

tesi di laurea

# **Tecniche di automazione del testing: applicazione ad un caso di studio reale**

Anno Accademico 2005-2006

**relatore**

Ch.mo prof. Porfirio Tramontana

**correlatore**

Sig. Mario Polverino

**candidato**

Fabrizio Leoncavallo

Matr. 534/1701

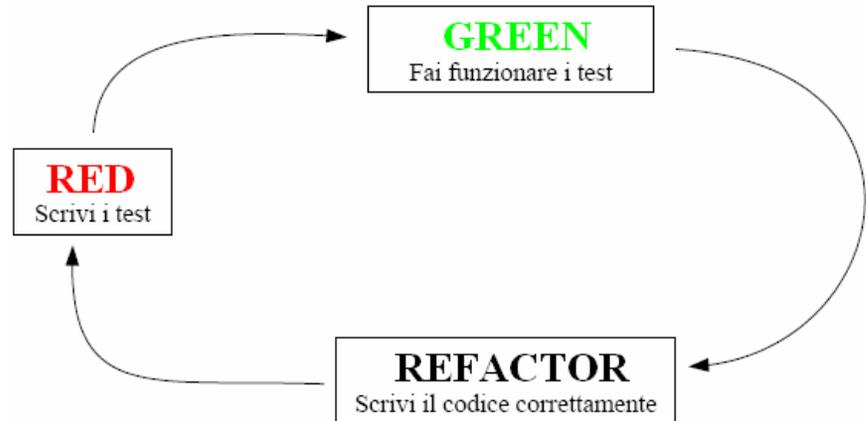
## Il testing del software

- Testare un prodotto software significa eseguirlo con l'obiettivo di scoprirne i malfunzionamenti
- *“Il testing può essere usato per dimostrare la presenza di difetti, ma mai per dimostrare la loro assenza”* (Tesi di Dijkstra)
- Automatizzare il testing ne aumenta efficienza ed efficacia
- Progettare ed eseguire casi di test di elevata qualità contribuisce a migliorare, parallelamente, anche la qualità del prodotto finale
- L'obiettivo è mettere in risalto il ruolo che la fase di testing riveste nello sviluppo di un software e l'importanza che ricopre la sua automazione
- Esempio pratico: caso di studio *PBX Gateway*

## Modello a cascata VS Sviluppo guidato dai test (TDD)

- Il testing si trova alla fine del ciclo di produzione: è possibile che venga eseguito in maniera superficiale o incompleta
- Processo alternativo: scrivere i test prima del codice (Test First)
- La struttura del sistema deve essere definita in anticipo
- Si ottiene “*clean code that works*” (Ron Jeffries)
- Però non è affatto semplice!
- Aumento del tempo stimato: 40%

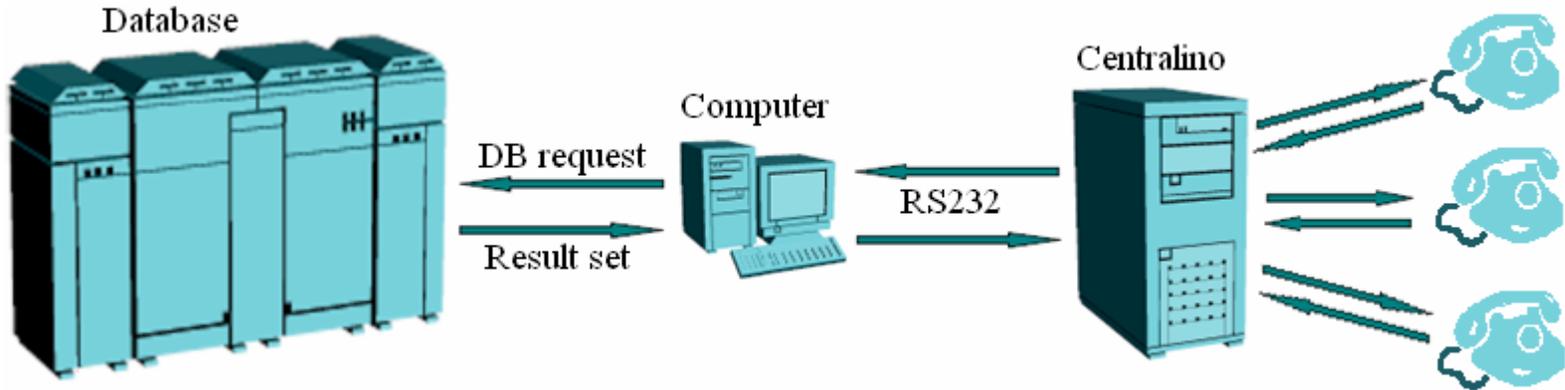
“Red/Green/Refactor” TDD Mantra



1. Non scrivere codice se non si dispone di un test automatico che abbia successo
2. Eliminare le eventuali duplicazioni in modo da mantenere il codice il più semplice possibile

