IV113 Validace a verifikace

Testování černé krabice

(Black-Box Testing)

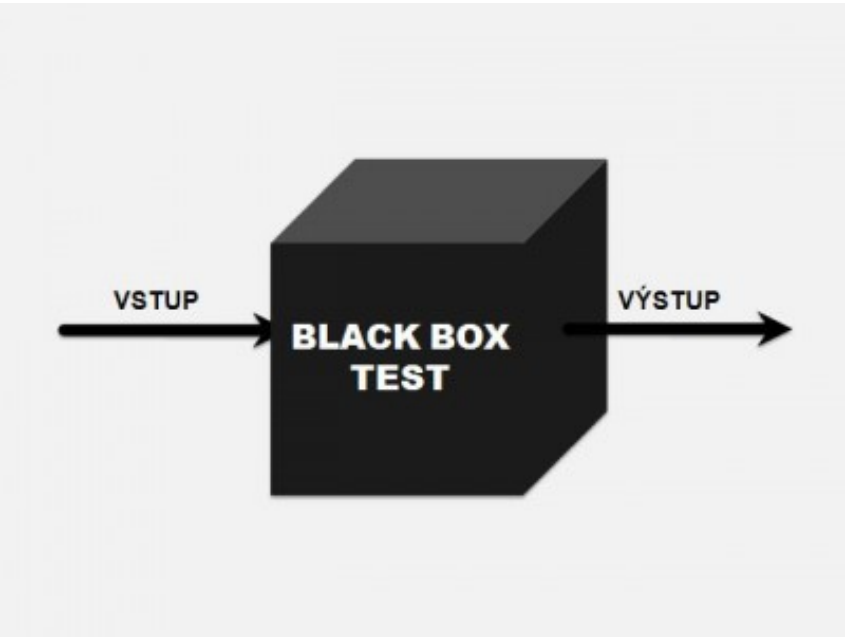
www.testingeducation.org/BBST

Jiří Barnat

Black-box testování

Black-box testování

- Na testovaný produkt je nahlíženo jako na černou skříňku.
- Vnitřní chování produktu je pro testera nepozorovatelné.
- Produkt je analyzován skrze jeho vstup-výstupní chování.
- Testování nezohledňuje zdrojový kód produktu.



White-box, Gray-box testování

White-box testování (Glass-box)

- Vnitřní chování produktu je možné pozorovat a využít.
- Tvorba a provádění testů vzhledem k vnitřnímu chování produktu, např. pro dosažení určitého stupně pokrytí.
- Vložení vnitřních chyb, kódu do produktu za účelem provedení vybraných testů.
- **Často rozšiřuje testování Black-box technikou.**

Gray-box testování

- Mezi Black-box a White-box testováním.
- Některými autory neodlišováno od White-box testování.

Techniky testování

Techniky testování

- Doménové testování
- Kombinatorické testování
- Regresní testování
- Scénářové testování
- Funkční testování
- Fuzz testování
- Risk-based testování
- ...

Je výčet úplný?

- Publikováno víc jak 100 technik $\Rightarrow$ výčet není úplný.

Příklad – adder

Je dána následující specifikace

- Program sčítá dvě celá čísla, která se mu zadají na vstup.
- Každé číslo by mělo mít jednu nebo dvě cifry.
- Program vytiskne součet.
- Program očekává "Enter" za každým číslem.
- Program se spouští příkazem `adder`.

Jaké jsou nedostatky specifikace?

Příklad – adder

Je dána následující specifikace

- Program sčítá dvě celá čísla, která se mu zadají na vstup.
- Každé číslo by mělo mít jednu nebo dvě cifry.
- Program vytiskne součet.
- Program očekává "Enter" za každým číslem.
- Program se spouští příkazem `adder`.

Jaké jsou nedostatky specifikace?

- Pracuje se zápornými čísly? (Ano)
- Jak se program ukončí? (Ctrl+C)
- Vytiskne kam? (Na display/obrazovku)

Příklad – adder

- Jaký bude základní test?

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?
  (celkem 199x199=39,601)
- Budete je testovat všechny?

