

Metodologie testování

Radek Mařík

CA CZ, s.r.o.

September 14, 2007



1 Základní terminologie testování

- Softwarová chyba
- Úrovně testování
- Základy návrhu testů
- Terminologie návrhu testů
- Postupy návrhu testů

6 zásad testování softwaru ^[Kit95, Het88]

Proces testování softwaru vyjadřuje, jakým způsobem jsou lidé, metody, měření, nástroje a zařízení integrovány za účelem testování softwarového produktu.

- 1 *Kvalita* testovacího procesu určuje úspěch testovacího úsilí.
- 2 Zabraň *migraci defektů* použitím technik testování v počátečních fázích vývoje.
- 3 Je čas začít používat *softwarové testovací nástroje*.
- 4 Odpovědnost za vylepšování testovacího procesu musí být nesena *lidmi*.
- 5 Testování je *profesionální disciplína* vyžadující trénované lidi s odpovídajícími vědomostmi.
- 6 Testování vyžaduje kultivovaný *pozitivní* postoj týmu ke kreativní destrukci.

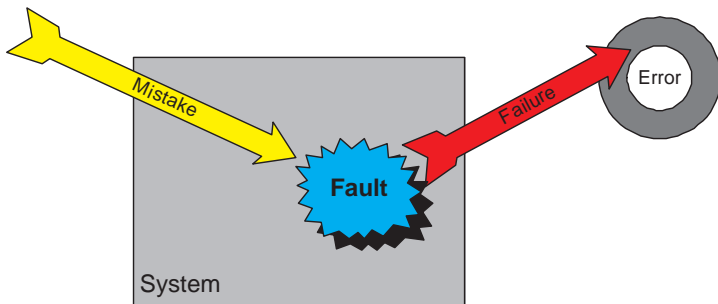
Principiální otázky testování ^[Kit95, Het88]

- Co by se mělo testovat?
- Kdy by mělo testování začít a kdy skončit?
- Kdo dělá testování?

Co je to softwarová chyba? ^[KFN93]

- Softwarová chyba je prezentace toho, že program nedělá něco, co jeho koncový uživatel předpokládá (Myers, 1976).
- Nemůže existovat absolutní definice softwarové chyby ani absolutní určení její existence. Míra přítomnosti chyb v programech odpovídá míře, podle které program přestává být užitečný. V základu lidská míra (Beizer, 1984).
- **ŠPATNĚ:** softwarová chyba je nesouhlas mezi programem a jeho specifikací.
 - Nesouhlas mezi programem a jeho specifikací je chybou pouze tehdy a jen tehdy, jestliže specifikace existují a jsou správné.

Softwarové chyby [Kit95]



Pochybení: Akce člověka, která produkuje nesprávný výsledek.

Vada: Nesprávný krok, proces nebo definice dat v počítačovém programu. Výsledek pochybení. Potenciálně vede k selhání.

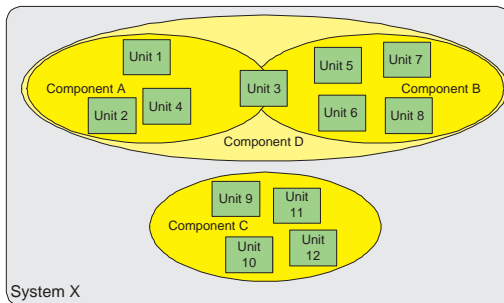
Selhání: Nesprávný výsledek. Projev vady.

Chyba: Kvantitativní vyjádření toho, na kolik je výsledek nesprávný.

Chybná víra testerů ^[Bei90]

- **Hypotéza laskavých chyb:** chyby jsou krásné, bezduché a logické.
- **Hypotéza lokality chyb:** chyba objevená v nějaké komponentě ovlivňuje pouze chování této komponenty.
- **Dominance chyb v řízení:** chyby v řídicích strukturách převládají (vs. chyby v toku dat a datových struktur)
- **Oddělení kódu a dat:** chyby respektují oddělení kódu a dat.
- **Lingua Salvator Est:** syntaxe a sémantika jazyka eliminuje většinu chyb (vs. prevence).
- **Opravy přetrvávají:** opravená chyba zůstává opravena. (A,B ovlivněné, skutečná chyba je v C)
- **Univerzální všelék:** X (jazyk, návrhová metoda, atd.) zaručuje imunitu vůči chybám,
- **Sadismus postačuje:** k vyhlazení většiny chyb. Obtížné chyby vyžadují metodologii a techniky.
- **Testeři - andělé:** tester je lepší při návrhu testů než programátoři při návrhu kódu.

