## 12pt2-V&V- Integration Testing 164

Test di unità dipendenti:

* Funzioni
* Classi e loro metodi

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteAlcune unità potrebbero necessitare di altre

* + **Faccio dependecy graph**

**Per testare dipendenze:**

* **STUB:** sostituisce un’unità
  + Immagine che contiene testo, schermata, diagramma, Carattere

    Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, linea

    Descrizione generata automaticamenteProva ad eliminare le dipendenze verso il basso e cerca di sostituire il comportamento della funzione che va a sostituire con un elemento più semplice che va a simulare il comportamento di quella funzione
    - Do per scontato che funzione4 funzioni e uso i suoi risultati per testare funzione 2
* **DRIVER**: pilota un’unità
  + Simulo di avere la funzione1 che passi dei parametri alla funzione2, e testo funzione 2

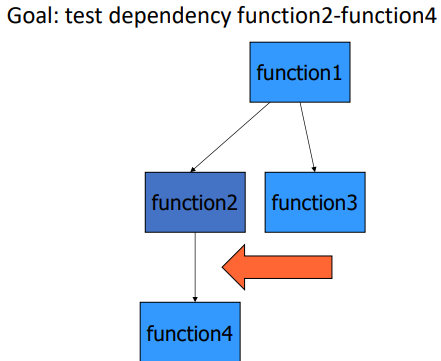
**Esempio di stub:**

* Creo uno stub che simuli il comportamento del db per testare il mio codice
  + In questo modo se poi ho problemi so che sono di configurazione di db

**Dipendenze:**

* **Immagine che contiene testo, linea, Carattere, schermata

  Descrizione generata automaticamente**Funzioni che separate vanno bene ma quando applicate insieme, si rompe tutto
  + Es: passo parametri male
  + Es:
    - passo tipo sbagliato
    - errori conversione
    - dati passati con misure diverse (uno usa metri, l’altro piedi)

**Immagine che contiene testo, Carattere, schermata, Rettangolo

Descrizione generata automaticamenteDunque**:

* **testo 2 unità in isolamento**
* le integro, testo nuovamente
* Immagine che contiene testo, schermata, Carattere, linea

  Descrizione generata automaticamentese ci sono più unità, incremento integrazione

**Altra opzione:**

* Immagine che contiene testo, schermata, diagramma, Carattere

  Descrizione generata automaticamenteTesto solo la funzione 4
* Integro la funzione 2 e testo

**Importante, cosa migliore è:**

* testare in isolation una alla volta
* pian piano unire una alla volta e testare

Ci sono **due approcci**:

* **Bottom-up**
  + Mentre scrivo le unità, prevedo dei test delle singole unità
  + I blocchi più interni sono già pronti, testati, funzionanti
    - Testo fin dall’inizio
    - Richiede driver
* **Top-down**
  + Permette di trovare prima errori in flussi architetturali
  + Codice aggiuntivo
    - Definizione di stubs per ogni unità di basso livello

***In pratica, si usa un mix dei due approcci in base a cosa serve.***

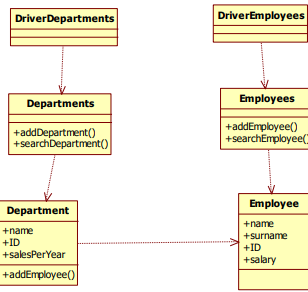
*Un altro approccio è quella della* ***bigBang Integration*** *ma non è buono in quanto diventa molto difficile trovare il baco in caso di test fallito.*

Immagine che contiene testo, schermata, Carattere, algebra

Descrizione generata automaticamenteImmagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamenteBOTTOM UP:**

Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

**→**   
 **→**  
**→**

**TOP DOWN**

Immagine che contiene testo, linea, schermata, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente  
  
**→**

**Problema in integration:**

* Nel caso in cui ci siano **loop**: due oggetti dipendono entrambi dall’altro e quindi è un problema
  + Testo singolarmente entrambi
  + Testo A con Bdriver
  + Testo B con Adriver
  + Unisco e testo

**Integrazione HW-SW**

* + Posso fare uno stub sia per sw che per hw
    - Faccio stub sw che sostituiscono hw
    - Provo ad unire hw