Produzione di software didattico: si volta pagina

Mauro Tavella
ITD-CNR, Genova

Nuove funzionalità e nuovi contesti di produzione per il software didattico.

Cambia il software didattico e cambiano anche il ruolo e le competenze richieste a chi lo deve progettare, realizzare e commercializzare.

La grande industria è entrata massicciamente nel campo e propone, a costi limitati, programmi educativi dalle caratteristiche realmente "professionali"; questi prodotti della nuova generazione sfruttano risorse tecnologiche di alto livello, rese disponibili dalle prestazioni sempre migliori dei computer, dalla elevata qualità dei software di sistema più recenti, dall'utilizzabilità di supporti di memorizzazione di grande capacità [Steele, 1996].

Dietro i prodotti più recenti e innovativi si intravedono non solo nuovi strumenti tecnologici e nuove funzionalità ma anche nuove idee, nuove energie e nuove professionalità.

Questa nostra riflessione riguarda direttamente proprio la produzione di software didattico; prende in considerazione da un lato le funzionalità più innovative rese possibili dalla disponibilità di tecnologie di avanguardia e, dall'altro, i nuovi contesti ed i nuovi modelli per la progettazione e la realizzazione di questi prodotti, per i quali si intravedono spazi di mercato sempre crescenti, visto l'incremento esponenziale della diffusione dei personal computer nelle case.

Cercheremo di dare uno sguardo a che cosa di nuovo c'è, o ci potrebbe essere, *dentro* al software didattico (caratteristiche e possibilità operative), ma cercheremo anche, contemporaneamente, di sottolineare chi e che cosa c'è, o ci dovrebbe essere, *dietro* la sua realizzazione (tipo di competenze richieste e di impianto strutturale necessario).

Accenneremo, giocoforza solo indirettamente, a tutte le problematiche connesse con gli aspetti di comunicazione che riguardano sia l'adattività dell'interfaccia sia l'efficacia della metodologia didattica utilizzata.

NUOVE FUNZIONALITÀ PER IL SOFTWARE DIDATTICO

Quando si parla di novità e di differenze fra il software didattico attuale e quello di qualche tempo fa, la prima cosa a cui si pensa è, inevitabilmente, l'interfaccia: pagine vivaci con disegni a colori, suoni, animazioni nei prodotti di oggi e, invece, pagine più grigie, statiche, a volte rigidamente testuali, in quelli di ieri.

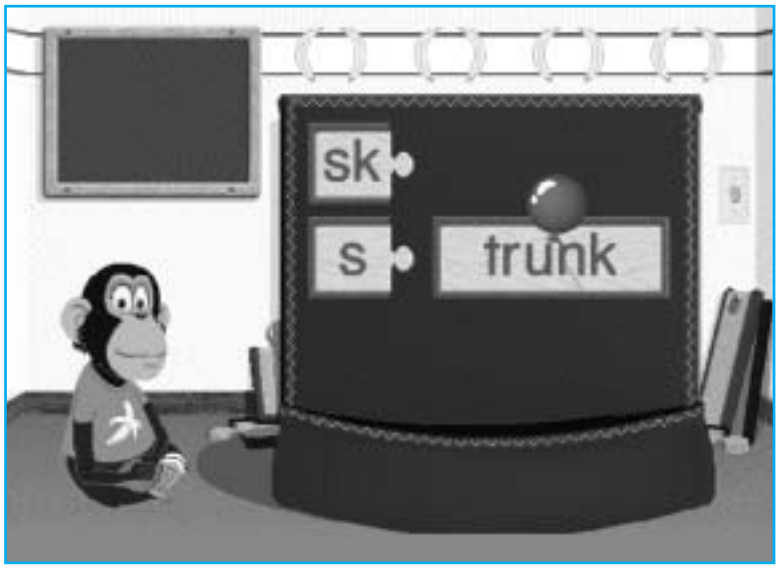
L'evoluzione delle tecnologie per lo sviluppo di software è stata cosi rapida da provocare un divario talmente abissale nei prodotti, che l'occhio esperto è oggi in grado di datarli, dando semplicemente uno sguardo ad alcune pagine - video.

