# Il Grande Libro dei Bug

Aaron Brancotti
Con la collaborazione straordinaria di Babele Dunnit

Supervisione Grafica di Luisella Brustolon

## Quarta di copertina:

Questo libro non contiene alcun CD. Se proprio non potete farne a meno concentratevi su questo, che tanto è la stessa cosa:

[disegno del CD]

Un CD vero non vi servirebbe comunque:
Il vostro computer non sta funzionando.

Se vi interessa avere almeno una vaga idea del perché, leggete questo libro.
Per il resto, Buona Fortuna.

## Note Sugli Autori

**[**Da fare. Con foto magari ritoccata o disegno, quello di Babele Dunnit è ovviamente un icosaedro]

# Indice

# Illustrazioni, Giochi e Fai-Da-Te

## *Riassunto delle Puntate Precedenti...*

*"Se Non Ce La Fai, Fingi."*
*Vecchio Detto Aerospaziale*[1]*.*

...il fatto è che l'Informatica ha fatto molta, troppa fatica a stare al passo con se stessa, mentre la Seconda Legge della Termodinamica[2] compiva il suo dovere in questi ultimi venti anni o giù di lì[3]. Il risultato lo possiamo vedere sulle nostre scrivanie: dispositivi con più potenza di calcolo di quanta ne sia servita alla NASA per spedire l'Uomo sulla Luna[4] che, per mille motivi che vedremo, arrancano faticosamente nel comportarsi come delle macchine per scrivere mentre ci chiedono ogni pochi secondi con faccia da graffetta se vogliamo installare lo stramaledetto Correttore Automatico (che sbaglia).

Sono centinaia i fattori che contribuiscono a provocare questa drammatica, paradossale situazione. Questo libro ci insegnerà, quantomeno, a riderne.

Nella prossima pagina – poiché qualcuno tempo fa ha detto che un'immagine vale più di mille parole – il lettore troverà un riassunto grafico di tutto quello che il concetto dei "bachi", ovvero i "bugs" in inglese[5], porta alla mente a chi ai computer ha dedicato vari anni della propria vita. Si tratta insomma di una sorta di libera associazione di idee, una cascata di concetti, un miscuglio di parole chiave che, forse, ci aiuteranno a dare una struttura a questo libro. O forse no, poiché, come vedremo, quando si tratta di far funzionare un computer, vale di più l'ispirazione e l'intuizione che non il puro ragionamento scientifico. Il punto è che il vostro computer **adesso** non sta funzionando e ne ha i suoi buoni motivi. Ed ora girate pagina.

---

[1] Come riportato a pag 31, "Non Siamo Mai Andati Sulla Luna", Bill Kaysing, Cult Media Net Edizioni, 1997. Si veda anche la nota XXX sul Capitano Kirk.

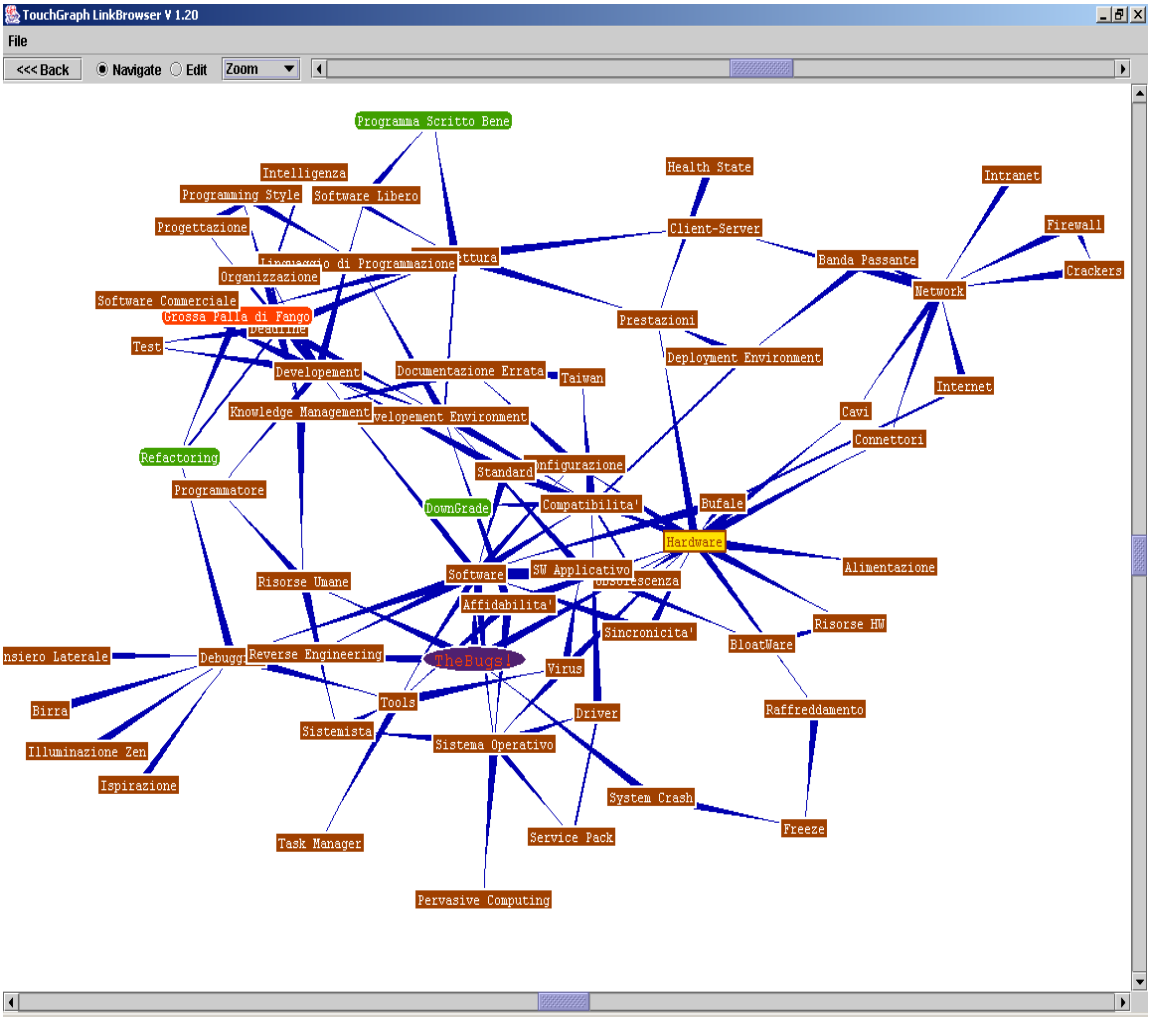
[2] La Seconda Legge della Termodinamica è una delle leggi più importanti della Fisica moderna. Famosa per aver definitivamente stroncato la millenaria ricerca del Moto Perpetuo da parte di sublimi ingegni tra i quali anche Leonardo da Vinci, questa legge afferma, in una delle sue versioni più semplici, che ogni volta che qualcuno fa qualcosa – *qualsiasi* cosa – una parte dell'energia utilizzata per compiere quel lavoro si trasforma, inesorabilmente e irreversibilmente, in calore. Questa, su scala cosmica, è la condanna a morte dell'Universo intero; nel suo piccolo invece questa Legge fornisce un fondamento fisico e filosofico alla ormai universalmente nota Legge di Murphy: "Se qualcosa può andare male, lo farà". Quindi il fatto che il vostro computer non funzioni è ampiamente giustificato e addirittura dimostrabile in maniera scientifica.

[3] In realtà la Seconda Legge della Termodinamica è da un bel pò di tempo che fa il suo dovere. Qui si parla strettamente dell'ambito tecnologico/informatico e di quello che in questo settore è successo dagli anni '80 in poi.

[4] Ammesso che la faccenda della Luna non sia una bufala, come afferma il Bill Kaysing di cui sopra. Ma questo è il primo di molti altri discorsi che il Libro dei Bug, per sua natura, sfiorerà.

[5] Qualcuno ha idea del perché uno scarafaggio si debba trasformare in un bruco quando si cambia lingua? Sono, è vero, entrambi animali che ai più provocano ribrezzo, ma il termine "Bug" ha un suo significato storico (che scopriremo più avanti) difficilmente riscontrabile nella italianizzazione anellide. Però ci viene in soccorso il termine "bacherozzo", che nel centro-sud è lo scarafaggio, ed ecco che il cerchio si chiude. Cionondimeno "Il Grande Libro Dei Bacherozzi" non suona bene.

[Pagina con pieghevole con grafo MOSTRUOSO. Tanto per avere una idea, ecco qui. Questo e' da vedere bene, per quanto riguarda la semantica dei link. Pensarci sopra. Ad esempio, link orientato = "implica", link non orientato = "richiama il concetto di"; si possono usare anche i colori… fare legenda!]



Quando vi siete ripresi, potete continuare a leggere.[6]

*[Disegno: disegnino che non c'entra un nulla, bucolico, rassicurante. Tipo fiorellini e orsetti che giocano in un prato.]*

Bene. Vi siete ripresi, a quanto pare. Allora è opportuno mettere bene in chiaro una cosa: il grafico della pagina precedente, che chiameremo ***ARGH***[7] è, nonostante la sua apparentemente inaccettabile complessità, una *semplificazione molto spinta* di quanto succede veramente Là Fuori. Il mondo dell'Informatica, dello sviluppo del Software, dell'Hardware, di Internet, di questa Tecnologia[8] che amiamo ed odiamo e della quale ormai non possiamo più fare a meno[9] in verità ormai sfugge a qualsiasi tipo di analisi di complessità. Non è pensabile che un essere umano possa conoscerne approfonditamente e razionalmente più di una piccola nicchia. Ecco perché in questo libro si parlerà più volte di ispirazione artistica, di intuizione, di Illuminazione Zen come di quelle qualità che "fanno la differenza" tra un esperto informatico, magari anche bravo, e un Imperatore del Sistema Binario.

D'altra parte, questo libro non ha certo la velleità di insegnare in poche pagine quello che può essere imparato, compreso ed intimamente assimilato solo dopo anni di combattimento sul campo. Sarebbe già un successo, nell'opinione dell'autore, insegnare al lettore a (ri)conoscere, almeno a grandi linee, il Terribile Nemico e, magari, ad evitare le situazioni più pericolose. Ancora meglio sarebbe imparare a sfruttare il Nemico stesso, senza combatterlo ed anzi ottenendo il massimo rendimento dalla sua stessa forza, così come i Grandi Maestri insegnano. La strada verso l'Illuminazione è lunga, ma cosparsa di grandi soddisfazioni.

Un altro scopo di questo libro è far capire al lettore che Il Bug, questa *Invariante Universale dell' Informatica*[10], raramente è ascrivibile da subito ad una singola categoria – hardware piuttosto che software, di livello applicativo piuttosto che sistemistico – e anzi molto spesso si cela nelle interazioni di sistemi tra loro apparentemente indipendenti. Ecco perché, a parte casi eclatanti, non individueremo univocamente la natura del Nemico: il buon combattente tiene occhi aperti e guardia alta.

---

[6] Ci sono un sacco di cose che potete fare a questo punto. Uscire a mangiare un gelato, partire per una gita al lago o concentrarvi sul disegno rassicurante che vedete qui sopra. Non è necessario continuare a leggere subito; è molto più importante metabolizzare il fatto che il mondo dei computer è un delirio e che non potrebbe essere altrimenti.

[7] Absurdo Resoconto Graphicamente Hobsceno

[8] "E quanta forza c'è in quest'ultima parola: le macchine non sono aridi strumenti, ma idee incarnate". G.Giorello, Corriere della Sera, 20/08/2005.

[9] Così sembrerebbe, almeno. Ma un sacco di gente sta barattando il proprio laptop con semi di granturco e piante di pomodoro. Un mio amico ha addirittura preso la Laurea in Scienze dell'Informazione solo per poter aprire una pizzeria che si chiama "O' Professore" – oppure "O' Processore", non ricordo bene – ed ha appeso il prezioso documento, incorniciato in stile Rococò, sopra al forno...

[10] Ovvero, *Tutto L'Universo È Un Grosso Baco*. Un grazie al Reverendo per la definizione.

Scordatevi quindi di far funzionare bene il vostro computer. Di quello non se ne parla. Piuttosto, andiamo ad incominciare.

## *Il Grande Libro Dei Bug*

*[Disegno: Il Nemico o qualcosa del genere]*

## Addio, Cartesio, Addio (Ci Siamo Voluti Bene)[11]

*Godel non poteva dimostrare di non esserci,*
*ma Cartesio sapeva di essere.*

Sulle prime potrebbe sembrare una buona idea affrontare il Nemico cercando di scinderlo nei suoi *wares*, neologismi che ormai tutti abbiamo imparato ad odiare con tutto il cuore. È questo un errore da evitare assolutamente; come Stanlio e Ollio, l'Hardware ed il Software sono assolutamente indissolubili. Entrambi contribuiscono alla realizzazione di fenomeni tali da farci sospettare dell'esistenza di vere e proprie forme di vita all'interno del nostro amato e per definizione obsoleto computer. Come afferma F. Von Taurus, filosofo di origini toscane e quindi in qualche modo "maledetto", durante il suo rapporto epistolare con Babele Dunnit:

- Per risolvere i problemi tecnici bisogna avere una buona preparazione spirituale;
- Per risolvere i problemi economici bisogna avere una buona preparazione agricolo-naturalistica;
- Per risolvere i problemi sentimentali si dovrebbe saper suonare uno strumento musicale ma io non son capace;
- Per risolvere i problemi spirituali dovresti riuscire ad usare un sistema operativo che apparentemente ha dei bug ma che poi funziona… Non so spiegare come.

Insomma, mai come in questo caso un approccio olistico e persino Zen[12] alla conoscenza del Nemico – contrapposto al classico metodo occidentale, riduzionistico, dualistico e cartesiano, è assolutamente preferibile. E poiché più di ogni altra cosa è importante del Nemico pronunciare il Nome, iniziamo con un po' di Storia.