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?
  (celkem 199x199=39,601)
- Budete je testovat všechny?
  (je to možné, ale ne)
- Budete testovat 3+8, 4+7, 2+7, 3+6, 3+3?

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?
  (celkem 199x199=39,601)
- Budete je testovat všechny?
  (je to možné, ale ne)
- Budete testovat 3+8, 4+7, 2+7, 3+6, 3+3?
  (zřejmě ne)
- Budete testovat 100 a více, a -100 a méně?

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?
  (celkem 199x199=39,601)
- Budete je testovat všechny?
  (je to možné, ale ne)
- Budete testovat 3+8, 4+7, 2+7, 3+6, 3+3?
  (zřejmě ne)
- Budete testovat 100 a více, a -100 a méně?
  (ano)

Příklad – adder

- Jaký bude základní test?
  (něco ala 3+7)
- Kolik je možných specifikovaných vstupů?
  (celkem 199x199=39,601)
- Budete je testovat všechny?
  (je to možné, ale ne)
- Budete testovat 3+8, 4+7, 2+7, 3+6, 3+3?
  (zřejmě ne)
- Budete testovat 100 a více, a -100 a méně?
  (ano)

= **Doménové testování**

Doménové testování

Doménové testování

Princip techniky

- Všech možných hodnot vstupů je příliš mnoho.
- Testování podobných hodnot vstupů je nezajímavé, testy jsou **ekvivalentní** např. vzhledem k průchodu grafem toku řízení.
- Je vhodné omezit provádění podobných testů, naopak je vhodné realizovat testy s hraničními a jinak zajímavými hodnotami vstupů.

"Matematický pohled"

- Relace ekvivalence na množině testů umožňuje rozdělit prostor všech možných testů na třídy ekvivalence.
- Produkt lze otestovat s použitím reprezentantů tříd rozkladu.

Analýza vstupních i výstupních proměnných

Tabulkou hraničních případů (THP)

Proměnná	Třída ekvivalence platných vstupů	Třída ekvivalence neplatných vstupů	Hraniční a zvláštní případy	Poznámky
První číslo	[-99,99]	< -99 > 99	-100,-99 99, 100	
Druhé číslo	[-99,99]	< -99 > 99	-100,-99 99, 100	
Součet	[-198,198]	< -198 > 198	(-99,-99) (99, 99) (-99,99) (99,-99)	Nelze vytvořit test s neplatnou hodnotou

Tabulka hraničních případů a testovací plán

Tabulka hraničních případů (THP)

- Nepřímá definice tříd ekvivalence.
- Nedílná součást testovací dokumentace.
- Klíčový nástroj pro tvorbu testovacího plánu.

THP jako součást testovacího plánu

- Stanovuje, co může být předmětem testu.
- Zahrnuje očekávané výsledky testů.
- Slouží jako kompaktní seznam testů.
- Umožňuje identifikaci testů.
- Lze použít i jako míru testování.

Budování THP v praxi

Zkušenosti z praxe

- Získat kompletní THP je náročné.
- V procesu testování je THP budována/doplňována během procesu testování.
- Kompletní THP se v praxi nedosahuje.
- THP obsahuje všechny proměnné, ale pouze u kritických proměnných je vyplněna.

Důvody

- Specifikace se tvoří za běhu. Samotný proces testování odkrývá nové vstupní proměnné.
- Proměnných je příliš mnoho, analýza mnohých proměnných by proběhla pouze za účelem vyplnění THP.

Hraniční hodnoty spojitých vstupů

Příklad

- Vstupní hodnota je desetinné číslo x z intervalu $(y, z]$.

Možné řešení

- Δ – nejmenší možná změna ovlivňující výsledek testu.
- Hraniční hodnoty: y, $y + \Delta$, z, $z + \Delta$

Δ ovlivněna následujícími faktory:

- Jaká je přesnost vstupního pole.
- Jaká je přesnost jednotlivých procedur, které s hodnotou nějakým způsobem pracují.
- S jakou přesností se hodnota projeví ve výsledku.

Uspořadatelné množiny

Uspořadatelné množiny

- Testování reprezentantů tříd ekvivalence a identifikace hraničních hodnot nejsou svázány pouze s číselnými doménami.
- Obecně lze doménovým přístupem analyzovat vstupy jejichž potenciální hodnoty lze uspořádat.

