

IST. DI RICERCA  
BIBLIOTECARIA  
Posto *Azzurro*

*Consiglio Nazionale delle Ricerche*

**ISTITUTO DI ELABORAZIONE  
DELLA INFORMAZIONE**

**PISA**

**CRITERI PER LA CERTIFICAZIONE  
PUBBLICA DI SOFTWARE TRANSAZIONALE**

Bertolino A., Carlesi C., Fusani M.

Nota Interna B4-56  
Dicembre 88

# CRITERI PER LA CERTIFICAZIONE PUBBLICA DI SOFTWARE TRANSAZIONALE

A.BERTOLINO, C.CARLESI, M.FUSANI

## 1. INTRODUZIONE

Questa nota presenta una rassegna di possibili metodologie per la certificazione pubblica del software.

Intendiamo per certificazione pubblica la verifica e validazione (V&V) di un prodotto software condotta da parte di un ente indipendente (cioè esterno al processo di sviluppo) e al di sopra delle parti allo scopo di accertare la congruenza con norme di legge; la validazione termina appunto con l'emissione di un certificato.

Nel corso di questo lavoro faremo riferimento al software che controlla i registratori di cassa, come noto [1] soggetti ad una normativa ministeriale, adoperando il termine "software fiscale". Questo software, prima di poter essere posto in commercio, deve infatti essere sottoposto a certificazione da parte dell'Amministrazione dello Stato, la quale si avvale della consulenza di enti pubblici di ricerca (fra cui l'IEI del CNR), e di Università'.

La certificazione del software fiscale costituisce a tutt'oggi uno dei pochi esempi reali (l'unico?) di certificazione pubblica in Italia, nel senso sopra dato a questo termine.

Nel paragrafo 2 viene fornita un'introduzione alla problematica della certificazione del software in generale, con alcuni cenni alla certificazione del software dei misuratori fiscali.

Il paragrafo 3 presenta una rassegna di possibili approcci per la certificazione, adeguatamente allo stato dell'arte.

Il paragrafo 4, infine, conclude indicando quali debbano essere i criteri alla base della scelta di una metodologia, fra quelle discusse nel paragrafo precedente.

## 2. IL PROBLEMA DELLA CERTIFICAZIONE DEL SOFTWARE

### 2.1 Considerazioni generali

Il software controlla fasce di applicazioni sempre piu' vaste e complesse e assorbe progressivamente sempre maggiori responsabilita' e poteri decisionali, precedentemente riservati ad operatori esperti.

Cosi', nelle attivita' di tutti i giorni, l'utente si trova ad interagire con sistemi software, che lo assistono e lo influenzano in svariati modi, e non puo' fare a meno di affidarsi nella maggior parte dei casi a questi "interlocutori" artificiali.

Si evidenzia sempre piu' la necessita' di garantire che tali sistemi siano progettati e realizzati secondo i criteri della "regola d'arte", cosicche' si possa ragionevolmente rivolgere ad essi un certo livello di confidenza.

Questa garanzia di affidabilita' viene perseguita dalle attivita' di assicurazione della qualita' del software [2], che si accompagnano al processo di sviluppo; un ulteriore controllo puo' essere costituito dalla verifica e validazione indipendenti (IV&V) [3] del prodotto finito.

Oltre alle esigenze di affidabilita', esistono anche motivazioni di carattere economico che giustificano la certificazione: in generale, un prodotto sottoposto ad una certificazione pubblica aumenta il proprio valore, sia nei confronti di chi lo vende che di chi lo compra. Infatti, il primo, attraverso il riconoscimento della validita' del proprio prodotto, dimostra di aver lavorato secondo i migliori canoni della tecnica corrente; il secondo vede, nel certificato associato al prodotto, una garanzia aggiunta.

Nella certificazione del software esistono problemi inerenti all'aspetto di indipendenza dell'ente certificante e problemi tecnici derivanti dalla relativa "giovinezza" del settore.

