

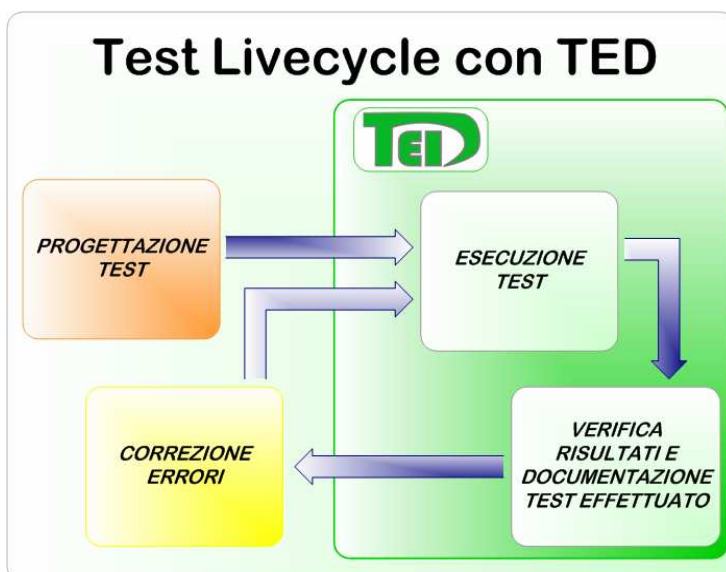
L'attività di test costituisce la fase più critica dell'intero ciclo di vita del software: dallo unit test, effettuato dal programmatore sul componente appena sviluppato, al system test effettuato dal capo progetto sull'intera applicazione, qualsiasi carenza o superficialità può indurre gravi inaffidabilità del software.

Per disporre di un software robusto e pronto a successive evoluzioni è sempre indispensabile un test accurato. Infatti solo l'esecuzione di un esauriente piano di test, dopo ogni modifica o evoluzione, è in grado di fornire garanzie sufficienti sulla non regressione del software in uso. Tuttavia, a causa del suo elevato costo, il test è spesso eseguito in modo superficiale ed incompleto, rendendo così inaffidabile il software ed inducendo una continua crescita dei costi di manutenzione ordinaria e straordinaria.



Make IT ha realizzato **TED** (*Test Execution & Documentation*), il prodotto software che semplifica e riduce i costi delle fasi di test per gli applicativi Java.

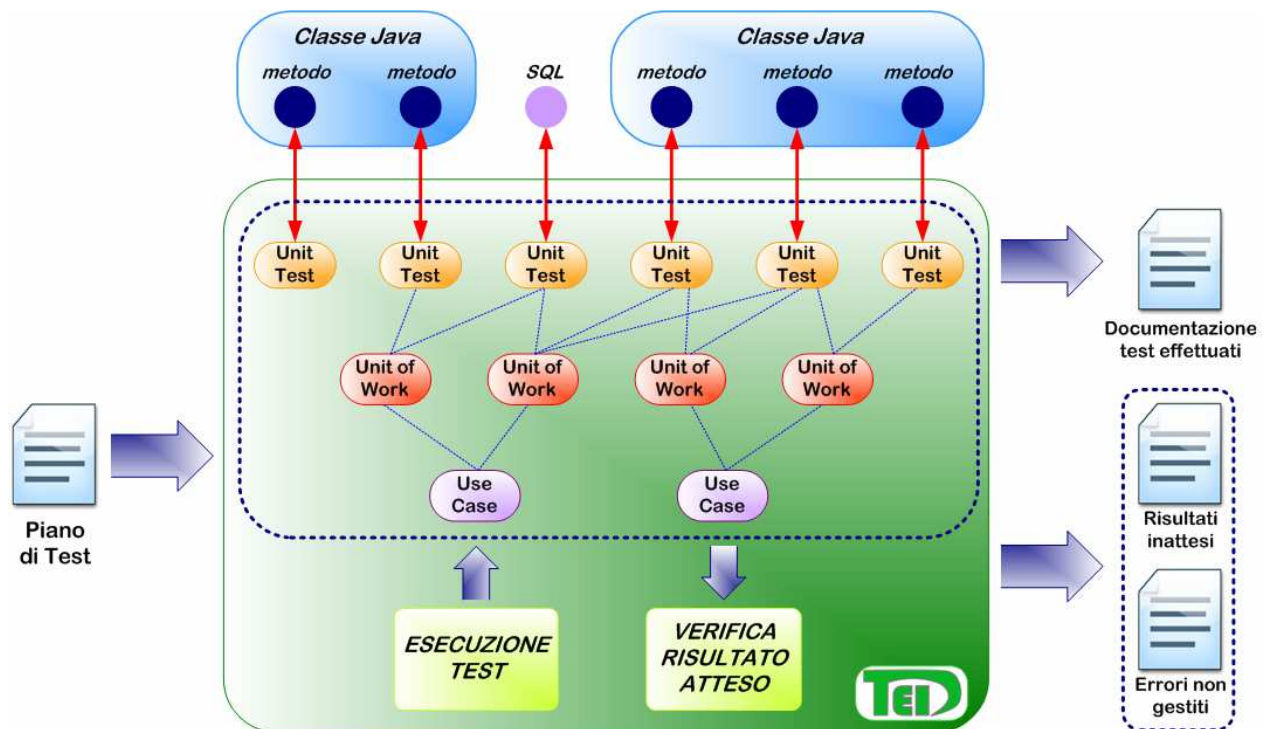
TED è lo strumento che consente di rendere il processo di test finalmente efficiente e produttivo: esegue i test e ne controlla i risultati in modo completamente **automatico**, quindi rapidamente, ripetutamente e senza il continuo intervento interattivo, ma con un semplice comando, rendendo veloce, completo e controllato l'intero ciclo di test. TED inoltre consente di **documentare** l'attività di test, sia come piano di test da allegare alle specifiche del progetto, sia come storico dei test effettuati. TED accompagna l'**intero team** di sviluppo durante tutta l'attività, in quanto può essere utilizzato sia dallo sviluppatore per eseguire il test di unità (*unit test*) sul componente appena sviluppato, sia dal capo progetto per seguire il test funzionale sull'intera applicazione (*system test*). Nelle attività di manutenzione del software in esercizio, grazie alla sua semplicità e velocità di utilizzo, TED consente inoltre di riproporre per ogni intervento di modifica l'intero piano di test, controllando così, nel modo più esaustivo possibile, la non regressione dell'applicativo aggiornato. L'adozione di TED come strumento di lavoro, oltre a produrre codice di maggiore qualità in quanto più robusto, permette anche di impiegare in modo **più produttivo** il team di programmazione. Il suo utilizzo costante e puntuale per la realizzazione di piani di test su tutto il codice sviluppato, costringe infatti il programmatore a definire i risultati attesi per ogni input e componente software, prevedendo così ogni possibile comportamento. La progettazione dei test diventa quindi una **guida allo sviluppo** che consente di ridurre i dispendiosi cicli di bug fixing. Inoltre la presenza di un accurato piano di test realizzato con TED consente di individuare immediatamente e con precisione la causa dei bugs nel momento in cui si manifestano, indicando l'istruzione in errore o la discrepanza con il risultato atteso.