Co lze testovat? [Bei90]



Jednotka je nejmenší testovatelný kus softwaru. Znamená to, že může být přeložen, sestaven, spuštěn a řízen testovacím přípravkem nebo řadičem.

Komponenta je integrovaný agregát jedné a více jednotek.

System je velká komponenta obvykle odpovídající celému produktu.

Úrovně testování ^[Bei90]

Testování jednotek - funkční a strukturní požadavky na úrovni jednotky,

Testování komponent - požadavky na úrovni komponenty,

Integrační testování - za předpokladu funkčních komponent
testování kombinace komponent,

Testování systému - zabývá se problematikou chování, ke kterému
dochází v plně integrovaném systému.

Typy testování ^[Het88]

Formální testování je proces provádění testovacích aktivit a hlášení výsledků testů podle odsouhlaseného testovacího plánu.

Akceptační testování je formální testování prováděného za účelem stanovit, zda systém splňuje akceptační kritéria a umožňuje zákazníkovi určit zda přijme systém či nikoliv.

Systémové testování je proces testování integrovaného systému za účelem ověření, zda vyhovuje specifikovaným požadavkům.

Regresní testování je částečné testování s cílem ověřit, že provedené modifikace nezpůsobují nechtěné vedlejší efekty nebo že modifikovaný systém stále splňuje požadavky.

Hodnocení výkonnosti - určení dosažení efektivnosti operativní charakteristiky.

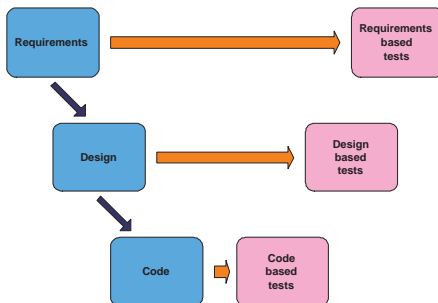
- identifikace problémů v návrhu,
- okolo 7 lidí.

Inspekce - formální hodnotící technika zahrnující detailní prozkouávání člověkem či skupinou jiným než autorem. Inspektoři kontrolují každou řádku návrhu proti každé položce kontrolního seznamu.

Demonstrace - inspekční proces, při kterém návrhář ukazuje ostatním pomocí simulace část návrhu nebo kódu, který napsal.

Technická porada - každý přinese seznam problémů. Účelem schůzky je vytvořit seznam problémů a zajistit, aby návrhář všemu rozuměli. Konečná rozhodnutí nejsou součástí této schůzky.

Vstupy návrhu testů [Het88, KFN93, Bei95]



Návrh testů

založený na požadavcích ... z externí specifikace,

založený na návrhu ... z architektury softwaru,

založený na kódu ... ze zakódované logiky a datových struktur.

Návrh testů [Het88, KFN93, Bei95]

Testování černé skříňky funkcionální testování:

strategie testování chování založené na požadavcích, program se chápe jako černá skříňka.

- Testování funkcí: funkce jsou testovány předložením vstupů a prověřováním jejich výstupů. Interní struktura programy se uvažuje pouze zřídka.

Testování bílé skříňky testování skleněné skříňky:

strategie testování struktur odvozených ze struktur testovaných objektů. Programátor využívá znalosti a přístup ke zdrojovému kódu k vývoji testovacích případů.

- Strukturální testování: Hlavní důraz je kladen na vhodný výběr cest skrz program nebo podprogram, které se procházejí při provádění sady testů.

Terminologie přípravy testů ^[Het88, Bei90]

Požadavek - podmínka nebo schopnost, kterou uživatel potřebuje k řešení problému nebo vyřešení úlohy.

Specifikace - vyjádření množiny požadavků, kterým by měl produkt vyhovět.

Testovací plán - dokument popisující zvolený přístup k zamýšleným testovacím aktivitám.

Testovací případ - specifická množina testovacích dat společně s očekávanými výsledky vztažené k vybranému cíli testu.

Návrh testu - výběr a specifikace množiny testovacích případů, které splňují úlohu testu nebo kritéria pokrytí.

Dobrý test - nezanedbatelná pravděpodobnost detekce dosud neobjevené chyby.

Úspěšný test - detekuje dosud neobjevenou chybu.

Terminologie testování ^[Het88, Kit95]

Testovací data - vstupní data a podmínky pro soubory asociované s daným testovacím případem.

Očekávané výsledky - predikované výstupní data a podmínky souborů asociované s daným testovacím případem.

Orákulus je jakýkoliv program, proces nebo objem dat, které specifikují očekávaný výsledek množiny testů, pokud jsou aplikovány na testovaný objekt.

Testovací procedura - dokument definující kroky směřující k pokrytí alespoň části testovacího plánu nebo běhu množiny testovacích případů.