Pozorování

- Pokud entity nelze uspořádat, lze využít obecných měřitelných a uspořadatelných vlastností.
  - Velikost a počet entit.
  - Časování událostí.
  - Pozorovaná granularita (např. rozlišení)
  - ...

Příklad – Trojúhelník

Specifikace (Gerald Weinberg, 1969; Glen Myers 1979)

- Program přečte tři čísla a interpretuje je jako délky stran trojúhelníku.
- Na výstup vypíše, zda se jedná o trojúhelník rovnostranný, rovnoramenný, či obyčejný (ani rovnostranný ani rovnoramenný).

Jaké jsou třídy rozkladu a hraniční hodnoty?

Příklad – Trojúhelník

Specifikace (Gerald Weinberg, 1969; Glen Myers 1979)

- Program přečte tři čísla a interpretuje je jako délky stran trojúhelníku.
- Na výstup vypíše, zda se jedná o trojúhelník rovnostranný, rovnoramenný, či obyčejný (ani rovnostranný ani rovnoramenný).

Jaké jsou třídy rozkladu a hraniční hodnoty?

- Obyčejný, rovnoramenný, rovnostranný trojúhelník.
- Délky, které netvoří trojúhelník.
- Délky, které tvoří trojúhelník s nulovou výškou.
- Případy s jednou a více nulovými délkami.
- Záporná délka strany.
- Zadány 2 nebo 4 hodnoty.
- Zadány nečíselné hodnoty.

Příklad – Kumulace a kombinace vstupů

Hodnoty vstupů více proměnných, jejichž kombinace je neplatná, avšak každá zvlášť je kombinovatelná s jinými hodnotami tak, aby tvořila platnou kombinaci.

Příklad – Trojúhelník

- [3,3,6], [3,6,3], [6,3,3]

Příklad – NHL computer game

- Podle bodů ze základní skupiny (vítěství 2, remíza 1, prohra 0) se určí 8 týmů, které postupují do play-off.
- 80 zápasů v základní části.
- Součet bodů uchováván v signed byte [-128,127].
- Jaká je pozorovaná veličina a jaká její hraniční hodnota?

Příklad – Další nečíselné vstupní domény

Příklad – Adder

- Prodleva před zadáním druhého čísla.
- Zadání příliš velkého čísla.
- Vedoucí nuly či mezery.
  - Nechť $009 = 09 = 9$. Platí, že $0000000000000009 = 9$?
- Nenumerický vstup.
- Špatné chování v obsluze špatného vstupu.
  (Je specifikováno?)

Příklad adder – rozšíření THP

Proměnná	Třída ekvivalence platných vstupů	Třída ekvivalence neplatných vstupů	Hraniční a zvláštní případy	Poznámky
První číslo	[-99,99]	< -99 > 99 desetinné nečísla výrazy	-100,-99 99, 100 2.5 '/' ':' 0, null	
Druhé číslo	– || –	– || –	– || –	

Nejasná specifikace hraničních případů – příklad

Pozorování

- Specifikace je nedokonalá a dává možnost býti interpretována různě (ač správně).

Příklad specifikace

- Uvažme algoritmus, který plánuje pro hypotetickou firmu rozvoz zakázek na následující den.
- Politika firmy je preferovat zákazníky, kteří mají platné speciální povolení od managementu býti přednostně obslouženi, nebo nebyli obslouženi už 30 dnů.

Nepřesnosti vzhledem k hraničním hodnotám

- Nebyli obslouženi víc jak 30 dnů, nebo 30 dnů včetně?
- 30 dnů měřeno v době plánování, či v době dodávky?
- Dny měřeny od půlnoci, či od hodiny dodávky?
- Kdy se posuzuje platnost povolení?

Reverzní hledání domén a hraničních hodnot

Princip

- Vycházíme od možného problému a zpětným postupem hledáme vstupní data, která vedou k vyvolání problému.
- Na základě nalezených hodnot redefinujeme třídy ekvivalence.