## C'Era Una Volta Un Piccolo Animale...

Primaditutto è giusto ricordare che per i primi programmatori il termine *Bug* si riferiva a qualcosa di intrinsecamente hardware e *veramente* ospitato da una macchina elettronica: a quegli insetti e altri piccoli animali che, attirati dal caldo delle valvole dei primi, giganteschi computer, provocavano cortocircuiti e malfunzionamenti vari passeggiando

[11] Il titolo del paragrafo si rifà ad una semisconosciuta canzone di un semisconosciuto gruppo: "Addio Cervello Addio", degli Acidi Tonanti.

[12]Ancora Von Taurus: "Dal punto di vista Zen il Bug è *fondamentale,* viene introdotto volontariamente e usato come strumento di lavoro... (il Bug) È quell'aspetto (dello Zen) che noi percepiamo come assurdo ed insensato, quello che ci fa sorridere con sguardo ebete di chi non capisce fino in fondo ma sa che gli sta sfuggendo qualcosa – forse tutto – e che ciononostante è convinto che "il cinese" si sia perso qualche tazza di riso per strada...".

tra cavi e connettori. Allora, se il 9 Settembre 1945[13] al posto degli scarafoni fossero entrate delle lucertole in quel termoionico Harvard Mark II, oggi questo libro si intitolerebbe "Il Grande Libro dei Lizard"? In verità no: lo stesso concetto di "mela bacata" ricorda qualcosa che non funziona a dovere, mentre già Thomas Edison, nel 1870, parlava di malfunzionamenti di circuiti elettrici chiamandoli "bugs", mentre il concetto di "bug" in quanto "oggetto mostruoso" si può fare risalire ai tempi di Shakespeare e del mitologico mostro di Albione chiamato "bugbear"[14].

Ma ancora più inquietante – ecco che sempre più si palesa la natura soprannaturale del Nemico – è quanto scrive Henley, traduttore del Vathek di Beckford[15]: "nell'inglese antico l'espressione del Salmo XIX 'Il Terrore della Notte' è tradotta con 'il Lamento dei Fantasmi ("bugge") della Notte'; infatti in alcune parti del Nord America, ogni lamentoso uccello notturno è genericamente chiamato 'bug' (cimice, insetto o fantasma); il termine "babau", ("spauracchio"), deriva proprio da quel termine. Beelzebub, ovvero Il Signore Delle Mosche, era il nome che in Oriente era dato al Diavolo; e il suono notturno (prodotto dagli insetti) chiamato dagli Arabi *Azif* si credeva – secondo alcune fonti – essere l'ululato dei demoni…"

## Saranno Bacosi

Ed ora, una breve galleria di bachi celebri, giusto per far capire al lettore fino a che punto la perfidia del Nemico possa spingersi e convincerlo ad intraprendere questa Guerra Santa:

**Criminali!**
Un Bug da ricordare è quello del sistema del tribunale di Parigi che nel 1989 comminò a circa 41.000 automobilisti multati altrettante denunce per traffico di droga, estorsione, sfruttamento della prostituzione ed altro. In casi come questo il difficile non è tanto mettere a posto il baco quanto riparare ai disastri politici e sociali che un simile errore, come è facile immaginare, sicuramente avrà causato.

**Mariner Killer**
Sicuramente da citare anche il Bug che il 28 Luglio 1962 ha interrotto il viaggio verso Venere della sonda spaziale Mariner I, facendola precipitare nell'Oceano Atlantico dopo soli quattro minuti dal lancio e bruciando così circa 10 milioni di dollari dell'epoca. Considerando che si trattò semplicemente del mancato utilizzo di un operatore di negazione, ossia di un singolo 1 al posto di uno 0...

[13] Data del rinvenimento del "Primo Bug" che, dopo essere stato opportunamente spiaccicato, venne incollato alle pagine di una agenda come testimoniano numerose fotografie reperibili su Internet.
[14] Beh, e poi c'è la Feroce Bestia Bugblatta di Traal.. the "Ravenous Bugblatt Beast of Traal" è, secondo Douglas Adams, l'animale più feroce dell'Universo ma anche il più stupido, tanto da credere di non poterti vedere se tu non puoi vedere lei. Questo è uno dei motivi per cui *bisogna* portarsi sempre dietro un asciugamano: quando incontrerete la Bestia Bugblatta (e prima o poi *la* incontrerete) vi basterà mettervelo davanti agli occhi e lei, credendo di non potervi più vedere, se ne andrà per la sua strada.
[15] Come riporta Sergio Basile ne "Il Necronomicon – Storia di un Libro che non c'è", Primo Volume, Tascabili Immaginario Fanucci, 2004, pagg 155-156, a proposito del nome arabo del Necronomicon, ovvero *Al Azif*. Se non sapete cosa sia il Necronomicon dovreste documentarvi.

**Mars Climate Kaboom**

Parlando di sonde spaziali, è indimenticabile la figuraccia della NASA con il Mars Climate Orbiter, al quale da Terra comunicavano usando il sistema metrico decimale e che rispondeva in misure anglosassoni. Quando fu il momento di atterrare sul Pianeta Rosso, a Novembre del 1999, l'errore di manovra accumulato dopo centinaia di correzioni di rotta impartite usando Metri e Chilogrammi al posto di Piedi e Once rese impossibile il controllo. Non si sa se l'MCO si sia schiantato al suolo tra gli strepiti degli autoctoni ivi convenuti o se si sia perso nello spazio. Quello che si sa è che non ha più contattato la Terra – nemmeno un segnale, una telefonata, una cartolina, un SMS, eddai... Quello che è sicuro è che i Marziani in quella occasione si sono *veramente* innervositi e non vedono l'ora di vaporizzare il nostro lussureggiante pianeta[16].

**Il Millennium Bug**

Il Bug forse più famoso di tutti  i tempi è stato chiamato Y2K, Il Baco Del Millennio. È stato un incubo per il mondo intero, a causa di una vera e propria campagna terroristica svolta all'unisono da tanto incompetenti quanto planetari mezzi di informazione in grado di citarsi l'un l'altro fino alla perdita di ogni senso[17]. A quanto pare, per l'ennesima volta, costoro la pelle dell'orso l'hanno praticamente regalata. Nulla di tutto quel che doveva succedere è accaduto; certamente, c'è stato grande lavoro e digrignar di denti e schiere di programmatori hanno passato notti insonni, ma non siamo tornati all'Età della Pietra. Invece, in una famosa società della quale non faremo il nome – ma il cui Ufficio Reclami occupa le terre emerse di tre pianeti[18] – il problema del Millennium Bug sarebbe stato elegantemente risolto portando indietro di trent'anni l'orologio di sistema di tutti i computer.

Questo introduce una interessante considerazione filosofica: molto spesso i bug non vengono davvero risolti; quando possibile, si preferisce ricorrere alla Mossa del PA[19] sapientemente miscelata al Principio di Impermanenza del Buddha[20]. Quello del Millennium Bug, per il sistemista di quella società, è ormai un Problema Altrui, che si ripresenterà solo quando egli sarà ormai in pensione. Addirittura, grazie alla naturale obsolescenza tecnologica, è possibile che tutto quel software vada a morire in qualche Verde Prateria risolvendo il problema alla radice[21].

---

[16] Cosa che peraltro stiamo facendo senza il loro aiuto.

[17] Variante del gioco del "telefono senza fili" al quale evidentemente molti giornalisti anche da adulti rimangono affezionati più di quanto siano disposti ad ammettere.

[18] Questa storia dell'Ufficio Reclami la rivedremo ed è ovviamente inventata, ma l'idea di portare indietro la data su tutti i computer invece è davvero stata proposta da un Grande Esperto ad una Grande Società… non lamentiamoci, per carità.

[19] Problema Altrui. Si manifesta in numerosissime forme, la più sibillina delle quali è la frase "Questo Me Lo Devi Dire Tu", in grado di annichilire chiunque.

[20] Il Principio di Impermanenza del Buddha dice "Tra cento anni, su tutta la Terra, nessuno di chi è vivo adesso lo sarà ancora", mentre il Nexus di Blade Runner dice: "Io ne ho viste cose che voi umani non potreste immaginarvi". Chissà cosa sarebbe venuto fuori se si fossero incontrati.

[21] Ma vista la già citata inerzia tecnologica di molti ambienti non saremmo poi così tanto fiduciosi. Tuttora il linguaggio di programmazione più comune in ambiente bancario è il COBOL, che ha più di 40 anni.Si

Non dimentichiamoci infine che di Mille E Non Più Mille ne abbiamo visti anche di più complessi. Frugando su Internet siamo riusciti a scovare un documento molto interessante… ecco che cosa diceva al proposito Plutanio al suo amico Cisio duemila anni fa[22]:

Caro Cisio,

Stai ancora lavorando a quel problema dell'Y0K? Questo cambio da A.C. a D.C., che oltretutto fa confusione con un gruppo di Heavy Metal, ci sta provocando non pochi mal di testa. Non ho ancora sentito nessuno con una buona idea su come risolvere la questione: abbiamo felicemente contato gli anni all'indietro fino ad ora e adesso dobbiamo rivoluzionare tutto il sistema. Qualcuno avrebbe dovuto pensarci prima, non mi sembra giusto scaricarci addosso questa storia all'ultimo minuto.

Ho parlato con Cesare due sere fa. È furente con Giuliano, che sembra si sia "dimenticato" della faccenda quando ha rimesso a posto il Calendario. E anche Bruto era un bel po' nervoso.

Cesare ha anche chiamato gli astrologi, ma gli hanno confermato che non c'è nulla da fare e che continuare a contare all'indietro alla lunga complicherebbe ancora di più la faccenda. Come al solito questi consulenti ti mettono fuori delle parcelle incredibili senza risolvere nulla. D'altra parte, io stesso non riesco ad immaginarmi una clessidra con la sabbia che va in su, quindi non so davvero cosa dire.

Qui si vocifera che ci siano tre esperti, orientali o arabi sembra, che ci stanno lavorando, ma non arriveranno prima di Gennaio, mentre addirittura il popolo dice che il Mondo intero così come lo conosciamo cesserà di esistere al momento del passaggio da Avanti Cristo a Dopo Cristo. Forse questi saggi verranno a spiegarci lo "zero" e i "numeri negativi" per contare all'indietro, come dicevo sopra. In ogni caso ti tengo informato.

Vale,
Plutanio.

Insomma, dal Millennium Bug ci siamo già passati, solo che nessuno se lo ricorda più.

## La Tosse Dell'Anima

Non è che c'entri particolarmente, ma uno stacchetto musicale non fa mai male, soprattutto prima di un capitolo impegnativo. Quindi, non si può non citare *The Bug* dei Soul Coughing, insolito gruppo musicale di New York. Non saprei se il Bug in questione sia un insetto o un errore di programma e cercare di capirlo dal testo è particolarmente arduo data la pronuncia inintelleggibile del cantante, ma il pezzo è molto bello. Altre

noti che i programmatori COBOL oggi sono tanto rari quanto profumatamente pagati

[22] Aneddoto liberamente ispirato e adattato da un messaggio anonimo circolante su Internet qualche tempo fa.

selezioni musicali appropriate possono essere "The Gold Bug" di Alan Parsons e "Life is a lemon (and I want my money back)" di Meatloaf, dall'album "Bat out of hell II"[23]

[23] “vero e proprio debugging esistenziale in chiave Schopenhaueriana”, mi assicura The One Phoenix. Direi che non possiamo non citarlo...

## Back To The Box

*Ci sono 10 categorie di persone al mondo:*
*quelle che capiscono il sistema binario e quelle che non lo capiscono.*

Tornando ai nostri scatolotti, è ora di dire le cose come stanno. Se proprio vogliamo parlare di hardware, ammettiamolo chiaramente una volta per tutte, i punti deboli sono sempre quelli: connettori molli e ossidati, cavi rotti, la lente del lettore CD da pulire, un alimentatore vecchio e stanco che non ce la fa più[24]… certo, ogni tanto si brucia davvero il processore, ma quanto spesso? E per il software? Quante volte è colpa nostra, e quante di un programmatore che ha sbagliato lavoro? La verità è che *raramente*, nella vita di tutti i giorni, un computer funziona male per problemi hardware. La colpa è *quasi sempre* del software, ovvero di qualcuno che quel software lo ha scritto.

Da un altro punto di vista, invece, non c'è nessun buon motivo per ritenere che un calcolatore debba funzionare come vogliamo noi. Il funzionamento corretto di un calcolatore è corretto dal nostro punto di vista, ma non necessariamente dal suo. La stramaledetta macchina ci fa spesso il favore di fare quello che noi ci aspettiamo, ma non è detto che questo debba succedere sempre, così come non è detto che domani sorgerà il sole. Bisogna finalmente prendere il coraggio a quattro mani e abbandonare il preconcetto secondo il quale si ritiene che un computer agisca in base a leggi di causa ed effetto. Ecco in che senso la cosiddetta "esperienza" nell'usare un dispositivo elettronico risulta essere simile all'Illuminazione Zen e la modalità con la quale giungervi sia puramente alchemica: si mescolano schede a caso, si spostano ponticelli, si provano driver e upgrade del BIOS e nuove schede, si spostano SIMM e connettori apparentemente senza nessun tipo di logica, seguendo uno schema che ad altri sembrerebbe assolutamente casuale ma che noi, Cavalieri dell'Elettrone, sappiamo essere analogo alla Forza di Guerre Stellari. Lentamente, seguendo l'ispirazione, con la mente sgombra come una lavagna sulla quale è scritto "Boh?", si arriva a diventare Esperti. Non c'è altro modo[25].

Viene a questo punto in nostro aiuto un altro importantissimo concetto da tenere sempre presente, che sulle prime potrebbe sembrare antitetico rispetto a quanto esposto finora: *Il*

[24] "È sempre colpa dell'alimentazione (elettrica)." Babele Dunnit, citazione. Il concetto si rifà in una certa misura ad un analogo ben noto in campo medico, ovvero che "si muore sempre per un arresto cardiaco".