Il ruolo dell'interfaccia non è, tuttavia, solo quello di proporre, in modo più o meno ameno e gradevole, i contenuti educativi. Al di là degli aspetti "coreografici" esteriori,

un'interfaccia, adeguatamente progettata, presenta degli aspetti di "sostanza" nel dialogo educativo [Cohen e al., 1995]. La facilità di interazione, la maneggevolezza di un prodotto e la sua comprensibilità sono fortemente connessi proprio con la plasticità e l'intelligenza dell'interfaccia, con la cura e l'attenzione dedicate alla sua progettazione [Frye e Soloway, 1987].

Il suono, per fare un esempio, può semplicemente svolgere la funzione di rendere più gradevole o simpatica l'interazione, ma può anche essere studiato in modo da consentire, o facilitare, nuove strategie di apprendimento (ad esempio, nel caso dell'insegnamento della lettura).

Allo stesso modo le icone, se opportunamente "disegnate", possono non soltanto guidare all'uso del prodotto, ma anche favorirne l'utilizzo da parte di certe fasce di utenza, attenuando l'impegno e lo sforzo cognitivo richiesto [Fullerton e Happ, 1993].



DIETRO L'INTERFACCIA

Il vero valore aggiunto che oggi qualifica un prodotto rispetto ad altri non sta dunque *dentro* l'interfaccia, ma *dietro* di essa, nello studio delle strategie di dialogo e di fruizione (che nell'interfaccia trovano poi il loro contesto applicativo).

Ancora *dietro* l'interfaccia stanno una serie di funzionalità accessorie che assumono diversa rilevanza a seconda degli obiettivi che un prodotto si propone ma che sicuramente ne estendono la fruibilità, il mercato e la pregnanza scientifico - progettuale.

Tra queste certamente l'accessibilità e la portabilità.

L'accessibilità

Per accessibilità si intende la capacità del prodotto ad adattarsi alle esigenze di utenti, che, per problemi specifici, non sono in grado di utilizzarlo nella sua configurazione standard e riguarda in particolare certe categorie di disabili (tipicamente i disabili motori ed i disabili della vista).

L'ideale, oggi finalmente emergente, di integrazione totale e completa del disabile nella scuola sottolinea, in maniera inequivocabile, la necessità che anche i prodotti educativi prevedano a priori la possibilità di uso da parte di qualunque categoria di utenti, senza discriminazioni.

Mentre ieri esisteva una categoria di software didattici "per" disabili, cioè creati appositamente per loro, con caratteristiche specifiche pensate e strutturate ad hoc, oggi l'idea di software "accessibile" sostituisce efficacemente quella di software "speciale" per disabili [Ott, 1997] ed è certamente in questo senso che la produzione deve muoversi, mostrando di aver compreso appieno la nuova dimensione della disabilità.

Un tempo le problematiche legate all'accessibilità potevano essere risolte esclusivamente con l'introduzione all'interno del programma di funzionalità specifiche per l'accessibilità[1]; in questo caso, ogni singolo realizzatore di software progettava in maniera diretta e autonoma l'accesso a quelle periferiche alternative che si ritenevano utili (interruttori, tastiere braille ecc...).

Oggi i sistemi operativi più recenti, come Windows95, WindowsNT e MacOS, includono funzionalità di accesso facilitato per disabili[2]; ciò consente che qualsiasi software sviluppato nel rispetto delle normative previste dal sistema operativo possa funzionare con strumenti per l'input alternativo e con modalità standard per tutte le applicazioni.

La standardizzazione e la generalizzazione di queste funzioni (che, purtroppo, vanno a risolvere soltanto alcuni dei problemi di accesso da parte dei disabili) innalza il livello di base dell'accessibiltà dei prodotti; ciò non esclude, ma supporta la realizzazione di strumenti per l'accessibilità ancora più raffinati, più adeguati, più completi e più funzionali.

1 *Esistono intere collane di prodotti di questo tipo come, ad esempio, quelli della Edmark Co., tradotti e distribuiti in Italia; essi prevedono l'uso di strumenti di input alternativi, i quali possono sostituire tastiera e mouse (Vedi TD 8/9 la descrizione del programma "Sammy's Science house" presentato nella Vetrina BSD).*

2 *Si veda TD n. 10: Rubrica TD e disabilità, pp. 67-68.*

La portabilità

Oggi i prodotti software di qualità hanno tempi e costi di progettazione ed realizzazione estremamente elevati e devono dunque necessariamente avere anche un'ampia utilizzabilità.

L'idea della portabilità di un software è direttamente connessa proprio con quello della sua utilizzabilità, nel numero più ampio possibile di contesti diversi.

