

# **Corso di Ingegneria del Software**

---

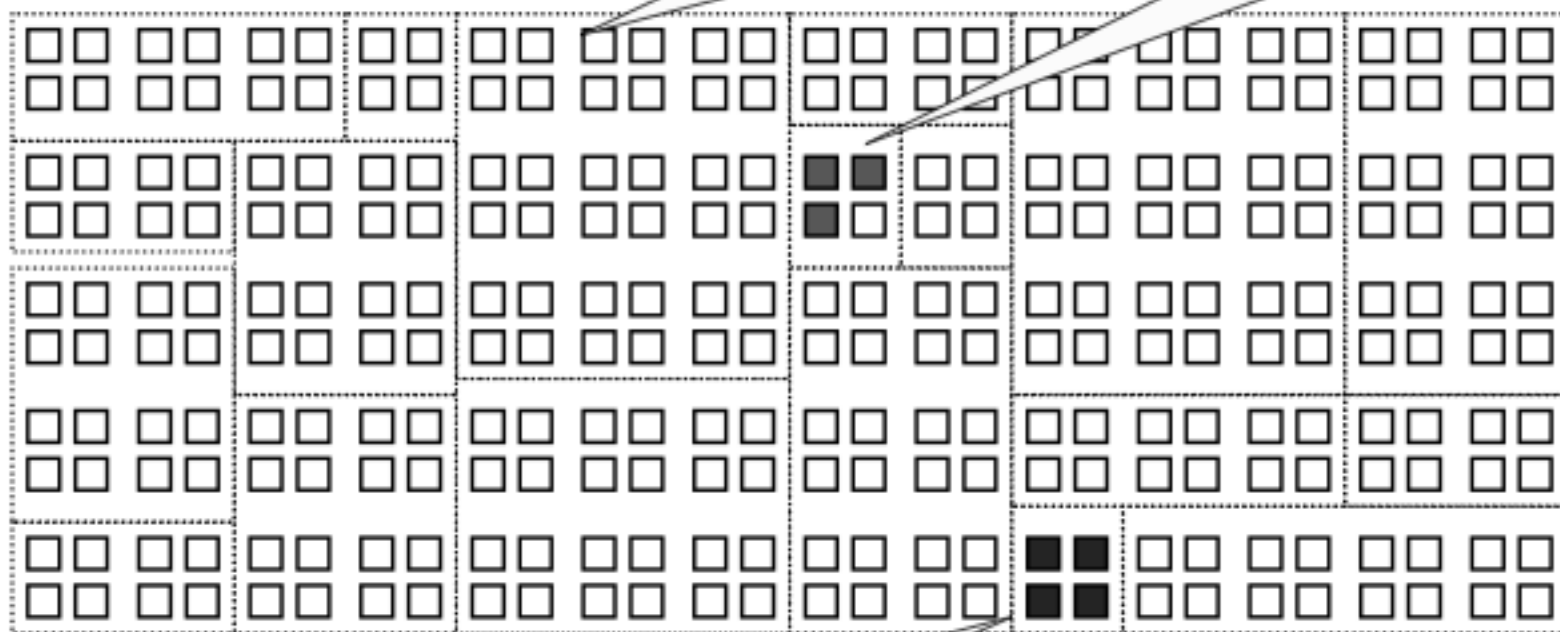
## **Esercitazione su Test Funzionale**

# Systematic Partition Testing

- Failure (valuable test case)
- No failure

Failures are sparse  
in the space of  
possible inputs ...

