

Ingegneria del Software

Testing

Corso di Ingegneria del Software
Anno Accademico 2012/2013

Definizione IEEE

- ❖ *Software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item*

Definizione IEEE

Risultati corretti e non corretti

- ❖ **Mistake** – a human action that produces an incorrect Result
- ❖ **Fault [or Defect]** – an incorrect step, process, or data definition in a program
- ❖ **Failure** – the inability of a system or component to perform its required function within the specified performance requirement.
- ❖ **Error** – the difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

Introduzione – Il Testing

- ❖ E' una pratica di **Verifica e Validazione (V&V)** del software
- ❖ **Verification** (the first V) is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (IEEE 1990)
 - ❑ Assicura che il software implementi correttamente una specifica funzione
 - ❑ Are we building the product right?

Introduzione – Il Testing

- ❖ ***Validation*** (the second V) is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements (IEEE 1990)
 - ❑ Assicura che il prodotto soddisfi le richieste del cliente
 - ❑ Are we building the right product?

Black Box & White Box

❖ **Black Box Testing**

- Validation
- Il tester non ha accesso al codice, il codice è una black box
- Si basa sui requisiti

❖ **White Box Testing**

- Verification
- Il tester ha accesso al codice

Livelli di Testing

- ❖ Ci sono sei livelli di testing che possono essere fatti su un sistema software:
 - ❑ **Unit testing** is the testing of individual hardware or software units or groups of related units (IEEE 1990)
 - ❑ **Integration test** is testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them (IEEE 1990)

Livelli di Testing

□ *Functional and System testing*

- *Functional testing involves ensuring that the functionality specified in the requirement specification works.*
- *System testing is testing conducted on a complete, integrated system to evaluate the system compliance with its specified requirements (IEEE 1990)*



Livelli di Testing

- ❑ *Acceptance testing* is formal testing conducted to determine whether or not a system satisfies its acceptance criteria (the criteria the system must satisfy to be accepted by a customer) and to enable the customer to determine whether or not to accept the system (IEEE 1990)
- ❑ *Regression testing* is selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements (IEEE 1990)

Livelli di Testing

- ❑ **Beta Testing** when an advanced partial or full version of a software package is available, the development organization can offer it free to one or more (and sometimes thousands) potential users or beta testers

Livelli di Testing

Testing Type	Specification	Kind of Testing
Unit	Low-Level Design Actual Code Structure	White Box
Integration	Low-Level Design High-Level Design	White Box Black Box
Functional and System	High Level Design Requirements Analysis	Black Box
Acceptance	Requirements Analysis	Black Box
Beta	Ad hoc	Black box
Regression	Changed Documentation High-Level Design	Black Box White Box

Il Testing nel processo di sviluppo

- ❖ **Eseguire i test il prima possibile e quante volte possibile**
- ❖ **La pianificazione e l'esecuzione dei test è un processo iterativo**
 - ❑ **Devono essere scritti appena si hanno a disposizione i requisiti**
 - ❑ **Devono essere aggiornati ogni volta che i requisiti cambiano**

Black Box Testing

- ❖ **Test Funzionali e Test Comportamentali**
- ❖ **Cerca di individuare 5 tipi di errori**
 - ❑ **Funzioni mancanti o non corrette**
 - ❑ **Errori nell'interfaccia**
 - ❑ **Errori nelle strutture dati usate dalle interfacce**
 - ❑ **Errori di performance e di comportamento**
 - ❑ **Errori di inizializzazione e terminazione**
- ❖ **Attraverso questi test si determina se il sistema è in accordo con le specifiche**

Black Box Testing

- ❖ Per eseguire il black box testing non è necessario conoscere la struttura interna del codice
- ❖ Meglio se **non** pianificati ed eseguiti dai programmatori che hanno scritto il codice
- ❖ Molte organizzazioni hanno gruppi di testing indipendenti e l'attività di test è spesso affidata a terze parti

Test Planning

- ❖ **A test plan is a document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency plans (IEEE, 1990)**
- ❖ **Un componente importante del test plan è l'individuazione dei casi di test**

Test Planning

- ❖ A **test case** is a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement (IEEE, 1990).