In realtà, tradizionalmente, il termine è sempre stato usato in senso strettamente tecnico, come *portabilità tecnologica* intendendo: "Il grado di mobilità di una applicazione informatica, che consente al prodotto di essere trasportato da un tipo di computer ad un altro cioè utilizzato su più tipi di computer" [Dahlstrand, 1984]; in questa accezione, in sostanza, il termine "portabilità" definisce se, e con quanta facilità, un prodotto software può essere utilizzato su piattaforme hardware diverse; si è molto spesso parlato, per fare un esempio, del livello di portabilità di prodotti software nati in ambiente Macintosh, rispetto alle piattaforme MS-DOS o Windows.

Per quanto riguarda gli aspetti di *portabilità tecnologica* del software, il problema oggi non è tanto quello di sviluppare un prodotto che si adatti a diverse piattaforme hardware (questo problema, è oggi tecnologicamente vicino alla soluzione, infatti esistono sia compilatori in grado di produrre codice per diversi ambienti, sia prodotti quali Macromedia Director, Ms PowerPoint etc. i cui documenti sono fruibili su diverse piattaforme, sia software Java, che consente una realizzazione indipendente dalla piattaforma) quanto quello di sviluppare prodotti "duraturi" anche all'interno della stessa piattaforma; al concetto di *portabiltà transpiattaforma* si aggiunge un concetto, più ristretto ma altrettanto rilevante, di portabilità all'interno della stessa piattaforma: dal momento che è facile ipotizzare che l'evoluzione del Sistemi Operativi e dell'hardware continui a ritmo incessante anche nei prossimi anni, è necessario tenerne conto in fase di progettazione dei prodotti software attuali, e seguire quindi stili e canoni di produzione che rendano facile la manutenibilità del software, la sua aggiornabilità, in questo modo si potrà consentire a ciascun prodotto una durata che giustifichi le energie richieste per la sua progettazione e realizzazione; in questo ambito, una regola, non sempre facile da attuare, è quella di separare, il più possibile, in fase di progettazione, ma, soprattutto di implementazione, i contenuti dal metodo di presentazione.

Estendendo l'idea di portabilità al software educativo, l'accezione stessa del termine si è allargata, sconfinando dall'ambito puramente tecnico - tecnologico a quello più genericamente umano e socio - culturale. Non tutti i prodotti noti in un determinato contesto culturale possono essere utilizzati direttamente così come sono in ambiti diversi, in quanto cambiano le abitudini, le tecniche didattiche, la lingua, i programmi, le dinamiche di studio e di apprendimento.

Collis e De Diana, a questo proposito, individuano tre fattori che giudicano "critici" relativamente alla possibilità che uno stesso prodotto didattico possa essere usato in contesti educativi diversi [Collis B., De Diana I., 1990]:

- *fattori strettamente educativi*, legati ai programmi didattici, ma anche alle abitudini educative locali (differenze particolarmente evidenti fra paese e paese);
- *fattori socio culturali* riferiti al tono, allo stile ed alla lingua che il prodotto usa per la comunicazione;
- *fattori organizzativi* che includono anche aspetti economici e di commercializzazione.

Portabilità più propriamente *educativa* o socio culturale dunque accanto a *portabilità tecnologica*, ma non soltanto…

Chiunque sia impegnato nella produzione di software educativi dovrebbe porsi dunque anche l'obiettivo della *portabilità transnazionale* [Kearsley G., 1990] e *crossnazionale*

[Collis e al. 1994]; esistono già alcune categorie o collane di prodotti che hanno affrontato queste problematiche[3]: si tratta in genere di prodotti a carattere europeo, realizzati per essere utilizzati indifferentemente nei diversi Stati membri della Comunità Europea. Per contro, si trovano anche prodotti che hanno subito un processo di localizzazione dell'interfaccia, ma non dei contenuti e che mostrano, quindi, evidenti riferimenti ad uno specifico ambito culturale.

La multilingualità è, di fatto, un'idea che sta prendendo corpo e concretizzandosi: stanno nascendo i prodotti europei che non prevedono "traduzioni", ma la cui *portabilità linguistica* è stata realizzata a priori ed introdotta come caratteristica di base del prodotto stesso.

È necessario che chi è impegnato nella produzione di materiale didattico rivolga un occhio attento alla *portabilità tecnologica*, ma anche a quella più strettamente *educativa* e *linguistica*, per garantire l'ampia utilizzabilità e, di fatto, estendere il mercato di quanto intende produrre.

