



Testiranje softvera

ETF BEOGRAD, 2019/2020.

VEŽBE #8, ASISTENT: DR DRAŽEN DRAŠKOVIĆ



Testiranje strategijama bele kutije

Mutaciono testiranje

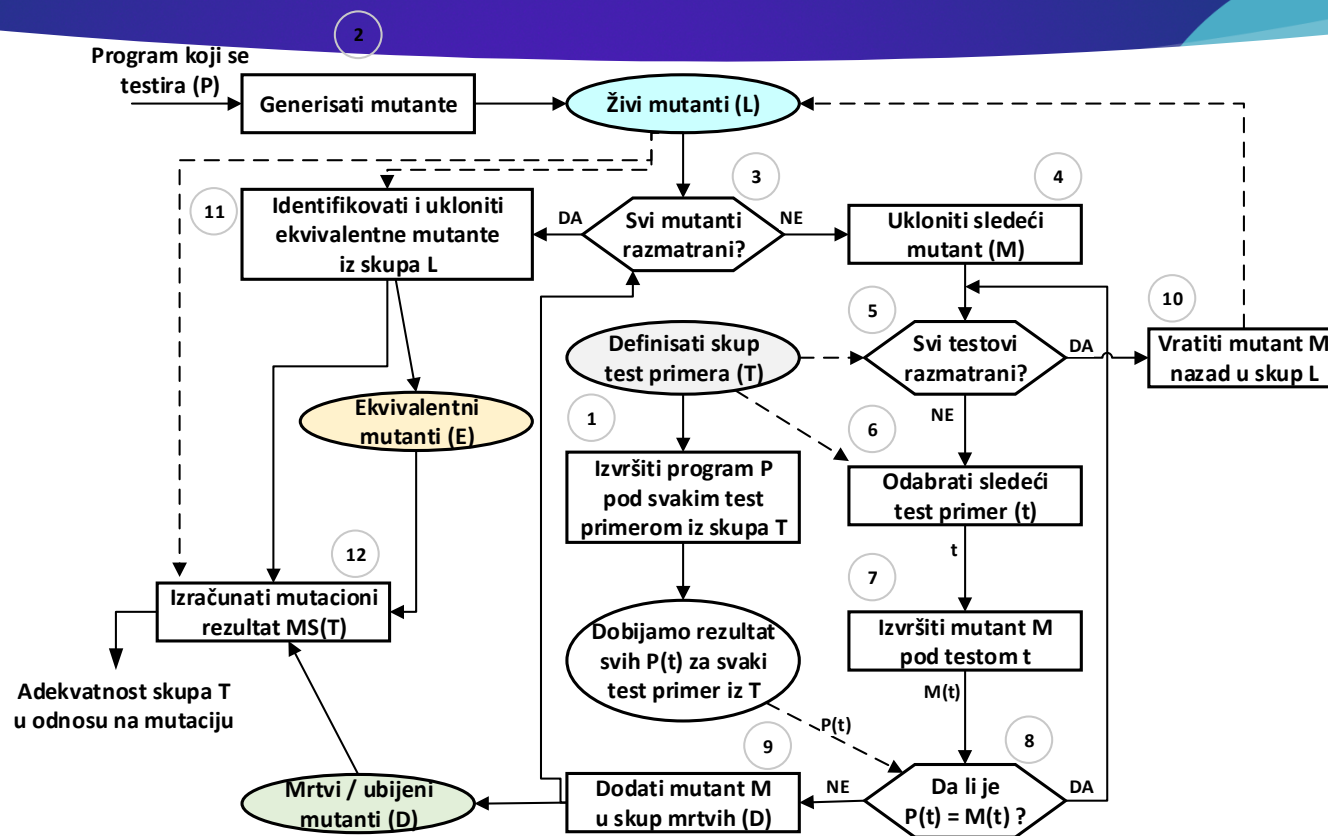
WHITE BOX TESTING - MUTATION TESTING

Mutaciono testiranje

- ▶ Tehnika testiranja zasnovana na defektima - proizvodimo hipotetički pogrešne programe
- ▶ Mutacija - male promene u programskom kodu
- ▶ Glavni program $P \Rightarrow$ Mutant program M
- ▶ Program mora da bude sintaksno korektan!
- ▶ Mutanti prvog reda i mutanti višeg reda
- ▶ Procena kvaliteta skupa testova - Mutacioni rezultat (skor)
- ▶ Mutacioni operatori - primenljivi u različitim prog. jezicima

$$MS(T) = \frac{|D|}{|L| + |D|}$$

Procedura mutacionog testiranja



Zadatak 1 - Aritmetičke operacije i mutanti

- ▶ U sledećem programu izvršiti mutaciono testiranje i formirati sledeće mutante za svaki izraz gde je to moguće:
 - ▶ kod aritmetičkih operatora, sabiranje zameniti oduzimanjem, a množenje deljenjem;
 - ▶ svaku celobrojnu promenljivu v , zameniti sa $v+1$.
- ▶ (a) Napisati četiri test primera sa različitim parovima ulaznih promenljivih x i y i pokazati da li su dati mutanti živi ili mrtvi.
- ▶ (b) Izračunati mutacioni skor nakon tačke (a).

```
1. begin
2.   int x, y;
3.   input (x, y);
4.   if (x < y)
5.     then
6.       output (x+y);
7.     else
8.       output (x*y);
9.   end
```

