
 - nėra sveikas skaičius (pvz. [2.5, 3.5, 5.5])
 - yra ne skaičius, o simbolis (pvz. [2, A, 3])
- Ar turite testą, skirtą patikrinti, ar visos trikampio kraštinės nėra lygios nuliui?

L. Vasiliauskienė. Pj testavimas

Savikontrolė (3)

- Ar turite testą, galintį patikrinti, ar dviejų kraštinių suma yra lygi trečios kraštinės ilgiui?
 - Ar šį testą atliekate visoms 3 įvestų skaičių kombinacijoms?
- Ar turite testą, galintį patikrinti, ar dviejų kraštinių suma yra mažiau, nei trečios kraštinės ilgis?
 - Ar šį testą atliekate visoms 3 įvestų skaičių kombinacijoms?
- Ar kiekvienam iš šių atvejų paruošėte laukiamą programos rezultatą (*expected output*)?

L. Vasiliauskienė. PĮ testavimas

Išvados

- Patyrę programuotojai ir testuotojai surinko vidutiniškai 7.6 taškus iš 14 galimų.
- Pavyzdžio esmė – pademonstruoti, kad net paprasčiausios programos testavimas nėra lengvas darbas 😊

L. Vasiliauskienė. PĮ testavimas

Testavimo psichologija

- Pagrindinė prasto testavimo priežastis – netinkamas termino apibrėžimas:
 - Testavimas – tai procesas, kuris parodo, kad programa klaidų neturi.
 - Testavimo paskirtis – parodyti, kad programa savo funkcijas atlieka gerai.
 - Testavimas – tai procesas, užtikrinantis, kad programa daro tą, ko iš jos ir tikimasi.

L. Vasiliauskienė. Pj testavimas

Testavimo psichologija

- Iš tikrųjų:
 - programą testuojame tam, kad pakeltume jos vertę
 - Testavimo metu vertė kyla, nes didėja programos patikimumas
 - Programos patikimumas didėja, kai randamos ir ištaisomos klaidos.
- Vadinasi, iš pat pradžių reikia padaryti prielaidą, kad programa turi klaidų ir tada ją testuoti siekiant surasti kaip galima daugiau klaidų.

L. Vasiliauskienė. Pj testavimas

Testavimo apibrėžimas

- Testavimas – tai darbas su programa siekiant surasti kaip galima daugiau klaidų.
- Žmonės orientuojasi į tikslą:
 - Jei tikslas – parodyti, kad programa klaidų neturi, testai bus parinkti taip, kad tikimybė rasti klaidą bus labai maža
 - Jei tikslas – parodyti, kad programa turi klaidų, testai bus ruošiami taip, kad tikimybė rasti klaidą bus labai didelė.

L. Vasiliauskienė. PĮ testavimas

Klausimas: kas yra “sėkmingas” ir “nesėkmingas” testai?

- **Sėkmingas** testas yra tas, kuris leidžia jums rasti klaidą (kur kas blogiau, jei klaidą randa užsakovas 😊).
- **Nesėkmingas testas** yra tas kuris neranda (arba jau neberanda) klaidų, nes tikėtis, kad PĮ nebeturi klaidų – nerealū.
- Gydytojo pavyzdys

L. Vasiliauskienė. PĮ testavimas

Testavimo "ekonomija"

- Ar įmanoma surasti visas programos klaidas?
- **Juodos dėžės** testavimas
- **Baltos dėžės** testavimas

L. Vasiliauskienė. PĮ testavimas

Juodos dėžės testavimas (*black box, data-driven, input/output-driven*)

- Vidinė programos struktūra šiuo atveju visai neįdomi
- Tikrinama, ar programa elgiasi taip, kaip parašyta specifikacijoje.
- Jei bandysite rasti visas programos klaidas pagal šį metodą, reikės patikrinti:
 - Visus teisingus įvedimo duomenis
 - Visus įmanomus įvedimo duomenis (astronominis kiekis testų)
- Tikslas: baigtiniu testų kiekiu rasti maksimalų klaidų skaičių
 - [2,2,2] – lygiakraštis, bet ir [3,3,3] taip pat lygiakraštis

L. Vasiliauskienė. PĮ testavimas

Baltos dėžės testavimas (white box, logic-driven)

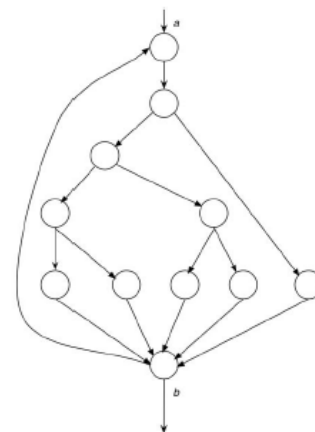
- Testavimo atvejai ruošiami remiantis programos logika
- Programa bus pilnai ištestuota, jei jūsų testo atvejis patikrins kiekvieną loginę programos šaką:
 - Unikalių loginių šakų programoje yra labai daug

L. Vasiliauskienė. PĮ testavimas

Baltos dėžės testų atvejų skaičiavimo pvz.