NUOVI CONTESTI DI PRODUZIONE PER IL SOFTWARE EDUCATIVO

Abbiamo visto come, di fatto, il software educativo che un tempo era prevalentemente fondato su pagine testuali interattive, oggi sia definitivamente approdato nella dimensione degli ipermedia e dei multimedia.

Produrre in modo economico software con caratteristiche di affidabilità, durevolezza e agilità tali da competere con il resto dei prodotti presenti sul mercato significa anche cambiare, rispetto al passato, tecniche e metodi di produzione.

Già in un articolo del 1992 Peter Owens intravedeva che l'era degli autori solitari dei software educativi era finita, e affermava che: "L'autore solitario di software educativo, che si è dimostrato la forza più innovativa nella produzione di software educativi negli ultimi dieci anni, è ormai completamente obsoleta. Gli ipermedia sono troppo dispendiosi, troppo complessi e richiedono il coinvolgimento di abilità e talenti troppo diversi, che non possono essere richiesti ad un singolo autore" [Owens, 1992].

Al lavoro di singoli si deve necessariamente sostituire il lavoro di gruppo; ancora Owens individua ben sette professionalità distinte che sono richieste per produrre, oggi, software di qualità:

1) Almeno un *autore - progettista* principale, con esperienza e preparazione specifica nella disciplina che il software intende trattare. Questa stessa persona deve essere esperto di pedagogia ma deve anche essere sufficientemente esperto di computer per poter sfruttare appieno le potenzialità dell'elaboratore. Egli ha il compito di produrre uno storyboard[4] iniziale del progetto e, per una migliore strutturazione dei contenuti, è consigliabile che si avvalga del confronto con più esperti prima di giungere alla stesura dello storyboard.
2) Un *responsabile artistico*, esperto di grafica e, in particolare, di grafica computerizzata il quale possa produrre e/o coordinare la produzione di tutte le icone, i disegni, le animazioni.

3 *Per un esempio di questo genere di prodotti si veda il programma EDUSEX descritto, in questo stesso volume, nella rubrica "Vetrina BSD".*

4 *Storyboard: il termine definisce un documento di base per la realizzazione di un prodotto software che è, in un certo senso, simile ad una scaletta molto dettagliata, ad una sceneggiatura, ad un brogliaccio articolato in cui si incastrano, evidenziandone le reciproche relazioni, tutti i "pezzi" di varia natura che costituiscono il prodotto definitivo.*

5 *Possono tuttavia, concretamente essere utilizzati per la produzione "domestica" di piccole applicazioni, anche didattiche, con obiettivi limitati e specifici. Il docente può produrre piccole unità e prototipi o può servirsene come strumento per avviare gli studenti alla produzione individuale di materiali.*

3) Un *esperto di produzione di video* con compiti di regista del prodotto, il quale si assuma la responsabilità della produzione di immagini video di qualità e che ne garantisca la coerenza con le specifiche prodotte dagli esperti istruzionali del progetto.
4) Un *programmatore* o una équipe di programmatori che produca il codice che controlla il modo in cui i vari media interagiscono e le modalità in cui saranno presentate al fruitore sulla base dello storyboard.
5) Un *tecnico di produzione* in grado di gestire le varie apparecchiature necessarie: scanner, apparati per la produzione di suoni e musiche, ecc.
6) Un *project manager* che coordini l'intero progetto e si assuma le responsabilità contrattuali e di marketing.
7) Un *direttore software* che, come il direttore di un film, si incarichi della supervisione e dell'assemblaggio dei vari pezzi che costituiscono il prodotto finale.

In questo quadro, naturalmente, Owens ammette che la stessa persona possa ricoprire uno o più ruoli; la figura del docente esperto di didattica, e di uso di software nella didattica, non rimane esclusa dal progetto complessivo di produzione di software ma entra a far parte, a pieno diritto, e con ruolo di leader, di una équipe di esperti con competenze ampie e diversificate.

L'evoluzione degli strumenti "di sviluppo" nel senso di una progressiva semplificazione è significativa, ma non sembra destinata, per il momento, ad aprire reali possibilità di sviluppo autonomo di prodotti "per il mercato"[5] per i quali rimane fondamentale l'aspetto di produzione collaborativa, di gruppo: lo stesso discutere, partendo da punti di vista e competenze diverse, sulla natura delle componenti del prodotto e sulle loro relazioni porta a determinare soluzioni alternative fondamentali per la definizione di un buon prodotto.