Zadatak 1 - Rešenje (1)

- ▶ Formira se 8 mutant programa prvog reda tako da pravimo jednu izmenu u jednoj liniji koda.
- ▶ Izmene su u linijama #4, #6 i #8.

Linija koda	Originalni program	Mutant ID	Izmena u Mutant programu
1	begin		-
2	int x,y		-
3	input (x, y)		-
4	if (x<y)	M ₁	if (x+1<y)
		M ₂	if (x<y+1)
5	then		-
6	output (x+y)	M ₃	output (x+1+y)
		M ₄	output (x+y+1)
		M ₅	output (x-y)
7	else		-
8	output (x*y)	M ₆	output ((x+1)*y)
		M ₇	output (x*(y+1))
		M ₈	output (x/y)
9	end		-

Zadatak 1 - Rešenje (2)

- ▶ Skup test primera:

$$T_P = \begin{cases} t_1 : (x = 0, y = 0) \\ t_2 : (x = 0, y = 1) \\ t_3 : (x = 1, y = 0) \\ t_4 : (x = -1, y = -2) \end{cases}$$

Napomene:

- ▶ ND = Nije definisan izlaz
- ▶ NI = Ne izvršava se (mutant ne mora da se izvrši)
- ▶ * = Prvi test kod koga smo utvrdili da se rezultat testa nad mutantom razlikuje od testa nad programom

Program	t ₁	t ₂	t ₃	t ₄	Mrtvi mutanti (skup K)
P(t)	0	1	0	2	{}
Mutant					
M ₁ (t)	0	0*	NI	NI	{M ₁ }
M ₂ (t)	0	1	0	2	{M ₁ }
M ₃ (t)	0	2*	NI	NI	{M ₁ , M ₃ }
M ₄ (t)	0	2*	NI	NI	{M ₁ , M ₃ , M ₄ }
M ₅ (t)	0	-1*	NI	NI	{M ₁ , M ₃ , M ₄ , M ₅ }
M ₆ (t)	0	1	0	0*	{M ₁ , M ₃ , M ₄ , M ₅ , M ₆ }
M ₇ (t)	0	1	1*	NI	{M ₁ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ }
M ₈ (t)	ND*	NI	NI	NI	{M ₁ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }

Zadatak 2 - Relacioni operatori u bankarskom softveru

- ▶ Neka je dat deo bankarskog softverskog sistema. U sledećem programskom Java kodu izvršiti mutaciono testiranje mutacionim operatorom ROR. Ulazni argument metode je prosečni iznos plate. Izračunati mutacioni rezultat, ako se na ulaz metode dovode sledeće vrednosti: 0 dinara, 15000 dinara i 35000 dinara.

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate >= 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```


Zadatak 2 - Rešenje (1)

Mutant M1:

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate > 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```

Mutant M2:

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate < 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```

Zadatak 2 - Rešenje (2)

Mutant M3:

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate <= 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```

Mutant M4:

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate == 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```

Zadatak 2 - Rešenje (3)

Mutant M5:

```
public String kreditnoSposoban(double iznosPlate) {  
    if(iznosPlate != 30 000) {  
        return "Kredit odobren!";  
    }  
    else {  
        return "Kredit nije odobren!";  
    }  
}
```

Zadatak 2 - Rešenje (4)

- Testiraćemo program sledećim skupom test primera:

$$T_P = \begin{cases} t_1 : iznosPlate = 0 \\ t_2 : iznosPlate = 15000 \\ t_3 : iznosPlate = 35000 \end{cases}$$

Program	t ₁	t ₂	t ₃	Skup živih (L)	Skup mrtvih (K)
P(t)	KN	KN	KO	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ }	{}
Mutant					
M ₁ (t)	KN	KN	KO	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ }	{}
M ₂ (t)	KO	-	-	{M ₁ , M ₃ , M ₄ , M ₅ }	{M ₂ }
M ₃ (t)	KO	-	-	{M ₁ , M ₄ , M ₅ }	{M ₂ , M ₃ }
M ₄ (t)	KN	KN	KN	{M ₁ , M ₅ }	{M ₂ , M ₃ , M ₄ }
M ₅ (t)	KO	-	-	{M ₁ }	{M ₂ , M ₃ , M ₄ , M ₅ }

$$MS(T) = \frac{|KILL|}{|LIVE| + |KILL|} = \frac{4}{1 + 4} = 0.8$$

Zadatak 3 - Konverzija decimalnog broja

- ▶ Formirati mutant programe za program *KonvertujDecimalniBroj*, ako se primenjuju sledeći mutacioni operatori:
 - ▶ a) ROR (relacioni operator zamene) koji zamenjuje relacioni operator sa $=$, $>$, $<$, \geq , \leq , \neq (odnosno $!=$).
 - ▶ b) AOR (aritmetički operator zamene) koji se primenjuje na izrazu:
*result = result + residue * multiplier*, odnosno aritmetičke operacije se menjaju sa $+$, $*$, $-$, $/$ i $\%$.

