

## Capitolo 1

# Craftsmanship: essere artigiani



Il sogno di volare è quasi certamente antico quanto la stessa umanità. L'antico mito greco che descrive la fuga di Dedalo e Icaro risale al 1550 a.C., circa. Nei millenni che seguirono, un certo numero di anime coraggiose, anche se folli, hanno legato goffi aggeggi al loro corpo e si sono lanciati da dirupi e torri verso il loro destino, inseguendo quel sogno.

Le cose cominciarono a cambiare circa cinquecento anni fa, quando Leonardo DaVinci disegnò schizzi di macchine che, sebbene non fossero veramente in grado di volare, esibivano quanto meno un pensiero ragionato. È stato Leonardo a capire che il volo poteva essere possibile perché la resistenza dell'aria funziona in entrambe le direzioni. La resistenza causata dalla spinta verso il basso sull'aria crea una forza di sollevamento della stessa entità. Questo è il meccanismo con cui volano tutti i moderni aeroplani.

Le idee di Leonardo andarono perdute fino alla metà del Settecento. Poi iniziò un'esplorazione quasi frenetica della possibilità del volo. Il Settecento e l'Ottocento furono un periodo di intense ricerche e sperimentazioni aeronautiche. Furono costruiti, provati, scartati e migliorati molti prototipi senza motore. La scienza dell'aeronautica iniziò a prendere forma, dopo l'identificazione e la comprensione delle forze di portanza, resistenza, spinta e gravità. Alcune anime coraggiose hanno fatto questo tentativo.

E alcuni si sono schiantati e sono morti.

Negli ultimi anni dell'Ottocento, e per il mezzo secolo che seguì, Sir George Cayley, padre dell'aerodinamica moderna, costruì piattaforme sperimentali, prototipi e modelli a grandezza naturale che culminarono con il primo volo con equipaggio di un aliante. E altri ancora si sono schiantati e sono morti.

Poi venne l'era del vapore e con essa la possibilità di concepire il volo con equipaggio e a motore. Furono realizzate decine di prototipi e condotti esperimenti. Scienziati e appassionati si unirono al gruppo di coloro che esplorarono le potenzialità del volo. Nel 1890, Clément Ader pilotò una macchina a vapore bimotores per 50 metri.

E altri ancora si sono schiantati e sono morti.

Il motore a combustione interna è stato il vero punto di svolta. Con ogni probabilità, il primo volo con equipaggio a motore e controllato venne effettuato nel 1901 da Gustave Whitehead. Ma furono i fratelli Wright che, il 17 dicembre 1903, a Kill Devil Hills, Carolina del Nord, condussero il primo volo veramente sostenuto, a motore e controllato di una macchina più pesante dell'aria.

E altri ancora si sono schiantati e sono morti.

Ma il mondo è cambiato da un giorno all'altro. Undici anni dopo, nel 1914, i biplani combattevano nei cieli sopra l'Europa.

E sebbene molti si siano schiantati e siano morti sotto il fuoco nemico, altrettanti si sono schiantati e sono morti solo per imparare a volare. I principi del volo ormai erano noti, ma la *tecnica* del volo era ancora agli albori.

Altri due decenni e i caccia e i bombardieri davvero terribili della Seconda guerra mondiale stavano seminando la distruzione in tutta Europa. Volavano a grandi altitudini, irti di bocche da fuoco e portavano un potere distruttivo devastante.

Durante la guerra furono persi 65.000 aerei americani, ma solo 23.000 di questi caddero in combattimento, e i loro piloti volarono e morirono in battaglia. Più spesso gli aerei volavano e i piloti morivano quando nessuno sparava. Ancora non sapevamo *come* volare. Un altro decennio e arrivarono gli aerei a reazione, il superamento della barriera del suono e l'esplosione delle rotte aeree commerciali e dei viaggi aerei civili. Era l'inizio dell'era del jet, quando solo i più ricchi (non per niente si chiamava il *jet set*) potevano saltare da una città all'altra e da un paese all'altro in poche ore.

E anche gli aerei di linea caddero a pezzi e precipitarono in numero terrificante. C'erano ancora così tante cose che non sapevamo sulla costruzione e sul volo.

Questo ci porta agli anni Cinquanta. Entro la fine del decennio i Boeing 707 trasportavano passeggeri in ogni angolo del globo. Altri due decenni e arrivò il primo jumbo jet a fusoliera larga, il Boeing 747.