L'intero lavoro di produzione di software, nel divenire marcatamente multidisciplinare ed interdisciplinare, richiede di essere articolato e strutturato in fasi distinte ma coerenti, richiede in sostanza una ingegnerizzazione del processo stesso.

INGEGNERIZZAZIONE DELLA PRODUZIONE

Per una efficace progettazione ingegnerizzata di applicazioni educative multimediali si possono ipotizzare le seguenti fasi di progetto:

Fase di pianificazione: comprende la definizione degli elementi e gli scopi generali del lavoro, lo studio di fattibilità, una valutazione del livello di difficoltà che si potranno incontrare nella fase di progetto e, in ultima

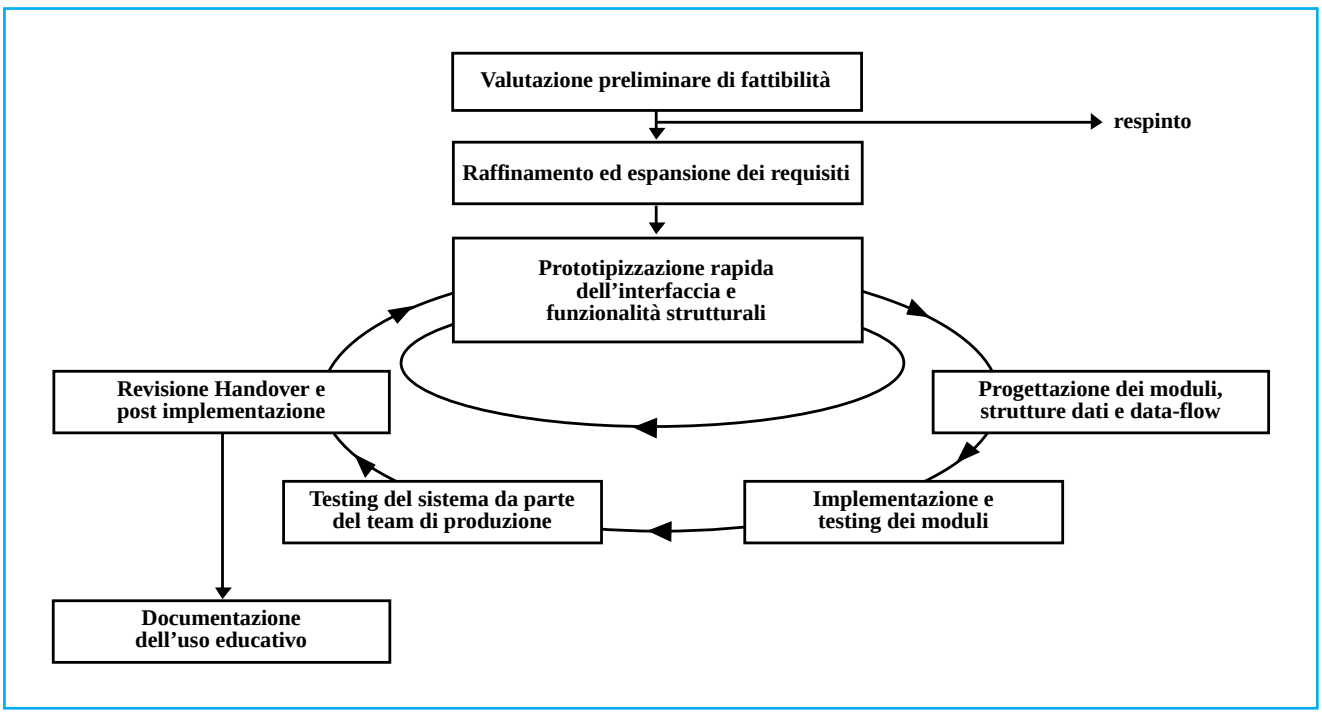


Figura 1.
Il ciclo di vita del software [Thomas, R. 1994].

analisi, la selezione degli strumenti per la successiva implementazione. Importante in questa fase è che si realizzi una diretta interazione tra gli esperti della materia e gli sviluppatori del prodotto.

Fase di progetto: comprende la definizione degli obiettivi puntuali e specifici dell'applicazione didattica e lo sviluppo dell'interfaccia, assicurandone la compatibilità con le norme dettate dall'ambiente scelto. In questa fase devono essere definiti e valutati anche gli aspetti motivazionali.