# PBX Gateway

Prodotto software sviluppato nell'azienda Alpha Service



## Funzionalità

- Automazione e controllo dei PBX
- Gestione della camera tramite l'esecuzione delle funzioni più comuni
- Gestione di soggiorni e clienti
- Filtraggio dei messaggi che rappresentano telefonate ed addebiti
- Gestione del conto delle camere
- Archiviazione dei dati

# Strategia di testing

- **Approccio incrementale**
- **Testing di unità**
  - White-box manuale (Structured-Basis Testing, Boundary Analysis)
  - Black-box automatico (VBUit, driver e stub)
- **Testing di integrazione**
  - Manuale con criterio bottom-up
- **Testing di regressione**
  - Automatizzato tramite VBUit
- **Testing di sistema**
  - Manuale tramite Classi di Equivalenza
- **GUI testing**
  - Manuale tramite checklist

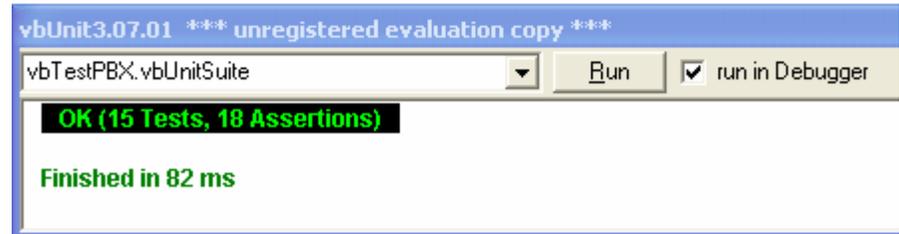
The screenshot displays the PBX Gateway software interface. At the top, there is a menu bar with 'File', 'Modifica', 'Visualizza', and 'Finestra'. Below the menu is a toolbar with icons for 'Camere', 'Telefonate', 'Addebiti', 'Svegli', 'Aggiorna', 'Piano Camere', and 'Prodotti'. The main area is divided into two panes. The left pane, titled 'mere', shows a grid of room cards for Piano 1, Piano 2, and Piano 3. The right pane, titled 'Camera 201', shows a detailed view of a room with various fields and buttons. At the bottom, there is a table titled 'Addebiti per la camera 104'.

Camera	Data	Ora	Reparto	Prodotto	Costo
104	10/11/2006	12.56.00	Bar	Caffè	1
104	12/11/2006	22.35.00	Minibar	Coca cola	0,75

# Testing di unità: VUnit

- Tool della famiglia xUnit
- Strumento Open Source per l'automazione del testing di unità e di regressione
- Tecniche black-box
- Ogni volta vengono eseguiti tutti i test progettati
- Aumento della qualità dei test e riduzione dei tempi di sviluppo
- La correttezza del codice viene verificata mediante le asserzioni

```
Public Sub TestStato()  
'Funzione che testa la funzione imposta_stato_pulizia  
  
Set cam = New clsCamera 'definisce cam  
Dim risultato As String 'definisce risultato  
  
'assegna a risultato la risposta della funzione  
'imposta_stato_pulizia(stato as String) as String  
risultato = cam.imposta_stato_pulizia("Pulita")  
  
'asserzione  
m_assert.StringsEqual "Pulita", risultato  
  
End Sub
```



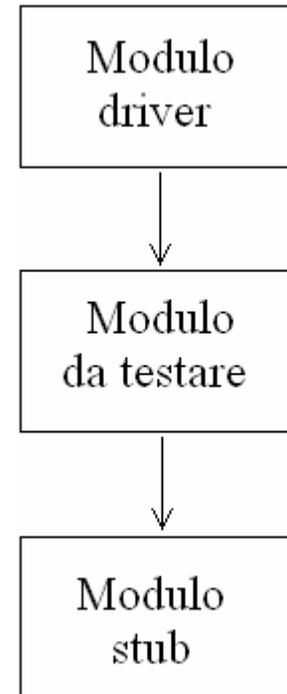
# Testing di unità: driver e stub

## Un driver:

- Prepara l'ambiente per il modulo chiamato
- Effettua una serie di chiamate di routine o di funzioni
- Verifica i risultati delle chiamate effettuate e li memorizza su file

