



Edizione italiana a cura di ALSI e Tecnoteca
<http://upgrade.tecnoteca.it>

Alla ricerca del Sacro Graal dell'Ingegneria del Software di Robert L. Glass

**(Traduzione italiana a cura di Paola M. Hayward dell'articolo
Searching for the Holy Grail of Software Engineering
pubblicato sul Vol. IV, No. 4, Agosto 2003
della rivista online UPGrade, a cura del CEPIS)**

*“Un metodo efficace, come una chiave fissa di misura specifica,
serve per aggiustare e fare un lavoro ottimale nel caso di un problema particolare;
un metodo non efficace, come una chiave inglese,
è concepito per modificarsi di fronte a problemi diversi,
ma non riesce a risolverne alcuno in modo ottimale.”*
Vassey e Glass, 1998

Riassunto: l'autore di questo articolo difende l'eclittismo nei metodi di sviluppo e il contributo che l'Ingegneria del Software apporterebbe in questo ambito, ogni qualvolta che la natura di un progetto, per risultare vincente, richiede metodologie flessibili.

Parole chiave: Soluzioni “ad hoc”, Ingegneria del Software, procedure software.

1. La “diversità del progetto” non costituisce una valida argomentazione

Sono ormai decenni che i teorici dell'informatica sono alla ricerca del Sacro Graal dell'Ingegneria del Software, cioè del vero e unico approccio nella costruzione di sistemi software da poter applicare universalmente a tutti i singoli programmi.

E da altrettanti decenni i programmatori software esaminano questi presunti “sacri graal” e commentano a denti stretti frasi che suonano come “ma il nostro progetto è diverso.”

I teorici dell'informatica, di sicuro, non sono molto soddisfatti da questo tipo di risposta. Si guardano e ammiccano a vicenda, come a voler dire che “ci riprovano, ancora una volta, questi programmatori ignoranti e rifiutano soluzioni ottime per loro.” E, il più delle volte, gli accademici borbottano qualcosa che riguarda l'uso del termine “soluzioni ad hoc”, usato dai programmatori. Quando si sente un accademico parlare di qualcosa “ad hoc”, si capisce che costui ha pronunciato la condanna definitiva. Il termine “ad hoc” nei circoli accademici suona come un insulto, una sorta di *j'accuse* finale di derisione.

Nel corso degli anni, in questo ambito, si è assistito a dei cambiamenti. Al momento, sempre più informatici universitari sono disposti a considerare la possibilità che problemi diversi richiedano approcci risolutivi diversi. C'è stata, per esempio, una risposta positiva da parte di molti lettori dell'articolo “Strong vs. weak approaches to systems development”. Qui si difende la posizione che soluzioni centrate sui problemi sono efficaci, mentre non lo sono se non tengono conto dei

problemi, all'interno della prospettiva storica della disciplina del *problem solving* (la citazione iniziale è tratta dall'articolo in questione). Ciononostante, sempre più spesso si nota quegli stessi accademici disposti a credere alla possibilità che diversi tipi di problemi richiedano approcci risolutivi diversi, ritornare sui propri passi e alla nozione di 'meta-approcci', cioè all'insieme di quegli approcci specifici al problema con un'apparenza indipendente dal problema. Nella disciplina del *method engineering*, per esempio, che è giunta alla conclusione che le metodologie debbano essere personalizzate ai problemi trattati, vi è sempre qualcuno che nella letteratura dell'argomento, propone un "meta" approccio per quelle metodologie personalizzate, come se vi potesse essere una soluzione generica se la si presenta in un modo interessante.

Lo stesso termine "ad hoc" rappresenta un caso interessante in materia. Come si è detto, molti esperti informatici lo usano in un senso ironico, interpretando il termine nel senso di "non strutturato, senza forma" o persino "caotico." Tuttavia il dizionario offre una definizione completamente diversa per questo termine. In tutti i dizionari che ho consultato, e ne ho consultati parecchi perché la definizione che ho trovato è così diversa dall'uso corrente nelle Scienze Informatiche, la locuzione significa "focalizzato, concentrato sul problema in questione". Ma allora è così? Il termine dispregiativo diventa qui un approccio accettabile nel *problem solving*, efficace secondo la citazione iniziale.