Postup

- Identifikujeme problém.
- Kategorizujeme testy na *způsobí/nezpůsobí* chybu.
- Pokud neexistuje test, který problém vyvolá, je třeba podniknout modifikaci definice domén, aby odpovídající test nemohl být vynechán.

Domény v neuspořadatelných množinách

Pozorování

- Existují případy, kdy jednotlivé možné hodnoty vstupní proměnné nelze uspořádat.
- Nelze identifikovat hraniční případy.
- Jak testovat vstupní proměnnou pokud může nabývat velkého množství hodnot?

Doménové testování neuspořadatelných množinách

- Rozlišení objektů na základě jejich atributů.
- Identifikace ekvivalence a odpovídajících tříd.
- Volba vhodných reprezentantů jednotlivých tříd.

Neuspořadatelné množiny – Tiskárny

Testování výstupu na tiskárnu

- Existuje víc jak tisíc různých tiskáren.
- Požadujeme kompatibilitu i se staršími typy.
- Třídy / Podtřídy
  - Třídy: LJ II, LJ III, Postscript Level I, Postscript Level II, Epson
  - Podtřídy LJ II: LJ II, LJ II+, LJ IIP, LJ IID

Pozorování

- Jednotlivé nedostatky se projevují zřetelněji na různých typech tiskáren. (Analogie s uspořádanými množinami.)

Doporučení

- Je vhodné volit reprezentanty tak, aby pokryly co možná nejvíce nedostatků.

Doménové testování – Teorie

Historie

- Domény (obory hodnot) a sub-domény byly v testování používány mnohem dřív, než bylo v rámci softwarového testování teoreticky doménové testování popsáno (1988, Testing Computer Software).

Teoretický pohled

- Doména je množina prvků, na kterých je pomocí nějaké relace ekvivalence definován rozklad.
- Testování domény probíhá pomocí vybraných reprezentantů jednotlivých tříd.
- Kreativita při definici relací ekvivalence je velmi vítaná.

Doménové testování – Volba reprezentantů

Pozorování

- Hraniční případy jsou dobrou volbou.

Nevýhody hraničních případů

- Volba hraniční hodnoty jako reprezentanta je heuristika.
- Nejlepší reprezentant v dané třídě je ten, který má největší pravděpodobnost vyvolat chybu.
- Neexistují v neuspořadatelných doménách.

Praxe

- Volí se hraniční a zajímavé nehraniční případy.

Doménové testování – Závislé vstupy

Pozorování

- Závislost vstupních proměnných může způsobit, že daná hodnota jedné proměnné je hraničním případem pouze při konkrétní hodnotě jiné vstupní proměnné.

Doporučení

- V takovém případě je vhodné zvolit všechny hodnoty, které mohou býti v nějaké konfiguraci hraničním případem.

Příklad závislých proměnných — měsíc a den v měsíci

- Měsíc: 2 – 28 až 29 dnů
- Měsíc: 4,6,9,11 – 30 dnů
- Měsíc: 1,3,5,7,8,10,12 – 31 dnů

Výhody a nevýhody doménového testování

Výhody

- Základní nástroj pro boj s neúplností testování zachovávající rozumné a rovnoměrné pokrytí stavového prostoru možných testů.
- Analýza hraničních hodnot domén vstupních proměnných dává přehled o rozsahu a množství nezbytných testů.
- Hraniční případy vstupních proměnných jsou případy, při kterých se často projeví chyby produktu.

Výhody a nevýhody doménového testování

Nevýhody

- Hraniční hodnoty mohou být nejasné nebo skryté, navíc dochází ke zjemňování rozkladu v průběhu procesu.
- Neredukuje množství kombinací způsobené binárními vstupy a počtem vstupních proměnných.
- Falešný pocit bezpečí.

Použití

- Samostatně jako technika se používá zřídka.
- Vhodně doplňuje ostatní metody testování.

Kombinatorické testování

Kombinatorické testování

Pozorování

- Při testování mnoha vstupních proměnných, počet možných testů roste exponenciálně.

Cíl kombinatorického testování

- Pokrýt exponenciální prostor možných vstupů rozumným způsobem s výrazně menším počtem testů.

