

Typy softwarových chyb

Radek Mařík

CA CZ, s.r.o.

September 14, 2007



- 1 Chyby uživatelského rozhraní
- 2 Chyby omezení
- 3 Procesní chyby
- 4 Chyby vedení
- 5 Chyby požadavků
- 6 Strukturální chyby
- 7 Datové chyby
- 8 Chyby implementace
- 9 Srovnávací testy

Chyby uživatelského rozhraní - funkcionality [KFN93]

Funkčnost

- Program má problém s funkčností, jestliže
 - nedělá něco, co by měl dělat nebo
 - to dělá nevhodně, zmatečným způsobem či neúplně,
 - lze některé operace provést obtížně,
- Konečná definice, co se “předpokládá” od programu, žije pouze v mysli uživatele.
- Všechny programy mají problémy s funkčností vzhledem k různým uživatelům.
- Funkční problém je chybou, pokud očekávání uživatele jsou rozumná.

Chyby uživatelského rozhraní - vstupy ^[KFN93]

Komunikace

- Jak lze nalézt, jak používat daný program?
- Jaká je nápověda, pokud uživatel udělá chybu či spustí *< Help >*?

Struktura příkazů

- Jak snadné je ztratit se v programu?
- Jaké chyby uživatel dělá, kolik je to stojí času a proč?

Chybějící příkazy

- Co chybí?
- Nutí program uživatele přemýšlet nějakým pevným, nepřirozeným nebo neefektivním způsobem?

Chyby uživatelského rozhraní - výstupy ^[KFN93]

Výkonnost

- Rychlost je základem interaktivního softwaru.
- Cokoliv vyvolává v uživateli pocit, že program pracuje pomalu, je problém.

Výstup

- Získá uživatel, co potřebuje?
- Mají výstupní reporty smysl?
- Může uživatel přizpůsobit výstup svým potřebám?
- Lze přesměrovat výstup podle výběru uživatele na monitor, tiskárnu, či do souboru daného formátu?

Chyby omezení ^[KFN93]

Chyby zpracování vyjímek zahrnují neschopnost

- předvídat možnost chyby bránit se jim,
- zpozorovat podmínky chyby,
- zpracovat detekovanou chybu rozumným způsobem.

Chyby hraničních podmínek

- Nejjednodušší hranice jsou numerické.
- Mezní nároky na paměť, za kterých program může pracovat.

Výpočetní chyby

- Chyby aritmetiky jsou časté a obtížně detekovatelné.
- Program ztrácí přesnost během výpočtu vlivem zaokrouhlovacích chyb a chyb ořezávání.
- Výpočetní chyby způsobené chybnými algoritmy.

Procesní chyby - sekvenční ^[KFN93]

Počáteční a jiné speciální stavy

- Funkce mohou selhat při prvním použití, např. chybějící inicializační informace či soubory.
- Nastaví se skutečně vše do výchozího bodu, vymažou se všechna data, jestliže uživatel provede reset programu?

Chyby řízení

- Chyba řízení nastane, pokud program provede chybný příští krok.
- Extrémní chyba nastane, pokud se program zastaví či naopak vymkne řízení.

Procesní chyby - paralelní ^[KFN93]

Chyby souběhu (angl. race errors)

- Jsou jedny z nejméně testovaných.
- Nastávají v multiprocesorových systémech a v interakčních systémech.
- Velmi obtížně se opakují.

Zátěžové podmínky

- Program se začne chovat chybně, pokud se přetíží.
- Spadají se chyby:
 - velkého *objemu*, tj. hodně práce za dlouhou dobu.
 - velkého *stresu*, tj. hodně práce v daném okamžiku.
- Všechny programy mají své limity. Je však důležité vědět, co nastane.

Chyby vedení I ^[KFN93]

Hardware

- Programy posílají chybná data na zařízení, ignorují chybové kódy přicházející zpět a zkouší použít zařízení, která neexistují či jsou právě vytížená.

Řízení zdrojů a verzí

- Staré problémy se opět objevují, pokud programátor zakomponuje do programu nějakou starou verzi komponenty.
- Ujistěte se, že program má správný copyright, vstupní obrazovky a čísla verzí.