Fra i problemi appartenenti alla prima classe, citiamo per tutti il problema di definire una procedura di certificazione di un prodotto che rimanga valida per diversi costruttori, e percio' per diverse metodologie di progettazione, sviluppo e documentazione.

Per quanto riguarda aspetti piu' tecnici, diremo che attualmente nel settore software sia la fase di produzione che quella di valutazione sono ancora in uno stato artigianale, in confronto con altre tecnologie; ad esempio nella produzione hardware sono disponibili tecniche e metriche gia' ben collaudate che consentono di valutare la qualita' intrinseca del prodotto [4]. Esiste un ramo molto attivo della ricerca scientifica che si occupa dei problemi della validazione del software, ma allo stato attuale non si dispone di risultati stabili e generalmente applicabili.

Bisogna riconoscere comunque che cominciano a diffondersi, da una parte, una maggiore sensibilita' al problema [5], e, dall'altra, alcune tecnologie e strumenti che

intervengono in fasi specifiche del ciclo di vita nella direzione di fornire garanzie sul "buon" funzionamento del software a partire dalla metodologia di progettazione.

Fondamentalmente, per garantire certe proprietà di comportamento di un prodotto software bisogna:

- i) definire rigorosamente le specifiche di funzionamento, in modo da consentirne la successiva verifica;
- ii) definire un metodo efficace di confronto fra le specifiche ed il comportamento effettivo.

Il processo di verifica della congruenza fra il prodotto effettivo e quello aspettato (ossia quello corrispondente ai requisiti richiesti dall'utente) è chiamato validazione [3],[6].

La validazione necessita di una documentazione precisa, strutturata e completa; un prodotto che non sia ben documentato crea difficoltà pesanti, e talvolta insormontabili, in fase di validazione.

Partendo perciò dal presupposto di disporre di un prodotto adeguatamente documentato, la validazione può essere condotta con tecniche di analisi statica e dinamica, applicate a tutti i prodotti intermedi (semilavorati) delle successive fasi del ciclo di sviluppo.

Fra le tecniche di analisi statica, le più semplici sono le tecniche di revisione [2] (o desk-checking), che sono adoperate in ambiente industriale. Si tratta di tecniche che non richiedono costosi investimenti o particolari specializzazioni, e hanno dimostrato nel tempo di essere abbastanza valide. La più efficace è la tecnica di Inspection [7], che consiste essenzialmente nell'analisi di ogni documento prodotto, guidata da una lista (checklist), che elenca i più comuni difetti, omissioni e inconsistenze, che è compilata e continuamente arricchita secondo il bagaglio culturale dell'azienda.

Le tecniche di analisi dinamica, invece, presuppongono la prova del prodotto in uso. L'analisi dinamica del codice è detta anche testing [6],[8]: un test consiste in una esecuzione del programma in esame e nell'osservazione dei risultati prodotti.

Esistono diverse metodologie di organizzazione dell'insieme dei test, che per un programma "reale" può comprendere anche migliaia di casi. Per esempio, si distinguono la tecnica di testing funzionale, o "a scatola nera", e la tecnica di testing strutturale, o "a scatola bianca". La prima è rivolta principalmente al controllo dei risultati senza entrare nel merito del come sono stati ottenuti; la seconda invece controlla la sequenza operativa attraverso cui si è ottenuto un certo risultato.

A supporto del processo di validazione, sono stati proposti e sviluppati strumenti rivolti all'automazione delle attività di analisi sopra esposte, che richiedono generalmente procedimenti ripetitivi e costosi, sia in termini di tempo che di risorse umane.

## 2.2 La metodologia dei "5 filtri"

Fra le tecniche di analisi statica o dinamica del software, accennate nel paragrafo precedente, ciascuna offre differenti vantaggi o limitazioni, a seconda della fase del ciclo di vita in cui viene applicata.