Princip techniky

- Identifikace podmnožiny možných testů, které splňují jisté kombinatorické vlastnosti.

Přístupy ke kombinatorickému testování

Mechanické (procedurální)

- Testované kombinace vstupů jsou určeny/vypočteny mechanickým postupem.

Založené na možném riziku

- Testované kombinace vstupů jsou určeny na základě možného rizika.

Založené na scénářovém testování

- Testované kombinace vstupů vyplývají ze zajímavých scénářů použití produktu.

Kombinace kombinatorického a doménového testování

Pozorování

- Každá jedna vstupní proměnná může sama o sobě mít mnoho možných vstupních hodnot.
- V případě testování kombinací vstupních proměnných, nelze testovat celou doménu každé proměnné.
- Domény hodnot jednotlivých proměnných se typicky předzpracují s využitím technik doménového testování.

Příklad volby reprezentantů v rámci jedné domény

- PM – příliš malá hodnota, NM – nejmenší platná hodnota
- NV – největší platná hodnota, PV – příliš velká hodnota

Pro N proměnných, je prostor možných testů

- $\{PM, NM, NV, PV\} \times \ldots \times \{PM, NM, NV, PV\}$

Mechanické kombinatorické testování

Slabé testování, verze 1

- Cílem je vybrat takovou množinu testů, aby každá proměnná byla alespoň jednou testována na každou hodnotu ze své domény.

Možné testy

	V1	V2	V3
Test1	NM	NM	NM
Test2	NV	NV	NV
Test3	PM	PM	PM
Test4	PV	PV	PV

Pozorování

- Jaký je smysl testů 3 a 4?
- Je možné, že testy 3 a 4 vůbec nelze provést.

Mechanické kombinatorické testování

Slabé testování, verze 2

- Cílem je vybrat takovou množinu testů, aby každá proměnná byla alespoň jednou testována na každou hodnotu ze své domény, tak aby žádný test neobsahoval neplatné vstupy u víc jak jedné proměnné.

	V1	V2	V3
Test1	NM	NM	NM
Test2	NV	NV	NV
Test3	PM	NM	NV
Test4	NV	PM	NV
Test5	NM	NV	PM
Test6	PV	NV	NM
Test7	NM	PV	NM
Test8	NV	NM	PV

Slabé testování, verze 3

- Cílem je vybrat takovou množinu testů, aby každá proměnná byla alespoň jednou testována na každou platnou hodnotu ze své domény.
- Obecně označována jako technika "All singles"

Možné testy

	V1	V2	V3
Test1	NM	NM	NM
Test2	NV	NV	NV

Mechanické kombinatorické testování

Silné testování, verze 1

- Cílem je otestovat všechny možné kombinace vstupů.

Možné testy

- 4x4x4=64 možností

Silné testování, verze 2

- Cílem je otestovat všechny možné kombinace platných vstupů.

Možné testy

- 2x2x2=8 možností

Mechanické kombinatorické testování

Silné testování, verze 3

- Cílem je otestovat všechny N-tice, tj. zvolit takovou sadu testů, aby každá podmnožina vstupních proměnných o N prvcích byla silně testována na všechny (platné) vstupy.
- Obecné označení: "all N-tuples", "all-pairs", "all triples".

Možné testy pro N=2 musí pokrýt následující kombinace

V1	V2
NM	NM
NV	NM
NM	NV
NV	NV

V1	V3
NM	NM
NV	NM
NM	NV
NV	NV

V2	V3
NM	NM
NV	NM
NM	NV
NV	NV

	V1	V2	V3
Test1	NM	NM	NM
Test2	NV	NM	NV
Test3	NM	NV	NV
Test4	NV	NV	NM

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D				
A	E				
B	D				
B	E				
C	D				
C	E				

- 6 testů

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F			
A	E	G			
B	D	G			
B	E	F			
C	D	G			
C	E	F			

- 6 testů

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H		
A	E	G	I		
B	D	G	I		
B	E	F	H		
C	D	G	H		
C	E	F	I		

- 6 testů

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H	J	
A	E	G	I	K	
B	D	G	I	K	
B	E	F	H	J	
C	D	G	H	J	
C	E	F	I	K	