Chyby vedení II [KFN93]

Dokumentace:

- Slabá dokumentace může způsobit, že uživatel přestane věřit, že software pracuje správně.

Chyby testování:

- Chyby udělané testery jsou nejčastějšími chybami objevenými během testování.
- Jestliže program navádí většinu uživatelů ke způsobení chyb, pak program není správně navržen.

Chyby požadavků, vlatností a funkčnosti ^[Bei90]

Požadavky a specifikace:

- neúplné, nejednoznačné, nebo vzájemně si odporující,
- hlavní zdroj drahých chyb.

Chyby vlastností:

- chybějící, chybné, nebo nevyžádané vlastnosti,

Interakce vlastností:

- nepredikovatelné interakce (např. přesměrování telefonních volání ve smyčce)

Preventivní opatření proti chybám ve specifikacích a vlastnostech:

- problémy s komunikací člověk-člověk,
- jazyky formálních specifikací poskytují krátkodobé řešení, avšak neřeší problém chyb v dlouhodobém horizontu.

Strukturální chyby I ^[Bei90]

Chyby v řízení a sekvencích:

- příkazy GOTO, kód ala špagety, kód ala pačinko,
- většina chyb řízení (v novém kódu) se dá snadno testovat a je chycena během testování jednotek,
- neupravený starý kód může mít řadu chyb v řídicím toku,
- stlačování za účelem kratšího prováděcího času nebo menšího nároku na paměť je špatná praktika.

Chyby zpracování:

- zahrnuje chyby vyhodnocení aritmetických, algebraických, či matematických funkcí, výběru algoritmu.
- řada problémy v této oblasti se váže k nesprávným konverzím z jedné reprezentace dat na druhou.



Strukturální chyby II ^[Bei90]

Chyby logiky:

- neporozumění jak se selekční či logické operátory chovají samostatně nebo v kombinacích,
- neporozumění sémantice uspořádání logických výrazů a jeho vyhodnocení specifickými překladači,
- chyby datového toku: nevztahují se k chybám v řízení,
- chyby toku řízení: část logického výrazu, která je použita pro ovládání toku řízení.

inicializační chyby:

- typické chyby: opominutí inicializace pracovního prostoru, registrů, nebo oblastí dat.

Strukturální chyby III ^[Bei90]

Chyby a anomálie v toku dat:

- Anomálie toku dat nastane, pokud existuje cesta, při které se udělá s daty něco neodůvodněného, např. použití neinicializované proměnné, použití proměnné, která ještě neexistuje.
- Anomálie datového toku jsou stejně tak důležité jako anomálie toku řízení.

Datové chyby I ^[Bei90]

Obecně:

- datové chyby lze nalézt ve specifikacích datových objektů, jejich formátů, počtu objektů nebo jejich počátečních hodnotách,
- software se vyvíjí k tabulkám obsahujícím řídicí a procesní funkce.
- trendy v programování vedou k zvýšenému používání nedeklarovaných, interních, speciálních programovacích jazyků.

Dynamické versus statické:

- protože efekt poškození dynamických dat se může projevit velmi vzdáleně od příčiny, nalézají se takovéto chyby jen velmi obtížně.
- základní problém zbytků ve sdílených zdrojích (např. vyčištění po použití uživatelem, sdílené čištění pomocí ovladače zdroje, žádné čištění).



Datové chyby II ^[Bei90]

Informace, parametr, řízení:

- údaj plní jednu ze tří rolí: jako parametr, jako řízení, jako zdroj informace.
- informace je obvykle dynamická s tendencí lokality pro danou transakci (nedostatek ochranného kódu validace dat)
- neadekvátní validace dat často vede k ukazování prstem.

Obsah, struktura, atributy:

- obsah - aktuální bitový vzor, řetězec znaků, nebo číslo vložené do datové struktury,
- struktura - velikost, tvar a počty popisující datové položky.
- atributy - specifikace významu (sémantika),
- základem je explicitní dokumentace obsahu, struktury a atributů všech datových objektů.