Zadatak 3 - Konverzija decimalnog broja (nastavak)

```
public class KonvertujDecimalniBroj {
    public void convert(int decimal, int base) {
        int result = 0;
        int multiplier = 1;
        while (decimal > 0) {
            int residue = decimal % base;
            decimal = decimal / base;
            result = result + residue * multiplier;
            multiplier = multiplier * 10;
        }
        System.out.println("Binarna predstava: " + result);
    }
}
```

```
public static void main(String args[]) {
    KonvertujDecimalniBroj conv =
        new KonvertujDecimalniBroj();
    //deo programa gde se ispravno ucitavaju decimal i base
    //kao celobrojne promenljive x i y
    conv.convert(x,y); //primer izvršavanja konverzije
}
}
```

Zadatak 3 - Konverzija decimalnog broja (1)

- ▶ U slučaju korišćenja relacionog operatora ROR:
 - ▶ while (decimal > 0)
- ▶ Mutanti prvog reda:
 - ▶ M1) while (decimal == 0)
 - ▶ M2) while (decimal < 0)
 - ▶ M3) while (decimal >= 0)
 - ▶ M4) while (decimal <= 0)
 - ▶ M5) while (decimal != 0)
- ▶ Test primeri:
 - ▶ TP1) decimal = 0, base = 2
 - ▶ TP2) decimal = 10, base = 2
 - ▶ TP3) decimal = -1, base = 2

Program	TP1	TP2	TP3	Skup LIVE	Skup KILL
P1 (t)	0	1010	0	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ }	{}
Mutanti					
M ₁ (t)	NI*	0	0	{M ₂ , M ₃ , M ₄ , M ₅ }	{M ₁ }
M ₂ (t)	0	0*	-1	{M ₃ , M ₄ , M ₅ }	{M ₁ , M ₂ }
M ₃ (t)	NI*	NI	0	{M ₄ , M ₅ }	{M ₁ , M ₂ , M ₃ }
M ₄ (t)	NI*	0	NI	{M ₅ }	{M ₁ , M ₂ , M ₃ , M ₄ }
M ₅ (t)	0	1010	-1*	{}	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ }

$$MS(T) = \frac{|KILL|}{|LIVE| + |KILL|} = \frac{5}{0 + 5} = 1$$

Zadatak 3 - Konverzija decimalnog broja (2)

- ▶ U slučaju korišćenja relacionog operatora AOR:

- ▶ result = result + residue * multiplier;

- ▶ Mutanti prvog reda kod operacije +:

- ▶ M1) result = result - residue * multiplier;
 - ▶ M2) result = result * residue * multiplier;
 - ▶ M3) result = result / residue * multiplier;
 - ▶ M4) result = result % residue * multiplier;

- ▶ Mutanti prvog reda kod operacije *:

- ▶ M5) result = result + residue + multiplier;
 - ▶ M6) result = result + residue - multiplier;
 - ▶ M7) result = result + residue / multiplier;
 - ▶ M8) result = result + residue % multiplier;

Program	TP1	TP2	TP3	Skup LIVE	Skup KILL
P1 (t)	0	1010	0	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }	{}
Mutanti					
M ₁ (t)	0	-1010*	0	{M ₂ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }	{M ₁ }
M ₂ (t)	0	0*	0	{M ₃ , M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }	{M ₁ , M ₂ }
M ₃ (t)	0	ND*	0	{M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }	{M ₁ , M ₂ , M ₃ }
M ₄ (t)	0	ND*	0	{M ₅ , M ₆ , M ₇ , M ₈ }	{M ₁ , M ₂ , M ₃ , M ₄ }
M ₅ (t)	0	1113*	0	{M ₆ , M ₇ , M ₈ }	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ }
M ₆ (t)	0	-1109*	0	{M ₇ , M ₈ }	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ , M ₆ }
M ₇ (t)	0	0*	0	{M ₈ }	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ }
M ₈ (t)	0	2*	0	{}	{M ₁ , M ₂ , M ₃ , M ₄ , M ₅ , M ₆ , M ₇ , M ₈ }

$$MS(T) = \frac{|KILL|}{|LIVE| + |KILL|} = \frac{8}{0 + 8} = 1$$