... but dense in some  
parts of the space



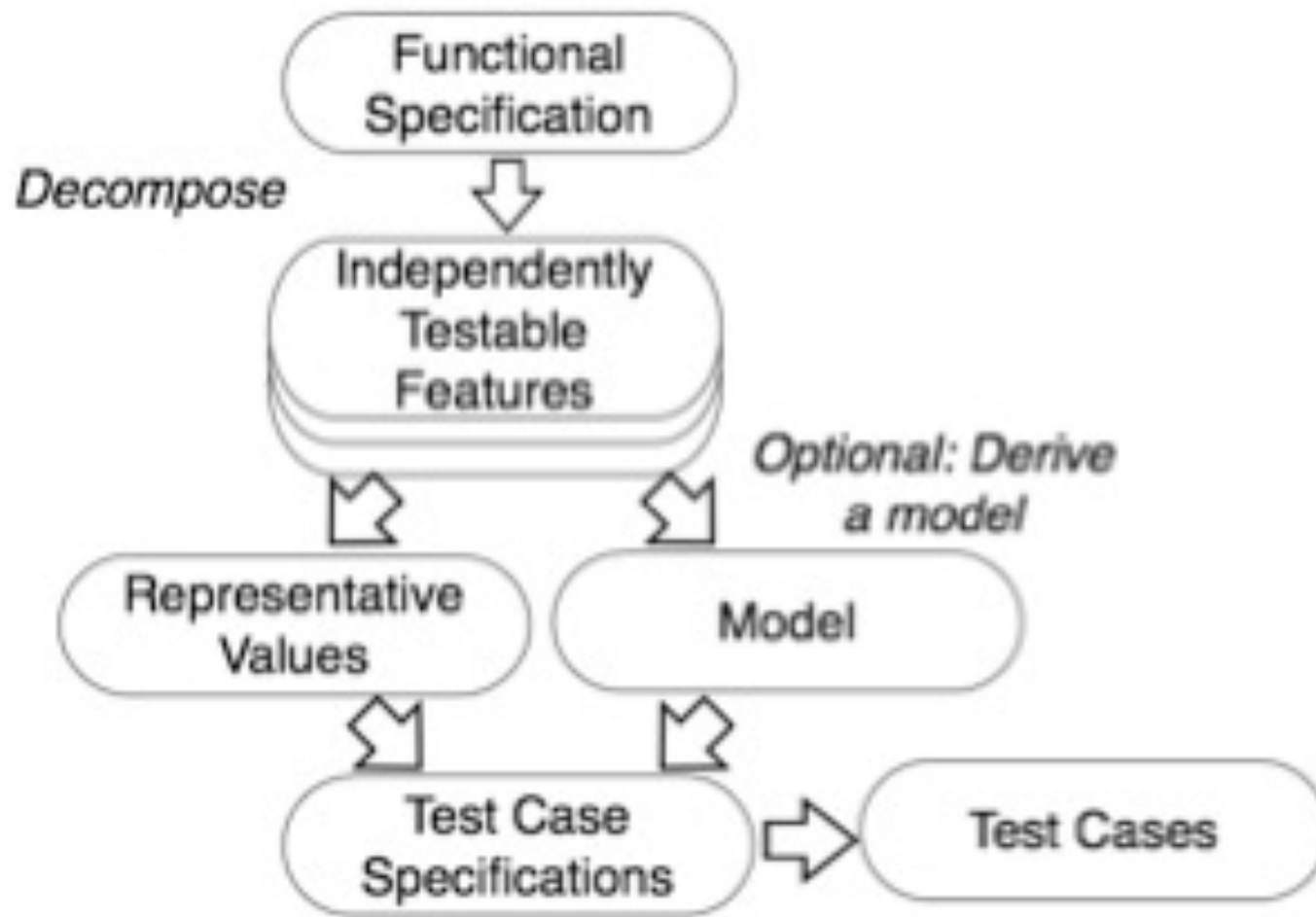
If we systematically test some  
cases from each part, we will  
include the dense parts

*Functional testing is one way of  
drawing pink lines to isolate  
regions with likely failures*

# Suddivisione in classi di equivalenza

- ♦ Il dominio dei dati di input è suddiviso dunque in classi di dati
- ♦ Un test-case ideale rileva da solo una classe di errori (es. una scorretta elaborazione di tutti i dati di tipo char), che altrimenti richiederebbe l'esecuzione di molti test, prima di intuire l'errore generico
- ♦ Una classe rappresenta un insieme di stati validi o non validi per le condizioni d'ingresso
- ♦ In generale una condizione di input può essere un valore specifico, un intervallo di valori, un insieme di valori, una condizione booleana

# Dalla specifica ai casi di test



# Dalla specifica ai casi di test

1. Decomporre la specifica
  - Se è grande, decomporla in *caratteristiche testabili indipendentemente*
2. Selezionare dei *rappresentanti*
  - Valori rappresentativi di ogni input oppure
  - Comportamenti rappresentativi di un modello
    - ◆ Spesso, semplici trasformazioni input-output non descrivono un sistema. Vengono usati modelli nella specifica del programma, nella progettazione e, nella progettazione dei casi di test
3. Formare le specifiche dei test
  - Tipicamente: combinazioni di valori di input, o comportamenti del modello
4. Produrre ed eseguire i test effettivi

# Approcci al test funzionale

- ◆ Data una specifica ci possono essere più approcci per derivare i casi di test
- ◆ Ad es., la presenza di molti vincoli nel domino di input suggerisce un metodo di partizionamento con vincoli (**category-partition** testing)
  - Al contrario, valori di input con pochi o nessun vincolo suggerisce un approccio combinatoriale (**pairwise testing**)
  - Se le transizioni tra un insieme finito di stati sono facilmente identificabili, un approccio basato su FSA può essere opportuno
  - Specifiche descritte in un dato formato suggeriscono approcci corrispondenti (basate sullo stesso formalismo)