L'aeronautica e il trasporto aereo si affermarono e diventarono i mezzi di trasporto più sicuri ed efficienti nella storia del mondo. Ci volle molto tempo e costò molte vite, ma finalmente avevamo imparato a costruire e pilotare in sicurezza gli aeroplani (nonostante il Boeing 737 Max).

Chesley Sullenberger nacque nel 1951 a Denison, Texas. Era un figlio dell'era del jet. Imparò a volare all'età di sedici anni e alla fine pilotava gli F-4 Phantom per l'US Air Force. Nel 1980 divenne pilota della US Airways.

Il 15 gennaio 2009, subito dopo il decollo da New York La Guardia, il suo Airbus A320 che trasportava 155 anime attraversò uno stormo di oche e perse la spinta di entrambi i motori. Il capitano Sullenberger, facendo affidamento su oltre ventimila ore di volo, pilotò il suo velivolo senza più spinta per compiere un "ammarraggio" nel fiume Hudson e, grazie a un'abilità eccezionale, salvò ognuna di quelle 155 anime. Il capitano Sullenberger era un'eccellenza nel suo campo. Il capitano Sullenberger era un "artigiano".

Il sogno di un'elaborazione e gestione dei dati veloce e affidabile è quasi certamente antico quanto l'umanità. Il conteggio su dita, bastoncini e perline risale a migliaia di anni fa. La gente costruiva e usava gli abachi più di quattromila anni fa. Dispositivi meccanici sono stati utilizzati per prevedere il movimento di stelle e dei pianeti circa duemila anni fa. I regoli calcolatori sono stati inventati circa quattrocento anni fa.

All'inizio dell'Ottocento, Charles Babbage iniziò a costruire macchine calcolatrici meccaniche. Si trattava di veri computer digitali, dotati di memoria e di funzionalità di calcolo aritmetico. Ma erano difficili da costruire con la tecnologia di lavorazione dei metalli dell'epoca e, sebbene abbia costruito alcuni prototipi, non furono un successo commerciale.

A metà Ottocento, Babbage tentò di costruire una macchina molto più potente. Sarebbe stata alimentata a vapore e in grado di eseguire veri programmi. La chiamò *Macchina analitica*.

La figlia di Lord Byron, Ada, contessa di Lovelace, tradusse gli appunti di una conferenza tenuta da Babbage e individuò un fatto che evidentemente non era venuto in mente a nessun altro in quel momento: che *i numeri in un computer non devono rappresentare necessariamente numeri, ma qualsiasi oggetto del mondo reale*. Per questa intuizione, viene spesso definita la prima vera programmatrice al mondo.

I problemi di precisione nella lavorazione dei metalli continuarono ad assillare Babbage e, alla fine, il suo progetto venne accantonato. Nessun ulteriore progresso venne compiuto sui computer digitali per tutto il resto dell'Ottocento e fino agli inizi del Novecento. Durante quel periodo, tuttavia, i computer meccanici *analogici* raggiunsero il loro apice. Nel 1936, Alan Turing dimostrò che non esiste un modo generale per dimostrare che una data equazione diofantea (un'equazione di interi) avesse soluzioni. Costruì questa dimostrazione ipotizzando l'esistenza di un semplice, ma infinito, computer digitale e poi dimostrando che c'erano numeri che questo computer non poteva calcolare. Come conseguenza di questa dimostrazione, inventò le macchine a stati finiti, il linguaggio

macchina, il linguaggio simbolico, le macro e le subroutine primitive. Inventò quello che oggi chiameremmo *software*.

Quasi esattamente nello stesso momento, Alonzo Church costruì una dimostrazione completamente diversa dello stesso problema e di conseguenza sviluppò il calcolo lambda, il concetto alla base della programmazione funzionale.

Nel 1941, Konrad Zuse costruì il primo computer digitale programmabile elettromeccanico, lo Z3. Era costituito da oltre 2.000 relais e funzionava a una frequenza di clock da 5 a 10 Hz. La macchina usava un'aritmetica binaria organizzata in word da 22 bit.

Durante la Seconda guerra mondiale, Turing fu reclutato per aiutare i "boffin" a Bletchley Park a decifrare i codici Enigma tedeschi. La macchina Enigma era semplice computer digitale che randomizzava i caratteri dei messaggi testuali tipicamente trasmessi utilizzando telegrafi radio. Turing aiutò a costruire un motore di ricerca digitale elettromeccanico in grado di trovare le chiavi di quei codici.