- Ciklas iš 20 iteracijų
- Kiekvienoje iteracijoje 5 šakos
- Reikia iš "a" nueiti į "b" visais galimais būdais
- Galimų variantų skaičius:

$$5^{20} + 5^{19} + \dots + 5^1 \approx 10^{14}$$



L. Vasiliauskienė. PĮ testavimas

Baltos dėžės metodo trūkumai

- Labai didelis testavimo atvejų skaičius
- Net ir “prasukus” visus įmanomus testus, negalėsime garantuoti, kad programa atitinka specifikaciją, nes
 - Netikriname algoritmo (Pvz. Rūšiavimo algoritmas)
 - Gali būti, kad programuodami praleidome kai kurias šakas
 - Netikriname sąlygų teisingumo:
 - Konvergavimas: `if (a-b < c)` – klaida,
 - Konvergavimas: `if (abs (a-b) < c)` – teisingai

L. Vasiliauskienė. PĮ testavimas

Pirmasis PĮ testavimo principas

- Būtina testavimo atvejo dalis – aprašyti, kokio rezultato tikimės (*expected output*)
 - Tai dažniausiai pasitaikanti klaida PĮ testavime
 - Matome tą, ką norime matyti
- Kiekvienam testavimo atvejui būtina nurodyti:
 - Paduodamų programai duomenų aprašymą
 - Galimus rezultatus pagal duotus įvedimo duomenis

L. Vasiliauskienė. PĮ testavimas

Antrasis PĮ testavimo principas

- Programuotojas neturėtų ruošti testų savo parašytai programai pats, nes:
 - Jei jis įdėjo daug darbo kurdamas programą, jam sunku kritiškai ją įvertinti ir sukritikuoti
 - Programoje gali būti klaidų, atsiradusių todėl, kad programuotojas neteisingai suprato specifikaciją, todėl jo ruošti testo atvejai automatiškai bus neteisingi.
- Pastaba:
 - šis principas netinka “debuginimui”!

L. Vasiliauskienė. PĮ testavimas

Trečiasis PĮ testavimo principas

- PĮ kurianti organizacija neturėtų testuoti savo pačios programų
 - PĮ kūrimo kainą ir laiko sąnaudas įvertinti daug lengviau, nei programos patikimumą
 - PĮ kurianti įmonė negali būti objektyvi testuodama savo pačios programas, nes testavimo procesas (jei jis organizuotas teisingai) gali reikšti, kad padidėja tikimybė nespėti užbaigti projekto laiku ir už nustatytą kainą.

L. Vasiliauskienė. PĮ testavimas

Ketvirtasis PĮ testavimo principas

- Atidžiai peržiūrėkite kiekvieno testo rezultatus

L. Vasiliauskienė. PĮ testavimas

Penktasis PĮ testavimo principas

- Testavimo atvejai turi būti ruošiami tiek su teisingais įvedimo duomenimis, tiek su neteisingais ir neleistiniais įvedimo duomenimis, nes neleistinų duomenų įvedimas duoda didesnę surandamų klaidų kiekį.

L. Vasiliauskienė. PĮ testavimas

Šeštasis PĮ testavimo principas

- Patikrinti programą, ar ji daro tą, ką reikia, yra pusė darbo
- Kitą darbo dalį sudaro patikrinimas, ar programa nedaro to, ko nereikia
 - Turime surasti “šalutinius efektus”

L. Vasiliauskienė. PĮ testavimas

Septintasis PĮ testavimo principas

- Venkite besivystančių testavimo atvejų, nebent pati programa būtų besivystančio tipo (*throwaway*)
 - Po kiekvieno naujo programos patobulinimo būtina keisti ir testavimo atvejus.
- Testavimo atvejų išsaugojimas ir paleidimas iš naujo po programos patobulinimo vadinamas **regresiniu** testavimu.

L. Vasiliauskienė. PĮ testavimas

Aštuntasis PĮ testavimo principas

- Neplanuokite laiko, reikalingo testavimui, darydami prielaidą, kad klaidų rasta nebus.

L. Vasiliauskienė. PĮ testavimas

Devintasis PĮ testavimo principas

- Programoje egzistuojančių klaidų skaičius yra tiesiogiai proporcingas jau surastų klaidų skaičiui
- Turime du modulius A ir B:
 - modulyje A rastos 4 klaidos
 - modulyje B rasta tik 1 klaida
 - pagal šį principą tikimybė, kad modulyje A vis dar yra klaidų yra didesnė, nei tikimybė, kad modulyje B yra klaidų.