Záznam testu - chronologický záznam všech významných podrobností testovací aktivity.

Platnost testu - stupeň, jak dalece test dosahuje specifického cíle.



První kolo testování ^[KFN93]

- ① Začni se zřejmým a **jednoduchým testem**.
- ② Poznamenej si, co dále je potřeba testovat:
 - Hledej **hraniční podmínky**.
 - Typicky se chyby nacházejí v blízkosti hranic.
- ③ Zkontroluj **platné případy** a pozoruj, co se děje.
- ④ Proveď testování “za letu”.
 - Vždy si zapisuj, co jsi udělal a co se děje, pokud provádíš **průzkumné testy**.
- ⑤ **Shrň**, co víš o programu a jeho problémech:
 - zpracování chyb,
 - datové typy,
 - skryté hranice.

Plán systémových testů I, Fáze 2, Krok 5 SPH [KJ96]

Příručka softwarového testování (SPH - software process handbook)

- definuje přístup rozložený na fáze,
 - Příručka by měla být tak krátká, aby se dala přečíst během jedné hodiny.
-
- **Účel:** Identifikovat a popsat testy požadované k tomu, aby produkt splnil funkční požadavky, pracoval tak, jak je specifikováno v dokumentaci produktu, a vyhověl jeho technickým omezením.
 - **Vstupy:** Specifikace softwarových požadavků, osnova dokumentu.
 - **Tým:**
 - Primární - skupina zajištění kvality softwaru,
 - Sekundární - techničtí vedoucí projektu, manažér projektu, manažér dokumentace.

Plán systémových testů II, Fáze 2, Krok 5 of the SPH [KJ96]

• Úlohy:

- ① Identifikuj hardwarovou a softwarovou konfiguraci testovacího prostředí.
- ② Popiš instalaci produktů, které se budou testovat a které budou řídit testovací prostředí.
- ③ Popiš jednotlivé testovací případy.
- ④ Odhadni časový plán a zdroje potřebné pro
 - ① vytvoření testovacích případů,
 - ② provedení testů,
 - ③ údržbu testovacích případů a příslušných testů.
- ⑤ Vytvoř návrh plánu systémových testů.
- ⑥ Zreviduj plán.
- ⑦ Identifikuj potencionální problémy.
- ⑧ Vyřeš tyto problémy.
- ⑨ Vytvoř referenční plán systémových testů.

• **Reference:** Příloha F - Vzor plánu systémových testů

• **Výstup:** Plán systémových testů

• **Výstupní kritéria:** Revize and podepsání odpovědnou osobou



Prohlubování testovacího plánu pomocí seznamů ^[KFN93]

Seznamy je jednoduché vytvořit, problémem bývá úplnost.

- Seznam zpráv a obrazovek vstupů dat.
- Seznam vstupních a výstupních proměnných.
- Seznam vlastností a funkcí.
- Seznam chybových hlášek.
- Seznam souborů programu.
- Seznam kompatibilního hardwaru.
- Seznam kompatibilního softwaru.
- Seznam kompatibilních operačních prostředí.
- Seznam komponent, které nalezne zákazník v krabici.
- Seznam veřejných dokumentů.

Prohlubování testovacího plánu pomocí tabulek ^[KFN93]

Tabulky dobře charakterizují vztahu.

- Tabulka zpráv.
- Tabulka vstupních a výstupních proměnných.
- Tabulka vztahu vstupů a výstupů.
- Rozhodovací tabulky a stromy.
- Tabulka kompatibility hardwaru/software.

IF	Rozlišující kód = 3	Y Y Y Y N N N N
	Označeno "Odloženo"	Y Y N N Y Y N N
	Vyřešeno v červnu	Y N Y N Y N Y N
THEN	Zahrň do červnové zprávy	Y N Y N Y N N N
	Zahrň do přehledové zprávy	Y Y Y Y Y Y N N

Literatura I



Boris Beizer.

Software Testing Techniques.

Van Nostrand Reinhold, New York, 2 edition, 1990.



Boris Beizer.

Black-Box Testing, Techniques for Functional Testing of Software and Systems.

John Wiley & Sons, Inc., New York, 1995.



Bill Hetzel.

The Complete Guide to Software Testing.

John Wiley & Sons, Inc., second edition, 1988.



Cem Kaner, Jack Falk, and Hung Quoc Nguyen.

Testing Computer Software.

International Thomson Computer Press, second edition, 1993.



Edward Kit.

Software Testing in the Real World.

Addison-Wesley, 1995.



Raymond Kehoe and Alka Jarvis.

ISO 9000-3, A Tool for Software Product and Process Improvement.

Springer, 1996.