

## 18. INTEGRACIONO TESTIRANJE

Nakon utvrđivanja ispravnosti određenih komponenti, tj. jediničnog testiranja osnovnih delova koda, isti se integrišu u veće celine softverskog sistema koji treba da funkcioniše prema postavljenim zahtevima. **Integracionim testiranjem** se proveravaju **jedinični moduli** koji su spojeni u veće celine i **kako oni zajedno funkcionišu**. Integracionom testiranju se ne poklanja dovoljno pažnje i u praksi se, često, ne uradi na zadovoljavajući način, što može dovesti do određenih problema. Najveći problem prilikom integracije komponenti jeste njihovo povezivanje preko softverskih interfejsa. **Interfejsi između komponenti mogu dovesti do sledećih problema:**

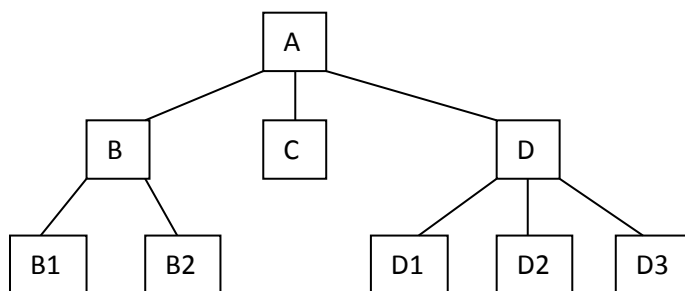
- Podaci se mogu izgubiti prilikom prolaska kroz interefejs između komponenti,
- Jedna komponenta može imati negativan uticaj na drugu,
- Kombinacije podfunkcija programa mogu imati nepovoljan uticaj na glavnu funkciju,
- Nepreciznosti u programiranju, prihvatljive na nivou pojedinačne komponente, nakon povezivanja daju neprihvatljive vrednosti (npr. promenljivih i složenijih struktura podataka).

**Strategija integracionog testiranja** definiše kako i na koji način se komponente kombinuju radi testiranja sistema, što utiče na vremenski raspored integracije, na redosled kodiranja i na troškove i sveobuhvatnost testiranja. Sistem se posmatra kao hijerarhija komponenti, u kome svaka komponenta pripada nekom sloju dizajna. **Cilj integracionog testiranja** jeste detaljno testiranje integracije i interfejsa između komponenti, kao i vezu sa operativnim sistemom, fajl sistemom, hardverskim i softverskim interfejsima i sl.

**Tehnike integracionog testiranja** su:

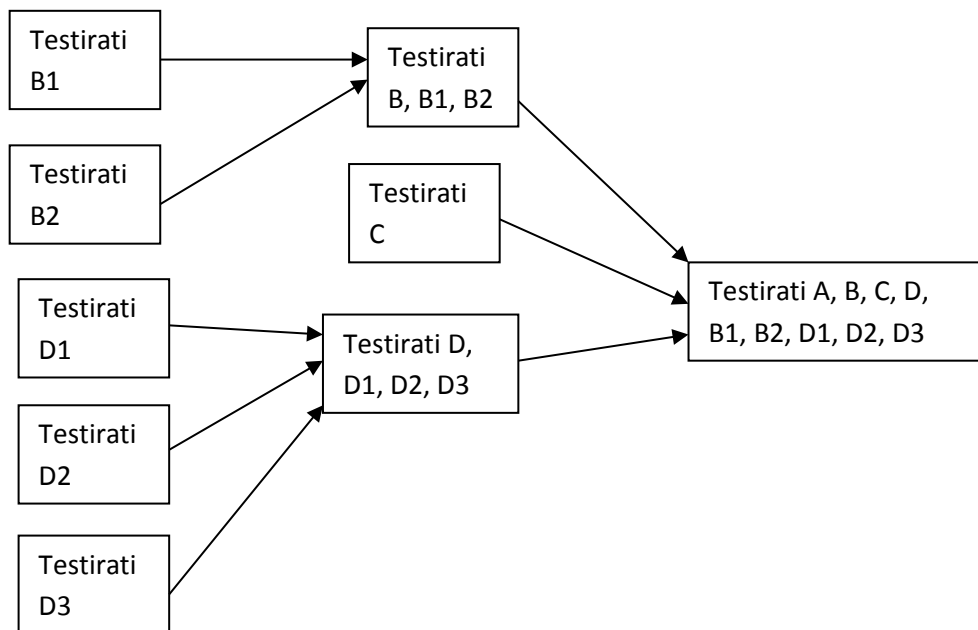
- Integracija od vrha prema dnu (Engl. „Top-Down“),
- Integracija od dna ka vrhu (Engl. „Bottom-Up“),
- Integrisanje po principu „velikog praska“ (Engl. „Big bang“),
- Sendvič integracija (Engl. „Sandwich“).