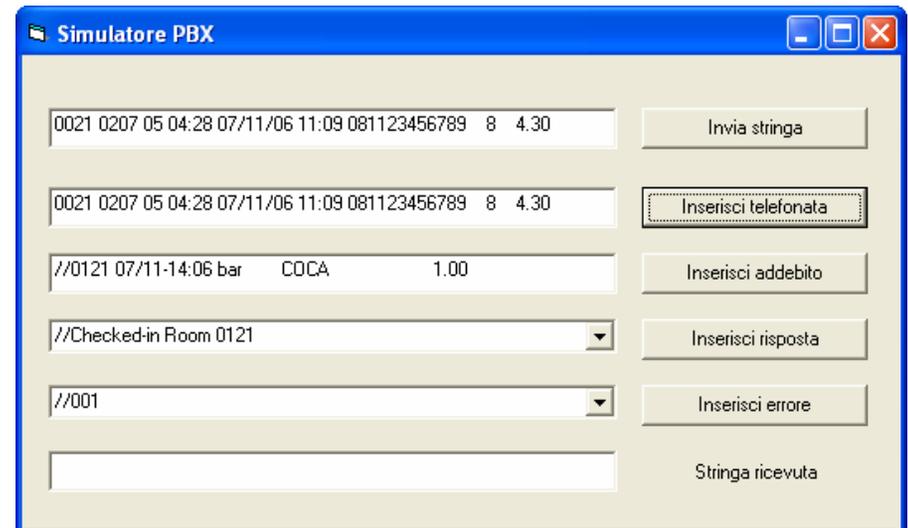
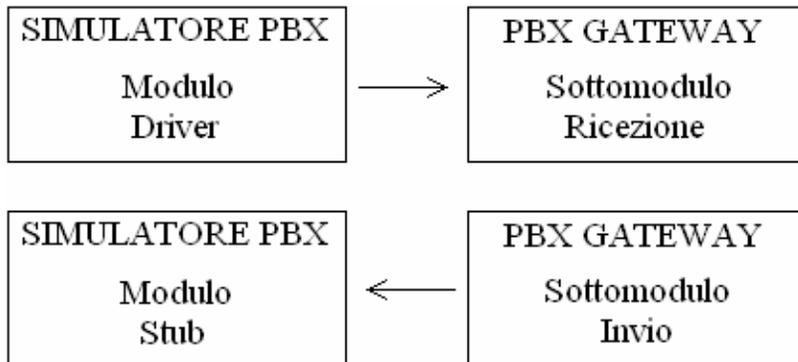
## Uno stub:

- Verifica l'ambiente predisposto dal modulo chiamante e l'accettabilità dei parametri
- Simula il funzionamento delle routine e delle funzioni richiamate
- Restituisce dei risultati, esatti rispetto alle specifiche o "accettabili"



# Simulatore PBX

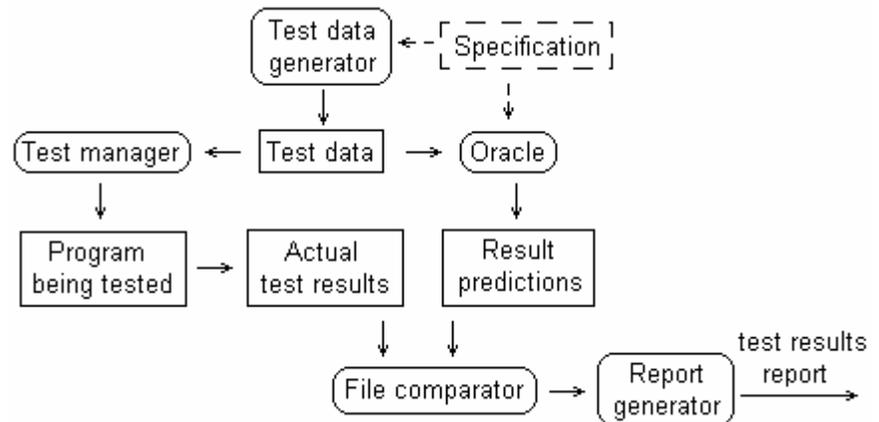
- Software semi-automatico di collaudo



- Driver generico (più costoso ma più riusabile)
- Stub specifico (meno costoso e meno riusabile)
- La realizzazione di driver e stub è gravosa
- Il costo per lo sviluppo viene però bilanciato dalla qualità dei casi di test

# Generazione automatica dei casi di test

- **Necessità di codice o specifiche formali**
- **Tecniche black-box**
  - **Model-checking**
- **Tecniche white-box**
  - **Algoritmi genetici**
- **Strumenti: pochi e non eccellenti**
- **JDocletUnit, CodeProTools, UTJML**
- **Tecniche manuali: Structured Basis Testing, Boundary Analysis, Classi di Equivalenza**



## Conclusioni e sviluppi futuri

- Il testing è una fase importante dello sviluppo software
- Non esistono una tecnica o un approccio migliori in generale
- Esistono varie tecniche più appropriate per lo specifico problema
- Le scelte, come quella fra testing manuale ed automatico, sono finalizzate al miglioramento dell'efficacia e dell'efficienza del testing.
- *PBX Gateway*: aggiunta di ulteriori moduli *Driver* atti alla comunicazione con PBX (tramite RS232 o TCP/IP) di altre marche
- Sviluppo di nuovi tool e miglioramento di quelli esistenti (famiglia xUnit)
- Discussione e proposizione di nuove tecniche e metodologie (MA, XP, TDD)
- La più grande sfida consiste nella generazione automatica dei casi di test