Nella validazione di un prodotto software già ultimato, ovvero del codice, una metodologia che riteniamo valida (e alla quale faremo riferimento nel seguito) consiste nell'applicazione in sequenza di 5 tecniche di analisi, che chiameremo "metodologia dei 5 filtri" [9].

I 5 filtri sono, nell'ordine:

- 1) Inspection del modulo.
- 2) Inspection di sistema.
- 3) Testing funzionale.
- 4) Misura di "copertura" a livello di modulo.
- 5) Misura di "copertura" a livello di sistema.

- 1) Inspection del modulo: la tecnica di analisi statica di inspection è stata brevemente descritta nel paragrafo precedente; tale revisione "a tavolino" deve essere condotta su ciascuno dei moduli costituenti il programma in esame. In questa fase si cercano soprattutto eventuali "difetti" di funzionamento, derivanti ad esempio da errori di specifica o di progettazione.
- 2) Inspection di sistema: si effettua la inspection del sistema integrato. In questa fase si cercano soprattutto eventuali "difetti" nelle interfacce fra i vari moduli.
- 3) Testing funzionale: si effettuano prove sperimentali del programma in base alle specifiche funzionali. In questa fase è auspicabile, anche se non strettamente necessario, il supporto di uno strumento automatico per il lancio dei test e la valutazione dei risultati, in modo da rendere questa attività veloce ed economica. Ci riferiamo agli strumenti noti in letteratura come "testbed" o "testdrivers": ne esistono ormai vari in commercio, a prezzi accessibili.
- 4) Misura di "copertura" a livello di modulo [10]: la misura di copertura è una delle tecniche di test che abbiamo classificato al paragrafo precedente come strutturale o a scatola bianca. Essa consiste molto brevemente nel tracciare il cammino percorso durante l'esecuzione di un modulo e nel misurare la percentuale di codice effettivamente eseguito.

5) Misura di "copertura" a livello di sistema [10]: si misura la percentuale dei moduli chiamati durante l'esecuzione del sistema integrato.

### 2.3 La certificazione del software fiscale

Il problema della certificazione del software nel caso dei misuratori fiscali si puo' formulare come l'esigenza di verificare che il programma che gestisce le risorse rilevanti dal punto di vista fiscale sia congruente con i requisiti espressi nelle relative norme [11]. Questa semplice definizione del problema necessita gia' di varie precisazioni. Intanto, le norme richiedono la certificazione solo di alcune parti del software di un misuratore fiscale, e cio' costituisce gia' un punto spinoso: e' infatti arduo riuscire ad analizzare in maniera separata il comportamento di moduli fra loro interfacciati. Cio' richiede notevole rigore durante la fase di progetto, e in particolare l'uso costante di criteri di progettazione quali modularita' e astrazione.

Un altro problema e' che i requisiti rispetto ai quali si deve effettuare la validazione sono definiti in testi di legge e quindi sono passibili di diverse interpretazioni.

Inoltre, tali requisiti di legge prescrivono delle funzionalita', ovvero dei requisiti di carattere positivo ("deve fare questo") e delle restrizioni, ovvero dei requisiti negativi ("non deve fare quello"). Mentre per i primi e' possibile accertare il soddisfacimento con prove che producano il funzionamento desiderato, per i requisiti negativi si puo' soltanto effettuare un determinato numero di prove, le quali daranno soltanto un certo grado di confidenza che l'evento indesiderato non si presenti, ma mai la garanzia assoluta. In altre parole, non possiamo mai sapere, dopo n tentativi rivolti a verificare un requisito negativo, che esso non sara' trasgredito al tentativo (n+1)esimo.

## 3. RASSEGNA DI POSSIBILI METODOLOGIE

Individuiamo anzitutto tre possibili linee di intervento:

- A: Certificazione indipendente del prodotto, ovvero condotta presso un Ente Esterno Accreditato (EEA).
- B: Certificazione da parte dell'EEA della validazione del prodotto condotta dallo stesso costruttore (autovalidazione).

C: Certificazione del processo.