Test Planning

- ❖ Il test plan dovrebbe essere fatto il prima possibile nel ciclo di sviluppo
- ❖ Il formato del test case è molto importante. Il template usato generalmente è il seguente:
 - ❑ **Test ID:** identificatore
 - ❑ **Description:** passi da eseguire e condizioni di ingresso
 - ❑ **Expected Results:** risultati attesi dal test
 - ❑ **Actual Results:** risultati ottenuti dal test

Test Planning

Test ID	Description	Expected Results	Actual Results

Test Planning

- ❖ **I campi del test case devono essere compilati in modo chiaro e non ambiguo**
- ❖ **Se si verifica un problema deve essere possibile ricrearlo**
- ❖ **I test case devono corrispondere ai requisiti espressi dal cliente**

Test early and often

- ❖ **Eseguire i test case il prima possibile è il modo migliore per ricevere feedback sul codice**
- ❖ **Per far girare i test case gli sviluppatori devono integrare i vari pezzi di codice in un codice di base**
- ❖ **Si raccomanda che il codice venga integrato una volta al giorno**

White Box Testing

- ❖ **Esamina la logica e la struttura del codice**
- ❖ **Structural testing, clear box testing, glass box testing**
- ❖ **Gli unit test utilizzano tecniche di white box testing per assicurare che le singole unità funzionino correttamente**
- ❖ **Una unit è un componente del codice**

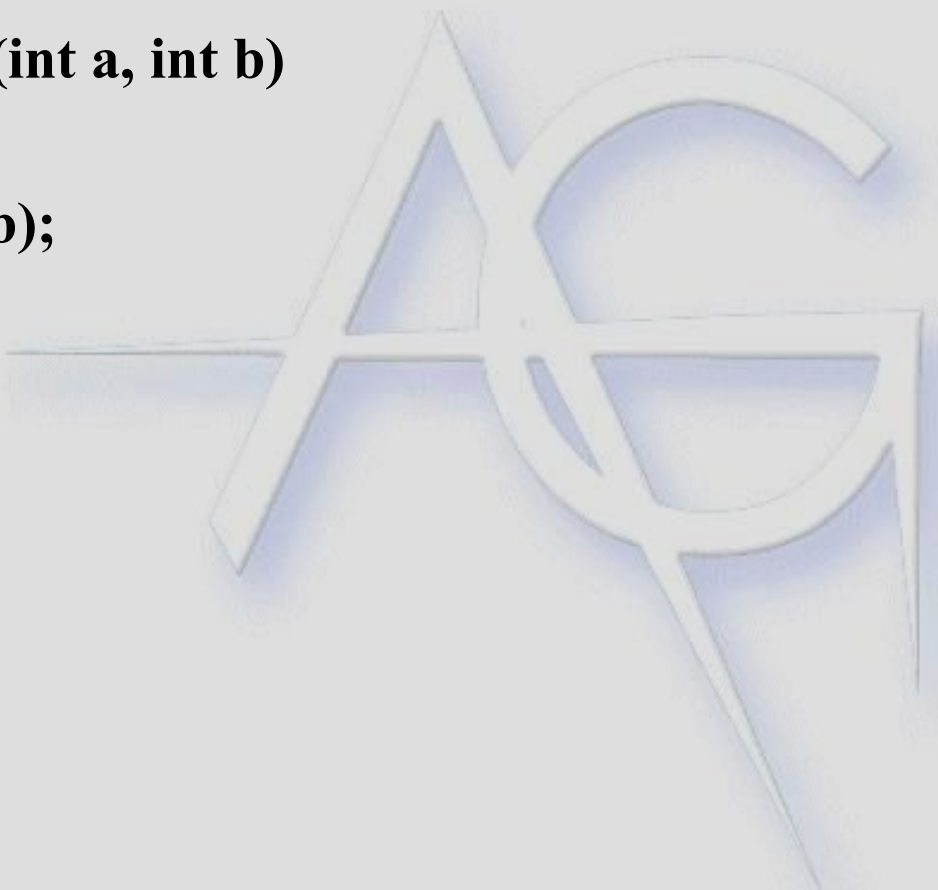
White Box Testing

- ❖ **Con i white box test è possibile:**
 - ❑ **Garantire che tutti i percorsi indipendenti di un modulo siano eseguiti almeno una volta**
 - ❑ **Eseguire tutte le decisioni logiche (sia true che false)**
 - ❑ **Eseguire tutti i loop compresi gli estremi**
 - ❑ **Fare il check delle strutture interne di dati**
- ❖ **Gli unit test permettono di catturare circa il 65% dei bug**

Esempio: Unit Test

```
/*Function that sum 2 integers*/
```

```
int addition(int a, int b)  
{  
return (a + b);  
}
```



Esempio-Unit Test

```
BOOL additionTest()  
{  
if(addition(2,3) != 5) return (FALSE);  
if(addition(0,0) != 0) return (FALSE);  
if(addition(6,0) != 6) return (FALSE);  
if(addition(-2,0) != -2) return (FALSE);  
if(addition(7,-2) != 5) return (FALSE);  
if(addition(-5,3) != -2) return (FALSE);  
if(addition(-5,-1) != -6) return (FALSE);  
return (TRUE);  
}
```