- 6 testů

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H	J	
A	E	G	I	K	
B	D	G	I	J	
B	E	F	H	K	
C	D	G	H	K	
C	E	F	I	J	

- 6 testů

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H	J	L
A	E	G	I	K	M
B	D	G	I	J	L
B	E	F	H	K	M
C	D	G	H	K	M
C	E	F	I	J	L

V1	V2	V3	V4	V5	V6
A	D	F	H	J	L
A	E	G	I	K	M
B	D	G	I	J	M
B	E	F	H	K	L
C	D	G	H	K	L
C	E	F	I	J	M

Příklad

Tvorba tabulky pokrývající všechny dvojice

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H	J	L
A	E	G	I	K	M
			H		M
B	D	G	I	J	M
B	E	F	H	K	L
			I		L
C	D	G	H	K	L
C	E	F	I	J	M

- 8 testů místo původních 3x2x2x2x2x2=96 testů

Příklad

Tabulka testů pokrývající jednotlivé případy (all singles)

- Pozorované proměnné V1,V2,V3,V4,V5 a V6
- V1:A,B,C V2:D,E V3:F,G V4:H,I V5:J,K V6:L,M

V1	V2	V3	V4	V5	V6
A	D	F	H	J	L
B	E	G	I	K	M
C					

All pairs – komplikace

Závislé proměnné

- Některé kombinace hodnot jsou neplatné, přestože hodnoty vyskytující se v neplatné kombinaci jsou platné v jiných kombinacích.
- Viz příklad s počty dnů v jednotlivých měsících.

Praxe

- Vypočítat all-pairs může být v praxi natolik komplikované, že se od all-pairs kombinatorického testování sklouzne k all-singles technice a náhodným kombinacím.
- Praktické řešení je používat all-singles plus kombinace, které dříve způsobily chyby, nebo jsou něčím zajímavé.

All-pairs shrnutí

All-pairs kombinatorické testování

- Významně redukuje prostor možných testů.
- Nástavba doménového testování.
- Vhodná pro verifikaci kombinací **nezávislých** proměnných.

Použití all-pairs na závislých proměnných

- Neplatnou hodnotu v dané kombinací je možné zaměnit jinou platnou hodnotou.

Náhodné kombinace

- Po zpracování jednotlivých domén vstupních proměnných bývá testování náhodných kombinací možných vstupů stejně efektivní jako testování metodou all-pairs.
- Získání náhodných kombinací je výrazně snazší, než výpočet all-pairs tabulek.

Scénářové testování

Scénářové testování

Princip techniky

- Testovat produkt pomocí fiktivních příběhů použití.

Základní cíl

- Otestovat produkt způsobem, který připomíná/napodobuje skutečné použití produktu.

Vlastnosti scénářů

Motivační

- Udává jak, ale také proč, je produkt takto používán.
- Odpovědná osoba by měla trvat na nápravě chyby, pokud se uvedeným scénářem nějaká chyba projeví.

Reálný

- Scénář by měl být věrohodný, tj. nevyprávět příběh, který by se mohl stát, ale příběh, který se pravděpodobně stane.

Netriviální

- Scénář by měl zahrnovat komplexní použití programu v komplexním prostředí na netriviálních datech.

Čitelný

- **Výsledky scénářového testu by měly být čitelné a zřejmé, tj. neměly by být skryty v komplexním výstupu programu.**

Scénářové testování a nejasnosti v požadavcích na produkt

Pozorování

- Tvorba scénářových testů je podobná procesu analýzy požadavků a tvorby specifikace.
- Scénářové testování vynáší na povrch "pod koberec zametená" nevyřešená rozhodnutí ohledně požadavků na produkt a specifikace.

Nedostatky požadavků a specifikace

- Snaha testera je poukázat v jakém kontextu se problém projeví a jaké to může mít následky.
- Je důležité udržet si roli testera.
- Tester nehledá řešení problému.
- Tester nedělá kompromisy v návrhu produktu (nepromíjí), naopak snaží se prokázat dopady možných kompromisů.