[25] "Da una parte, anche se il buonsenso e la ragione (nei casi in cui si dimostrano efficaci) suggeriscono ottime soluzioni, c'è forse qualcuno che non sappia quanto sia frustrante fare del proprio meglio nel senso suggerito dalla logica e poi vedere le cose andare di male in peggio? Dall'altra parte ogni tanto si assiste al cambiamento desiderato in qualche difficile situazione di stallo, anche se il modo con cui il cambiamento si è verificato ci sorprende per la sua 'illogicità'. In realtà la soluzione inspiegabile, *uncommonsensical*, è un tema archetipico e la troviamo nel folklore, nelle fiabe, nello humour e in molti sogni". Watzlawick-Weakland-Fisch (Mental Research Institute of Palo Alto), "Change – Sulla formazione e la soluzione dei problemi", Astrolabio, 1974. Interessantissimo anche il meccanismo di affermazione contemporanea della negazione di qualcosa e del suo non-contrario ("non A ma anche non non-A"), che permette di costruire a tavolino soluzioni "paradossali" e che sta alla base di alcuni noti aneddoti Zen.

*Computer e' un Idiota Veloce.* Può benissimo darsi che Svariati Folletti Irlandesi abbiano deciso di sedersi sul vostro disco rigido, ma questo non modifica il fatto inequivocabile che la ferraglia rimanga ferraglia. In ultima analisi, se il vostro problema è *indubbiamente* un guasto hardware, allora da qualche parte un cavo rotto, un connettore molle, una ventola che non gira li troverete sicuramente. *Se e solo se* dopo un esame approfondito nessuna di queste cause viene alla luce, possiamo ricominciare a parlare di Svariati Folletti Irlandesi e riconsiderare l'"indubbiamente" di cui sopra.

*[Disegno del Folletto che maltratta il computer]*

Ora, come possiamo cercare di far convivere la visione del computer abitato da misteriose bastarde entità intelligenti con quella totalmente meccanicistica dell'*Idiota Veloce*? La questione assomiglia singolarmente a quella che i fisici d'inizio secolo si sono ritrovati a dover affrontare quando numerosi esperimenti di laboratorio dimostrarono per la luce ora la sua natura ondulatoria ed ora quella particellare. Un secolo dopo, grazie a menti che hanno pensato e dimostrato l'impossibile, finalmente sappiamo che la luce gode, incredibilmente, di questa doppia natura. Per quanto sia difficile da credere, il computer ha analoghe caratteristiche di duplicità. Ancora di più: l'Informatica non potrà essere considerata una *scienza esatta* fino a quando una Teoria Generale Della Complessità Informatica non unificherà le due visioni, così come è successo per la Fisica Quantistica. Fino ad allora brancoleremo nel buio e il Debugging continuerà ad essere un'arte tramandata di padre in figlio (per modo di dire). Quanto sopra esposto è comunque intuitivo: il computer, in quanto Sistema Estremamente Complesso, implica una grande quantità di comportamenti emergenti anche se, a livello teorico è e rimane un oggetto rigidamente meccanicistico e quindi soggetto alle leggi della fisica classica. La stessa cosa, da un punto di vista puramente neurofisiologico e rigorosamente non-mistico, vale per il nostro cervello. Eppure noi, miracolosamente, pensiamo[26]. Non vorremmo arrivare a confrontare la complessità del cervello umano con quella di un ammasso di metallo e silicio e lungi da noi iniziare a parlare di Anima in questa sede, ma il concetto è lo stesso. Più o meno. Quasi.

Purtroppo non è per nulla semplice arrivare a percepire con facilità dove risieda il Problema. I sopraccitati Folletti, lo Spirito nella Macchina, i Bug insomma tendono a manifestarsi con un grado di *sincronicità* addirittura fenomenale. Per fare un esempio, un malfunzionamento apparentemente dovuto ad un aspetto hardware potrebbe essere invece causato (si veda l'ARGH) da una *configurazione* errata, da un *virus* o addirittura da una *bufala,* un problema di disinformazione creato e manipolato più o meno *ad hoc* nel quale siamo cascati come degli allocchi. Qualche esempio tratto dalla pluriennale esperienza di Babele Dunnit e dei suoi Amici ci verrà in aiuto.

[26] Tutti tranne la Zia Carmela, si intende. Lo stesso autore è molto geloso del contenuto del suo cervello (per i curiosi, si tratta di pasta di mandorle).

### *Le Storie Di Babele: (1) L'Incredibile Caso della Tastiera di Ashwà*

Il Computer è uno Strumento del Demonio. Andremo ora a dimostrare questo teorema pur senza dare necessariamente una spiegazione scientifica di questo nostro convincimento; piuttosto, basteranno i fatti. Ad esempio, vorrei raccontarvi cosa è successo ad Ashwà, un mio amico dedito alla negromanzia e a Linux. Dopo aver assistito impotente alla morte del suo Disco Rigido, con la conseguente perdita di due giorni di lavoro (e i danni sono stati limitati solo grazie all'Incantesimo delle Sette Sorelle del Backup) e dopo altri due giorni passati a piangere sul software versato, in un malaugurato tentativo di spostare la tastiera, Ashwà la fece cadere. Questa, di rimbalzo, colpì il tastino del Reset del suo calcolatore, protetto ma non abbastanza. Lo giuro. È successo *veramente*. Ora, tutti sanno che *non va bene* resettare un computer in funzione, anche se non tutti sanno perché. A questo proposito ci sono due spiegazioni, una che potete trovare sui libri e l'altra invece tramandata oralmente di Maestro in Allievo. La spiegazione ufficiale dice che, in mancanza di adeguata quantità di memoria centrale, il Sistema Operativo inizia a scaricare delle copie dalla stessa sul disco rigido, in modo da liberarne una certa quantità e poter continuare a funzionare anche se molto più lentamente. Per inciso, questo meccanismo è chiamato Memoria Virtuale. Chiaramente, un Reset in questo istante è fatale, perché disallinea la RAM reale, che viene cancellata a causa del reset, e quella virtuale copiata su disco, che invece rimane immutata. Se nella operazione vengono perse informazioni riguardanti, ad esempio, la struttura del disco rigido stesso, il Sistema Operativo non ci si raccapezza più, con conseguenze potenzialmente nefaste. Inoltre il Sistema Operativo mantiene delle copie dei dati più usati del disco in memoria centrale, in modo da potervi accedere più velocemente; solo ogni tanto riallinea questi dati con quelli effettivamente scritti sul disco, riscrivendoli su quest'ultimo. Quest'altro meccanismo viene chiamato Cache Memory ed anche questo, in caso di Reset, si guasta irrimediabilmente per gli stessi motivi di cui sopra. Questa spiegazione, anche se teoricamente corretta, risente di una visione meccanicistica del computer, decisamente occidentale e cartesiana, per non dire aristotelica. In una versione più consona ai tempi moderni, che solitamente si può proporre solo a chi sia in qualche modo un Iniziato e che non esiterei a definire olistica, al momento del Reset all'interno del computer si spengono le luci e i vari componenti, CPU e DSP inizino a vagare confusi e felici, provocando danni più o meno gravi. A questo proposito lo Sciamano M'Bhutu Ghoile dice testualmente: "Esistono Duecentomila Algoritmi tra il Cielo e la Terra, e solo di Due di Essi io Ignoro l'Origine: La Memoria Virtuale e il Protocollo X-25".

*Commento:*

Come non ammettere davanti ad un caso simile che il computer in questione non fosse posseduto da qualche Malefica Entità? A fronte di simili eventi nulla si può fare, è ovvio[27]. L'unica strategia possibile in questi casi è giocare d'anticipo. Nel caso specifico,

[27] Tranne fotocopiare, ritagliare e costruire Bastet-Con-Le-Sfigh, il Demone Antiurto, appoggiandolo vicino al computer. Il Demone Antiurto protegge il computer nella sua fisicità da tutti gli urti, rovesciamenti di caffè, calci involontari e molte altre simili nefaste occorrenze. Nulla può contro i gatti, però, essendo esso stesso una divinità felina.

Ashwà avrebbe dovuto ricordarsi che se qualcosa sarebbe potuto andare male lo avrebbe fatto[28] e avrebbe dovuto quindi evitare di comprare un computer con il tasto del reset alla mercé dell'Universo, o quantomeno avrebbe dovuto metterlo in una posizione più protetta. Appoggiare il computer per terra *non va bene*. Quando vi sedete una volta su due gli tirate un calcio. E poi ci entrano polvere e topi e i secondi si divertono molto correndo sulla ventola, in mezzo a nuvole della prima. Detto questo, potete benissimo lasciare lì il vostro computer, ma sappiate che prima o poi vi succederà qualcosa di analogo a quanto è successo ad Ashwà: all'interno del vostro computer si spegneranno le luci e i vari processori inizieranno a far baldoria. E non è questione di sfiga: è *statistica*.

*[DIY: Il Demone Antiurto, che è un gatto di qualche tipo]*

Ad onor del vero, bisogna anche dire che ci sono situazioni senza speranza: ad Owen, un altro amico di Babele, è successo che la gatta Biba schiacciasse con il naso il tasto del Reset cercando di capire cosa fosse quello strano rumore che faceva il floppy disk. Con i gatti c'è poco da fare: il computer lo potete anche inchiodare al soffitto, ci arriveranno lo stesso e si posizioneranno a testa in giù sulla tastiera in modo da digitare esattamente i tasti Ctrl-Alt-Del, la password di amministratore e il comando di formattazione totale globale del disco rigido. L'unica accortezza possibile in una situazione gatto-computer è chiudere uno dei due in una cassaforte e dare dei croccantini all'altro[29].

L'unica soluzione a questi problemi è quella auspicata dal Padre Fondatore XXX Tanenbaum, che ha praticamente dedicato la vita a chiedersi perché i computer si incartano e a pensare come migliorarli:

> I will consider the job finished when no manufacturer anywhere makes a PC with a reset button.

Forse un giorno ci arriveremo. Nel frattempo non trattenete il respiro, però...

[28] Certo. La cara vecchia Legge di Murphy. In ognuna delle sue diecimila e più varianti, questa Verità Assoluta è da tenere sempre presente. Probabilmente tutto questo libro potrebbe essere riassunto con l'enunciato della Legge di Murphy, ma sarebbe meno divertente, quindi noi si va avanti a scrivere e voi a leggere.

[29] Nell'improbabile caso in cui il vostro computer disdegni i croccantini provate con tonno e riso, che è una vera leccornia per qualsiasi processore.

## *Le Storie Di Babele: (2) L'Incredibile Caso del Connettore Bastardo*

Due mattine or sono ho dovuto fare una copia di backup su uno ZIP-Drive. Attacco, carico il software, non funziona. "Non sono state trovate unità removibili". Mi altero un pochino: è evidente che l'unità removibile sia presente e connessa e accesa, la vedrebbe anche un dueottosei col CP/M (purché munito di telecamera e opportuno driver). Smanazzo, trovo conflitti, li risolvo, scarico i driver nuovi da Internet alla fantastica velocità di centootto bytes al secondo, alla fine l'accrocchio parte e compare un disco rimovibile H: in tutta la sua gloria. Però non va più il mouse. Ipotizzo conflitti, ma faccio finta di niente e copio quello che devo copiare usando unicamente la tastiera. Impresa epica, ci sono alcuni comandi a più tasti contemporanei per i quali ho dovuto chiedere aiuto al mio amico Froppz di Andromeda, un polipo mutante pianista con centootto dita su ognuna delle sue otto mani. Adesso potrei schiacciare noci a mani nude, sono diventato un vero culturista almeno per quanto riguarda pollici ed indici. Stacco lo zippotto, spengo e riaccendo, niente mouse. Ipotizzo conflitti (ancora!), rimuovo il driver parallelo-SCSI usato dallo ZIP-Drive. Spengo e riaccendo, niente mouse. Rimuovo il driver del mouse, spengo e riaccendo, reinstallo il driver del mouse, spengo e riaccendo, niente mouse. Controllo esternamente i cavi, tutto a posto. In un impeto di disperazione e prima della giubilazione definitiva mediante immissione di corrente industriale a 380V trifase direttamente sui connettori di alimentazione 5-12V della piastra madre, subodoro il Bug Malefico e apro il computer. Dentro, incredibilmente, ci trovo il connettore della seriale alla quale è normalmente collegato il mouse che fluttua liberamente nello spazio. Come ha fatto a sganciarsi? Soprattutto, come ha fatto a scegliere esattamente il momento in cui io sono riuscito a fare partire lo ZIP-Drive per staccarsi, in modo da farmi pensare che il tutto fosse un ennesimo problema di incompatibilità tra driver? Io insisto col dire che queste macchine non solo sono intelligenti, ma anche un pò bastarde. Non lasciatele mai accese ed incustodite a lungo e badate che non sto parlando di risparmio energetico.

*Commento:*

Ora per fortuna con USB, FireWire e compagnia bella la faccenda è *molto* più semplice, quindi per la comprensione dei molti termini tecnici peraltro vecchi di anni ci rifaremo ai principi della maieutica socratica[30]. Cionondimeno siamo in presenza di un tipico caso di sincronicità hardware che però ci porta a pensare a malfunzionamenti tipicamente software. Come si vede, anche un vero hacker[31] come Babele Dunnit davanti a simili incredibili coincidenze si ritrova spiazzato e brancola nel buio. Quello che lo distingue da altri, forse meno esperti, è la quantità di tempo perso: nel suo caso relativamente poco. Una situazione simile potrebbe invece protrarsi per molto tempo in mancanza di una vera *intuizione*. In effetti esistono anche circostanze nelle quali un *problema tecnico* banale

[30] La conoscenza è dentro di te, tutto quello che devi fare è cercarla ed improvvisamente capirai tutto. Maieutica. Comodo. Troppo comodo. Bella la vita, eh? In verità, come dice Guzzanti, "A' Risposta che cerchi è ddentro de te. Solo che è *sbajata*".

[31] Attenzione, non spaventiamoci. Il termine "hacker" qui è inteso nel suo significato originario. Torneremo su questo concetto più avanti. Niente paura. Niente paura.

diventa irrisolvibile perché porta con sé un *problema politico*. Questa è una situazione aziendale tipica, che tende a manifestarsi in maniera direttamente proporzionale alle dimensioni dell'azienda stessa. Vedremo meglio questo fenomeno quando affronteremo i bug relativi al *PeopleWare*, ovvero alle *Risorse Umane*.