L. Vasiliauskienė. PĮ testavimas

Dešimtasij PĮ testavimo principas

- PĮ testavimas – ypatingai kūrybinis ir intelektualus darbas 😊

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimas, tikrinimas bei apžvalga

- Inspection, Walkthroughs, Review
- Tai periodinis PĮ kodo testavimas, atliekamas PĮ kūrimo komandos
- Šie būdai PĮ vertę ir produktyvumą didina dviem aspektais:
 - Kuo anksčiau surasta klaida, tuo mažesnė jos ištaisymo kaina
 - Taisydami vienas klaidas labai dažnai programuotojai padaro kitas klaidas. Tas ypač skausminga, kai klaida randama PĮ kūrimo pabaigoje atliekant galutinius automatinius testus

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimas ir tikrinimas

- Tikslas – peržiūrint parašytą kodą surasti klaidas, bet ne sprendimus
- Komandinis tikrinimas:
 - 3-4 žmonės paruošia kodo apžvalgą
 - Tik vienas iš tų trijų žmonių yra programos autorius
- Šiais būdais surandama apie 30-70 % PĮ kodo klaidų
- Šiais būdais nesurandamos projektavimo klaidos, padarytos dar ruošiant reikalavimų specifikaciją

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimas (1)

- Tai procedūrų ir klaidų paieškos keliems žmonėms skaitant PĮ kodą technikų rinkinys
- Grupę sudaro 4 žmonės
 - Labiausiai patyręs programuotojas paskiriamas moderatoriumi (išskyrus patį programos autorių)
 - Išdalina medžiagą, vadovauja peržiūrai
 - dokumnetuoja visas rastas klaidas
 - Patikrina, ar visos rastos klaidos buvo ištaisytos
 - Likę nariai: vienas programuotojas, vienas testuotojas ir vienas projektuotojas

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimas (2)

■ Pirmas darbas:

- Programuotojas smulkiai pasakoja savo programos logiką
- Kiti grupės nariai užduoda klausimus ir bando nustatyti, kur programoje gali būti klaidos
- Dažniausiai pats programuotojas beskaitydamas kodą randa savo klaidas

■ Antras darbas:

- Programos kodas tikrinamas remiantis istoriškai paruoštomis dažniausiai aptinkamų klaidų anketomis

■ Po peržiūros programuotojui atiduodamas klaidų sąrašas

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimo privalumai

- Periodiškai atnaujinamas dažniausiai pasitaikančių klaidų sąrašas
- Didesnė tikimybė, kad klaida bus aptikta
- Be paties programuotojo dar bent trys žmonės aptaria egzistuojančias tos programos dalies problemas
- Programuotojas sužino kitų žmonių nuomonę apie savo programavimo stilių

L. Vasiliauskienė. PĮ testavimas

Programų inspektavimo trūkumai

- Jei programuotojas į inspektavimą pažiūrės kaip į kritiką jo atžvilgiu, procedūra bus neefektyvi
- Siūloma inspektavimo rezultatų neviešinti

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (1)

- Kintamųjų deklaravimo klaidos
 - Ar kiekvienas kintamasis yra reikiamo tipo? Ar jis inicializuotas?
 - Ar inicializuotos rodyklės?
 - Ar yra kintamųjų su panašiais vardais?
 - Ar sutampa kintamųjų ir funkcijų parametru tipai?

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (2)

■ Skaičiavimo klaidos

- Ar skaičiavimai atliekami su to paties tipo kintamaisiais?
- Ar yra veiksmų, atliekamų su to paties tipo, bet skirtingo ilgio kintamaisiais?
- Ar nėra dalybos iš nulio?
- Ar aiški veiksmų atlikimo tvarka?

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (3)

■ Palyginimo klaidos

- Ar lyginami kintamieji yra to paties tipo?
- Ar visi palyginimo operatoriai apibrėžti teisingai?
 - Daugiau, daugiau arba lygu, ne daugiau...
- Ar loginės operacijos naudojamos tinkamai?
 - AND ar OR?
- Ar lyginant atsižvelgiama į ženklų skaičių po kablelio?
- Ar loginių operacijų atlikimo tvarka aiški?
 - `if((a==2) && (b== 2) || (c==3))`

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (4)

■ Ciklų klaidos (`for()`, `while()`, `do - while`, ir pan.)