Dopo la guerra, Turing fu determinante nella costruzione e programmazione di uno dei primi computer elettronici a valvole al mondo: l'*Automatic Computing Engine*, ACE. Il prototipo originale utilizzava 1.000 valvole e manipolava numeri binari a una velocità di un milione di bit al secondo.

Nel 1947, dopo aver scritto alcuni programmi per questa macchina e aver esplorato le sue possibilità, Turing tenne una conferenza in cui fece queste dichiarazioni, preveggenti:

Avremo bisogno di un gran numero di matematici in grado di elaborare i problemi in una forma adatta al calcolo.

Una delle nostre difficoltà sarà il mantenimento di una disciplina adeguata, in modo da non perdere il senso di ciò che stiamo facendo.

- A.M. Turing's *ACE Report of 1946 and Other Papers* - Vol. 10, *In the Charles Babbage Institute Reprint Series for the History of Computing*, editor B.E. Carpenter, B.W. Doran, The MIT Press, 1986.

E così il mondo cambiò da un giorno all'altro.

Nel giro di pochi anni, venne sviluppata la memoria centrale. La possibilità di avere centinaia di migliaia, se non milioni, di bit di memoria accessibili in pochi microsecondi divenne realtà. Allo stesso tempo, la produzione di massa di valvole rese i computer sempre più economici e affidabili. La produzione di massa stava lentamente diventando una realtà. Entro il 1960, IBM aveva venduto 140 computer modello 70x. Si trattava di enormi macchine a valvole del valore di milioni di dollari.

Turing aveva programmato la sua macchina direttamente in codice binario, ma tutti capivano che la cosa non era pratica. Nel 1949, Grace Hopper coniò la parola *compilatore* e nel 1952 creò il primo esempio di compilatore funzionante: A-0. Alla fine del 1953, John Bachus presentò la prima specifica del FORTRAN. Nel 1958 seguirono l'ALGOL e il LISP. Nel 1947, John Bardeen, Walter Brattain e William Shockley crearono il primo transistor funzionante. Cominciarono a occuparsi di computer nel 1953. La sostituzione delle valvole con i transistor cambiò completamente le cose. I computer divennero più piccoli, più veloci, più economici e molto più affidabili.

Entro il 1965, IBM aveva prodotto 10.000 computer modello 1401, noleggiati per 2.500 dollari al mese, una spesa alla portata delle imprese di medie dimensioni. Ma quelle aziende avevano bisogno di programmatori, e così la domanda di programmatori iniziò ad accelerare. E chi programmava tutte queste macchine? Non esistevano corsi universitari. Nessuno andava a scuola per imparare a programmare, nel 1965. Questi programmatori proveni-

vano dal mondo del lavoro. Erano persone mature che avevano svolto altre attività per un certo tempo. Avevano intorno ai 30, 40 o 50 anni.

Nel 1966, IBM produceva 1.000 computer modello 360 al mese, e le aziende ne chiedevano sempre di più. Queste macchine avevano una memoria anche superiore ai 64 kB e potevano eseguire centinaia di migliaia di istruzioni al secondo.

Nello stesso anno, lavorando su un Univac 1107 presso il Norwegian Computer Center, Ole-Johan Dahl e Kristen Nygard inventarono Simula 67, un'estensione dell'ALGOL. Fu il primo linguaggio a oggetti.

*La conferenza di Alan Turing si era svolta solo vent'anni prima!*

Due anni dopo, nel marzo 1968, Edsger Dijkstra scrisse la sua famosa lettera su "Communications of the ACM" (CACM). L'editor diede a quella lettera il titolo *Go To Statement Considered Harmful* (Edsger W. Dijkstra, *Go To Statement Considered Harmful*, in "Communications of the ACM 11", n. 3, 1968). Nacque la programmazione strutturata. Nel 1972, ai Bell Labs, New Jersey, Ken Thompson e Dennis Ritchie erano impegnati in vari progetti. Lavoravano su un PDP 7 impiegato da un altro team e inventarono lo UNIX e il C.

Da questo momento il ritmo accelerò a velocità quasi vertiginose. Vi darò alcune date chiave. Per ognuna di esse, domandatevi quanti computer c'erano nel mondo? E quanti programmatori? E da dove venivano quei programmatori?