# Esercizio 1

- ◆ E' richiesta la realizzazione di un software in grado di gestire spedizioni on-line. L'applicazione dovrà permettere l'utilizzo da parte dell'utente del servizio di spedizione di pacchi di diverso tipo tramite un interfaccia semplice ed intuitiva. L'applicazione prevederà una suddivisione in categorie per i diversi servizi offerti:
- ◆ Spedizione – Ricerca – Assistenza - Gestione Account - Gestione Pagamenti – Registrazione
- ◆ Tali servizi saranno disponibili per gli utenti registrati al sito. Un ospite (utente non registrato) potrà visionare i vari metodi di spedizione possibili con relativi costi e tempi ma non potrà effettuare nessun'altra azione di cui sopra (eccetto la registrazione).

# Esercizio 1

- ◆ E' prevista una vetrina offerte per cui in determinati periodi sarà possibile usufruire di offerte speciali sulla spedizione dei pacchi. Sono possibili diversi tipi di pagamenti: *Bollettino postale o bonifico bancario – postepay - Pagamento in contanti alla consegna del pacco alla sede più vicina - Carta di credito - Carta prepagata – Contrassegno*. Si accettano spedizioni con fatturazione in contrassegno solo nel caso in cui il destinatario sia un cliente registrato.



# Individuazione caratteristiche testabili indipendentemente

Esempio 1:

## SPEDIZIONE:

- ✦ L'utente potrà scegliere se consegnare il pacco a mano alla sede più vicina, oppure richiedere che venga preso in consegna da un corriere.
- ✦ La dimensione e il peso del pacco influiranno sull'importo finale così come la tempistica che consisterà nella scelta dei tempi di consegna; ovvero si potrà scegliere tra diversi tipi di spedizione in relazioni ai tempi di invio che potranno essere di 1, 3 o 5 giorni dalla consegna del pacco.
- ✦ In questo caso i tempi di consegna influiranno sul costo della spedizione in modo inversamente proporzionale, ovvero la spedizione in 1 giorno costerà di più della spedizione in 3 giorni. Il prezzo varierà, ovviamente, anche in funzione della tratta.

# Parametri e categorie

- ✦ Nome Utente
- ✦ Password
- ✦ Spedizione
  - Nome
  - Scelta consegna pacco
  - Scelta tipo (tempi) di spedizione
  - Indirizzo Origine
  - Indirizzo Destinazione
- ✦ Pagamento
  - Modalità
- ✦ Pacco
  - Numero
  - Dimensione
  - Peso

# Choices

## ✦ Nome Utente

- Valido (Registrato)
- Non Valido

## ✦ Password

- Valida
- Non Valida

## ✦ Destinatario

- Registrato
- Non Registrato

## ✦ Spedizione – Nome

- Valido
- Non Valido (caratteri non ammessi)
- Non Valido (>128 caratteri)

## ✦ Spedizione – Indirizzo origine

- Indirizzo Valido
- Indirizzo Non Valido: Caratteri non Validi
- Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi)
- Indirizzo Non Valido: > 256 caratteri

## ✦ Spedizione – Tipo

- 1 giorno
- 3 giorni
- 5 giorni

# Choices

## ★ Spedizione – Indirizzo destinazione

- ★ Indirizzo Valido
- ★ Indirizzo Non Valido: Caratteri non Validi
- ★ Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi)
- ★ Indirizzo Non Valido: > 256 caratteri

## ★ Spedizione – Consegna

- Manuale
- Prelievo dal Corriere

## ★ Pacco – Numero

- 0
- 1
- Many

## ★ Pacco – Dimensione

- Consentite
- Non Consentite

## ★ Pacco – Peso

- Consentite
- Non Consentite

## ★ Pagamenti

- ★ Bollettino postale
- ★ Bonifico bancario
- ★ Postepay
- ★ Pagamento in contanti alla consegna del pacco alla sede più vicina
- ★ Carta di credito
- ★ Carta prepagata
- ★ Contrassegno

# Numero di casi di test

Numero di casi di test senza vincoli (prendendo un 1 caso di test per test case specification):

$$2*2*2*3*4*3*4*2*7*3*2*2 = \mathbf{193536}$$

# Vincoli Errore

## ◆ Nome Utente

- Valido (Registrato)
- Non Valido [**ERROR**]

## ◆ Password

- Valida
- Non Valida [**ERROR**]

## ◆ Destinatario

- Registrato
- Non Registrato

## ◆ Spedizione – Nome

- Valido
- Non Valido (caratteri non ammessi) [**ERROR**]
- Non Valido (>128 caratteri) [**ERROR**]

