

# CodeFest 2021: una grande manifestazione per capire come funziona il digitale

di Stefano Penge



Digitale ovunque. Nella scuola, nel lavoro, nel divertimento. Digitale come servizio, come risorsa a disposizione di tutti, appena girato l'angolo di Google.

Tempo fa [\[1\]](#) scrivevo che ci sono tre miti che circolano sulla rete: 1) che sia un ambiente naturale, 2) che le risorse che offre sono gratuite e 3) che siano infinite. Un supermercato infinito, con scaffali pieni di ogni ben di dio, senza nessuno alla cassa. Tanta formazione alla "didattica digitale" si poggia proprio su questi miti.

C'è ancora chi proprio non riesce ad adeguarsi al flusso mainstream e si domanda: cosa rende possibile tutto questo? I computer, certo, i cavi e i satelliti, senz'altro. Ma cos'altro c'è, sotto la superficie?

In un senso, sotto ci sono interessi, soldi, potere, che governano questo come altri campi, con buona pace di chi vedo solo futuri rosei in cui intelligenze artificiali e persone andranno a braccetto. Non è per amore della condivisione della

conoscenza che possiamo fare ricerche, scambiarsi email, tradurre, condividere agende, parlarci a distanza. Ed è curioso che quasi nessuno – in un mondo così attento al profitto, alla conquista di ogni possibile mercato – metta in questione tutta quest'abbondanza di risorse gratuite; non solo quelle create e condivise da docenti-artigiani di buona volontà, ma anche quelle che richiedono enormi centri di calcolo solo per essere distribuite. Cosa ottengono le grandi imprese in cambio di questi servizi gratuiti? Da dove traggono le risorse economiche per restare in piedi? Per quanto tempo questi servizi resteranno gratuiti?

In un altro senso, sotto tutti questi servizi c'è il codice sorgente: un testo che fa funzionare i computer, raccoglie e trasforma dati, inventa mondi e connette persone. Anche oggi, nell'era delle interfacce ammiccanti, dei podcast, dei videotutorial, quello che c'è sotto è sempre un *testo*. Proprio così: solo lettere, numeri e segni di interpunzione, perché è di questo che sono fatti tutti i programmi che fanno girare il mondo, dagli smartphone ai satelliti.

Oggi, molto più di ieri, siamo letteralmente (!) immersi in un mondo di testi. Solo che non lo sappiamo, o non vogliamo saperlo. La superficie è molto più rassicurante. Secoli di divisioni tra tecnica e umanesimo, tra estetica ed etica ci hanno abituato a disinteressarci di quello che c'è sotto il cofano della macchina. Il design dei dispositivi che ci rendono possibile lo stile di vita occidentale punta a nascondere sotto superfici lisce, dai colori tenui, tutto quello che potrebbe disturbarci perché ci ricorda che non lo capiamo.

Perché questi testi che governano il nostro mondo sono incomprensibili per il 99,6% della popolazione mondiale (oggi ci sono circa trenta milioni di programmatori nel mondo, che su 8 miliardi di abitanti del pianeta circa fa appunto lo 0,4%). Questi trenta milioni sono gli unici esseri umani in grado

di verificare (leggendolo) cosa fa un programma. Se, per ipotesi, venissero tutti sostituiti da software in grado di programmare al loro posto, [\[2\]](#) avremmo un mondo perfetto di cui non saremmo più in grado di riprendere il controllo. Questo secondo senso si incastra perfettamente col primo: se otteniamo tutto quello che ci serve gratis, non c'è più bisogno di capire come funziona.

Ma sapere cosa davvero c'è sotto è importante anche per avere una conoscenza di una parte di attività umana che ci sfugge. Il codice sorgente non è *fatto di 0 e di 1*, come ancora si legge da qualche parte, magari sottintendendo che siccome usano un codice binario i computer possono scegliere solo tra due cose, bianco e nero: insomma sono stupidi. Sarebbe come dire che le lingue umane sono composte solo da suoni. 0 e 1 sono le lettere di un alfabeto, con il quale si compongono parole, frasi, testi e intere biblioteche. Non è vero che computer capiscono solo 0 e 1: non li capiscono affatto. Invece capiscono istruzioni complesse, descrizioni di situazioni, condizioni, specifiche di azioni, purché appartenenti ad uno degli ottomila linguaggi di programmazione oggi esistenti, e rappresentate con l'alfabeto più semplice possibile: quello appunto a due simboli.

