

Le gioie del mestiere

Perché programmare è divertente? Che piaceri può aspettarsi il professionista come ricompensa?

Prima di tutto c'è la pura gioia di costruire delle cose. Come il bimbo s'incanta davanti alla sua torta di fango, l'adulto è felice quando costruisce delle cose, specialmente cose che ha progettato lui stesso. Penso che questo piacere debba essere l'immagine del piacere di fare le cose che ha Dio, un piacere evidente nella diversità di ogni foglia e nella novità di ogni fiocco di neve.

Secondo, c'è il piacere di costruire cose che siano utili ad altre persone. In fondo, vogliamo che altri adoperino il nostro lavoro e lo trovino utile. In questo senso il sistema programmato non è così tanto diverso dal primo portamatite d'argilla fatto dal bimbo "per l'ufficio di papà".

Terzo, c'è la magia del modellare oggetti complessi, quasi come orologi, composti di parti mobili congiunte tra loro e vederli lavorare a cicli impercettibili, mentre svolgono gli effetti di principi presenti fin dall'inizio. Un calcolatore da programmare ha tutto il fascino del flipper o del meccanismo di un jukebox, portato all'ennesima potenza.

Quarta viene la gioia di non smettere mai d'imparare, che scaturisce dalla natura non ripetitiva dell'opera. In un modo o nell'altro il problema è sempre nuovo, e chi lo risolve impara qualcosa, a volte qualcosa di pratico, a volte qualcosa di teorico, e a volte tutti e due.

Infine c'è il piacere di lavorare con un mezzo così duttile. Il programmatore, come il poeta, lavora a poca distanza dalle regioni del pensiero puro. Costruisce dall'aria i suoi castelli in aria, creando con la pura volontà della sua immaginazione. Pochi mezzi creativi sono così flessibili, così facili da limare e rimodellare, così capaci di realizzare immediatamente grandiose strutture concettuali (come vedremo dopo, questa stessa duttilità ha i suoi problemi).

Eppure un programma, a differenza delle parole del poeta, è reale nel senso che si muove e funziona, produce risultati visibili dotati d'esistenza indipendente da quella del programma stesso. Stampa soluzioni, disegna figure, produce suoni, muove braccia. La magia del mito e delle leggende si è avverata nel nostro tempo. Si scrive l'incantesimo giusto sulla tastiera e uno schermo prende vita, mostrando cose che mai prima furono o mai sarebbero potute essere.

Programmare, dunque, è divertente perché soddisfa i desideri creativi insiti nell'intimo di ognuno di noi e compiace sensibilità che tutti gli esseri umani hanno in comune.

I dolori del mestiere

Non tutto è gioia, comunque, e conoscere i dolori che ci aspettano rende più semplice sopportarli quando si manifestano.

Prima di tutto, si deve essere perfetti nell'esecuzione. Il calcolatore assomiglia alla magia delle leggende anche in questo senso. Se un solo carattere, o una pausa dell'incantesimo non è precisamente come deve essere, la magia non funziona. Gli esseri umani non sono abituati ad essere perfetti e poche aree dell'attività umana lo richiedono. Adattarsi alla necessità di perfezione è, credo, la parte più difficile dell'imparare a programmare.

Inoltre, sono altri che decidono gli obiettivi, procurano le risorse e forniscono le informazioni. Difficilmente si controllano le circostanze del proprio lavoro, e neppure i suoi obiettivi. In termini di management, non si ha abbastanza autorità per quelle che sono le proprie responsabilità. Pare, comunque, che in nessun campo i lavori in cui bisogna davvero fare le cose abbiano mai ricevuto

un'autorità formale commisurata alle responsabilità. In pratica, un'autorità di fatto (e non formale) la si acquisisce con la forza stessa dei risultati prodotti.

C'è un caso in cui la dipendenza dagli altri è particolarmente dolorosa per il programmatore di sistemi: quando dipende dai programmi fatti da altre persone. Questi ultimi sono spesso mal progettati e mal implementati, consegnati incompleti (niente codice sorgente o niente test) e scarsamente documentati. A questo punto, il programmatore è costretto a passare ore a studiare e ad aggiustare delle cose che in un mondo perfetto sarebbero complete, disponibili e utilizzabili.

Un altro dolore consiste nel fatto che progettare strutture concettuali importanti è divertente, ma stanare piccoli *bug* gretti è solo una faticaccia. Ad ogni attività creativa si accompagnano ore monotone di lavoro tedioso, pignolo e faticoso, e la programmazione non fa certo eccezione.

L'ultimo dolore, a volte la goccia che fa traboccare il vaso, è vedere che il prodotto sul quale s'è lavorato tanto a lungo diventa obsoleto quando lo si sta per completare (se non prima). I colleghi e i concorrenti sono già all'inseguimento di idee nuove e migliori. Non solo si può intravedere il pensionamento del figlio del proprio ingegno, si può anche scriverne la data sul calendario.

Sembra sempre peggio di quanto sia davvero, questo fatto. Il prodotto nuovo e migliore di solito non è ancora disponibile quando si completa il proprio, se ne parla soltanto. Anche quello avrà bisogno di mesi di sviluppo. La tigre vera non è mai all'altezza di quella di carta, a meno che non si voglia lavorare sul serio. A quel punto i vantaggi della realtà danno del tutto proprie.

Certo, la piattaforma tecnologica sulla quale si costruisce è sempre in evoluzione. Non appena un progetto viene stabilito, diventa obsoleto in termini concettuali. Ma l'implementazione di un prodotto reale richiede un'azione metodica e cadenzata. L'obsolescenza di un'implementazione va misurata in confronto ad altre implementazioni esistenti, non a concetti irrealizzati. La sfida e la missione consistono nel trovare soluzioni reali a problemi reali, nei tempi previsti e con le risorse disponibili.

Questa è dunque la programmazione, sia una pozza di catrame nella quale molti tentativi si sono arenati che un'attività creativa con gioie e dolori tutti suoi.

Di Frederick Brooks, The Mythical Man Month, 1975: Essays on software engineering

traduzione di Federico Caboni