Scénáře jako Telenovela (mýdlová opera)

Scénář pro test programu pro výpočet daně z příjmů

Na začátku loňského roku byl Pavel rozvedený a z předchozího manželství měl 2 dospělé děti, z nichž jedno v březnu zemřelo ve věku 19 let při autonehodě, druhé dosáhlo věku 26 let v srpnu. Pavel celý rok pracoval na částečný (2/3) úvazek jako dělník v zavedené firmě s hrubým ročním příjmem 360.000Kč. Ze začátku roku se seznámil s pohlednou Helenou, od února žili ve společné domácnosti, v březnu se vzali a v říjnu se jim narodila dcera. Aby mohl Pavel novou rodinu uživit, začal v červnu podnikat. Na rozjezd podnikání si vzal půjčku ze Stavebního spoření, kterou však splatil v prosinci, když konečně skončilo vleklé dědické řízení, ve kterém Pavel zdědil 421.456 Kč. Na úrocích z půjčky Pavel zaplatil 37.210 Kč. Na konci roku byla finanční bilance Pavla jako podnikatele záporná, konkrétně -13.000 Kč. Vzhledem k rizikovému těhotenství Helena pracovala pouze první dva měsíce v roce a její příjmy nepřesáhly 25.000 Kč. Pavel s Helenou uplatňují společné zdanění manželů.

Nevýhody scénářového testování

Nevhodné pro testování v rané fázi vývoje

- Scénář by měl být komplexní, dosud neimplementované funkce brání provedení testu.
- Jednotlivé funkce má smysl testovat nejprve samostatně.

Nevhodné pro metriku pokrytí kódu programu

- Pokrytí kódu scénářem je náročné a vede scénář do netypických situací.

Znovupoužití scénářů může být neefektivní

- Příprava dobrého zcela nového scénáře je náročná.
- Tvorba nových scénářů modifikací existujících nemusí odhalit zcela jiné typy chyb.
- Na odhalené chyby lze připravit jiné typy testů a ty například použít v regresním testování.

Testování odvozené od možných rizik (Risk-based testing)

Motivace

Pozorování

- Jednotlivé dosud probrané metody testování se příliš zaměřují na jednotlivé aspekty/způsoby použití produktu.
- Důkladná realizace jedné z metod se postupem času stává nákladná a neúčinná v detekci jiných nedostatků produktu.

Pragmatický pohled

- Jednotlivé testy předepsané důkladnou aplikací daných testovacích metod je třeba podrobit analýze a zvážit, zda jejich provedení pomůže naplnit misi.
- Volbu správné strategie, metody testování a realizovaných testů je možné provést na **základě možných rizik**.

Riziko

Riziko

- Možnost utrpět ztrátu či být postižen nějakou škodou.

Dimenze rizika v softwarovém inženýrství

- Jakým způsobem může program selhat.
- Jak pravděpodobné je, že program selže.
- Jaké jsou důsledky selhání programu.

Princip techniky testování rizikem

- Uvědomění si, jakým způsobem může program selhat.
- Návrh testů, které odhalí myšlené (potencionální) selhání.

Využití rizik

Testování odvozené od možných rizik

- Přirozeně (nevědomky) používaná metoda, často již samotnými vývojáři systému.
- Typická metoda pro hledání chyb.
- Snadno se prolíná s ostatními metodami.

Příklad – využití rizika v doménovém testování

- V číselných doménách je často používána 0 jako jeden z hraničních případů. Důvodem je mimo jiné i to, že existuje riziko dělení nulou.
- I na volbu hraničních případů lze nahlížet jako na volbu zdůvodněnou rizikem záměny $<$ a $\leq$.

Rizikové testování – cyklus

Testovací cyklus rizikového testování

- Analýza selhání
- Zdokonalení procesu analýzy
- Určení priorit jednotlivých rizik
- Provedení odpovídajících testů
- Ohlášení chyb
- Náprava některých problémů

Klíčová otázka rizikového testování

Otázka

- Jakým způsobem může program selhat?

Hledání odpovědi

- Mnoha způsoby, úplného výčtu není možné efektivně dosáhnout.
- Pro identifikaci rizik se používají zejména zkušenosti z předchozích projektů, a různé heuristiky.