Quando oggi gli esseri umani scrivono il codice sorgente di un programma non usano 0 e 1, né quelle strane parolette incomprensibili (MOV, PTR, JMP) che sono solo delle etichette più facili da ricordare per ordinare al processore di spostare dati da un registro all'altro. I linguaggi che si usano oggi sono flessibili, complessi. Permettono di scrivere regole e fatti, di definire funzioni, di costruire classi di oggetti che ereditano conoscenze. Non sono nemmeno tutti ispirati all'Inglese, come ancora qualcuno crede: ci sono linguaggi basati sul francese, sul finnico, sull'arabo, sul giapponese.

I linguaggi di programmazione vengono inventati continuamente, a differenza delle lingue naturali. Spesso un linguaggio viene inventato per fornire ai programmatori uno strumento più

veloce e pratico, o più flessibile e potente; a volte per fornirne uno più elegante e piacevole. Perché i linguaggi sono creati da esseri umani per altri esseri umani, non per i computer. Tant'è che ci sono linguaggi inventati per gioco, per divertimento, per mostrare quanto si è intelligenti e sfidare gli altri, ed hanno nomi come Malebolge, Shakespeare, LOLCAT, Chef o Cow. Questa classe di linguaggi, detti pomposamente "esoterici", sono praticamente inutili, ma svolgono una funzione puramente estetica, alla faccia di chi sostiene che l'informatica è una tecnica senz'anima. L'esistenza di questi linguaggi dice anche molto sul fatto che gli esseri umani riescono a divertirsi anche facendo le cose più serie, e non resistono a prendersi in giro da soli.

Anche i codici sorgenti non sfuggono a questa legge: i testi dei programmi non sempre sono "algoritmi codificati in un linguaggio per risolvere un problema", come vorrebbero i manuali. Ci sono programmi che non risolvono nulla, ma sono opere d'arte, scherzi, poesie. Sì, poesie: perché l'estetica si infila in ogni attività umana, in particolare quando si mette di mezzo un linguaggio.

Certo viene la curiosità, soprattutto per una scuola che ospita iniziative di Coding e che propone l'apprendimento del pensiero computazionale per preparare le generazioni future di programmatori (o almeno così pensa): ma chi sono allora questi programmatori? Sono sacerdoti della Macchina Divina che si aggirano in camice bianco? Sono hacker quindicenni e brufolosi che scatenano guerre nucleari spiattellando i segreti della CIA? Sono nerd bianchi etero che vivono solo per schiacciare tutti gli altri con la forza della loro conoscenza ultra-specialistica? Questi sono i luoghi comuni che costituiscono il nostro immaginario collettivo e che provengono dal cinema e dalla letteratura di fantascienza. Ma i programmatori sono anche altro: sono ragazzini e vecchietti, cinesi e russi, donne (poche, purtroppo) e altri generi, geni e principianti, creativi e precisini. Scrivono in maniera diversa, anche per

ottenere lo stesso risultato, in base alla loro storia, alla loro lingua madre, alle preferenze stilistiche, alla fase della loro vita. Esattamente come ciascuno di noi scrive diversamente un messaggio d'amore e una velenosa lettera all'amministratore di condominio.

Il fatto è che ancora se ne sa poco: è un campo che non è studiato allo stesso modo, poniamo, di quello della letteratura tradizionale. Non esiste una "sociolinguistica" del codice sorgente.

Di tutti questi temi si occupa CodeFest 2021, il festival online che si apre il 27 Settembre e si chiude l'8 ottobre. Un Festival unico al mondo, organizzato insieme dall'Università di Torino e dall'associazione Codexpo.org. Riflessioni, concerti, spettacoli, laboratori in cui si affrontano l'estetica digitale, la conservazione del patrimonio, le sfide sociali, le infinite possibilità della scrittura. Tutto per cercare di sfatare il mito della programmazione come roba da macchine, per cercare di raccontare un'altra avventura dell'umanità alla ricerca della perfezione.

Per saperne di più e per partecipare

[Calendario aggiornato](#)

[Form per registrarsi e partecipare al festival](#)

[1] *Le tre fallacie della rete*, in "Una scuola per la cittadinanza. Volume 2: Gli orizzonti di senso", a cura di Mario Ambel. PM Edizioni, 2020, pagg. 178-184.