Ovviamente ci sono poi anche i casi di T.I.S.I., Totale Incapacità Sistemistica Invalidante, davanti ai quali nulla si può. Lasciamo la parola ancora una volta a Babele Dunnit e al suo amico Rob, questa volta con un salto notevole dal punto di vista della complessità:

Babele:

```
Ho visto la cosa più scema del mondo.
Qui hanno messo in Work Load Balancing due processi gemelli che FANNO
LA STESSA COSA E GIRANO SULLO STESSO PROCESSORE.
```

Rob:

```
è al livello di un "sistema di sicurezza" che ho visto:
Volume C: mirrorato sul D: ...che stanno OVVIAMENTE sullo stesso disco
fisico. Per cui non solo sicurezza aggiuntiva zero ma anche
prestazioni di accesso ai dati quasi dimezzate...
```

*Parafrasi e Commento* :

Il Work Load Balancing è un sistema grazie al quale un lavoro particolarmente pesante viene distribuito su più processori[32]. Ovviamente bisogna avere un computer multiprocessore, per poter utilizzare il Work Load Balancing. Ora, se si prende un lavoro particolarmente pesante e lo si scinde in due lavori separati, ma poi questi due lavori si fanno comunque fare allo stesso processore, ovviamente non si ha alcun vantaggio. Anzi, è ancora peggio, perché adesso c'è anche il lavoro del Work Load Balancing a dover essere svolto, nonostante non serva assolutamente a nulla. A complicare ulteriormente le cose in questa situazione abbiamo addirittura due lavori tra loro identici. Sarebbe come prendere due persone A e B, assegnare ad ognuno la trascrizione in bella calligrafia delle prime 200 pagine dei Promessi Sposi (perché?) e, alle rimostranze sulla dimensione del compito assegnato, far trascrivere ad A le sue prime 100 pagine e le seconde 100 pagine di B (che sono comunque uguali alle seconde 100 pagine di A) e agire simmetricamente per B. Insomma, qualcosa di *veramente* inutile.

[32] Il processore di un computer, detto anche CPU ovvero Central Processing Unit, è in parole povere *Quello Che Fa I Calcoli*. E, gente, li fa *in fretta.* Come per le patate arrosto, più ce n'è e meglio è; ovvero, è meglio avere un computer con uno o – meglio ancora – più processori veloci che una vecchia stufa a gas che ci mette due giorni a partire. Attenzione, però: vedremo anche che le *prestazioni* di un computer sono il risultato di una interazione complessa tra i vari componenti del computer stesso; in particolare, vedremo che la velocità di una macchina è pari a quella del suo componente *più lento*. Si parla in questo caso di *bottleneck*, ovvero "collo di bottiglia". Torneremo su questi temi quando parleremo di *ottimizzazione*.

Quanto ha visto Rob è, se possibile, ancora più demenziale. Spesso si usa, per i computer sui quali risiedono Dati Particolarmente Importanti[33], utilizzare una tecnica di salvaguardia degli stessi chiamata *mirroring*[34], che consiste nell'utilizzare in parallelo più di un Hard Disk[35] ridondando le informazioni scritte. Ovviamente utilizzare un sistema di sicurezza di questo tipo su *un solo* HD fisico, facendo credere al Sistema Operativo[36] che i dischi siano invece più d'uno è una mossa di una stupidità sconcertante, ovvero un caso di T.I.S.I. L'unico motivo che ci può faticosamente spingere, in casi come questo, a dare una giustificazione a tanta incompetenza è un *problema politico* soggiacente, ovvero il fatto che in una Grande Azienda si è in tanti e bisogna mangiare tutti e che spesso il sistemista più incapace è anche il cugino del coniuge dell'Amministratore Delegato, se non addirittura il figlio "appassionato di computer". In questo ultimo caso se ne vedono davvero delle belle, come ad esempio videogiochi installati su server pieni di virus e tutti che fischiettano facendo finta di niente.

Infine, ecco cosa dice Remote, a rimarcare queste proteiformi capacità del Nemico:

> Non ricordo se sia nella Fisica Quantistica o nella Relatività Generale che si dice che la presenza di un osservatore modifica l'evento che si sta osservando. Tu lo saprai sicuramente[37].
>
> Questa cosa è altretttanto vera nel Debugging.

[33] Il concetto di Dato Particolarmente Importante è uno dei più elusivi e soggettivi di tutta l'informatica. Quello che a una persona qualsiasi appare come una collezione di files liberamente cestinabili, per il suo superiore rappresenterà una collezione di informazioni strategiche di importanza tale che una malaugurata perdita degli stessi farà precipitare la situazione a *Defcon3*.

[34] In prima approssimazione, non storcano il naso gli hacker. Da *mirror*, "specchio" in Inglese. Il perché questa tecnica si chiami così è ovvio, se andate avanti a leggere. Il fatto poi che in italiano diventi "mirrorare" effettivamente fa schifo a tutti.

[35] Spesso abbreviato con HD, "disco rigido" in Italiano. E' il principale supporto di memoria non-volatile (ovvero che rimane inalterata anche a computer spento) dei computer attuali (A.D. 2005). Il motivo di questo nome si perde nella notte dei tempi: gli HD si chiamavano così per distinguerli dai *Floppy Disk*, "dischi flessibili", che una volta erano davvero pieghevoli. Poi invece i floppy sono stati rimpiccioliti e racchiusi in custodie di plastica diventando i dischetti da 3.5'', che con il formaggio genericamente definito "sottiletta" condividono sia dimensioni che affidabilità ma non più la consistenza.

[36] Il *Sistema Operativo*, abbreviato a volte come SO in Italiano ma molto più universalmente OS (Operative System) in Inglese è quel considerevole insieme di istruzioni che trasformano lo Scatolotto da inutile cubo di latta in, appunto, un Computer, che però non è detto sia sempre più utile di un cubo di latta. Se dovete nascondere i croccantini perché c'è in giro il gatto Biba del mio amico Owen, ad esempio, non vi serve alcun Sistema Operativo: vi basta una scatola di latta. Torneremo sui SO più avanti, nella parte dedicata al *Software*.

[37] Siccome l'autore *odia* smentire i suoi amici, deve ammetterlo: si tratta del *Principio di Indeterminazione di Heisenberg*, che afferma – nella sua formulazione più semplice – che a livello quantistico è possibile conoscere la velocità di una particella subatomica *oppure* la sua posizione, ma non tutte e due contemporaneamente. Se poi volete finalmente capire qualcosa di Meccanica Quantistica, leggetevi il bellissimo "Tao della Fisica" di Fritjof Capra, Adelphi: un altro libro pieno di tarli...

Se un bug viene rilevato da un utente, questo bug non sarà riproducibile o sarà comunque diverso se osservato da un sistemista o dall' autore del programma. In pratica esiste una sorta di empatia tra il Bug e chi lo sta osservando, che fa in modo che il grado di complessità del Bug sia proporzionale alle capacità dell'osservatore.

In questa ottica il "Non mi si apre la finestra!" dell'utente finale equivale a un mancato doppioclick secondo l'helpdesk, a un problema a livello di Sistema Operativo per un analista e a una non corretta inizializzazione di un puntatore o qualsivoglia bastardata di stringa nel registro per il Samurai del Codice.

Il che vale a dire che il Combattente deve avere la capacita di vedere il Nemico da molti punti di vista e scegliere il migliore per attaccarlo. Ma dell'aspetto strategico parleremo poi approfonditamente; piuttosto, scusate se questa ultima parte è stata piuttosto pesante. Per farci perdonare, ecco un simpatico RAID fai-da-te. Rilassatevi.

*[DIY: Un RAID da Ritagliare e Costruire]*

### *Le Storie Di Babele: (X) L'Incredibile Caso del Piedino Termoelettrico*

Qualche anno fa vinsi un notebook. Alla faccia, direte. È vero, rispondo io, anche se – non essendo una estrazione casuale ma un concorso di scrittura al quale partecipai sotto pseudonimo – possiamo a mia discolpa affermare che qualcuno decise che me lo meritavo. D'altra parte, a concorsi basati sulla pura fortuna - dal *Se Lo Sai Rispondi* di *Topolino* al Superenalotto - non ho mai vinto niente. Ma stiamo divagando. Insomma, il nuovo arrivato era una macchina notevole, sicuramente più bella del mio *OmniBook* dell'epoca, ormai già vecchio di un paio di anni. Peccato che non funzionasse. Si incartava. Semplicemente, si bloccava tutto. Tastiera, mouse, persino le lucine del Caps Lock a un certo punto non funzionavano più, in un muto *memento mori* digitale. A volte andava per ore, altre non faceva neanche in tempo a finire il bootstrap. Vari trattamenti, da quello psicologico a quello esorcistico – passando per tutte le formattazioni e reinstallazioni del caso – non migliorarono mai la situazione, mentre il supporto tecnico si dimostrava particolarmente inetto e il concetto del Caval Donato, al quale notoriamente non si guarda in bocca, unito alla mancanza di una vera necessità contingente, la facevano da padroni impedendomi di incazzarmi seriamente. Insomma, pur tecnologicamente superiore, questo portatile (un Compaq 705EA – e facciamo questi nomi!) è sempre rimasto una macchina secondaria rispetto all'indistruttibile Hewlett-Packard OmniBook 4150, mentre il mio senso tecno-etico mi ha sempre impedito di vendere una macchina che sapevo non essere perfettamente funzionante.

Così passano vari anni e un bel giorno, vagando su Internet, trovo per caso una pagina di un utente americano che racconta la sua personale odissea con un portatile uguale: crash di sistema immotivati e assolutamente casuali, formattazioni, cambio di sistemi operativi, applicazioni critiche etc. Anche lui arriva ad una diagnosi che avevo io stesso ipotizzato, ovvero che fossero problemi di surriscaldamento e che per qualche motivo la ventola non facesse in tempo a partire, o qualcosa di simile. Segue una ridda di interventi di smanettoni vari da tutto il mondo che spiegano come aprire la macchina e modificare il circuito stampato in modo da svegliare il sensore di temperatura, propongono patch fatte in casa al sistema operativo o altre simili soluzioni più o meno esoteriche. Ricordo addirittura un intervento che affermava che il modello in questione fosse stato ritirato dal mercato perché termodinamicamente bacato.
"Aha" penso io, ecco perché hanno messo in palio una macchina così bella, non potevano certo vendere un computer che si incarta ogni due per tre... fino a quando non leggo l'ultimo messaggio, a fronte del quale rido per una settimana:
"Tutto vero, ma c'è una soluzione molto più semplice: sotto al computer ci sono dei piedini estraibili. Apriteli, in modo che la base rimanga sollevata dal tavolo e possa circolare aria ..."

In seguito, il Reverendo avrebbe commentato:
"usa i fottuti piedini, no? Se ci sono serviranno pure a qualcosa..."
mentre Remote avrebbe a sua volta fatto notare:
"In effetti la migliore posizione per usare un laptop (dal punto di vista del comfort del laptop, s'intende) risulta essere quella semisdraiata con cuscinone dietro la nuca, ginocchia flesse sulla cui punta s'appoggiano gli angoli finali del portatile in precario equilibrio per lasciare libere le feritoie di ventilazione, cavo alimentazione che passa sotto la schiena, frontale del laptop piantato nello sterno, braccia raccolte nella posizione dello struzzo e diciotto gradi di libertà per i movimenti dei polsi.
Il tutto in un freezer.
Almeno... così il mio non si incarta.
P.S. Ho tralasciato le complicazioni dovute a periferiche USB, cavi modem, mouse, e cuffie eventualmente connesse."

Insomma, era vero. La faccenda dei maledetti piedini era vera. Da allora il piccolo mostro non ha più fatto una piega, e quando il masterizzatore del desktop è morto di vecchiaia il 705EA ne ha preso il posto, senza fallire mai una copia. Peccato che nel frattempo sia diventato vecchio pure lui. Avremmo potuto essere amici.

L'elenco dei malfunzionamenti puramente hardware non si ferma qui; ci ritorneremo ancora ed ancora, ma giusto per focalizzare brevemente l'argomento, ricordatevi che in questi casi la soluzione di solito è *banale*. L'ipotesi che il vostro computer (o la stampante, o lo scanner, o il mouse, o la tastiera, o l'ormai onnipresente teletrasporto) non funzionino più perché "si sono rotti" è da prendere in considerazione solo dopo aver esaminato cavi e connettori, controllato alimentatori e batterie, lanciato sale dietro le spalle e appeso una treccia di aglio all'uscio di casa. Poi, il fatto che, statisticamente

parlando, l'installazione di nuovo hardware venga meglio se fatta da Jackie Chan o da Obi Wan Kenobi – insomma da qualcuno che sa come usare la Forza – è un altro discorso.

## Fast And Furious

Continuando questa nostra libera associazione di concetti fondamentali arriviamo a parlare di *Prestazioni*. Ed iniziano davvero i problemi. Certamente, nel caso di una macchina inesplicabilmente *lenta* non si può parlare di un vero e proprio bug, ma nemmeno di un computer in *buona salute*. Come vedremo, da questo punto in poi siamo obbligati ad iniziare a parlare di *Software*, complicando la faccenda in maniera esponenziale.

Bisogna infatti ricordare che esistono decine di programmi per misurare le prestazioni dei calcolatori ed altrettante unità di misura, dai vecchi Drystones ai Winmark ai piu' crudi MIPS, MFlops, GIPS, Gigaflops, Megapixel e Megatexel, Furlong per Forthnight e tutto un coacervo di "quantità di cose al secondo" che sfugge all'umana comprensione. All' incirca, ogni costruttore di CPU ha una *suite* di benchmark e una unità di misura nella quale la propria CPU surclassa in modo eclatante tutte le altre, mentre ogni costruttore di hardware ha una macchina più veloce e potente di tutte le altre messe insieme. Questo è un fenomeno interessante che ci riserviamo di analizzare in futuro, ma è in qualche modo collegato al fatto che, quando l'acqua diventa non potabile, basta elevare la soglia di schifezze ammissibili in un litro della stessa per farla ridiventare, appunto, potabile.