Analizziamo di seguito questi tre possibili approcci, descrivendo per ognuno alcune proposte, non necessariamente contrapposte fra di loro, valutando per ogni proposta vantaggi e svantaggi.

Nell'esposizione che segue ci riferiamo al caso già più volte citato della certificazione del software fiscale, senza per altro perdere di generalità.

### 3.A. *Certificazione indipendente del prodotto*

#### 3.A.1- Software Fiscale fornito dall'Amministrazione dello Stato su ROM.

L'Amministrazione dello Stato, tramite eventualmente consulenti tecnici di fiducia, si fa carico delle seguenti azioni:

- definisce le specifiche funzionali riguardanti le sole funzioni fiscali.
- Provvede alla codifica dei relativi programmi, con gli adeguati attributi di qualità.
- Distribuisce il codice su ROM (Read-Only-Memory) ai costruttori, insieme alle interfacce che ne consentano un uso conforme ai requisiti di legge.

Questo tipo di intervento presenta da una parte evidenti difficoltà di carattere economico e politico. La realizzazione di un software fiscale universale ed universalmente usabile potrebbe inoltre richiedere lunghi tempi di attuazione. Va infine considerato quanto possa essere conveniente congelare una legge di carattere fiscale, e che fino ad oggi è stata in continua evoluzione, in un progetto di tali proporzioni.

Dall'altra parte, si otterrebbero i seguenti vantaggi:

- garanzia di adeguatezza e affidabilità del software fiscale, essendo stato prodotto sotto la supervisione della stessa Amministrazione dello Stato;
- riduzione dei controlli alla sola verifica che non siano state apportate variazioni alle risorse fiscali; trattandosi di software residente su ROM, la verifica di congruenza può essere ad esempio un semplice controllo di "checksum", ripetibile anche presso lo stesso utente finale.

#### 3.A.2- Software Fiscale prodotto dal costruttore.

Il Software Fiscale è prodotto interamente dal costruttore, che ne genera le specifiche funzionali a partire dal testo della legge.

L'intervento dell'Amministrazione dello Stato si limita alla valutazione e approvazione del prodotto finale in sede di apposite commissioni che consultano l'EEA.

In questo caso si possono distinguere due strategie:

1) La strategia attuale, non regolata da alcuna direttiva, per cui i laboratori (EEA) attuano l'attività di certificazione con modi e metodi basati principalmente sull'esperienza del personale impiegato.

2) L'Amministrazione dello Stato definisce una normativa (operativa e di documentazione) per la certificazione da adottarsi da parte di tutti i laboratori accreditati. Ad esempio, una strategia di intervento adeguata allo stato dell'arte potrebbe essere l'applicazione della metodologia dei 5 filtri, descritta nella prima parte, con parametri opportuni che tengano conto dei costi derivanti.

I vantaggi derivanti da una standardizzazione della procedura di verifica e validazione attraverso una normativa ministeriale sono numerosi e correlati fra loro:

- normalizzazione della documentazione di accompagnamento al codice fornita dal costruttore con un adeguato, seppur minimo, livello di qualità;
- comparabilità fra le validazioni condotte in diversi laboratori, difficile da raggiungere in assenza di normative;
- normalizzazione della documentazione redatta dal laboratorio, con conseguente diminuzione dei tempi di esame da parte delle commissioni ministeriali preposte all'emissione del certificato finale;
- un globale aumento del livello di qualità e affidabilità del prodotto certificato, con vantaggi di ritorno anche verso lo stesso produttore.

Per quanto riguarda i laboratori, questi dovrebbero adeguarsi alla suddetta normativa, potenziando le proprie strutture, al minimo, in termini di personale e know-how, o meglio con l'adozione di strutture di supporto fornite dall'avanzamento della tecnologia del settore (archivi automatici, strumenti per l'analisi e il testing, documentazione automatica).