[2] Come provo a raccontare in *La fine della formazione*, <https://www.stefanopenge.it/wp/?p=912>

---

# La videoconferenza è nata per comunicare, non per insegnare

di Stefano Penge



La DaD è un altro caso di \*misuse\*, di uso in ambito educativo e scolastico di tecnologie nate per altri scopi e contesti. La videoconferenza è nata per comunicare, non per insegnare. Gli ambienti come MS Teams o GSuite sono nati per la collaborazione dei gruppi, non delle classi di bambini. Sono nati per fare altro ma vengono forzati ad un uso molto lontano da quello per cui erano stati progettati e realizzati. Non ne faccio un problema etico: è che quando si progetta un software si parte dai casi d'uso, dai bisogni degli utenti, e si realizzano funzionalità per soddisfare quelli. Quello stesso software che è perfettamente adatto in un contesto funziona malissimo in un altro. Per esempio: voglio fare un quiz e uso uno strumento che serve a fare i sondaggi. Sembra quasi uguale, ma è pensato in maniera completamente diversa: punta ai risultati aggregati, alle statistiche; probabilmente permette solo tipi di domande molto semplici, non quelle a cui avrei pensato (cloze, corrispondenze, trascinamenti).

Non è la prima volta che succede in ambito digitale: basta ricordarsi la LIM, che è nata per fare presentazioni in azienda ed è stata trasportata – quasi senza trasformazioni – nelle aule di scuola elementare. Prima ancora, le suite di software per ufficio (scrittura/calcolo/database/presentazione) spacciate come software educativo.

Qualche rara volta è successo che si pensasse al mondo dell'educazione come ad un mercato sufficientemente unico e grande per meritare di sviluppare dei prodotti o servizi apposta, da zero. Ma è rischioso e non è andata sempre bene.

Il riuso ha enormi vantaggi per chi produce: riapplicare un prodotto in mercati diversi da quello originale permette di entrare in lizza prima dei concorrenti che perdono tempo a progettare qualcosa di specifico; ma soprattutto permette di recuperare gli investimenti, sia di lavoro che di infrastrutture. Amazon aveva un sacco di server e di banda inutilizzata e si è inventata prima AWS e poi Amazone Chime. Microsoft con Azure e Teams ha fatto lo stesso.

Solo che questo fenomeno ha avuto, ed ha oggi, delle enormi controindicazioni per gli utenti. Piano piano ci siamo convinti che il software educativo, in fondo, non è diverso da quello per la produttività in ufficio. O che una lavagna serve a presentare, non a collaborare. Che una classe è un gruppo di persone che condividono un'agenda e un po' di file.

---

## **L'imbuto di Norimberga:**



# attenti alle metafore!

di Stefano Penge



Stanotte mi è tornata in mente una lezione di Bruno Cermignani, il mio amato professore di Filosofia della Scienza, che ci avvertiva dei rischi che si corrono quando si usa una metafora astratta per comprendere un'esperienza concreta.

L'esempio che faceva era quello della conoscenza come specchio del mondo.

Astratta, perché del fenomeno del rispecchiamento prendeva solo l'idea, ma non la realtà. Uno specchio non riflette sempre l'oggetto che gli sta di fronte, e l'immagine non è affatto realistica. Ci sono le leggi dell'ottica (la riflessione e l'inversione dell'immagine), ma anche altre condizioni reali, come l'umidità (lo specchio appannato) e la temperatura che può distorcere lo specchio, fino a fonderlo. Uno specchio in un forno non riflette un bel niente..

L'insegnamento come riversamento della conoscenza nella testa di uno studente tramite un imbuto è un caso dello stesso fenomeno. Prima di ridere dell'ingenuità di chi parla di riempire gli imbusti, proviamo a prenderla sul serio (la metafora) e a toglierle la dose di astrattezza. Se l'imbuto fosse una buona metafora dell'insegnamento, allora...



Prima di tutto: quando si usa un imbuto? Per esempio: da una damigiana di vino o di olio voglio riempire tante bottiglie più piccole: si chiama travaso. E' utile per distribuire un bene, per farlo viaggiare, o conservarlo meglio. Magari perché da un Sangiovese volgare si vuole ottenere un magnifico Brunello, che come ognuno sa deve stare in bottiglia almeno cinque anni.