Quindi, ecco alcune Regole d'Oro sulle Prestazioni del Sistema Complesso:

- La velocità di un Sistema Complesso è dettata dalla velocità del suo componente più lento.
- Le operazioni più veloci sono quelle che non vengono svolte.
- Non provare. Fai. O non fare. Non esiste provare. (Yoda, maestro Jedi)
- Ogni Sistema Complesso è un insieme di N sistemi semplici non compresi e/o male documentati. (Assioma di Remote[38])
- Tutto è bacato. (Postulato di Cortez)

Infine, dopo aver esaminato e commentato questi concetti chiuderemo con l'importantissima "Regola 80-20".

[38] Riassumibile con uno degli acronimi più noti del gergo informatico e più usati nel mondo dello sviluppo del software: RTFM, "Read The Fuckin' Manual", leggiti il fottuto manuale (...prima di fare queste domande idiote, farci perder tempo e perderlo tu stesso in prove inutili)

## La Legge del Collo di Bottiglia

"*La velocità di un Sistema Complesso è dettata dalla velocità del suo componente più lento*" è nota come Legge del Collo di Bottiglia. Quello che afferma è assolutamente intuitivo, infatti:

- Se prendete una Ferrari e le montate il motore di una Panda, l'oggetto risultante andrà alla velocità di una Panda.
- Se prendete una Panda e le montate il motore di una Ferrari, l'oggetto risultante arriverà comunque a velocità minori di quelle di una Ferrari per motivi aerodinamici, poi si staccherà dal suolo e si vaporizzerà nella ionosfera.

e questo ovviamente vale anche per sistemi con più di due componenti in grado di sopportare velocità diverse – come ad esempio una Panda con le gomme di una Ferrari e un Campini–Caproni stellare tredici cilindri come motore, praticamente un idrovolante a forma di utilitaria ma con le gomme da corsa. Inoltre vale anche per diverse definizioni del concetto di "velocità": se mangiate dell'insalata e delle cotiche di cammello fritte, quella spiacevole sensazione di pesantezza durerà fino a quando non avrete digerito entrambe le portate – e non è detto che digeriate prima l'insalata[39].

Insomma, state attenti quando dovrete ampliare la memoria del vostro computer[40]: se comprate due banchi di memoria in due momenti diversi, può succedere che i vari componenti elettronici non si piacciano reciprocamente quando vengono obbligati a funzionare insieme, costringendo altre parti del computer a rallentare per mantenere una coerenza tra sincronismi e segnali elettrici assortiti. Il tutto si traduce in una efficienza non ottimale del sistema, che pur avendo più memoria di prima sembra non andare molto più veloce.

## La Legge del Processore Pigro

"*Le operazioni più veloci sono quelle che non vengono svolte*". Questa legge assume numerose incarnazioni: ad esempio, parlando di grafica, le figure più veloci ad essere disegnate sono quelle che non vengono disegnate del tutto. Sarà idiota come consiglio, ma se dovete disegnare una sfera composta da un milione di poligoni conviene togliersi dai piedi il mezzo milione di poligoni che sta "dietro" e che quindi comunque non si vedrebbe. Questo concetto è analogo alla cosiddetta *Lazy Evaluation* e sta alla base di molte situazioni prettamente informatiche, come l'utilizzo di un *proxy*[41] – ovvero di una

[39] Un amico di Babele Dunnit noto come Dado L'Uomo di Legno digerisce *tutto* tranne che rucola e soncino.

[40] Prima o poi vi succederà. "Dato un software, questo aumenterà le proprie dimensioni fino ad occupare tutta la memoria disponibile". Vale anche per gli armadi, le scarpiere e in generale le case.

[41] Si, il concetto di "proxy" non si applica solo a Internet. È un concetto generico. Cionondimeno, quando il lavandino è pieno e i piatti sono obiettivamente da lavare, non si può invocare un concetto – per quanto

entità che aspetta a fare quello che è stato richiesto fino a quando non è assolutamente indispensabile. In generale, è inutile e antieconomico fare un lavoro prima che questo effettivamente serva[42]. Purtroppo è quel *prima* che solitamente è difficile da stimare, perché succede molto, molto spesso che quando finalmente ci si decide sia ormai troppo tardi. Come dire: alzatevi e bevete quella birra *ora*. Tra dieci minuti il mondo potrebbe non esistere più.

## Il Consiglio di Yoda

"*Non provare. Fai. O non fare. Non esiste provare.*" riassume in poche parole quello che dovrebbe essere il comportamento di un vero Ninja Esadecimale. Purtroppo sappiamo benissimo che la realtà non funziona così ed infatti a darci questo consiglio è un Cavaliere Jedi[43]. Insomma, è ammesso anche un minimo di dubbio, in misura inversamente proporzionale alla nostra esperienza nel campo in questione. Ad esempio, Babele Dunnit non ha dubbi per quanto riguarda i computer, ma provate a metterlo su un canotto...

## L'Assioma di Remote

Con il suo Assioma – *Ogni Sistema Complesso è un Insieme di N Sistemi Semplici Non Compresi e/o Male Documentati* – il grande Remote cerca di ricondurre il concetto di complessità su un piano scientifico. Il razionale dietro a questa considerazione è che se qualcosa ci sembra complesso, la colpa è nostra. Probabilmente non siamo ancora pronti, non abbiamo ancora capito, non è ancora scattata quella scintilla che improvvisamente fa esplodere la camera satura di neuroni allo stato gassoso nella quale ci troviamo, illuminandoci subitaneamente. Non male per uno che, in altre occasioni, disse di un prodotto che "Se funziona è l'uovo del Tenente Colombo".

## Postulato di Cortez

"*Tutto è Bacato*" *non è* semplicemente l'ennesima riscrittura della Legge di Murphy, anche se in un certo senso ci ricorda il già citato Reverendo quando dice che "L'Universo è un grosso baco". Le due visioni sono invece profondamente diverse: il dott. Francisco Cortez – noto analista informatico – non si limita ad una considerazione globale, ma ci dice qualcosa in più: ci dice che le cose andranno male *perché c'è un bug*, passando dalla valutazione qualitativa di Murphy a quella quantitativa e ponendo le basi per la nascita di

generico come un proxy – per continuare a guardare i Simpsons in televisione.

[42] Oppure rifarlo da capo quando lo si è appena fatto. In questo caso i concetti di *proxy* e di *cache* si mescolano, giusto per incasinare la faccenda.

[43] Dalla generalità di questa legge si desume che Yoda non conosce intimamente solo i computer, ma l'universo intero. Altrimenti non farebbe lo spaccone in quel modo.

una nuova Scienza[44], la *Bacologia*. Queste sono parole di speranza: è implicito che, una volta risolto il bug – anzi, gli infiniti bug che permeano la Realtà, ovvero Il Bug Primordiale – le cose inizieranno ad andar bene. Lasciamo Cortez a crogiolarsi nella sua beata illusione e procediamo.

**Gran Finale: La Regola 80-20**

La "Regola 80-20", conosciuta anche come "Principio di Pareto[45]", afferma che in qualsiasi cosa il 20% e' importante e l'80% non lo è. Incredibile ma vero. Ad esempio, in un gruppo di lavoro, l'80% sono cialtroni che svolgono il 20% di quello che c'è da fare nell'80% del tempo – sbagliando l'80% di quello che fanno – mentre il restante 20% del gruppo svolge l'80% del lavoro in un 20% del tempo sbagliando solo il 20% di quello che fa. E non provate a dire di no perché lo sappiamo benissimo tutti che è vero.

In particolare, in campo informatico, l'80% delle risorse, del tempo macchina, della memoria, degli accessi al disco, del tempo di manutenzione etc di un programma sono usati, eseguiti, dedicati al 20% del programma[46] stesso. Questa regola empirica è ormai ben nota ad ogni Samurai della Tastiera che si rispetti: prima di ottimizzare qualcosa, è opportuno capire se ne vale la pena. Id Software[47] con Doom ha stupito il mondo dei videogiochi raggiungendo velocità fino a quel momento impensabili per le macchine dell'epoca semplicemente riscrivendo in maniera ottimale e più accurata una piccolissima parte del videogioco in questione: quel famoso 20% (e neanche tutto).

Si noti che quelle appena viste non sono altro che considerazioni generali *sempre* valide quando siamo in presenza di una situazione dove ogni singolo componente interagisce con molti altri (ovvero nel 99% dei casi, da cui si deduce che l'Universo è Intrinsecamente Incasinato[48]). In particolare il Principio di Pareto dovrebbe insegnarci a gestire meglio le nostre forze, concentrandoci sul 20% che importa veramente, ma questo

[44] È questo un cambio di paradigma analogo a quello che ha segnato la nascita della Fisica Moderna, quando Galileo Galilei ha iniziato a misurare il Mondo parlando di riproducibilità dell'Esperimento Scientifico (anche se già Leonardo e Bacon avevano dimostrato di avere le idee abbastanza chiare sulla faccenda).

[45] Vilfredo Pareto, economista italiano, nel 1906 notava come l'80% delle risorse economiche fosse sotto il controllo del 20% della popolazione, ma solo nel 1940 il dr. Joseph M. Juran utilizzava questa osservazione rendendola famosa come "Principio di Pareto".

[46] Ok, ma cosa è veramente un programma? In prima approssimazione, diciamo che un processore è in grado di fare un certo insieme di operazioni matematiche e che una sequenza ben precisa di queste, decisa da un programmatore, rappresenta appunto un programma. Per quanto possa sembrare incredibile, una *grande quantità* (milioni) di operazioni matematiche estremamente semplici svolte in una sequenza ben precisa e a *grande velocità* (milioni di operazioni al secondo) riesce a trasformare il nostro ferrovecchio in una macchina per scrivere o a permettergli di predire (sbagliando, ma è colpa del programmatore e/o della inesattezza intrinseca dei dati in ingresso) che tempo farà domani.

[47] ID Software è la società di videogiochi che ha entusiasmato con Wolfenstein Castle e successivamente sconvolto con l'ormai mitico Doom, per finire con Quake. Di fatto hanno dimostrato che certe cose, anche con un PC, si *potevano fare*. Mica ce lo aspettavamo.

[48] È degno di nota il fatto che nel caso della complessità dell'Universo la regola dell'80-20 non valga. Questo Yoda lo sa, ma fa finta di nulla.

non è un vademecum per l'accrescimento dell'autostima o una qualche cavolo di rivista *New Age* con accluso CD di musica rilassante/pallosa, quindi possiamo tranquillamente continuare a incasinarci la vita.

## Coperchi Tondi e Pentole Quadrate

É opportuno spendere qualche parola a proposito di una delle spine nel fianco dell'Informatica, nonché di un fenomeno ad essa strettamente collegato: *Compatibilità* e *Obsolescenza*. Il tempo passa, le cose cambiano e davanti a questi elementari dati di fatto si possono tenere due atteggiamenti antitetici: cercare di fare proseguire le cose così come stanno o buttare tutto alle ortiche e ripartire da zero. In mezzo ci sono tutte le sfumature del caso che, per quanto riguarda l'informatica, nel tentativo di salvare capra e cavoli hanno dato luogo a soluzioni pressoché inspiegabili per problemi pressoché insolubili[49].

*[DIY: un oggetto da ritagliare e costruire oppure un disegno, un incastro tipo gioco per bambini con un cubo che dovrebbe entrare in un cubo più grande che pero' ha dei fori rotondi sulle facce]*

Davanti all'obsolescenza dell'Universo non c'è nulla da fare, se non rassegnarsi; è una Legge di Natura. Il Tempo scorre invariabilmente in una sola direzione[50]. Davanti all'obsolescenza del vostro computer però c'è il dovere di innervosirsi almeno un po', perché non è vero che, per girellare su Internet e scrivere lettere, fatture e qualche Email abbiamo bisogno di un nuovo computer ogni sei mesi. Purtroppo questo è quanto vogliono farci credere e così, ogni pochi minuti, esce una nuova versione del Fantastico Sistema Operativo, dell'Irrinunciabile Programma, del Paradisiaco Plugin e di altri ventimila tipi di software senza i quali, ci dicono, proprio non possiamo stare. Infine, subdolamente e lentamente, questi nuovi programmi diventano incompatibili con quelli vecchi – non apertamente, per carità: semplicemente, iniziano a "funzionare male"...

Trinity: Ho aperto il tuo documento ma non c'è dentro nulla!
Zimo: Ma cosa stai dicendo? Da me funziona perfettamente!
Trinity: Mah… non capisco. Aspetta! Ma che versione di Wozz hai usato?

[49] Assolutamente da citare qui il modello di architettura software cosiddetta *a lasagna*, nella quale, strato dopo strato, "i difetti di funzionamento sostanziali sono completamente occultati dai difetti di funzionamento superficiali". Questa strepitosa citazione risalente al 1984 è di Douglas Noel Adams, autore della famosa *Hitchhiker's Guide To The Galaxy*. Originariamente si riferiva ai prodotti perennemente malfunzionanti della Società Cibernetica Sirio, il cui Reparto Reclami occupa tre pianeti (l'avete già sentita?). Non c'è dubbio che Adams avesse raggiunto una profonda conoscenza della Vita, dell'Universo e di Tutto Quanto e possiamo solo piangere la sua prematura scomparsa.

[50] In prima approssimazione. Einstein avrebbe un po' di cose da dire a proposito, ma per ora facciamo finta di nulla.

Zimo: Aspetta… la 19.7.2 per Ossix Maggintoc, il Sistema Operativo Più Fico Del Mondo…

Trinity: Questo spiega tutto. Io lo sto aprendo con la 21.2 per CiccioIcsPi, il Sistema Operativo Più Grasso Del Mondo, e il file sembra vuoto. Lo devi salvare con l'opzione "Mantieni Compatibilità Davanti, Dietro, Sopra E Sotto E Anche Con La Nonna In Salamoia" mentre ciucci il becco a un peluche portafortuna raffigurante un pinguino verde, altrimenti qui non ne usciamo vive.

