

Atributy kvality

Atributy kvality

- Proč se zajímat o atributy kvality?
 - Porovnávací kritéria pro posouzení dvou a více systémů.
 - Vyhodnocení systému
- Příklad KWIC
 - Kritéria:
 - Změny v algoritmu – jak posunujeme
 - Změny v datové reprezentaci – jak uloženy řádky
 - Vylepšení funkcí – interaktivní systém
 - Výkonnost – časová a paměťová
 - Opětné použití v jiných systémech
 - Problém lze řešit například pomocí:
 - Hlavního programu a podprogramů se sdílenou pamětí.
 - Abstraktním datovým typem – ADT
 - Událostmi řízený systém
 - Roury a filtry

Atributy kvality

■ Run-time kvality

- Jak systém při běhu zajistí požadavky chování?
- Dodává požadované výsledky?
- Dodá výsledky s vhodným časem?
- Jsou výsledky správné?
- Funguje systém tak jak má i pokud je propojen s dalším systémem?

■ Statické kvality

- Jak snadno můžeme systém integrovat?
- Jak snadno testovat?
- Jak modifikovat/udržovat?
- Kolik to bude stát?
- Načasování

Zamyšlení se

■ Esej na téma symbolismus

- Napíšeš esej ať je na jakémkoliv téma
- Až pak se zaměřím na úpravy, abych doplnil prvky týkající se symbolismu

==

- Napíšeš program a pak se zaměřím na dodání atributů kvality.

??

- Je to správné??

Atributy kvality

■ Run-time kvality

- Výkon
- Bezpečnost
- Dostupnost
- Spolehlivost
- Přenositelnost
- Funkčnost
- Použitelnost

■ Statické kvality

- Modifikovatelnost
- Znovupoužitelnost
- Integrovatelnost
- Testovatelnost

Výkon

- Cíl
 - Responsiveness - čas za transakci
 - Dávkové zpracování - transakce za čas
- Souvislosti
 - Latency, queue size, složitost úloh (volba algoritmů), arrival rate, processing time, distribution of service..
 - Synchronizace
 - Komunikace mezi komponentami, volání metod
 - Profilování

Bezpečnost

- Autentifikace - identita uživatele
- Autorizace - práva uživatele
- Integrita systému
- Odolnost proti útokům
 - Změna dat vs. odepření služby
- Inverzně
 - Autorizovaný uživatel musí mít přístup
- Strategie
 - Autentifikační služba
 - mezi uživatelem a podmnožinou služeb
 - Monitorování sítě, logování
 - Firewall, proxy
 - Zkonstruováno na vrchu zabezpečených jader (SSL)
 - Role based, AOP, Kontext Role, Temporal

Dostupnost a spolehlivost

- Uptime / (uptime + downtime)
- Odolnost proti chybám, prevence chyb (testovatelnost)
- Error reporting & handling
- Failover
- Mechanismy obnovy po pádu

Funkčnost

- Pracuje tak, jak bylo zamýšleno?
- Vykonání úkolu vyžaduje více komponent
 - koordinovány
 - každá komponenta má svoji zodpovědnost
- Funkčnost je ortogonální ke struktuře
- Struktura implementuje funkčnost
- Systém může existovat jako monolitická komponenta
- SW architektura alokuje funkcionalitu na strukturu a

Funkčnost

- Validace, verifikace
- Efektivní implementace funkčnosti strukturou

- Proces validace
 - ověření, že systém nabízí chtěnou/požadovanou funkcionalitu
 - ověření, že model reality je v požadovaných aspektech věrný

- Proces verifikace
 - ověření, že poskytované funkce systému pracují dle očekávání

Použitelnost (Usability)

■ Learnability

- Jak dlouho trvá uživateli naučit se systém?

■ Efficiency

- Odpovídá systém uživateli dostatečně rychle?

■ Error avoidance & handling

- Zamezuje systém možným chybám?

Použitelnost (Usability)

■ Memorability

- Dokáže si uživatel zapamatovat jak používat systémové operace na delší čas?

■ Satisfaction

- Zjednodušuje systém práci?

Statické atributy kvality

- Modifikovatelnost
- Znovupoužitelnost
- Integrovatelnost
- Testovatelnost

Modifikovatelnost

- Velké ovlivnění architekturou – GRASP vzory
- Kolik komponent přepsat/přidat

- Změny dělíme na
 - Rozšíření funkčnosti, změna dovedností
 - Smazání nechtěných dovedností
 - Adaptace na nové operační prostředí
 - Procesor, HW, I/O device – portability
 - Restrukturalizace
 - Nové moduly, jiné komponenty
 - Někdy nazýváme i **maintainability**

Přenositelnost (Portability)

- Cíl
 - Schopnost běžet v různých výpočetních prostředích (HW, SW, kombinace)

- Řešení
 - Zapouzdření platformě specifických rozhodnutí
 - Nepřímý přístup (DAO, ORM, Bridge, Facade,..)

Znovupoužitelnost (Reusability)

- Cíl
 - Znovupoužití komponenty/systému/návrhu v jiném systému
- Souvislosti
 - Low coupling, high cohesion
 - Jak velká změna znovupoužité komponenty
 - Kolik dalších komponent je třeba dopsat/znovupoužít
 - Stavba modifikovatelného systému == snadný reuse

Znovupoužitelnost (Reusability)

- Reusability
 - Systém S byl postaven ze 100 komponent, S1 až S100. Nový systém T je postaven také ze 100 komponent. T1 až T100. Při budování T vývojář objevil identitu **T1 s S1**, S1 proto bude použita namísto stavby nové. Nová by musela projít návrhem, implementací a testy. Také zjistili že komponenta **T2 je identická s S2** a bude také znovu použita. Také zjistily že **dalších 'n'** komponent je identických proto je použijí pro stavbu T.

Znovupoužitelnost (Reusability)

- Reusability
 - Systém S byl postaven ze 100 komponent. Tento systém musí projít modifikací. Modifikace zasahuje jen do komponenty **S100**. Je tedy znovu napsána a nová komponenta se jmenuje T100. Nový systém se také jmenuje T.

Znovupoužitelnost (Reusability)

- Reusability
 - Poučení z příkladu
 - Reuse a modifikovatelnost
 - dvě strany jedné mince
 - Stavba modifikovatelného systému
 - == snadný reuse

Integrovatelnost

- Cíl
 - Jak moc náročné je zaintegrovat komponentu
- Souvislosti
 - Rozdělení zodpovědnosti
 - Komunikační protokol, mechanismy interakce
 - Externí složitost komponenty
 - Interoperabilita - schopnost spolupracovat s dalšími
 - Standardy

Testovatelnost

- Snadnost testování
- Pozorovatelnost
- Kontrolovatelnost komponent (stavy, I/O)
- Úroveň zapouzdření
- Dokumentace