Karakteristike lošeg i dobrog softvera

Karakteristike lošeg softvera:

- *Rigidnost* (*rigidity*) — teško ga je promeniti, jer svaka promena jednog utiče na suviše drugih delova softvera:
  - delovi softvera su međusobno jako zavisni, pa svaka promena u jednom zahteva promene u drugim delovima
  - zato je efekte i troškove promene teško predvideti, programeri ne mogu da ih procene, pa rukovodioci teško odobravaju promene
- *Krhkost* (*lomljivost*, *fragility*) — tendencija da promena u softveru izaziva neočekivane probleme i otkaze u mnogim delovima tog softvera:
  - zbog jake isprepletanosti i zamršenosti, svaka promena u jednom delu izaziva problem u drugom - *efekat domina* (*domino effect*)
  - problemi su često u delovima koji nemaju konceptualne veze sa onim na koje se promena odnosi
  - degradira se kredibilitet softvera i poverenje u njegov kvalitet
- *Neprenosivost* (*immobility*) — delove jedne aplikacije teško je ponovo upotrebiti u drugoj, jer se ne mogu rasplesti od ostatka koji nije potreban, opet zbog prevelikih zavisnosti:
  - napor potreban za takvo raspetljavanje je prevelik, pa se lakše odlučuje za pravljenje istih stvari iznova

Karakteristike lošeg i dobrog softvera

- Dobar softver ne poseduje ove karakteristike, što znači da je:
  - *fleksibilan* (*flexible*): lako ga je održavati (menjati i proširivati)
  - *robustan* (*robust*): otporan na promene, tj. promene ne utiču na njegovu ispravnost
  - *prenosiv* (*portable*) i *ponovno upotrebljiv* (*reusable*): delovi se mogu iskoristiti u drugim aplikacijama bez mnogo ulaganja
- Naravno, ne postoji idealan softver i nijedan softver ne može biti neograničeno fleksibilan, robustan i prenosiv, ali je cilj da bude takav u opsegu *predvidivih* promena uslova i načina korišćenja u određenom domenu primene. Dobar i iskusan arhitekta softvera i poznavalac njegovog domena mogu ovo da predvide i procene
- Osnovni preduslovi za kvalitetan softver sa ovakvim karakteristikama:
  - jasna, jednostavna, pravilna, razumljiva, fleksibilna *arhitektura*
  - prave apstrakcije, dobra raspodela odgovornosti
  - modularnost, enkapsulacija
  - slabe zavisnosti između delova (apstrakcija, modula), jednostavni interfejsi
- Sve ovo ima svoju cenu, pa je u dobru arhitekturu i dizajn softvera potrebno uložiti više napora i vremena u početku, da bi se kasnije efekti videli sve većom brzinom; kod lošeg softvera je obrnuto
- Karaketeristika nepreporučljivog ponašanja: *žurba ka kodovanju* (*rush to code syndrom*)