Primer modula programa, označenih sa: A, B, C, D, B1, B2, D1, D2, D3, koji će ilustrovati tehnike integracionog testiranja:



## Integracija od dna ka vrhu

Kod korišćenja ovog metoda, prvo se jedinično testira svaka komponenta na najnižem nivou hijerarhije sistema. Zatim sledi testiranje komponenti koje pozivaju prethodno testirane komponente. Ovakav postupak se ponavlja sve dok testiranjem ne budu obuhvaćene sve komponente. Metod od dna ka vrhu (odozdo nagore) je koristan kada većina komponenti na najnižem nivou predstavlja pomoćne rutine opšte namene koje se pozivaju iz drugih komponenti, kao kod objektno orijentisanog dizajna ili kada sistem integriše veći broj nezavisnih ponovo iskoristivih komponenti. Mana ovakve vrste testiranja je to što je teško testirati sistem odozdo nagore ako u velikom delu obrade, u sistemu, postoji vremenska zavisnost. Često je ova tehnika najprimerenija za objektno-orijentisane programe. Objekti se kombinuju jedan po jedan sa objektima ili kolekcijama prethodno testiranih objekata.

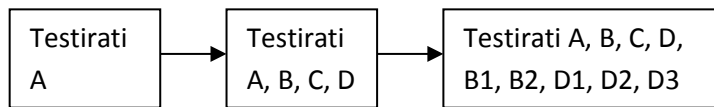


## Integracija od vrha ka dnu

Integracija od vrha ka dnu (odozgo nadole) je omiljena metoda mnogih projekatara i testera. Na najvišem nivou obično postoji jedna komponenta koja se izolovano testira. Nakon toga se komponente koje ona poziva kombinuju i testiraju kao veće celine. Taj postupak se ponavlja dokle god ne budu uključene sve komponente. Komponenta koja se testira možda poziva drugu koja još nije testirana i zato se kreira lažna rutina (stab), program specijalne namene koji simulira aktivnost komponente koja nedostaje zato što je još uvek u fazi razvoja. Stab glumi nepostojeći modul, odgovara na poziv, prihvata ulazne podatke i vraća izlazne podatke tako da postupak testiranja mogao da se nastavi. Stab je komponenta koji simulira rad modula nižeg nivoa, a drajver je komponenta koji simulira rad modula višeg nivoa.

Na sledećoj slici je prikazano kako bi se sistem testirao metodom odozgo nadole. Komponenta A, na najvišem nivou, je jedina koja se testira izolovano, dok su za B, C i D potrebne lažne rutine. Kada se komponenta A istestira, kombinuje se sa sledećim nivoom pa

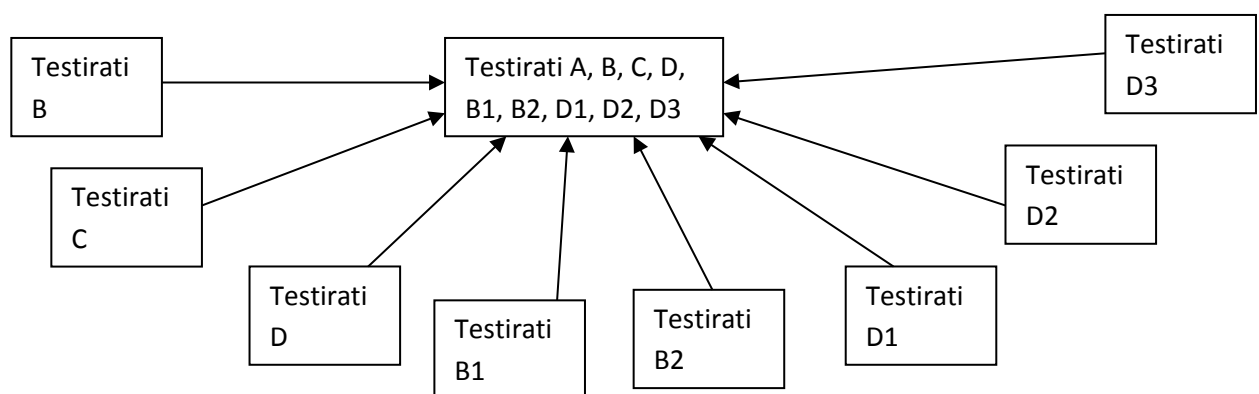
se zajedno testiraju B, C i D. U ovoj fazi testiranja biće potrebne lažne rutine za B1, B2, D1, D2, D3. Na kraju se testira ceo sistem.