Quante volte avete assistito ad una conversazione simile a quella riportata qui sopra? Oltretutto non c'è neanche bisogno di introdurre appositamente questi problemi, cosa che comunque qualche produttore di software fa di proposito: in ogni caso, il Bug dell'Incompatibilità e dell'Obsolescenza ha tutta la forza per manifestarsi da solo. Famosa è ad esempio la *Profezia dei 640 KiloBytes*, che si dice sia stata pronunciata da Bill Gates in persona: "640K di memoria saranno più che sufficienti". Oramai una macchina con Mezzo Giga di RAM, ovvero mille volte tanto, è considerata una entry-level. Anche Babele Dunnit, durante uno dei suoi viaggi nel tempo, quando incontra i progettisti Intel cerca inutilmente di cambiare il corso della Storia:

### *Le Storie Di Babele: (3) Il Senno Di Poi (excerpt)*

"Ragazzi, rimanga tra noi, ma guardate che queste idee dei segmenti, dei sedici bit, dei 64K per volta e degli interrupt cablati in quel modo non sono il massimo. Porteranno grande sventura". Rafforzo questi concetti in sé banali comparendo dal nulla vestito come la Dea Kalì, con una serie di braccia sintetiche servoassistite che ho comprato su Altair quell'anno che a carnevale volevo travestirmi da Dottor Octopus. Provoco un certo scompiglio, i progettisti indiani si prostrano ai miei piedi, c'è qualche altro sviluppatore che si fa il Segno della Croce ed inizia ad urlare "Vade Retro" o "Orribile Mostro" o "Hic Stantibus Rebus" mentre i project manager giapponesi si limitano a buttarsi al suolo e scattare delle fotografie. Mi siedo, per terra anch'io per non essere da meno, e in breve traccio una visione apocalittica dove tuttologi che non sanno nulla chiamano "Realtà Virtuale" anche l'atto di mangiare la pizza mentre designer di videogiochi pensano alla boxe virtuale dove si usa un solo braccio per risparmiare sui costi dei sensori. Mentre guardo questi eroi del passato mi vengono in mente le Soyuz, praticamente degli scaldabagni con un propulsore a reazione attaccato sotto, e me li immagino con la scritta "Gatorade" sul fianco.

A migliaia di chilometri di distanza la statua di Yuri Gagarin (Colonna sonora: "Starman" di David Bowie) svetta impavida nel cortile del Museo delle Realizzazioni Sovietiche del Cosmo. Lo sguardo terrorizzato sembra urlare: *"Cosa? Voi Mi Volete Infilare Lì Dentro E Sparare Tra Le Stelle??"*

Alla fine lascio tutto come sta e torno indietro, cioè avanti, insomma a questo esatto momento. Quel che è fatto è fatto, mi disse una volta il mio maestro di Viaggio Nel Tempo.

E ancora:

```
Ho comprato un contenitore per computer ATX e una piastra madre ATX e
ho messo la piastra madre nel relativo contenitore (detto
anglofonicamente "case"). Le due, guarda un po', non combaciano. Non
finirà mai? Non si metteranno mai d'accordo?
```

A complicare ulteriormente il quadro contribuisce il fatto che, sulla lunga strada che porta ad essere un Vero Hacker, prima o poi la *Sindrome dell'Ultima Versione* se la prendono tutti[51]. Si consideri ad esempio il seguente frammento, ad opera di Babele Dunnit:

```
"È pronta la Versione 74 di Uorrd!" urla il Waz. "Fa anche la pasta! E
i pizzoccheri! Per il Negroni ci vuole un'interfaccia particolare, uno
shaker montato su un plotter, ma fa anche quello!". Mi chiedo a cosa
serva tutto questo nell'ambito del Word Processing, poi mi ricordo
degli Insegnamenti del Reverendo, il mio Maestro di Programmazione
Zen: "Il codice ASCII come lingua adamitica. I cartoons della Warner
Brothers come fulgido esempio di schema narrativo non-lineare. Il
futuro nei caratteri diacritici".
```

La legge di Murphy in effetti dichiara: "There is Always One More Bug", c'è sempre un Altro Baco. Ma il concetto è estensibile a "There is Always One More Version"[52]. Come

[51]Ed è *esattamente* ciò che Loro vogliono: sviare menti potenti obbligandole a cambiare il pannolino al computer due volte al giorno anziché permettere a sconosciuti, brillanti pippolatori di scrivere applicazioni killer in grado di sovvertire l'ordine costituito. Google, Skype, Ebay, Flickr... avete idea di *quanti soldi* certe intuizioni possano portare via a IBM, a Microsoft e agli altri colossi dell'industria informatica? Questo è quello che Joel Spolsky (http://www.joelonsoftware.com) chiama "Sparare e Muoversi". Cambiare *tutto* ogni sei mesi serve solo a fare in modo che un sacco di smanettoni impazziscano nel tentativo di tenersi aggiornati senza poter pensare ai propri prodotti, a nuove geniali funzionalità da implementare e alle cose davvero importanti da fare.

[52] Non si può non citare lo schema di numerazione delle versioni utilizzato da TeX, il linguaggio di rendering per documenti scientifici del Padre Fondatore Donald Knuth: a partire dalla versione 3, ogni nuovo rilascio è stato numerato aggiungendo una nuova cifra decimale in modo da avvicinarsi asintoticamente a Pi Greco. Questo indica anche che TeX è molto stabile: si prevedono solo modifiche di poco conto e, come dice lo stesso Knuth, il "cambio definitivo (da farsi dopo la mia morte)" sarà

salvarsi? Ecco la parola finale di Babele Dunnit sul tema scottante dell'evoluzione del software[53]:

> Io credo nel Grande Sistema Operativo Che Muove l'Universo, del quale Wizzo è la più infima delle implementazioni. È questa visione induistico-informatica che mi aiuta a sperare in un futuro migliore. Verrà il giorno in cui non sarà necessario passare l'ottanta per cento del proprio tempo a far funzionare queste Macchine Infernali e il venti a usarle davvero. Dopo un processo quasi infinito di debuggamento e test (detto Software Samsara 1.0, 1.01, 1.01.01, ..., 1.1, ..., 2.0, ... etc) il Software raggiungerà il suo Nirvana, diventando improvvisamente affidabile.
>
> ***Ed In Quel Giorno Tutto Funzionerà.***

Insomma, meno male che ci sono gli *standard*. Ognuno ha i suoi, ovviamente[54].

## Un Baco Per Ogni Cosa, Ogni Cosa Ha Il Suo Baco

Non pensiate di essere al sicuro dal *baco compatibilorum* solo perché usate poco il computer: *qualsiasi* dispositivo elettronico ne è affetto. Non vi è mai capitato di ricevere dei messaggi SMS assolutamente incomprensibili sul vostro cellulare? Lunghe sfilze di caratteri senza senso, più o meno ripetuti? Bene, si tratta di ASCII Art[55] composta su un cellulare dal display dotato di un ben preciso numero di righe e colonne. Chiaramente lo stesso messaggio visto su un cellulare diverso, con un display non identico a quello sul quale il messaggio è stato composto, renderà quest'ultimo assolutamente illeggibile... ecco una simpatica faccina composta su un display largo otto caratteri:

```
12345678

 \\!//
  o o
 | ^ |
  \-/
   |
```

Quando questo disegno viene trasmesso da un cellulare all'altro, viene implicitamente trasformato in una sequenza di caratteri, ovvero una riga singola (si notino gli spazi):

confermare come numero di versione il simbolo "π", al che tutti i rimanenti bachi diventeranno *features*.

[53] Si noti la Regola dell'80-20.
[54] Ringraziamo il Reverendo Pampamoon per questa geniale precisazione.
[55] Per ASCII Art si intendono quelle immagini composte utilizzando caratteri semigrafici, ovvero appartenenti appunto al set più o meno standard dei caratteri ASCII.

```
\\!//    o o    | ^ |    \-/      |
```

Cosa succede quando viene ricevuto da un cellulare che ha un display con una larghezza diversa da otto caratteri? Ad esempio, un display da sei caratteri? Ecco qui:

```
123456

   \\!
//
o o
 | ^ |
    \-
/
 |
```

La simpatica faccina già inizialmente lasciava a desiderare come espressività, ma dopo il trattamento sembra inaugurare un nuovo periodo artistico in qualche modo analogo al futurismo. Non siamo comunque certi che Boccioni XXX fosse un fanatico degli SMS.

Come difendersi da questi onnipresenti problemi di compatibilità? Difficile trovare una regola aurea. Il lettore tenga presente che in questi casi l'unica via di fuga, l'unica speranza di uscire vincitori senza passare i classici due giorni compiendo rituali informatici casuali e dal sapore smaccatamente alchemico[56] è rappresentata dalla *conoscenza storica*, il che è effettivamente poco incoraggiante se avete iniziato a smanacciare con un computer in tempi relativamente recenti. Sapere cosa sia successo in questi ultimi venti anni e perché si sia arrivati al punto in cui Uorrd per CiccioIcsPi e IcsPress per MacOssix non si parlano neanche a martellate – anche se te lo fanno credere esportando amichevolmente documenti uno nel formato dell'altro – equivale a decifrare la Stele di Rosetta. È un po' come interpretare quelle frasi assolutamente incomprensibili di brochure e manuali che abbiamo imparato ad amare:

---

CANALE STATO CIVILE: mostrare corrente VIDEOREGISTRATORE preselezione canali. Premere DISPLAY bottone fino a canale numero apparire.[57]

[56] Disinstalla e Reinstalla - Esci e Rientra - Spegni e Riaccendi – Cambia il Cavo – Apri e Chiudi – Sposta il Ponticello – Sostituisci il Connettore etc.

[57] Dal manuale del videoregistratore Sharp VC-MH770SM, pagina I-16. Solitamente è possibile venire a capo di questi deliri linguistici solo traducendoli all'indietro e analizzando l'errore originario. Forse che il traduttore fosse di natura digitale? Con le traduzioni automatiche, e quindi con l'Intelligenza Artificiale, bisogna stare attenti. Le IA, anche e soprattutto per la difficoltà insita nel definire il concetto stesso di Intelligenza, raramente fanno esattamente quello che ci aspettiamo... tanto che questa caratteristica è stata usata proprio come definizione della disciplina stessa: "tutto quello che, in campo informatico, non è ancora stato fatto". La visione di Jaron Lanier è ancora più radicale: "non è possibile distinguere l'intelligenza delle macchine dalla stupidità umana". Ovvero, nel tenere a bada la Stramaledetta Graffetta Di Word che vuol correggere (sbagliando) quello che scriviamo, è lei ad essere non abbastanza intelligente oppure noi un po' troppo stupidi nell'insistere a tenercela tra i piedi?

oppure quelle, molto più divertenti, dovute ad eccesso di zelo... eccone una, attribuita ad un oscuro genio della linguistica di IBM, che così spiegava cosa fare qualora i primi mouse meccanici  forniti con quegli ormai storici PC non funzionassero a dovere:

Le palle dei topi sono da oggi disponibili come ricambio. Se il topo funziona a scatti, è possibile che abbia bisogno di una palla di ricambio.

Insomma, l'Archeologia Informatica è spesso la Via migliore per trovare delle risposte – oltre al sistema più usato che è quello di utilizzare dadi icosaedrici.

*[DIY: dado icosaedrico da costruire, magari con delle facce di bug al posto dei numeri, oppure malfunzionamenti vari oppure consigli tipo esci e rientra, controlla il cavo etc]*

1. spegni e riaccendi il computer
2. controlla il cavo di alimentazione
3. esci e rientra dal programma
4. disinstalla e reinstalla il programma
5. cambia il mouse
6. aggiorna l'antivirus
7. aggiorna il driver della scheda video
8. aggiorna il driver della scheda di rete
9. aggiorna il driver della scheda sonora
10. cerca su Internet se è già successo a qualcuno
11. controlla il cavo del disco
12. formatta tutto
13. resetta tutto
14. cambia la tastiera
15. compra un nuovo computer.
16. buon momento per una pizza o una birra o ancora meglio tutte e due.

Oppure con due dadi:

1. compra

2. aggiorna
3. controlla
4. reinstalla
5. resetta
6.

1. computer
2. disco
3. scheda

etc.

A questo punto dovremmo smettere di elencar problemi e iniziare a cercar risposte. É quindi il momento di introdurre un concetto fondamentale in questa nostra Ricerca della Soluzione:

> Internet è il Nostro Salvatore e Google[58] è il suo Profeta.

Nel caso particolare della conversione di documenti che angustia Trinity e Zimo una ricerca del tipo "convert files <nome della prima applicazione> <nome della seconda applicazione>" ci potrebbe fornire utili informazioni e magari anche l'indicazione di qualche programma dedicato a risolvere il nostro problema. Certo, bisogna sapere l'inglese, ma anche per guidare l'automobile ci vuole la patente, no?

Se ancora non siete convinti, ecco cosa si spinge a dire Remote su Google:

```
Hello, Aaron Brancotti,

Google è IL motore di ricerca, ormai questa cosa è assodata, quindi,
applicando un minimo di retorica...

Google è LO strumento per trovare informazioni sulla rete

La RETE è il contenitore della quasi totalità dello scibile umano.

Quindi, interrogando Google si interroga la quasi totalità dello
scibile umano.
```

[58] Almeno per ora, ma dopodomani potrebbe nascere un motore di ricerca ancora più efficiente. Si vedrà.

Chi detiene la quasi totalità dello scibile umano ha molte più probabilità di delineare scenari futuri rispetto a un mazzo di Tarocchi.

Possiamo quindi affermare che Google è UN ORACOLO.

Ti ho mai parlato dell'Indice di Popolarità Personale a Culo (IPPC)?

È basato su Google e rappresenta il rapporto tra l'incidenza del tuo nome o di una cosa rispetto al culo.

Come funziona?

Si va su Google, si digita : culo

Risultati 1 - 10 su circa 1.160.000 per culo. (0,08 secondi )

Si ritorna su Google, si digita : <vero nome di battesimo di Remote>

Risultati 1 - 10 su circa 5.190.000 per <vero nome di battesimo di Remote>.

Quindi 5.190.000 / 1.160.000 = 4.5
Ebbene, il mio solo nome è 4.5 volte più famoso del culo.

Ma se digiti : <vero nome e cognome di Remote>

Risultati 1 - 10 su circa 290.000 per <vero nome e cognome di Remote>

290.000 / 1.160.000 = 0.25

Io personalmente sono 4 volte meno famoso del culo.