Tutte queste riflessioni si agitano sotto la superficie delle mie opinioni nel campo dell'informatica. Posso così spiegare la mia convinzione nella pratica del software, per esempio, davanti alle accuse tremende da parte di alcuni teorici che lo definiscono un campo in crisi. Riesco a spiegare il rovesciamento della mia posizione nel campo del *method engineering*: ho sempre nutrito dei dubbi a proposito delle metodologie "Capital-M", così definite da Tom DeMarco. Sulla base di tali metodologie, i suoi seguaci esigono che i fruitori delle metodologie stesse aderiscano sia alla lettera che allo spirito delle proscrizioni. Ma ora capisco che il *method engineering* promuove la nozione di metodologie su misura (del tipo che DeMarco chiama con la emme minuscola), riesco a credere che delle metodologie hanno iniziato a offrire concetti tangibili e utili che i programmatori potranno mettere in pratica, in modo ottimo e, specialmente, creativo.

Come ho già accennato, questi pensieri si muovono sotto la superficie. Ma di tanto in tanto vengono a galla. Posso fornire un esempio recente, un episodio che ha rinvigorito la mia critica sull'argomento degli approcci risolutivi centrati sul problema, l'elemento scatenante che mi ha spinto a scrivere questo articolo.

Il *Wall Street Journal* è impegnato nel genere di giornalismo che più preferisco. I suoi editoriali talvolta riflettono opinioni di estrema destra, ma le pagine di cronaca sono ricche di meravigliose e introspettive storie d'interesse umano, molte delle quali trattano di argomenti fondamentali per i nostri giorni.

Recentemente, di fronte al fiorente interesse per tutte quelle problematiche sorte dall'*e-commerce*, il *Wall Street Journal* ha pubblicato un inserto speciale sull'argomento. E, tra i numerosi argomenti trattati nell'inserto, vi era un'analisi sul perché così tanti e-business falliscono [1]. Molte erano le ragioni fornite per il fallimento, le stesse che si ritrovano in altre pubblicazioni: scarso *business planning*, attenzione concentrata sull'ideale più che sul profitto, l'aspetto manageriale affidato agli esperti informatici, il mancato riconoscimento del potere dello status quo, sottocapitalizzazione, troppe uscite, attese non realistiche. E via dicendo. (Per un elenco esaustivo, cfr. [3]).

Ma alla fine di un elenco così lungo e dettagliato, veniva esposta un'altra ragione, quella più vicina alle mie convinzioni, che peraltro sono molto parziali: la mancata comprensione del dominio di applicazione dei problemi di base. Ad esempio, siti che vendono giocattoli che non accettano i bassi margini e la necessità di acquistare in anticipo [2]. Siti che vendono prodotti per la sanità che non riescono a cogliere le esigenze particolari della professione medica (e, credetemi, sono innumerevoli). *B2B business* che danno per scontato che i potenziali clienti siano disponibili ad un aggancio elettronico tra compratore e venditore, senza capire che "aggancio" ha significati diversi a seconda dei vari campi degli affari. Una società ha notato che il proprio personale "deve essere capace di mettere in contatto compratori e venditori di centinaia di categorie," e legare il successo al

“proprio curriculum come fa un'azienda tradizionale” (questa società ha costruito strumenti automatizzati, per esempio, che colgono la forza tradizionale nell' “aiutare i compratori a comprare il motore giusto dalle specificazioni corrette”).

Rieccoci ancora una volta. Fornire soluzioni nel campo dell'informatica significa essere a metà strada nel diventare esperti di informatica. Bisogna sapere anche come affrontare i problemi. E bisogna comprendere, sottolineo ancora questo punto, che gli approcci risolutivi devono tenere in considerazione il problema in esame. Anche se qualcuno potrebbe ironizzare su questo sistema “ad hoc” di condurre gli affari.

[1] J. Anders, *B2B: Yesterday's darling*, Wall Street Journal, Ott. 23, 2000.

[2] L. Bannon, *Toys: Rough Play*, Wall Street Journal, Ott. 23, 2000

[3] R.L. Glass, Ed. *ComputingFailure.com*. Prentice-Hall, Englewood Cliffs, NJ, 2001

[4] I. Vessey and R.L. Glass. *Strong vs. weak approaches to systems development*, Commun. ACM 41, 4 (Apr. 1998), 99-102.

Robert L. Glass è l'editore di *Software Practitioner newsletter*, ed editor emeritus di *Journal of Systems and Software* pubblicato da Elsevier. <rlglass@acm.org>