Mana ovakve vrste testiranja jeste potencijalna potreba za velikim brojem lažnih rutina, tj. stabova. To se može dogoditi ako najniži nivo sistema sadrži veliki broj rutina opšte namene. Jedan od načina da se taj problem izbegne je promena strategije. Umesto da se istovremeno uključuje jedan ceo nivo, u modifikovanom pristupu odozgo nadole se pre objedinjavanja jedinično testiraju komponente svakog nivoa. Sistem se može testirati po modifikovanom pristupu tako što se prvo testira A, zatim B, C i D, i tek tada se te četiri komponente integrišu radi testiranja prvog i drugog nivoa. Zatim se B1, B2, D1, D2, D3 testiraju pojedinačno. Na kraju se ceo sistem spaja za testiranje. Jedinično testiranje komponenti svakog nivoa prouzrokuje još jednu poteškoću. Za svaku komponentu su potrebne lažne rutine i programi rukovaoci, što rezultira sa više kodiranja, a time i više potencijalnih problema.

### Integrisanje po principu „velikog praska“

Nakon jediničnog testiranja svih komponenti, sledi finalnaintegracija sistema i provera da li sve radi kako je specificirano i dizajnirano. Autor, Myers je 1979. godine to nazvao integrisanje „po principu velikog praska“. Mnogi testeri primenjuju ovaj način kod manjih sistema, dok je njegova primena kod velikih sistema nepraktična sa potencijalnim problemima. Prvi problem predstavlja to što je za testiranje pojedinačnih komponenti potrebno pisati i stabove i drajvere. Drugi problem predstavlja to, pošto se sve komponente integrišu odjednom, teško je pronaći uzrok nepravilnosti, a greške u interfejsima je teško razdvojiti od ostalih vrsta grešaka. Na sledećoj slici je prikazana integracija „po principu velikog praska“ ili spajanje komponenti odjednom.



## Sendvič integracija

Sendvič integracija je nastala kao kombinacija tehnika od vrha ka dnu i od dna ka vrhu. Autor sendvič integracije je Myers, 1979 godine. Po njemu sistem se posmatra u tri sloja, kao sendvič. Ciljni sloj se nalazi u sredini, a sa donje i gornje strane se nalaze po jedan sloj. Na gornji sloj se primenjuje pristup odozgo nadole, a za donji sloj pristup odozdo nagore. Testiranje se približava ciljnom sloju koji se bira na osnovu karakteristika sistema i strukturne hijerarhije komponenti. Stabovi i drajveri su potrebni za jedan manji broj modula/komponenti (kao npr. "Testirati A" na slici dole, potreban je stab za A). Na sledećoj slici je prikazan redosled testova za sendvič integraciju, gde je ciljni sloj u sredini, tj. komponente: B, C i D. Ovakva vrsta integracije omogućava da se testiranje obavlja što je moguće ranije prilikom implementacije softvera. Sendvič integracija ima prednost u odnosu na pristupe odozgo nadole i odozdo nagore tako što se na početku testiraju kontrolne komponente i pomoćni programi. Ne vrši se sveobuhvatno testiranje pojedinačnih komponenti pre integracije. Jedna varijacija, modifikovano sendvič testiranje, predviđa testiranje komponenti višeg nivoa pre njihovog spajanja sa ostalim.