### 3.A.3- Definizione delle specifiche funzionali del Software Fiscale da parte dell'EEA.

A partire dai requisiti di legge, l'EEA genera un documento di specifiche funzionali, ad un livello di astrazione indipendente dalla realizzazione, che possa essere poi utilizzato secondo due diversi approcci:

- a) - L'Amministrazione dello Stato potrebbe imporre tale documento come specifiche preliminari di progetto.
- b) - Le specifiche potrebbero essere un allegato alla legge, da utilizzare come riferimento durante la validazione da parte dell'EEA, senza essere per questo imposte come specifiche di progetto.

Nel caso di cui al punto a), i vantaggi sarebbero:

- semplificazione del processo di certificazione da parte del laboratorio accreditato, in quanto, viene saltata la fase di controllo di conformita` delle specifiche del costruttore ai requisiti di legge. L'unica verifica da eseguire e` la conformita` del prodotto alle specifiche date.

- per quanto detto sopra, una piu` immediata verifica delle funzionalita`.

Rimane aperta la questione della fattibilita` di un tale approccio. Da una parte l'Amministrazione dello Stato avrebbe l'autorita` necessaria ad imporre una precisa linea di progettazione e cio`, da un punto di vista teorico, potrebbe contribuire all'attuazione di un sistema rigoroso di validazione.

D'altra parte, questo tipo di intervento, per motivazioni principalmente di ordine "politico", non e` forse facilmente attuabile, anche perche` puo` essere interpretato dalle aziende come una interferenza non strettamente necessaria sul proprio modo di lavorare. Infatti, si puo` rimanere confacenti ai requisiti di legge anche con specifiche funzionali differenti da quelle proposte.

Nel caso di cui al punto b) si avrebbe che il produttore puo` scegliere di fare riferimento alle specifiche durante la fase di progettazione, o di prescindere da esse, a propria discrezione.

Nel primo caso, ci riconduciamo ad un tipo di certificazione di cui al punto a); nel secondo l'EEA o lo stesso produttore dovranno validare la conformita` alla legge delle specifiche adottate riconducendo il problema ad un tipo di certificazione di cui al punto 3 .A.2.

#### 3.A.4- Piano di V&V.

In alternativa ad un intervento di V&V sul prodotto finale, come prospettato nel paragrafo 3.A.2, si puo' pensare che l'EEA effettui la procedura di verifica e validazione, intervenendo durante le varie fasi del ciclo di vita del software.

A tale scopo l'EEA produce un piano di V&V [12] che contenga al minimo:

- a)- indicazioni generali sul modo di procedere;
- b)- criteri di V&V della specifica dei requisiti;
- c)- criteri di V&V del prodotto finale;
- d)- elenco dei rapporti di documentazione sulle prove e verifiche condotte.

A questo punto sono possibili due linee di intervento:

- a) l'EEA effettua la verifica e validazione, secondo il piano di V&V, parallelamente al processo di sviluppo, ovvero instaura una stretta linea di collaborazione con il

produttore; la V&V possono in questo modo essere condotte al termine di ciascun passo del ciclo di vita ovvero sui prodotti intermedi delle varie fasi.

Questa soluzione presenta serie difficoltà di realizzazione, in quanto l'EEA verrebbe a far parte integrante dell'attività aziendale e in alcuni casi a sovrapporsi ad organismi aziendali come quelli preposti al controllo qualità; non ultimo il problema della reticenza delle aziende ad aprirsi verso organismi esterni.

Il vantaggio è evidentemente una elevata tempestività di eventuali interventi correttivi; d'altra parte questo tipo di intervento presenta costi elevati che non riteniamo giustificati considerando che il prodotto non presenta caratteristiche di elevata criticità.

b) L'EEA sottopone il piano di V&V all'azienda produttrice come "guideline" da osservare nelle varie fasi del ciclo di sviluppo del prodotto.

