

Dizajn u razvoju software-a

Izazovi prilikom dizajniranja software-a

- Definicija dizajna u razvoju software-a
- Dizajn software-a je težak problem
- Dizajn software-a je neuredan proces
- Dizajn software-a zahteva kompromise i prioritete
- Dizajn software-a uključuje restrikcije kao sastavni deo svog procesa
- Dizajn software-a nije deterministički proces
- Dizajn software-a je pojavni proces (emergent)

Ključni koncepti u dizajnu softwer-a

- Slučajni i esencijalni problemi (osobine) prilikom razvoja
- Tehnički imperativ: upravljanje kompleksnošću

Slučajni i esencijalni problemi

- Pojam slučajnog (opcionog) u kontekstu entiteta
- Pojam esencijalnog (suštinskog) u kontekstu entiteta
- Terminologija i samo razmišljanje datira iz vremena Aristotela

Upravljanje kompleksnošću

- Značaj upravljanja kompleksnošću
- Tipovi problema koji proističu iz skupog i neefektivnog dizajna
 - Kompleksno rešenje trivijalnog problema
 - Trivijalno neispravno rešenje kompleksnog problema
 - Neodgovarajuće kompleksno rešenje kompleksnog problema
- Načini upravljanja kompleksnošću
 - Minimizovati količinu esencijalne kompleksnosti o kojoj bilo ko mora da razmišlja u bilo koje vreme
 - Nedoovoliti da se slučajna kompleksnost nepotrebno usložnjava

Osobine kvalitetnog dizajna

- Minimalna potrebna kompleksnost
- Pristupačno i lako održavanje
- Minimalna povezanost
- Proširivost
- Ponovna upotreba
- Visok fan-in
- Nizak-srednji fan-out
- Portabilnost
- Skromnost
- Stratifikacija
- Korišćenje standardnih dobro utemeljenih tehnologija

Nivoi dizajna

- Nivo sistema
- Nivo podsistema (software-skih paketa)
 - Česte forme podsistema:
 - Poslovna logika
 - Korisnički interface
 - Pristup bazi podataka
 - Systemske zavisnosti
- Podela na klase unutar podsistema
- Podela podataka i rutina unutar klasa
- Interni dizajn rutina

Osnovni gradivni elementi dizajna - heuristike

- Pravilo nedeterminisanosti
- Pojam heuristike u dizajnu software-a
- Pokušaji i greške kao način rešavanja problema

Objektno orijentisani pristup - heuristika

- Standardni koraci:
 - Identifikovati entitete i njihove osobine
 - Odrediti šta je moguće raditi sa entitetima
 - Odrediti šta prepoznati entiteti mogu da rade u vezi sa drugim entitetima
 - Odrediti nivoe vidljivosti medju entitetima (šta koji entitet treba da izloži drugim entitetima u kontekstu informacije)
 - Odrediti interface svakog entiteta

Pojam apstrakcije u kontekstu OO dizajna

- Generalizacija u smislu klase
- Specijalizacija u smislu objekta
- Nedostaci lošeg apstrahovanja
- Pozitivan učinak kvalitetnog apstrahovanja

Pojam enkapsulacije u kontekstu OO dizajna

- Definicija enkapsulacije
- Zašto koristiti enkapsulaciju (dobit)
- Da li postoji alternativa

Pojam nasledjivanja u kontekstu OO dizajna

- Šta je to nasledjivanje
- Zašto i kada koristiti nasledjivanje

Pojam sakrivanja informacija

- Tajnost i pravo na privatnost
- Dobit i nedostatci sakrivanja informacija
- Tipovi tajni:
 - Sakrivanje kompleksnosti sa svrhom da implementator ne mora da birne o detaljima prilikom implementacije na višem nivou
 - Sakrivanje izvora promena sa svrhom da kada do promene dodje njeni efekti budu lokalizovani

Identifikovanje oblasti koje su podložne promenama

- Pojam promenljivosti
- Heuristika:
 - Identifikovati elemente koji imaju tendenciju da se menjaju. Ukoliko je dokumentacija dovoljno iscrpna u njoj će se nalaziti elementi koji su podložni promena kao njihove verovatnoće. U ovom slučaju prvi korak se lako realizuje, u suprotnom videti sledeće korake.
 - Razdvojiti postojeće elemente na elemente koji su skloni promenama po određenom kriterijumu. Grupisati prepoznate elemente u pojedinačne klase ili ako je to potrebno grupisati više promenljivih elemenata sa istim tipom i tendencijom promene u jednu klasu.
 - Izolovati identifikovane elemente. Dizajnirati inter-klasni interface tako da bude neosetljiv na potencijalne promene. Dizajnirati interface tako da promene imaju uticaj samo na unutrašnjost klase, interakcija klase sa spoljašnjim svetom ne sme trpeti posledice promena. Svaka druga klasa koja komunicira sa klasom koja je podložna promena ne sme da trpi posledice njenih promena.

Identifikovanje oblasti koje su podložne promenama – nastavak

- Oblasti podložne promenama:
 - Poslovna logika
 - Hardware-ske zavisnosti
 - Ulazno/Izlazni podsistemi
 - Nestandardne jezičke mogućnosti
 - Oblasti koje je bilo teško dizajnirati
 - Statusne promenljive
 - Ograničenja veličine podataka

Uvezivanje (coupling)

- Pojam uvezivanja
- Ukoliko je moguće treba nastojati da se koristi slabo uvezivanje (loose coupling)
- Kriterijumi uvezivanja:
 - Veličina
 - Vidljivost
 - Fleksibilnost
- Tipovi uvezivanja:
 - Uvezivanje zasnovano parametrima podataka
 - Objektno uvezivanje
 - Objektno-parametarsko uvezivanje
 - Semantičko uvezivanje

Projektni uzorci (Design patterns)

- Definicija projektnog uzorka
- Pozitivne karakteristike projektnih uzoraka:
 - Redukovanje kompleksnosti korišćenjem već postojećih mehanizama apstrakcije
 - Smanjenje mogućnosti da se napravi greška, projektni uzorci su provereni jer su nastali kao posledica da se reše česti problemi u dizajnu (predstavljaju uglavnom najbolje rešenje za česte kako strukturne tako i funkcionalne zahteve)
 - Projektni uzorci pružaju heurističku vrednost time što sugerišu postojeća rešenja
 - Ukoliko se koriste olakšavaju komunikaciju medju projektatnima

Lista projektnih uzoraka koji se često koriste

- Abstract Factory
- Adapter
- Bridge
- Composite
- Decorator
- Singleton
- Iterator
- Template Method

Još neke heuristike

- Ostvarivati što je moguće veću spregnutost
- Izgradnja hierarhija
- Posmatrati interface-e kao ugovor klase sa ostatkom sistema
- Dodeljivanje odgovornosti
- Dizajnirati sistem da bude testabilan
- Pažljivo birati vreme vezivanja vrednosti za variable
- Praviti centralne kontrolne tačke
- Razmatrati korišćenje grube sile
- Praviti dijagrame
- Ostvariti što je moguće veću modularnost

Prakse u dizajnu software-a

- Iterativno napredovanje
- Zavadi pa vladaaj (divide and conquer)
- Top-down pristup (najčešće)
- Bottom-up pristup
- Eksperimentalno pravljenje prototipova
- Kolaborativni dizajn
- Praćenje dopunjavanje i ažuriranje projektne dokumentacije