Fase di produzione: prevede la creazione di tutti gli oggetti necessari: immagini, sequenze video, audio, testi, file eseguibili, ecc., e il codice che lega tutti questi oggetti in forma coordinata per gli scopi di presentazione e gli scopi didattici.

Le fasi fin qui citate sono riassunte in un grafico che rappresenta il ciclo di vita del software [Thomas, R. 1994] il quale dettaglia, in particolare, la fase di produzione vera e propria.

Ciascuna delle box del grafico rappresenta, in realtà, una attività complessa, articolata e diversificata, all'interno della quale possono essere evidenziati alcuni punti chiave:

Valutazione preliminare di fattibilità

- Analisi dell'esistente e, in particolare, verifica del fatto che non esista già altro software sovrapponibile.
- Verifica della validità educativa del progetto in generale e del prodotto in particolare.
- Valutazione che la materia sia adatta ad essere insegnata con un computer.

Raffinamento e espansione dei requisiti

- Esame critico del progetto.
- Analisi del rapporto costi - benefici nell'ambito della portabilità del prodotto.
- Valutazione della durabilità del software.

Prototipizzazione rapida dell'interfaccia e funzionalità strutturali

- Definizione, ad uso dei programmatori, delle specifiche, che è opportuno siano ben comprensibili a tutti i membri dell'équipe, alcuni dei quali non essendo professionisti del computer possono avere difficoltà ad immaginare il prodotto finito.
- Prototipizzazione dell'interfaccia; ogni prototipo deve richiedere pochi giorni di lavoro, preferibilmente solo poche ore, così che tutti i membri del gruppo possano sentirsi liberi di criticare e apportare modifiche, questo modo di procedere ha il vantaggio di mantenere vivo l'interesse e il coinvolgimento nel progetto.
- Produzione di un prototipo definitivo, dettagliato, dell'interfaccia e dei flussi di controllo, con il consenso di tutti i membri del gruppo su ciò che si è ottenuto.

Progettazione dei moduli, strutture dati e data flow

- Definizione di tutti i dettagli realizzativi; il lavoro di questa fase è limitato alla sola équipe di implementazione, che studia, definisce e struttura l'organizzazione e le relazioni reciproche dei vari moduli che costituiranno il prodotto.

Implementazione e testing dei moduli

- Scrittura del codice sorgente, commentandolo rigorosamente e seguendo delle linee guida di stile univoche, per consentirne una agevole manutenibilità futura.
- Testaggio dei singoli moduli; eventualmente può essere presa in considerazione l'introduzione deliberata di errori per verificare la qualità del testaggio stesso.

Testing del sistema da parte dell'équipe di produzione

- Testaggio del prodotto nella sua globalità; deve essere effettuato da tutti i membri del gruppo nelle condizioni normali d'uso.

Revisione Handover e post implementazione

- Programmazione di successive revisioni per risolvere eventuali errori segnalati dall'utenza, (anche se le fasi di testing dovrebbero assicurare un basso tasso di errori).
- Previsione di nuove funzionalità e possibili raffinamenti da implementare in nuove versioni; ciò può garantire più lunga vita al prodotto.

Documentazione dell'uso educativo

- Redazione, oltre che dei manuali d'uso, di note per il docente, le quali esplicitino le modalità di impiego educativo del prodotto.

L'intero processo di produzione del software non si conclude con la produzione, ma rimangono aperte tutte le problematiche connesse con la valutazione e la commercializzazione di quanto prodotto; l'ingegnerizzazione del prodotto richiede, quindi in realtà, almeno altre tre fasi:

Fase di valutazione: consiste nel provare il prodotto realizzato, su una popolazione campione, comparando i risultati ottenuti rispetto ad altri metodi tradizionali. Implica anche ri-progettare ri-produrre e/o ri-valutare, se necessario.

Fase di autorizzazione: consiste nell'assicurarsi, prima del rilascio, che il prodotto non violi diritti d'autore e societari, nell'ottenere tutte le necessarie autorizzazioni, licenze, ecc. per il materiale preesistente usato nella creazione dei moduli del prodotto e consiste anche nell'individuazione delle modalità di registrazione del copyright.
Fase di rilascio: una volta che il prodotto è stato sviluppato e testato può essere rilasciato e immesso sul mercato; naturalmente, ciò implica l'uso di adeguate strategie di commercializzazione.