- 1970: Digital Equipment Corporation ha prodotto 50.000 computer PDP-8 dal 1965.
- 1970: Winston Royce scrive l'articolo sul *waterfall* dal titolo *Managing the Development of Large Software Systems*.
- 1971: Intel rilascia il microcomputer single-chip 4004.
- 1974: Intel rilascia il microcomputer single-chip 8080.
- 1977: Apple rilascia l'Apple II.
- 1979: Motorola rilascia il 68000, un microcomputer single-chip a 16 bit.
- 1980: Bjarne Stroustrup inventa il *C con Classi* (un preprocessore che rende il linguaggio C simile al Simula).
- 1980: Alan Kay inventa il linguaggio Smalltalk.
- 1981: IBM rilascia il PC IBM.
- 1983: Apple rilascia il Macintosh 128K.
- 1983: Stroustrup rinomina il *C con Classi* in C++.
- 1985: Il Dipartimento della Difesa degli Stati Uniti adotta il *waterfall* come processo ufficiale di sviluppo software (DOD-STD-2167A).
- 1986: Stroustrup pubblica *The C++ Programming Language* (Addison-Wesley).
- 1991: Grady Booch pubblica *Object-Oriented Design with Applications* (Benjamin/Cummings).
- 1991: James Gosling inventa Java (all'epoca chiamato *Oak*).
- 1991: Guido Van Rossum rilascia il linguaggio Python.
- 1995: Erich Gamma, Richard Helm, John Vlissides e Ralph Johnson pubblicano *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley).
- 1995: Yukihiro Matsumoto rilascia il linguaggio Ruby.
- 1995: Brendan Eich crea JavaScript.

- 1996: Sun Microsystems rilascia Java.
- 1999: Microsoft inventato C#/.NET (allora chiamato *Cool*).
- 2000: Y2K! Il Millennium Bug.
- 2001: viene scritto il *Manifesto Agile*.

Tra il 1970 e il 2000, le frequenze di clock dei computer sono aumentate di tre ordini di grandezza. La densità è aumentata di quattro ordini di grandezza. Lo spazio su disco è aumentato di sei o sette ordini di grandezza. La capacità della RAM è aumentata di sei o sette ordini di grandezza. I costi sono scesi da tot dollari per bit a tot dollari per gigabit. L'evoluzione dell'hardware è difficile da visualizzare, ma anche solo raggruppando tutti gli ordini di grandezza che ho appena citato ci porta a un incremento delle capacità di circa trenta ordini di grandezza.

E tutto questo in poco più di cinquant'anni dalla conferenza di Alan Turing.

Quanti programmatori ci sono adesso? Quante righe di codice sono state scritte? Quanto è buono tutto quel codice?

Confrontate questa linea temporale con la linea temporale dell'aeronautica. Notate la somiglianza? Vedete il graduale espandersi della teoria, la frenesia e il fallimento degli appassionati, il graduale incremento delle competenze? I decenni interi in cui non sapevamo neppure che cosa stavamo facendo?

E ora che la nostra società dipende, per la sua stessa esistenza, dalle nostre capacità, abbiamo quei Sullenberger di cui la nostra società ha disperatamente bisogno? Abbiamo preparato programmatori che comprendano il loro mestiere con la stessa profondità con cui i piloti di linea comprendono il loro? Abbiamo quegli artigiani di cui sicuramente avremo bisogno?

“Artigianalità”, *craftsmanship*, significa saper fare bene qualcosa ed è il risultato di buoni insegnamenti e tanta esperienza. Fino a poco tempo fa, l'industria del software aveva troppo poco di entrambi. I programmatori tendevano a non rimanere programmatori a lungo, perché consideravano la programmazione solo come un trampolino di lancio verso qualcosa di più. Ciò significava che c'erano pochi programmatori che avevano acquisito abbastanza esperienza per insegnare il loro mestiere ad altri. A peggiorare le cose, il numero di nuovi programmatori che entrano nel mondo del lavoro raddoppia ogni cinque anni circa, cosa che contribuisce a mantenere troppo basso il rapporto tra programmatori esperti e inesperti.

Il risultato è che la maggior parte dei programmatori non impara mai le discipline, gli standard e l'etica che potrebbero definire il loro mestiere. Nel corso della loro carriera relativamente breve di programmatori, rimangono principianti senza apprendistato. E questo, ovviamente, significa che gran parte del codice da loro prodotto è scadente, mal strutturata, non sicura, ricca di bug e, in buona sostanza, un disastro.

In questo libro, descrivo gli standard, le discipline e l'etica che credo ogni programmatore dovrebbe conoscere e adottare, per acquisire gradualmente le conoscenze e le abilità che il mestiere richiede.