Credi non sia attendibile?

Risultati 1 - 10 su circa 13.200.000 per Bill Gates.
IPPC = 11.4

Risultati 1 - 10 su circa 999.000 per Silvio Berlusconi
IPPC = 0.86

Risultati 1 - 10 su circa 35.600.000 per George Bush
IPPC = 30.7

Risultati 1 - 10 su circa 335 per Aaron Brancotti
IPPC = 0.000289

Caxxo Aaron, DEVI FARE QUALCOSA!!!!

## Patrimonio dell'Umanità

In verità parleremo poco di Internet, anche se il tema potrebbe permetterci di riempire libri e libri; proprio per questo, già altri lo hanno fatto[59]. Il concetto che ci limiteremo a rimarcare qui è che La Madre Di Tutte Le Reti, in quelle manifestazioni che vanno dal WWW alle mailing list fino ai forum e alla possibilità di sviluppo collaborativo che offre, è una risorsa pressoché inesauribile di munizioni. Quindi, è Nostro Sacro Dovere mantenere pulito questo mare digitale.

**Filtra La Cozza Virale**

A questo proposito, mantenere in buona efficienza il nostro computer è anche una forma di rispetto verso tutti gli altri ai quali siamo vicendevolmente collegati. Un virus che prende il controllo della nostra macchina può infettarne altre migliaia. Uno schifo. Insomma, come già detto è *assolutamente consigliato* munirsi dello stretto indispensabile quando si tiene una macchina attaccata alla Rete.

**Reti e Catene?**

Parlando di pulizia ed efficienza di Internet, non possiamo non citare una forma di infezione particolare: quella delle cosiddette "Catene di Sant'Antonio". Se è vero che "il linguaggio è un virus", allora la propagazione di messaggi il più delle volte tanto tristi e catastrofici quanto fasulli ha una sua giustificazione sociologica profonda. A cosa servono? A raccogliere indirizzi di posta elettronica da bersagliare successivamente con messaggi indesiderati, ovvero da "spammare". Come difendersi? Esistono numerosi "servizi antibufala" su Internet[60]; solitamente basta digitare "Antibufala <una parola tipica del messaggio del quale si vuore verificare la veridicità>" in Google per scoprire cose insospettabili. A dire il vero, spesso basta un pizzico di buonsenso: pensate davvero che Bill Gates vi manderà mai un assegno a casa? Suvvia!

Inoltre, una tanto trascurata quanto utile forma di rispetto nei confronti del Prossimo Digitale è utilizzare il campo BCC[61] per spedire un messaggio a più persone che tra loro non si conoscono. In questo modo i rispettivi indirizzi di posta elettronica non vengono propagati indiscriminatamente, la privacy è salva e lo spammer digrigna i denti.

Questa faccenda delle catene di Sant'Antonio digitali, se non altro, ha dato luogo a qualche simpatico lazzo:

| |
|---|
| Questo è un e-mail virus manuale. |

[59] Con esiti abbastanza variabili, in verità…
[60] Al momento della stesura di questo libro, merita la menzione di onore il Servizio Antibufala di Paolo Attivissimo su www.attivissimo.net.
[61] Blind Copy Carbon, nota nelle versioni italiane dei programmi di posta come Copia Carbone Nascosta (CCN)

Il suo ideatore non ha purtroppo ne' tempo, ne' idea di come sviluppare un vero virus. Quindi scegliete i primi 50 indirizzi dal vostro address book e inoltrate loro questo virus. Poi cancellate alcuni dati dalla system directory. In caso oggi sia venerdi' 13, formattate anche il vostro hard disk.

Grazie per la collaborazione.

Che, a dire il vero, circolò anche in versione meno politically-correct, ma decisamente più divertente:

In questo momento voi ricevuto il "virus albanese".

Siccome nella Albania noi non ha esperienza di softuer e programmazione, nostro virus albanese funziona su principio di fiducia e cooperazione. Allora, noi prega voi adesso cancella tutti i file di vostro ard disc e spedisce questo virus a tutti amici di vostra rubrica (prima che cancella tutto, pero).

Grazie per fiducia e cooperazione.

Se questo vi diverte, molte altre Catene le potete trovare, raccolte e dissezionate, sul sito di One Phoenix (http://www.fenice.org) XXX. Happy Chaining!

## I Virus

Parlando di Internet non possono non venirci in mente le Grandi Pestilenze del mondo informatico, alle quali la TV è tanto affezionata: I virus possono essere considerati dei Bug?

È una questione interessante. Prima di tutto, capendo cosa siano questi virus informatici e come funzionino non rischieremo la morte digitale ogni tre giorni. Inoltre i media a questo proposito affermano tali idiozie che non avere gli strumenti per riderne a crepapelle sarebbe come andare dal MacDonald di Chernobyl, mangiare le patatine fritte e farsi dare anche la maionese.

Allora: un virus è un programma, ovvero una sequenza di istruzioni. Come tutti gli altri. Fissiamo questo concetto. Un virus, per infettare il nostro amato scatolotto di latta, deve essere eseguito – come tutti gli altri. I virus possono essere scritti in molti linguaggi – dal

linguaggio macchina a linguaggi cosiddetti di "scripting", che mettono con molta[62] facilità a disposizione le risorse di un sistema informatico – ma per espletare lo scopo per il quale sono stati concepiti, sia esso semplicemente stampare una frase ingiuriosa sul video o formattare tutti i dischi rigidi dell'isolato, devono essere in qualche modo lanciati. Il fatto che ormai i nostri computer eseguano automaticamente un sacco di compiti e che quindi, potenzialmente, un virus possa usare questa caratteristica a proprio favore è un altro discorso. Ad esempio, ricevere un virus sotto forma di allegato di posta elettronica non è pericoloso: basta non eseguire l'allegato, non toccarlo e cancellare subito il messaggio dell'untore. Però un programmatore particolarmente abile e bastardo potrebbe scrivere un virus sapendo che alcuni computer, configurati in maniera non del tutto sicura, eseguono gli script nascosti in pagine HTML spedite per posta elettronica in maniera automatica e senza avvertire l'utente.

Questo implica una informazione interessantissima: è impossibile che un virus si nasconda in un file di dati puro – ad esempio una pagina HTML nuda e cruda o in un foglio di calcolo – ma d'altro canto molti programmi nascondono insospettabili capacità di automazione. Ovvero: con un *editor* che è capace *solo* di scrivere puro testo ASCII[63] è impossibile nascondere un virus nel testo o fare in modo che questo venga eseguito; d'altra parte moltissimi programmi di elaborazione testi, oltre a far quello per cui sono stati originariamente progettati[64], sono anche capaci di eseguire delle sequenze di comandi[65]. Queste possono, potenzialmente, essere dei virus e vengono salvate con il testo stesso[66].

---

[62] Forse troppa. Sta di fatto che le questioni di sicurezza intrinseche alle specifiche di ogni linguaggio sono sempre tra le più discusse caratteristiche, proprio per l'uso potenzialmente sconsiderato che se ne potrebbe fare.

[63] Per ASCII si intende un codice che associa ogni carattere alfanumerico della tastiera di un computer ad un numero ben preciso e che permette quindi, in ultima analisi, di scambiare files di testo tra computer di marche diverse. Mentre i primi 128 caratteri sono determinati in maniera univoca (ad esempio, ormai su tutti i computer del mondo alla lettera "A" maiuscola è associato il numero 65), i successivi sono tutt'altro che standardizzati. A complicare il tutto c'è che ormai il codice ASCII è vecchio e inadeguato, usando un solo byte per carattere e permettendo quindi un set di soli 256 caratteri diversi; ormai nello sviluppo di nuovo software viene utilizzato il codice Unicode, che utilizza invece due bytes – quindi un numero da 0 a 65535 – e che quindi offre abbastanza spazio per la maggior parte degli alfabeti del mondo. I word processor che scrivono e leggono puro codice ASCII sono solitamente chiamati *editor* e fanno parte dell'armamentario del Power User: UltraEdit, cEdit e il caro vecchio NotePad ne sono alcuni esempi.

[64] Ovvero dopo che sono stati contagiati dalla "Sindrome di Word", il Programma più Ciccione dell'Universo. Word ha una tal quantità di opzioni inutili da oscurare anche la fama di SendMail, mitico programma UNIX di gestione EMail che opportunamente istruito, secondo Il Riglio, "fa anche la pizza". Addirittura, alcune versioni di Word hanno al loro interno... un flipper! Questi sono tra gli scherzetti che i programmatori burloni si divertono a fare e sono meno rari di quanto si pensi. Vengono chiamati, in gergo, *Easter Eggs*, ovvero "Uova di Pasqua".

[65] Solitamente chiamate *Macro* e normalmente utilizzate per riunire in un unico comando una serie di operazioni altrimenti ripetitive e tediose. Ad esempio, andare a capo e inserire il carattere "-" su 3000 righe di testo può necessitare ore a "mano libera", ma solo pochi secondi con una macro composta di tre o quattro comandi. Sicuramente comodo, ma potenzialmente devastante se il set di comandi utilizzabili in una macro si estende fino a poter creare o distruggere files, directories e interi dischi...

[66] Tra i files "puro ASCII" ci sono quelli con estensione TXT, BAT, LOG, INI e quasi tutti i sorgenti di programmi scritti in C, Java, C++, Python e centinaia di altri – ma torneremo presto su questa faccenda dei linguaggi di programmazione, purtroppo per il lettore. Gli stessi files HTML, RTF, XML sono di puro testo

Non è finita qui: sono anche comparsi dei virus che sembrano violare quanto appena affermato, poiché si propagano attraverso immagini JPG. In realtà è una variante di un vecchio trucco chiamato *buffer overflow*, dove il computer si aspetta una sequenza di dati che rispetti certe regole e ne viene fornita una lievemente diversa, tipicamente troppo lunga per essere gestita correttamente. In questi casi – che comunque si basano su un baco del programma che dovrebbe interpretare i dati puri in questione – è possibile eseguire codice arbitrario e quindi un virus o un umano malintenzionato in una situazione del genere ci sguazzano. Come dire: per *visualizzare* una immagine bisogna eseguire del codice che interpreta una serie di numeri e li trasforma in puntini colorati. Se quel codice non è scritto in maniera più che blindata, è possibile farlo impazzire dandogli in pasto dei dati indigesti e fargli fare quel che si vuole. Incredibile ma è così[67].

Comunque, un computer correttamente configurato e un programma con tutti i bit al suo posto dovrebbero avvertire l'utente che in un file di dati è stata trovata una qualsiasi diavoleria eseguibile e dovrebbero chiedere se la si voglia eseguire o meno. Ma molto spesso non si ha la volontà o il tempo per decifrare messaggi di allerta, magari sibillini e/o addirittura scritti in inglese…

*In order to read the rest of this book you must download and run an ActiveX Control to overwrite the FAT. Would you like to proceed?*

[inserire qui tre bottoni YES-NO-CANCEL]

Il lettore sarebbe disposto a scaricare sul suo computer un controllo ActiveX che sovrascrive la FAT pur di andare avanti a leggere questo libro? Ehi, chi sa *esattamente* cosa sia una FAT[68]? E un controllo ActiveX? Il punto è che un utente medio, abituato ad usare il computer come un normale elettrodomestico, non si aspetta niente di "cattivo", e clicca *Yes*.

Per citare il famoso analista americano Paul S.Boreele, "Pensiamo veramente che si possa essere utenti di Internet[69] senza conoscerne minimamente i meccanismi? Il phon funziona con l'elettricità e non conviene usarlo mentre sei nella vasca da bagno. Questo lo sanno tutti". Eh già. Ci possiamo schiantare allegramente con automobili di serie che viaggiano ben oltre i 200 Km/h, possiamo comprare armi in maniera più o meno legale per

anche se, aperti con gli opportuni programmi, mostrano testo formattato nei modi più disparati. Ovvero: non bisogna confondere il dato con il programma che *visualizza* quel dato...

[67] Il problema del buffer overflow è tra i più amati dagli informatici malintenzionati, poiché può presentarsi (e si è effettivamente presentato) praticamente ovunque. Chi si ricorda il buffer overflow di una delle prime implementazioni dello stack TCP/IP di MicroZgrunt?

[68] La File Allocation Table (FAT) è una zona dell'HD dove il computer tiene le informazioni su come e dove i vostri files vengono memorizzati fisicamente. Se si danneggia, potete dire addio ai vostri dati.

[69] O, anche molto più generalmente, utenti di un computer.

ammazzarci l'un l'altro, ma per qualche motivo l'utente informatico medio ritiene che *il compiuter* dovrebbe in qualche modo funzionare sempre e comunque in un ambiente protetto, praticamente innocuo e a prova di idiota. Purtroppo non è così, anche se – ad onor del vero – in molti stanno provando a rendere l'Informatica più accessibile. Solo che la cosa è molto, molto complessa. Molto più di un phon, un'automobile o una pistola[70].

Una seconda caratteristica dei virus è che sono capaci di propagarsi. I meccanismi mediante i quali si riproducono sono molteplici: ad esempio, un virus può essere in grado di "attaccarsi" ad un altro programma. In questo modo, quando il "portatore" viene eseguito, il virus ha la possibilità di copiare se stesso incollandosi ad altri programmi del computer ospite. Quando l'accesso a Internet non era diffuso come lo è attualmente, questo era il meccanismo classico di diffusione dei virus – anche perché non ne esistevano molti altri. Per passare da un computer ad un altro il virus doveva necessariamente usare il procedimento di scambio di programmi attraverso i floppy disk. Ora per un virus è molto più semplice, anche grazie a linguaggi, servizi e sistemi operativi nuovi e molto più potenti, andare a spulciare l'agenda degli indirizzi e spedire se stesso per posta, magari camuffandosi con qualche trucco che induca il destinatario ad eseguire l'allegato ricevuto dal presunto amico.

Una terza caratteristica, questa volta positiva per noi, è che i virus risentono delle stesse limitazioni che tutto il software ha: in particolare la poca compatibilità tra sistemi diversi. È *molto* difficile scrivere un virus che funzioni su qualsiasi tipo di computer. Tipicamente i virus che attaccano macchine Windows non hanno alcun effetto su sistemi Linux o Apple o altro, poiché ognuno di questi sistemi non è in grado di eseguire dei programmi pensati per le altre piattaforme[71].