## ◆ Spedizione – Indirizzo origine

- Indirizzo Valido
- Indirizzo Non Valido: Caratteri non Validi [**ERROR**]
- Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi) [**ERROR**]
- Indirizzo Non Valido: > 256 caratteri [**ERROR**]

## ◆ Spedizione – Tipo

- 1 giorno
- 3 giorni
- 5 giorni

# Vincoli Errore

## ★ Spedizione – Indirizzo destinazione

- Indirizzo Valido
- Indirizzo Non Valido: Caratteri non Validi [ERROR]
- Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi) [ERROR]
- Indirizzo Non Valido: > 256 caratteri [ERROR]

## ★ Spedizione – Consegna

- Manuale
- Prelievo dal Corriere

## ★ Pacco – Numero

- 0
- 1
- Many

## ★ Pacco – Dimensione

- Consentite
- Non Consentite [ERROR]

## ★ Pagamenti

- ★ Bollettino postale
- ★ Bonifico bancario
- ★ Postepay
- ★ Pagamento in contanti alla consegna del pacco alla sede più vicina
- ★ Carta di credito
- ★ Carta prepagata
- ★ Contrassegno

## ★ Pacco – Peso

- Consentite
- Non Consentite [ERROR]

# Numero di casi di test

Numero di casi di test con vincoli errore (prendendo un 1 caso di test per test case specification):

$$(1*1*2*1*1*3*1*2*7*3*1*1)+12 = 264$$



# Vincoli Single

- ★ Pacco – Numero
- 0 [SINGLE]
- 1
- Many

Numero di casi di test con vincoli single  
(prendendo un 1 caso di test per test case  
specification):  $(1*1*2*1*1*3*1*2*7*2*1*1)+13 =$   
**181**

# Vincoli Property

## ◆ Nome Utente

- Valido (Registrato)
- Non Valido [**ERROR**]

## ◆ Password

- Valida
- Non Valida [**ERROR**]

## ◆ Destinatario

- Registrato [property **REG**]
- Non Registrato

## ◆ Spedizione – Nome

- Valido
- Non Valido (caratteri non ammessi) [**ERROR**]
- Non Valido (>128 caratteri) [**ERROR**]

## ◆ Spedizione – Indirizzo origine

- Indirizzo Valido
- Indirizzo Non Valido: Caratteri non Validi[**ERROR**]
- Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi) [**ERROR**]
- Indirizzo Non Valido: > 256 caratteri[**ERROR**]

## ◆ Spedizione – Tipo

- 1 giorno
- 3 giorni
- 5 giorni

# Vincoli property

## ★ Spedizione – Indirizzo destinazione

- Indirizzo Valido
- Indirizzo Non Valido: Caratteri non Validi [ERROR]
- Indirizzo Non Valido: Indirizzo Inesistente (uno qualsiasi dei campi) [ERROR]
- Indirizzo Non Valido: > 256 caratteri [ERROR]

## ★ Spedizione – Consegna

- Manuale
- Prelievo dal Corriere

## ★ Pacco – Numero

- 0 [SINGLE]
- 1
- Many

## ★ Pacco – Dimensione

- Consentite
- Non Consentite [ERROR]

## ★ Pagamenti

- ★ Bollettino postale
- ★ Bonifico bancario
- ★ Postepay
- ★ Pagamento in contanti alla consegna del pacco alla sede più vicina
- ★ Carta di credito
- ★ Carta prepagata
- ★ Contrassegno [if REG]

## ★ Pacco – Peso

- Consentite
- Non Consentite [ERROR]

# Numero finale di test

Numero di casi di test con vincoli single  
(prendendo un 1 caso di test per test case  
specification):

$$(1*1*1*1*1*3*1*2*(2+2+2+2+2+2+1)*2*1*1) + 13 = 169$$

*Equivalentemente*

$$(1*1*2*1*1*3*1*2*6*2*1*1) + 12 + 13 = 169$$

# Osservazioni

- ✦ In molti casi, la definizione dei casi di test porta a raffinare le specifiche (ad es., il limite sull'indirizzo a 256 caratteri non era specificato; è venuto fuori a valle di un raffinamento, si suppone dopo un confronto con l'analista)
- ✦ E' importante rimarcare che la “granualrità” delle classi (ad es., quanti sotto-casi considerare per “indirizzo non valido”?) dipende dal limite sul numero massimo di casi di test che si vogliono eseguire, o meglio dal trade-off desiderato tra il grado di completezza del test (numero di test) ed il costo. In definitiva ciò è determinato dal budget a disposizione.
- ✦ Questa ultima considerazione determina anche la scelta dei vincoli