TED semplifica e velocizza le fasi di test.

- ☞ Esegue in modo automatico sia test di unità che test funzionali e ne controlla i risultati.
- ☞ Consente di documentare l'attività di test, sia come piano di test da allegare alle specifiche del progetto, sia come storico dei test effettuati.
- ☞ Consente di produrre codice di maggiore qualità, perché più robusto e pronto a successive evoluzioni.
- ☞ Consente di verificare velocemente la non regressione degli applicativi in seguito alle piccole, ma molto critiche attività di manutenzione.
- ☞ E' espandibile ed utilizzabile per qualsiasi sistema applicativo, anche già esistente.

La soluzione



Caratteristiche tecniche:

- TED è un framework di test scritto in Java; è utilizzabile mediante una web application e le sue API sono a disposizione per essere richiamate in modalità batch o come libreria.
- E' espandibile ed adattabile al sistema software esistente.
- E' interamente configurabile mediante file XML, quindi può essere utilizzato sia dai programmatori per i test di unità, sia dai non programmatori per i test funzionali.
- Memorizza i risultati dei test in file XML, consentendo di mantenere uno storico dei test effettuati.

TED esegue tre tipi di test: *unit test*, *unit of work* e *use case*.

- Lo **unit test** è un test di unità, cioè un test effettuato su una singola funzionalità di un componente. TED in uno unit test testa il metodo di una classe Java o uno statement SQL. In uno unit test si specificano i parametri di input, il valore atteso e l'operazione con cui confrontare il valore atteso e il valore restituito (uguale, maggiore, minore...). Ad ogni funzionalità si associano tanti unit test quanti sono i suoi possibili comportamenti. Al termine della realizzazione di ogni nuova funzionalità si lanciano gli unit test associati alla stessa e quelli associati alle funzionalità precedentemente sviluppate per verificare il comportamento del componente in seguito alle nuove modifiche. Se lo unit test ha successo si ha una garanzia sul comportamento dinamico del codice; se il test non ha successo si hanno ottime probabilità che l'errore sia facilmente localizzabile. Se il test smette di funzionare in seguito ad una modifica del codice, con grande probabilità l'errore è dovuto alla modifica stessa e là va ricercato.
- La **unit of work** è utilizzata per testare la collaborazione tra componenti diversi. E' costituita da uno o più unit test attivabili in cascata e ciclicamente; gli unit test possono condividere e scambiarsi parametri. Al termine dell'esecuzione di ogni unit test è valutato il suo risultato; in caso di fallimento si prevedono due comportamenti: generazione di un warning o terminazione con fallimento dell'intera unit of work.
- Lo **use case** è utilizzato per testare i casi d'uso del sistema e quindi per effettuare i test funzionali. E' costituito da una o più unit of work attivabili in cascata.

Ambiente tecnologico:

| | |
|--------------------|--|
| Java | versione 5.0 o successive |
| Application server | Tomcat 5.5 o successivi. |
| DB | Microsoft SQL Server, Oracle, MySQL, DB2 e tutti i DB con driver JDBC. |
| Browser | Internet Explorer 6.0 e successivi, Mozilla Firefox 1.5 e successive. |