In questo contesto l'EEA dovrà poi verificare che il produttore si sia attenuto alle disposizioni contenute nel piano; a questo scopo l'EEA potrà avvalersi sia dell'analisi della documentazione presentata, sia di tecniche di analisi statica o dinamica con cui riprodurre alcune delle attività di V&V raccomandate, come ad esempio alcuni dei test. A tale proposito si potrebbe presupporre che il piano di test sia definito direttamente dall'EEA come descritto nel punto seguente.

L'approccio b), perciò, per certi aspetti, si può ricondurre ad una certificazione dell'autovalidazione del produttore, come descritto nella parte al sottoparagrafo 3.B della presente nota.

### 3.A.5- Definizione del piano dei test da parte dell'EEA.

L'EEA definisce un piano dei test da condurre in fase di analisi dinamica del codice.

Tale piano di test è di tipo funzionale, o "black box", ed è generato a partire dai requisiti di legge. Poiché questi ultimi possono dar luogo ad interpretazioni tecniche differenti, potrà rendersi necessaria, come passo intermedio, la definizione da parte dell'EEA delle specifiche funzionali. Questa possibilità è stata trattata al sottoparagrafo 3.A.3.

Ovviamente l'EEA prenderà in considerazione per l'analisi soltanto quelle funzionalità che sono contemplate dalla legge.

L'analisi funzionale da sola tuttavia, non garantisce una completa analisi del codice. Opportuni criteri dovranno perciò essere stabiliti, tramite cui ottenere un livello accettabile di confidenza dell'adeguatezza dei tests condotti. Una possibilità potrebbe essere quella di condurre parallelamente un'attività di testing strutturale, o "white box".

### *3.B. Certificazione della auto-validazione.*

3.B.1-Il costruttore esegue in fabbrica, durante lo sviluppo, la validazione del software fiscale in base a criteri e procedure definiti dall'EEA, ad esempio, tramite un piano di V&V, come accennato al paragrafo 3.A.4.

In altre parole, si ritorna al caso di definire una procedura di validazione adeguata allo stato dell'arte, come prospettato nel paragrafo 3.A.2; tuttavia, tale procedura, anziché essere condotta presso laboratori esterni accreditati, è applicata dallo stesso produttore. L'EEA interverrà soltanto successivamente alla fase di validazione, per certificare, questa volta, che il processo di validazione sia stato condotto in accordo alle specifiche. La certificazione del processo di validazione potrà essere condotta, ad esempio, su adeguata documentazione, redatta sempre in accordo a normative EEA.

3.B.2- Senza entrare in dettaglio sulla procedura di validazione, questa conterrà comunque una fase di analisi dinamica del codice o testing, come già delineato al paragrafo A-5. Questa analisi potrà essere condotta utilizzando strumenti automatici appropriati, che il produttore potrà scegliere adeguatamente alle proprie esigenze e modalità di sviluppo. Non si ritiene infatti opportuno imporre da parte dell'EEA l'uso di determinati prodotti e strategie. L'EEA potrà però richiedere, quale garanzia di un valido intervento, un certo livello di copertura del testing, come criterio di terminazione per questa fase.

3.B.3- È quasi superfluo osservare che il costruttore dovrà garantire la ripetibilità dei test condotti; a questo scopo l'EEA avrà ovviamente la facoltà di intervenire mediante alcuni test campione.

### *3.C. Certificazione del processo.*

Anziché validare di volta in volta il prodotto software, l'Amministrazione dello Stato potrebbe decidere di validare il processo di sviluppo del costruttore. Ciò può essere effettuato con diverse modalità di intervento, di livello gradatamente meno restrittivo nei confronti del costruttore.

#### *3.C.1- Definizione e standardizzazione del processo.*

L'EEA definisce un processo di sviluppo standard, ad esempio prendendo come riferimento uno standard internazionale già circolante (DoD, IEEE,...) ed applicandolo al caso specifico. In questo caso, il costruttore dovrà adottare tale modello. L'EEA verificherà la conformità al modello mediante ispezioni in fabbrica oppure tramite una

procedura (da stabilirsi) di approvazione del processo, ad esempio tramite analisi della documentazione.