Detekce rizik

Správa projektu

- Nové nebo modifikované části produktu.
- Nová technologie v produktu.
- Proces učení.
- Části vyvíjené v časové/finanční tísni.

Lidský faktor

- Distribuovaný tým.
- Osobní vztahy.
- Nečekané “features”.
- Práce odvedená třetí stranou.
- Práce odvedená zdarma.

Detekce rizik

Specifikace a požadavky

- Nepřesná/konfliktní specifikace.
- Záhadná mlčenlivost.
- Požadavky vyvíjející se za běhu.

Výskyt chyb

- Chybové části.
- Opravené části.
- Komplexní části.

Proces testování

- Málo testované/netestované části.
- Nedostatečná různorodost testovacích technik.
- Neopravitelné chyby.

Detekce rizik

Prevence

- Testování částí, které v případě projevu chyby mohou mít důsledky.

Možné důsledky projevení chyby

- Špatná publicita,
- právním následky,
- výrazné škody,
- nesoulad s požadavky,
- zneužití produktu,
- zklamání většiny uživatelů,
- ohrožení strategického postavení,
- neuspokojení V.I.P. osob, ...

Management rizikového testování

Dimenze rizika v softwarovém inženýrství

- Jakým způsobem může program selhat.
- **Jak pravděpodobné je, že program selže.**
- **Jaké jsou důsledky selhání programu.**

Odhad míry rizika

- Ohodnocení pravděpodobnosti výskytu a závažnosti důsledků číslem v rozmezí [1-10]
- Míra rizika = [Pravděpodobnost] x [Důsledky]

Management

- Preference odstranění chyb s větší mírou rizika.

Míra rizika může být zavádějící

Nedostatky

- Jaká je pravděpodobnost výskytu chyby v produktu?
- Jaký je rozdíl mezi známkou 3, 4 a 5?

Příklad – Borland's Turbo C++

- Chyba: poškození projektového souboru
- Pravděpodobnost: 1 (very rare)
- Závažnost: 10 (critical)
- Míra rizika uvedené chyby: 10
- Nejzávažnější chyba projektu, ale v měřítku míry rizika dosahuje pouze 26% maxima.
- Chyba [5]x[5] je o 50% závažnější.

Sekce

Další techniky testování černé krabice

Nejpoužívanější techniky black-box testování

- Doménové testování
- Testování na základě rizika
- Scénářové testování

- Fuzz testování
- Testování vůči specifikaci (Specification-based)
- Testování zátěží (Stress)
- Testování uživatelem (User)

Fuzz testování

Princip

- Tvorba nových testů náhodnou modifikací (mutační testování) nebo vypočtenou modifikací (whitebox fuzz testování) vstupních dat existujících testů.
- Genetické algoritmy pro tvorbu nových testů.

Motivace

- V black-box variantě levná a překvapivě úspěšná metoda.
- Docílit průchod jiných částí kódu (zvýšit pokrytí).

Problémy

- Ve white-box variantě je systematický výpočet modifikace vstupních dat náročný.

Testování vůči specifikaci

Princip

- Testování jednotlivých tvrzení ve specifikaci.

Motivace

- Chceme soulad se specifikací.
- Prokážeme souladu se specifikací.
- Metoda správy testování a testovacího úsilí jako celku.

Problémy

- Nevyjádřená, implicitní a nejasná fakta ve specifikaci.
- Zaměřeno na pokrytí specifikace, spíše než na eliminaci možných rizik.

Testování zátěží

Princip

- Pozorovat chování produktu při enormním zatížení.
- Vynutit chybu v produktu způsobenou jeho zahlcením.

Motivace

- Získání důvěra v chování produktu v kritických momentech.
- Prevence odhalení nových chyb masovým používáním produktu uživateli.

Problémy

- Generování odpovídající zátěže.

Testování uživatelem

Princip

- Simulace uvolnění produktu.
- Uvolnění produktu omezené skupině uživatelů, kteří mají za úkol ověřit funkčnost produktu.
- Beta test.

Motivace

- Kvalitu posuzují sami cíloví zákazníci. Ti odhalí skutečné problémy produktu.

Problémy

- Vyžaduje téměř finální verzi produktu.
- Získávání beta testerů.