Chyby kódování ^[Bei90]

Charakteristiky

- dobrý překladač chytne syntaktické chyby, nedeklarovaná data, nedeklarované procedury, nedefinovaný kód a mnoho inicializačních problémů,
- častou chybou kódu jsou dokumentační chyby (komentáře).
- úsilí programování je dominováno údržbou.

Chyby správy paměti ^[Pur99]

Charakteristiky

- nejobtížnější chyby z hlediska lokalizace,
- nejdůležitější chyby z hlediska opravy,
- projevy nesprávného obsahu paměti jsou nepredikovatelné,
- chyby v obsahu paměti se typicky projevují vzdáleně od jejich příčiny.
- chyby zůstávají často nedetekované dokud nejsou náhodně spuštěny.

Typy chyb

- chyby hranic polí,
- přístup přes nedefinovaný ukazatel,
- čtení z neinicializované paměti,
- chyby alokace paměti,
- chyby ztráty paměti (memory leaks).

Chyby správy paměti - Nástroje ^[Pur99]

Vkládání kódu - Object Code Insertion (OCI)

program a každá knihovna, která se zavolá, se zkopírují a poté pozmění. Proces změn vkládá instrukce, které validují každé čtení, zápis, alokaci či dealokaci paměti.

Example 1

PURIFY

Slabá místa výkonnosti ^[Qua99]

- kolekce vyčerpávající přesné množiny dat pro výkonnostní testy programu a každé jeho komponenty (profilování).
- zaměření se na kritická data,
- sběr správně vybraných dat:

řádka ... počítání kolikrát se každá řádka provedla během běhu programu. Poskytuje nejvíce přesné a detailní údaje, ale vyžaduje nejvíce času ke sběru.

funkce ... tato úroveň poskytuje méně podrobné údaje než čítání řádek. Je užitečné, pokud se nezabýváme přesnou výkonností jednotlivých řádek.

čas ... data se sbírají z údajů časovaných běhů funkcí. Data jsou správná pro daný běh, ale závislá na stavu mikroprocesoru a paměti. Nejméně náročná na sběr.

Slabá místa výkonnosti - Nástroje ^[Qua99]

Vkládání kódu - Object Code Insertion (OCI)

program a každá knihovna, která se zavolá, se zkopírují a poté pozmění. Proces změn vkládá instrukce, které zaznamenávají náročnost průchodu kódem.

Example 2

QUANTIFY

Srovnávací testy - benchmarking ^[Kol95]

Srovnávací údaj je neformálně definován jako vztažný bod nebo standard, vůči kterému se provádí měření nebo posudky.

Formální definice:

Srovnávací testování je spojitý proces měření produktů, služeb a praktik vůči nejsilnějším konkurentům nebo firmám s vedoucím postavením na trhu.

Pracovní definice:

Srovnávací testování je hledání nejlepších praktik průmyslu, které vedou nadprůměrné výkonnosti.

Strategie srovnávacích testů se koncentruje na

- nejlepší praktiky (výkonnost) a
- metriky (měření).

Kroky k úspěšným srovnávacím testům ^[Kol95]

- Plánování
 - identifikuj produkt/proces,
 - identifikuj porovnatelné organizace,
 - identifikuj potřebná data.
- Analýza
 - sběr dat,
 - vypočítej mezery ve výkonnosti,
 - navrhni výkonnost v budoucnosti.
- Integrace
 - předej ostatním nálezy,
 - nastav funkční cíle,
 - navrhni akční plány.
- Akce
 - implementuj akční plány,
 - monitoruj progres,
 - překalibruj srovnávací testy, dosáhni vedoucí pozici.
- Vyspělost
 - integruj srovnávací testy do firemní kultury.

Literatura I



Boris Beizer.

Software Testing Techniques.

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



Cem Kaner, Jack Falk, and Hung Quoc Nguyen.

Testing Computer Software.

International Thomson Computer Press, second edition, 1993.



William J. Kolarik.

Creating Quality: Concepts, Systems, Strategies, and Tools.

McGRAW-HILL, INC., 1995.



Gettin ahead with Rational Purify, pinpoint and eliminate run-time errors.

Rational Software Corporation, 1999.



Gettin ahead with Rational Visual Quantify, pinpoint and eliminate application performance bottlenecks.

Rational Software Corporation, 1999.