E perché ha quella forma? Il flusso che esce dalla damigiana (che ha una bocca più larga) è maggiore di quello che potrebbe assorbire la bottiglia, che ha un collo piccolo. Se non ho una buona vista e una mano più che ferma rischio di rovesciare e disperdere il prezioso liquido. L'imbuto ha una bocca ancora più larga di quella della damigiana, e poi si restringe. Così si può sbagliare, sia in termini orizzontali (se sbaglio la direzione il liquido non va nel buco) che verticali (se esagero con l'inclinazione verso un flusso troppo importante che la bottiglia non riesce ad assorbire). L'imbuto ha senso perché ha una tolleranza all'errore maggiore di quello del collo della bottiglia; e ha senso se il liquido che si vuole travasare è prezioso.

Che tipo di contenuto si può travasare con un imbuto? Beh, non uno qualsiasi. Deve avere delle caratteristiche fisiche precise: i liquidi vanno bene, i gas no. I solidi? Devono essere aggregati in particelle molto piccole (come lo zucchero); ma la farina o la sabbia, soprattutto se umidi, tendono a creare degli intasamenti. Dei tappi, appunto. Quindi bisogna fare attenzione anche alle condizioni meteorologiche. Certo non basta averlo, l'imbuto: bisogna anche usarlo bene. Bisogna assicurarsi che l'imbuto stia ben fermo, che sia pulito (l'olio tende a restare appiccicato sulle pareti, il vino diventa aceto). Alla fine dell'operazione bisogna tappare la bottiglia, altrimenti è tutto inutile.

Che possiamo concludere? Che l'imbuto non è una (buona o cattiva) metafora dell'insegnamento, ma degli metodi e degli strumenti che si usano quando si vuole travasare conoscenza

per distribuirla nel mondo. Metodi che servono a controllare il travaso, a evitare che ci sia dispersione. Strumenti che funzionano con certi tipi di conoscenze, o conoscenze in certa forma (fluida, cioè non strutturata fortemente, o molto parcellizzata). Strumenti che vanno mantenuti, verificati, controllati. Ah, e poi c'è il tappo: a cosa corrisponde un tappo? Per esempio ad un ritmo esagerato di versamento; o a una condizione sociale invasiva, oppure ad una conoscenza che tende a coagulare, a fare "mappazza".

E se uno non vuole travasare conoscenza? Si poteva usare una metafora diversa? Certo. Per esempio, mescolare farina burro e uova per fare una torta. Oppure piantare un seme nel terriccio e innaffiarlo. O covare un uovo. Far cadere un granello di sabbia in una soluzione sovrassatura.

E per ogni metafora si poteva andare a scavare sulle sue condizioni reali d'uso.

---

## Valutazione on line, se ne parla da anni



di Stefano Penge

Ieri sera mi domandavo: dove sono finiti tutti i docenti che hanno partecipato ai convegni sull'e-learning, sulla valutazione online, all'inizio del millennio?

La valutazione autentica, alternativa, tra pari etc. sembravano delle acquisizioni, tant'è che ci si poneva il

problema di come applicarle al nascente campo dell'e-learning. I quiz sembravano strumenti giurassici e si sperimentavano a fini valutativi mappe, portfolio, analisi quantitativi dei commenti (Thinking Types) e dei testi liberi.

Oggi invece siamo da capo: c'è bisogno di distinguere tra valutazione e verifica, tra assessment e evaluation, fra valutazione dell'apprendimento e del processo. C'è bisogno di \*parlare\* della valutazione online come se l'avessimo scoperta ora. Ma come?

Stamattina mi sono dato una risposta: "vent'anni fa, o giù di lì" (cit Guccini) quelli erano concetti con cui giocare, ma nessuno pensava veramente di metterli a sistema, di standardizzarne l'applicazione quotidiana fuori dai convegni e dalle riviste. Erano giocattoli divertenti, finché restavano confinati nei discorsi tra ricercatori, a scuola e fuori.

Per cui era facile accettare anche ipotesi folli come quella che diceva che nell'e-learning i dati raccolti dalle piattaforme, le interazioni tra corsisti, e non i risultati dei test, erano la fonte primaria per valutare il processo di apprendimento. I rari tentativi di applicare queste idee a grandi moli di dati (mi riferisco, per esempio, alle analisi dei forum INDIRE fatte nel 2005 per conto di Scienze della Comunicazione della Sapienza) non sono mai riusciti a diventare prassi consolidate.

Ripesco allora una bibliografia di un libro sulla valutazione online che poi non ho mai scritto e la posto qui, in modo che chi ha voglia possa farsi un giro. Non è completa e qualcuno di quelli che ci hanno lavorato sarà sorpreso di non ritrovarsi (aggiungetevi pure) ma può essere comunque utile. Ho eliminato tutti i testi non italiani.