- Ar kiekviena programos kilpa turi išėjimą?
- Ar programa, modulis ar funkcija kada nors baigiasi?
- Ar numatytas atsarginis variantas, jei kartais ciklo baigimo sąlyga būtų neįvykdoma?
- Ar iteracijų kiekis nurodytas korektiškai:
 - `for(int i = 0; i <= 10; i++)`
 - `for(int i = 0; i <= 9; i++)`
 - `for(int i = 0; i < 10; i++)`

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (4)

■ Sąsajos klaidos:

- Ar funkciją pasiekusių parametų eilės tvarka atitinka funkcijos apibrėžime nurodytą parametų tvarką
- Ar gaunamų parametų tipai atitinka reikiamų argumentų tipus?
- Ar gautų parametų matavimo vienetai sutampa su tais vienetais, kuriuos naudoja konkreti funkcija?
- Ar aukščiau išvardinti punktai tinka ir kitoms funkcijos viduje iškviečiamoms funkcijoms?
- Ar funkcija keičia reikšmę kintamojo, kuris turėtų būti tik parametras?

L. Vasiliauskienė. PĮ testavimas

Dažniausiai pasitaikančių klaidų sąrašas (5)

- Įvedimo – išvedimo klaidos:
 - Ar operacijos OPEN atributai apibrėžti korektiškai?
 - Ar failo formatas atitinka I/Ospecifikaciją?
 - Ar užtenka atminties nuskaitytam failui laikyti?
 - Ar visi reikalingi failai buvo atidaryti prieš naudojimą?
 - Ar visi panaudoti failai buvo uždaryti?
 - Ar failo pabaigos sąlygos apibrėžtos ir apdorotos korektiškai?
 - Ar yra kokių nors rašybos klaidų tekste, kurį į ekraną išveda programa?

L. Vasiliauskienė. PĮ testavimas

Kiti patikrinimai

- Jei programa sukompilijuojama, bet kompiliatorius pateikia “warning” ar “informational” pranešimus, reikia tuos pranešimus patikrinti ir panaikinti.
- Patikrinkite, ar nėra trūkstančių funkcijų

L. Vasiliauskienė. PĮ testavimas

Programos tikrinimas (Walkthroughs)

- Komandą sudaro 3 - 5 žmonės, kurių rolės yra tokios:
 - Vienas moderatorius – valdo visą procesą
 - Vienas sekretorius – užrašinėja visas rastas klaidas
 - Vienas testuotojas – ruošia medžiagą susitikimui
 - Likę du žmonės gali būti:
 - labai patyręs programuotojas
 - programavimo kalbos ekspertas
 - naujai priimtas programuotojas
 - žmogus, užsiimantis programos palaikymu
 - kas nors iš visai kito projekto
 - koks nors programuotojas iš to paties projekto.

L. Vasiliauskienė. PĮ testavimas

Programos tikrinimas (Walkthroughs) (2)

- Programos kodas dalyviams išdalinamas prieš dvi dienas – kad visi galėtų susipažinti
- Dalyviai “žaidžia kompiuterį”
- Testuotojas paruošia aibę testų – galimų įvedimo/išvedimo duomenų rinkinius
 - Jie turi būti paprasti, kad žmonės mintinai galėtų skaičiuoti
- Susitikimo metu įsivaizduojamai pereinama visa programa, jos būseną fiksuojama popieriuje ar ant lentos.
- Tikrinant paruoštus testus, iškyla daug klausimų, į kuriuos programuotojas turi atsakyti, paaiškinti savo logiką ir prielaidas. Tokiu būdu aptinkama daugiausia klaidų.

L. Vasiliauskienė. PĮ testavimas

Kolegų įvertinimas (1)

- Anoniminis programos kodo kokybės, panaudojamumo bei aiškumo patikrinimas
- Tikslas – paskatinti programuotoją tobulintis
- Parenkama 6-20 panašią patirtį turinčių programuotojų
- Kiekvienas iš jų turi pateikti dvi savo programas:
 - Viena, kurią jis laiko geriausia
 - Kita, kurią jis laiko blogiausia
- Programos atsitiktine tvarka išdalinamos dalyviams
- Kiekvienas dalyvis gauna 4 programas

L. Vasiliauskienė. PĮ testavimas

Kolegų įvertinimas (2)

- Kiekvienas programuotojas gauna po 30 minučių kiekvienos programos kodo skaitymui
- Kiekvienai programai pateikiami vertinimai nuo 1 iki 7:
 - 1 – reiškia “taip”; 7 – reiškia “ne”
- Klausimai:
 - Ar programa buvo lengvai suprantama?
 - Ar aukštesnio/žemesnio lygio dizainas buvo aiškiai matomas ir pagrįstas?
 - Ar jums būtų lengva tą programą modifikuoti?
 - Ar jūs didžiutumėtės, jei būtumėt pats parašęs tą programą?

L. Vasiliauskienė. PĮ testavimas

Klausimai ir pasiūlymai

L. Vasiliauskienė. Pj testavimas