3.C.2- Definizione di un minimo set di regole cui il costruttore si deve attenere.

L'EEA definisce una normativa soltanto per alcune sottofasi del ciclo di sviluppo, cui il costruttore dovrà attenersi. Ci si riconduce, perciò, ad un sottocaso del paragrafo precedente, ovviamente meno restrittivo.

Tipicamente, si potrebbe richiedere l'inspection delle specifiche, con la produzione di un documento standard, oppure un adeguato livello di copertura nel testing dei moduli software.

3.C.3- Inspection del processo.

L'approccio meno restrittivo, infine, è quello in cui l'EEA non fornisce indicazioni sul procedimento da seguire, ma effettua la validazione del processo interno del costruttore tramite ispezioni.

Queste ispezioni possono essere schedate in maniera periodica e concordata, oppure essere effettuate senza preavviso (spot-check).

#### **4. CRITERI PER LA SCELTA DI UNA METODOLOGIA DI INTERVENTO.**

L'EEA di cui si è parlato dovrà essere un Ente o Istituto altamente qualificato, che offra insieme garanzia di imparzialità e un adeguato know-how nel campo della validazione del software.

Un criterio fondamentale da perseguire nella definizione di una procedura di validazione del software è che qualunque documentazione dovrà essere orientata verso il massimo grado possibile di standardizzazione. Questo criterio è stato enfatizzato più volte anche nella rassegna delle possibili linee di intervento descritte nel paragrafo 3.

I vantaggi conseguenti sono notevoli, specie in ambito di una certificazione sottomessa a commissioni istituite dall'Amministrazione dello Stato, in quanto la validazione e l'approvazione finale sarebbero basate sull'analisi di documenti standard per tutti i produttori, con economia massima di tempo e risorse.

La documentazione, inoltre, dovrà essere concepita in modo da rendere ogni tentativo di "aggiramento" il più costoso e difficoltoso possibile mentre il processo di validazione richiesto dovrà essere, paradossalmente, la scelta meno costosa.

Pensiamo, infine, che nessuno dei diversi approcci prospettati fornisca da solo il miglior metodo di intervento. Sicuramente, una procedura di validazione adeguata sarà ottenuta

da un approccio misto, che tragga vantaggio proprio dalla diversificazione degli interventi. Così, sarà senz'altro opportuno intervenire con una validazione e del prodotto e del processo, poiché ciascuna delle due adoperata isolatamente fornisce garanzie limitate relativamente al soddisfacimento di requisiti di legge.

## 5. BIBLIOGRAFIA

- [1] Bertolino A., Fusani M., "Software Validation: A Government-imposed Challenge to the State of the Art in Certification", *Computer Standards & Interfaces*, Volume 6, No. 4, 1987.
- [2] Deutsch M.S., Willis R.R., *Software Quality Engineering*, Prentice-Hall, 1988.
- [3] ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology .
- [4] Manuale MIL 217-D.
- [5] "Atti del Convegno CQS-88, Osservatorio sulla Qualità del Software e sul CASE", Milano, 14-16 Settembre 1988.
- [6] Miller E.F., Howden W.E., *Tutorial: Software Testing and Validation Techniques*, Second Edition, IEEE Computer Society, 1981.
- [7] Fagan M.E., "Design Code Inspections to Reduce Errors in Program Development", *IBM Systems Journal*, Vol.15, No.3, 1976.
- [8] DeMillo R.A., et al., *Software Testing and Evaluation*, Benjamin/Cummings, 1987.
- [9] Miller E.F., Private communication.
- [10] Miller E.F., "Software Testing Technology: An Overview", in Vick C.R., Ramamoorthy C.V. (Eds.), *Handbook of Software Engineering*, Van Nostrand Reinhold, 1984.
- [11] Legge N.18/83 della Repubblica Italiana, 26 gennaio 1983.
- [12] ANSI/IEEE Std 1012-1986, IEEE Standard for Software Verification and Validation Plans.