19 Dynamické metody testování

Jedná se o testování běhu programu - analýza výsledků zkušebních běhů a simulací (testování a prototypování). Několik příklady dynamických metod:

- Funkční testování - Spuštěním systému se zkoumá, zda jsou naplněny požadavky uživatelů.
- Analýza mezních hodnot - Detekce chyb na mezních vstupech (nuly, prázdná pole a řetězce, první a poslední položka).
- Testování rozhraní - Ověření reakcí na vstupy, hledání chyb v I/O.
- Testování výkonu - Porovnává skutečný výkon s požadavky. Zkoumá zatížení komponent.

29 Jak uchovávat autorizace

- Adresář (directory) - Metodu popíšeme pro případ uživatelů systému v roli subjektů a souborů coby objektů, lze ji však snadno rozšířit na libovolné objekty a subjekty. Každý soubor má svého vlastníka, který k němu vlastní veškerá práva včetně práva určovat rozsah oprávnění ostatních uživatelů k tomuto souboru. S každým uživatelem je spojena speciální struktura (adresář) obsahující odkazy na všechny soubory, k nimž má daný uživatel nějaké oprávnění, včetně popisu tohoto oprávnění. žádný uživatel nesmí zapisovat do svého adresáře. Nevýhodou může být velký rozsah adresářů a velmi obtížná správa a úpravy takto přidělovaných oprávnění. Rovněž udržení přehledu o tom, kdo k danému souboru má jaká práva může být problematické.
- Seznam oprávnění (Access Control List) - Využívá opačný přístup k problému. Tentokrát je s každým objektem udržován seznam informací, které subjekty k němu mají jaká oprávnění. Metoda umožňuje snadno přidělovat implicitní práva subjektům případně skupinám subjekt° u. Při vhodném označení subjektů a použití expanzních znaků může být tato metoda dostatečně pružná. Seznamy zpravidla bývají udržovány setříděné tak, že záznamy s expanzními znaky jsou na konci. Tak stačí hledat první shodu s identifikací subjektu a použít tímto záznamem specifikované oprávnění.
- Přístupová matice (Access Control Matrix) - řádky matice odpovídají jednotlivým subjektům, sloupce objektům. V políčku daném řádkem a sloupcem je záznam o úrovni oprávnění odpovídajícího subjektu k příslušnému objektu. Přístupová matice je zpravidla velmi velká záležitost, zhusta řídká.
- Způsobilost (Capability) - Způsobilost budeme chápat jako nefalšovatelný token, jehož vlastnictví dává vlastníkovi specifická práva k danému objektu. Lze chápat jako lístek do kina. Jednou z metod zajištění nefalšovatelnosti je, že tokeny se nepředávají přímo subjektům, ale jsou udržovány v chráněné oblasti paměti, přístupné pouze systému.

Při přístupu k objektu tak systém zkontroluje existenci příslušného tokenu, tento postup lze urychlit tím, že zvlášť udržujeme seznam způsobilostí právě běžícího procesu. Výhodou metody je, že dovoluje definovat nové dosud neznámé způsoby používání objektů a přidělovat odpovídající oprávnění. Nevýhodou je opět poněkud obtížná správa těchto tokenů, zejména odebrání způsobilosti je netriviální operace.

- Security Labeling - S každým subjektem a objektem asociujete bezpečnostní label popisující pověření/klasifikaci entity Podpora víceúrovňové modely.
- Procedurálně orientovaný přístup - Namísto přidělování obecného přístupu k subjektu (čtení, zápis) můžeme přidělovat právo používat některých funkcí z rozhraní, prostřednictvím kterého je objekt zpřístupňován. Metoda podporuje koncept skrývání a zapouzdřování informací popsaný v minulé lekci. Nevýhodou je jistá ztráta efektivity a rychlosti přístupu.

50 Spojení dvou sítí

- gateway
- filtrování datového toku
- spojení sítí přes veřejné komunikační sítě (VPN)

55 Statické metody testování

Přímo zkoumají struktury a formu produktu bez jeho spuštění (reviews, inspekce, data-flow). Příklady statických metod:

- Analýza algoritmů - Zpětným přepisem do běžného jazyka nebo formalismu je ověřována logika a správnost návrhu. Zahrnuje znovuodvození vztahů, vhodnost návrhu, stabilitu, časování, přetypování
- čtení kódu - Expertní pročítání cizího kódu.
- Rozhodovací tabulky - Analýza komplexních logických vztahů a rozhodování.
- Analýza rozhraní - Ověření správnosti rozhraní modulů, reakcí na vstupy, dodržení norem.

57 Testy - životní cyklus

Je obtížné prokazovat správnost programu. To, že nebyly nalezeny chyby, může pouze znamenat, že byla použita nevhodná metodika testování. Testování by měl provádět nezávislý tým. Snižuje se tak nebezpečí, že výsledný program obsahuje nežádoucí kód, či že nebyl dodržen původní cíl projektu. Testování během vývoje programu:

1. Validace požadavků na software - ověření, zda požadavky nejsou v rozporu s platnými standardy, zda nejsou vnitřně sporné.
2. Verifikace a validace návrhu - ověření, jestli návrh úplně a bezchybně implementuje požadavky.
3. Verifikace a validace kódu - ověření, použití předepsaných standardů a postupů vývoje.
4. Testování - (modulů, integrace, systému, instalace) na jednotlivých úrovních se zkoumá, jestli se výsledný program chová přesně podle zadání, zvláště pod tlakem.
5. Verifikace a validace při používání programu - co dělat při změně v systému.

67 Výběr dat ze statistické databáze

Statistická databáze obsahuje velké množství položek (statistická data), přičemž jednotlivé položky jsou senzitivní. Není možné číst pouze jednotlivé položky.

- Potlačení malých výsledků (limited response suppression) - Systém nevydá odpověď', pokud výsledná hodnota nepřekračuje stanovený limit, případně závisí na příliš malém, nebo příliš velkém oboru (tzn. < k nebo > n ? k řádků z celkových n pro zvolené k).
- Kombinování výsledků - Systém nevydává výsledky týkající se jednotlivých hodnot z dané domény, ale pouze souhrnné výsledky pro intervaly těchto hodnot.
- Modifikace výsledků(response modification) - Systém spočítá přesný výsledek, který následně mírně poškodí možností je zaokrouhlování výsledků, navracení průměru výsledků pro interval hodnot z dané

domény apod.

- Náhodný šum - Před vyhodnocením výsledku je ke každé použité položce připočtena malá náhodná chyba, chybové hodnoty volíme jako náhodnou veličinu se střední hodnotou 0.
- Náhodný výběr (random sample) - Systém neodpovídá na základě všech hodnot v databázi ale pouze na základě provedeného náhodného výběru ze všech relevantních položek, aby nebylo možné počítáním průměrných hodnot výsledků opakovaných dotazů získat přesné hodnoty, měl by se pro ekvivalentní dotazy používat stále stejný výběr.
- Náhodné zmatení (random data perturbation) - Ke každé položce v databázi přičteme náhodnou chybu e, přičemž narozdíl od náhodného šumu pro opakované dotazy systém zajišťuje, že použitá chyba je stále stejná pokud je chyba z okolí nuly, je vliv této úpravy na statistické výsledky typu součet, průměr malý. Metoda je vhodnější než předchozí, neboť lze snáze zajistit stejné výsledky ekvivalentních dotazů.