CONCLUSIONI

Abbiamo cercato di affrontare le problematiche connesse con la produzione di software didattico mettendoci dalla parte del "realizzatore", di chi è interessato ad una produzione specialistica, ma contemporaneamente "di mercato". In realtà, già da qualche tempo, siamo consapevoli che il nostro ruolo di ricercatori del settore richiede oggi una impostazione diversa e una riflessione più approfondita che vanno al di là, superano e forse prescindono, dalla realizzazione autonoma di materiale didattico.

Anni fa abbiamo pensato, implementato e sperimentato package educativi che hanno costituito parte integrante delle nostre ricerche non solo applicative, ma anche teoriche.

Oggi guardiamo a quei prodotti come ad una esperienza utile e fruttuosa, ma conclusa; sentiamo la necessità di orientare la nostra attività ad una riflessione progettuale più globale che possa realmente dare corpo e utilizzare appieno tutti i mezzi che l'evoluzione tecnologica ci mette a disposizione (e che non sempre sono sfruttati in maniera adeguata per produrre un reale miglioramento/incremento delle capacità di apprendimento).

Poco importa se, per tradurre in concreto i risultati delle nostre ricerche, avremo bisogno di altri: la forza della ricerca sta anche nel saper quando voltare pagina.

1 Solo negli Stati Uniti, nel censimento del 1990, sono stati stimati circa quarantasette milioni di disabili. Secondo altre fonti, quelli visivi, negli U.S.A. e in Europa, sono circa ventisette milioni. In Italia mancano, invece, stime globali sul numero dei disabili. Si tratta di numeri che continuano a crescere anche in relazione all'allungamento della vita media della persona.

2 Gli utenti ipo o non vedenti possono accedere a un terminale attraverso i cd "screen reader", dispositivi che consentono l'esplorazione testuale dello schermo attraverso sintetizzatori vocali o dispositivi di uscita in codice Braille/Labile.

3 L'URL del progetto è il seguente: http://www.trace.wisc.edu/world/world.html

4 A chi potrebbe rilevare una presunta funzione estetico-ornamentale, e quindi, inessenziale, delle immagini, lo spirito del Progetto sostiene il diritto alla completa e totale percezione e fruizione delle informazioni presentate, indipendentemente dalla loro natura e dal giudizio di rilevanza del progettista.

5 Analoghe soluzioni possono essere adottate per le immagini che necessitano di viewer esterni, per audioclip e filmati.

6 Questo comando consente di visualizzare, al posto dell'immagine che non viene caricata, una breve descrizione del suo contenuto, leggibile da qualunque screen reader.

7 La cui URL è http://-www.boston.com/wgbh/ncam

Riferimenti Bibliografici

Cohen S., Tsai F., Chechile R. (1995), A model for assessing student interaction with educational software, *Behavior Research Methods, Instruments, & Computers*, Vol. 27, n. 2, pp. 251-256.

Collis B., De Diana I. (1990), The portability of computer related educational resources, *Journal of Research in Computing in Education* vol. 23, n. 2, pp.147-59.

Collis B.A., Ji-Ping Z., Stanchev I. (1994), Investigating Cross-national Educational Software Portability: the Case of Electronic Workbenches, *ETTI*, Vol. 31, n.1, pp. 44-58.

Dahlstrand I. (1984) Software portability and standards, *Computers and their applications* Vol. 27, pp 1-147.

Frye D., Soloway E. (1987), Interface Design: A Neglected Issue in Educational Software, *Proc. of ACM, CHI + GI 1987*, Apr. Toronto-Canada, North Holland publishers, Amsterdam, pp. 93-97.

Fullerton S., Happ A.J. (1993), A User-Oriented Test of Icons in an Educational Software Product, *Proceedings of the fifth International Conference on Human Computer Interaction*, vol.2, pp.44-49.

Kearsley G. (1990), Designing Educational Software for International Use, *Journal of Research on Computing in Education*, Vol. 23, n.2, pp.242-250.

Ott M. (1997), Educational Software for Disabled Children, *ERCIM News* n. 28.

Owens P. (1992), Multimedia Educational Software: a Radical New Era for Software Design and Authorship, *Journal of Computing in Higher Education*, vol.3, n.2, pp.3-20.

Steele J. (1996), Producing CD-ROMs for Schools, EMI Educational Media International Vol. 33 n.3 pp.119-122.

Thomas R., (1994), Durable, low-cost educational software,