## **Bibliografia su valutazione online – 2012**

### **I. In generale: verso la valutazione online**

- Varisco B. M., Metodi e pratiche di valutazione, Milano,

Guerini, 2000.

- Varisco, B.M. "Tecnologie didattiche, apprendimento e valutazione", in D. Persico (a cura di), Tecnologie didattiche e scuola. Atti del Convegno TED, Genova, 12-14 febbraio 2001, Ministero della Pubblica Istruzione – CNR, Istituto Tecnologie Didattiche, 2001.
- Colace F., De Santo M., Vento M., Un modello per la valutazione di ambienti per on-line learning, Atti di Didamatica, AICA, Napoli, 2002.
- Galliani L., Costa R., Valutare l'e-learning, Pensa Editore, Lecce, 2003.
- Guglielman E., Vettraino L., Guspini M., Complex Learning, in Apprendimento e nuove tecnologie. Modelli e strumenti, a cura di V. Castello e D. Pepe, Franco Angeli, Milano, 2010
- Trincherò R., Valutare l'apprendimento nell'e-learning, Erickson, Trento, 2006.

## **II. Strumenti di valutazione delle interazioni didattiche**

- Bocconi S., Midoro V. e Sarti L. (1999), Valutazione della qualità nella formazione in rete. Una metrica della qualità nei processi di formazione collaborativi in rete, "TD – Tecnologie didattiche", n. 16, vol. 1-1999, pp. 24-40.
- Talamo A., Zucchermaglio C., Inter@zioni. Gruppi e tecnologie. Roma, Carocci, 2003
- Penge S., "Valutare un gruppo d'apprendimento online", Atti del Convegno E-learning: formazione, modelli e proposte, Abbazia di Fiastra, Aprile 2004 (Armando Armando, Roma, 2006).
- Penge S., Valutare online, in Atti del convegno Didamatica 2004 (a cura di Andronico A., Frignani P., Poletti G.), Omnicom Editore, Ferrara, pp. 651-662, 2004
- Mazzoni E., Bertolasi S., "La Social Networks Analysis (SNA) applicata alle comunità virtuali per l'apprendimento: analisi strutturale delle interazioni all'interno dei Web forum", in Je-LKS – Journal of e-Learning and Knowledge Society, n° 2, p. 243-257, 2005.
- Manca, S., Sarti, L., Pozzi, F., Persico, D. "A mixed model for evaluating CSCL environments", Colloquio TICEMED, Genova,

Facoltà di Scienze della Formazione, Maggio 2006

- Ranieri M., "Osservare e valutare gruppi di apprendimento online: soggetti, strumenti, strategie di analisi e intervento", Colloque TICEMED, Genova, Facoltà di Scienze della Formazione, 2006.

### **III. Strumenti di valutazione delle interazioni verbali**

- Maffei R., "Questioni di stile – L'influenza dello stile di conduzione sui gruppi collaborativi online", Colloque TICEMED, Genova, Facoltà di Scienze della Formazione, Maggio 2006.

- Mancini I., Baroni B., Analisi Conversazionale e Analisi Sequenziale: applicazioni possibili alla CMC asincrona, su Form@re, n. 27, Maggio 2004.

- Mazzone E., La Social Network Analysis: analisi strutturale delle comunità virtuali, in A. Calvani, Rete, comunità e conoscenza, pp. 193-207, Trento, Erickson, 2005

- Molino M., Costruzione del discorso nei web forum collaborativi, in A. Calvani, Rete, comunità e conoscenza, pp. 225-35, Trento, Erickson., 2005.

- Tasso C., Rossi P.G., Morandini A., Virgili C., Personalizzazione basata sull'analisi dei contenuti per strumenti di comunicazione asincrona: il sistema iFFORUM, Atti 43° Congresso AICA, AICA, Udine, 2005

### **IV Altri strumenti di valutazione delle interazioni**

- Comoglio M., Insegnare e apprendere con il portfolio, Fabbri, Milano 2003

- Guastavigna, M. Mappe per capire, capire per mappe, Carocci, Roma, 2004

- Rossi P.G., Progettare e realizzare il portfolio, Carocci, Roma, 2005

- Varani A., Valutazione, portfolio formativo e TIC, Informatica & Scuola -1 3/2004