Allora possiamo considerare questi Virus dei bug? Se consideriamo che, solitamente, un virus provoca un malfunzionamento del nostro computer certamente la risposta è positiva; d'altro canto, visto che lo scopo del virus è proprio quello di romperci le scatole nere, non possiamo fare altro che far idealmente i complimenti a denti stretti agli autori di questi piccoli capolavori della Ars Programmatoria prima di passare al contrattacco.

In particolare, non dovrebbero mancare un *Antivirus* e un *Firewall* all'arsenale dell'Utente Autocosciente. Entrambi si trovano anche in versioni a basso costo se non addirittura gratuite – più avanti parleremo di *utilities*, di software gratuito e di qualità dello stesso dando alcune utili indicazioni – e servono il primo a bloccare qualsiasi

[70] Infatti phon, automobili e pistole fanno una sola cosa e la fanno bene. Qui probabilmente bisognerebbe parlare del concetto di "computer invisibile", di David Norman, di "infodomestici" e del fatto che, molto probabilmente, l'errore risieda proprio nel tentare di usare *il computer* – ovvero un dispositivo universale e quindi intrinsecamente complesso - per fare una serie di operazioni teoricamente semplici ma molto diverse una dall'altra. Rimandiamo gli interessati alla lettura di XXX.

[71] In realtà un programma scritto nel linguaggio macchina di un certo processore può essere teoricamente eseguito su qualsiasi dispositivo che abbia quel processore, indipendentemente da tutto il resto e in particolare dal Sistema Operativo. Però un virus deve dare per scontati alcuni aspetti tipici del SO ospite, ad esempio la struttura di memorizzazione dei files o il formato di un file eseguibile, al variare dei quali il virus perderà in varia misura le sue funzionalità. Nondimeno, come già accennato, ci sono ormai linguaggi e applicazioni multipiattaforma; questo apre la strada anche a ipotetici virus in grado di propagarsi attraverso sistemi diversi anche a livello di hardware. Sempre più difficile, insomma...

tentativo di interazione con programmi e dati sospetti e il secondo a filtrare il traffico di rete da e verso il nostro computer, chiedendo esplicitamente la nostra autorizzazione ogni volta che "qualcosa" cerca di connettersi a noi o viceversa.

**La Spia Che Veniva Dal Sito**

Non si può non parlare a questo punto dei cugini più sfigati dei Virus: i cosiddetti *SpyWare*. Non hanno nulla a che fare con James Bond: sono, molto più prosaicamente, piccoli programmi che raccolgono informazioni dal nostro computer e le spediscono via rete, con lo scopo di informare qualcuno delle nostre abitudini informatiche. Anche se non avete lasciato dentro *My Documents* il numero della vostra carta di credito e senza stare a scomodare il Grande Fratello[72] sarete d'accordo che non è bello che qualcuno frughi tra le vostre cose. Così, come è opportuno avere un buon antivirus, è auspicabile anche installare un analogo antispyware. Ancora una volta, ne esistono di tutti i tipi e misure, comprese versioni gratuite perfettamente adeguate all'utilizzo casalingo[73].

*[Disegno di virus, bug, spyware e altre schifezze]*

Bene. Dopo questo riscaldamento introduttivo è finalmente arrivato il momento di affrontare frontalmente il nemico, cercando di capire cosa sia il software e come questo venga prodotto onde continuare con l'opera di totale disorientamento del lettore e procedere nel lungo e difficile cammino che ci porterà all'Illuminazione.

## Sistema Operativo e Software Applicativo

Possiamo classificare il Software in due grandi famiglie: *Software di Base* e *Software Applicativo*.

Il software di base è quello che non può che essere classificato che come un pezzo di *Sistema Operativo*. Il software applicativo è tutto il resto. Entrambi sono bacati per definizione e concorrono a non fare funzionare correttamente il nostro computer.

Del primo abbiamo già parlato affrettatamente poco sopra, del secondo abbiamo solo intuito l'esistenza quando si parlava di come, a volte, il nostro computer-meraviglia da N gozillioni di operazioni al secondo faccia fatica a trasformarsi in una banale – che poi tanto banale non è – macchina per scrivere. Come funziona quindi la faccenda? Più o meno così: all'inizio c'è il nulla, o quantomeno un ammasso di roba elettronica, il che non fa molta differenza. Provate ad assaggiare un elettrone e ne converrete. Per fare più o

[72] Quello di Orwell, si intende.
[73] Una vera miniera di consigli a proposito di virus, spyware e salute generale del PC è il libro di Paolo Attivissimo "L'acchiappavirus", edito da Apogeo e caldamente raccomandato da Babele Dunnit.

meno qualsiasi operazione con questa roba elettronica ci vuole una certa considerevole massa di istruzioni – sulla cui natura torneremo tra poco – che indichi all'ammasso di roba elettronica[74] come accendere e spegnere varie sue componenti affinché vengano eseguite alcune operazioni fisiche. Ad esempio, con che frequenza accendere e spegnere un laser e con che velocità fare girare un motorino per riuscire a leggere i microscopici buchi di un CD, che poi è abbastanza simile a far girare un motorino e muovere un piccolo braccio meccanico sul quale è montata una microscopica testina in grado di capire da che parte son girati i domini magnetici[75] sulle varie porzioni di un disco rigido. Oppure in che modo inviare dei segnali su un filo in modo da scambiare dei dati con un computer che si trova dall'altra parte del filo, ammesso che dall'altra parte del filo ci sia un computer e non il terribile Kovolvoorok, predone spaziale di rara cattiveria[76].

Ok. Sto divagando. Se iniziamo da così lontano non la finiamo più.

Allora: le istruzioni di cui sopra, che in ultima analisi dicono ad alcune parti del computer[77] come far funzionare altre parti sono molto, molto vicine all'hardware e si chiamano *driver*. È intuibile come, affinché l'hardware del nostro computer funzioni correttamente, siano necessari i giusti driver e come ad ogni minima modifica di un dispositivo hardware da parte del costruttore sia solitamente necessaria anche una modifica del relativo driver.

Ecco infatti come il Reverendo definisce il concetto di driver:

> Componente software dedicato che mette in sinergia i difetti di un computer con le imperfezioni di altri dispositivi elettronici. Di **driver** ce ne sono sempre cento e, come a tombola, quello che viene estratto non va bene per la vostra scheda.

Spesso chi costruisce l'hardware sviluppa anche il software. Questo fatto ha due forti controindicazioni: la prima è che spesso è difficile fare bene due cose allo stesso tempo[78]

[74] O meglio a quella parte dell'ammasso di roba elettronica in grado di eseguire sequenze di istruzioni. Il Lettore Più Attento riconoscerà in questo identikit la ormai famosa CPU.

[75] Il concetto è più o meno lo stesso del magnetofono: le variazioni di magnetizzazione di un supporto sono in grado di immagazzinare informazioni. Là era musica, qui sono uni e zeri, e un "dominio magnetico" è la più piccola area di materiale magnetico necessaria a immagazzinare un numero.

[76] Se dall'altro lato del filo c'è davvero il terribile Kovolvoorok voi e il vostro computer siete nei guai. Se c'è una cosa che il terribile Kovolvoorok non sopporta è essere preso a colpi di scarica elettrica. Essendo composto di impalpabile polvere magnetica, il terribile Kovolvoorok quando viene sottoposto a tensioni maggiori di pochi Volt praticamente si atomizza sparpagliandosi tutto intorno per centinaia e centinaia di chilometri.

[77] La CPU, certo, ma anche componenti dedicati a lavori specifici come grafica tridimensionale interattiva iperquaternionica avanzata, suono polifonico poliritmico poliedrico o i nuovi chip per la lettura del pensiero che vanno tanto di moda nei salotti della Andromeda bene.

[78] Questo è un concetto molto Samurai. Ad esempio, mai mangiare e camminare contemporaneamente. Avete mai provato a mangiare un Buondì in venti passi? Provate.

e la seconda è che questo è un ottimo escamotage per non rendere pubbliche le caratteristiche hardware di un certo prodotto, il che taglia le gambe a chi magari sarebbe capace di scrivere dei driver migliori per quell'hardware[79].

Questa storia dei driver è anche una metafora della vita: se vostro nipote disdegna la matematica o la storia, probabilmente ha dei driver vecchi. Cambiateglieli. Ma questo è un altro discorso.

Insomma, dovrebbe essere a questo punto ben visibile il ginepraio nel quale stiamo per infilarci in maniera del tutto cosciente e consenziente. Solo dei pazzi potrebbero pensare che un simile, complicatissimo sistema possa funzionare senza degenerare in una inestricabile PDF[80] di driver sbagliati tra loro pressoché identici e hardware misteriosamente malfunzionante dalla quale anche un Grande Maestro uscirà con le ossa rotte.

Infatti siamo dei pazzi: la faccenda funziona proprio come appena descritta. Non solo: questa è solo la punta di un enorme *iceberg* di letame. In effetti, sopra ai servizi basilari che questi driver forniscono, vengono costruite altre considerevoli masse di istruzioni che aumentano le capacità e le potenzialità del nostro computer, dando per scontato che tutto quello che c'è sotto funzioni (!) e astraendo sempre di più il funzionamento del computer dall'hardware che effettivamente c'è sotto la scrivania – o dall'altra parte del cavo[81]. Dopo innumerevoli operazioni di occultamento dei livelli più bassi di questi insiemi di istruzioni si inizia quindi, ad esempio, a parlare di *file* senza più preoccuparsi di tutti i movimenti che il braccino dell'hard disk deve fare per leggere tutti quei dati, o addirittura senza chiedersi se il file risieda su un disco rigido o su un CD – domanda invece lecita se il file non vogliamo leggerlo ma scriverlo pur non disponendo di un masterizzatore[82]. Ecco quindi che iniziamo a parlare di protocolli, di periferiche virtuali, di processi e thread e altri magici impalpabili concetti che aleggiano intorno a noi in maniera indipendente dall'hardware e che non funzionano mai come dovrebbero. Ecco. La magia si è compiuta ancora una volta: l'insieme di tutte queste considerevoli masse di istruzioni che, con effetto cipolla, ci portano da un duro nocciolo[83] hardware di motorini, braccini meccanici, testine magnetiche, cavi e misteriosi componenti elettronici fino ad un mondo etereo e quasi ideale di files e icone[84] è lo stramaledetto Sistema Operativo.

---

[79] Paradossalmente, a volte c'è chi i driver li scrive anche in barba ai costruttori di hardware e alla loro incapacità nello sviluppo del software. Parleremo più avanti di questa razza di sviluppatori alieni.

[80] Palla Di Fango, da non confondere con i documenti scritti con Adobe Acrobat. Architettura verso la quale qualsiasi struttura complessa tende.

[81] Sempre sperando che, almeno questa volta, dall'altra parte del cavo ci sia un computer e non il terribile Kovolvoorok.

[82] È comunque possibile scrivere un file su un CD anche senza masterizzatore, ovvero senza un dispositivo hardware con un laser abbastanza potente da bucherellare in maniera opportuna un supporto CD vergine. Però ci vuole un microscopio elettronico, uno scalpello finissimo e molta, molta pazienza.

[83] La parola "nocciolo" non è qui utilizzata casualmente. Il nucleo più interno del Sistema Operativo viene chiamato *Kernel*, che vuol dire proprio…?

[84] Il motivo per cui si è deciso di utilizzare delle "icone", ovvero delle immagini sacre, per visualizzare le informazioni a livello di Sistema Operativo è che il fatto che lo stesso funzioni è il più delle volte un

È ovvio che di sistemi operativi ce ne sono di tutti i gusti, tipi e misure, poiché in qualche misura rispecchiano l'hardware sul quale girano, così come i cani e i loro padroni che si assomigliano in maniera inquietante:

- Quelli ridotti all'osso dei telefoni cellulari[85] e dei sistemi a controllo numerico come torni, frese, frullatori programmabili e levigatrici orbitali;
- Quelli orientati ai processi industriali o realtime come QNX;
- Quelli che girano sui dispositivi palmari come Symbian o PalmOS o Windows CE, che sono a volte versioni dimagrite di altri SO più impegnativi;
- Gli innumerevoli prodotti Microsoft – che vanno dal vetusto DOS a Windows XP ed oltre passando per Windows 2.0, 3.0, 3.11, W95, W98, Millennium[86] Windows 2000 nelle varie versioni Workstation, Server, Advanced Server, 2003…ho perso il conto.
- La linea di sistemi operativi Apple, ovvero i vari MacOS, dei quali l'ultimo OS X è, a detta di molti, "Il miglior sistema operativo mai scritto";
- Il venerabile Unix in tutti i suoi *flavors*[87], il cui viaggio trentennale va da AT&T e SCO ad AIX a BSD, e finisce nel fenomenale e gratuito Linux, esso stesso disponibile in numerosissime distribuzioni e configurazioni: Mandrake, Red Hat, Slackware, SUSE, Dyne:bolic, Ubuntu, Gentoo...ne esce una alla settimana, impossibile riportarle tutte...
- Solaris, tipico dell'ambiente SUN e anch'esso derivato da Unix;
- Tanti altri che, pur avendo caratteristiche avanzate, hanno avuto meno fama e fortuna di altri: l'incredibile BeOS, ad esempio, o il TOS degli Atari ST, l'AmigaOS, il PS2...
- Tutti quelli morti, o quelli che resistono ancora sui mainframe che nessuno si sogna di mandare in pensione: gli IBM AS400 delle banche di tutto il mondo, il VMS dei mitologici VAX della Digital… e come dimenticare l'Exec8 di Sperry-Univac?
- Quelli che risorgono: come Haiku, ovvero "Beos Colpisce Ancora"...

Potremmo continuare per molto ad elencare Sistemi Operativi. Più avanti vedremo una situazione simile anche per i Linguaggi di Programmazione. Sarebbe una operazione alquanto sterile: ci basti ora sapere che esistono molti, molti Sistemi Operativi, ma che l'ambito dell'utilizzo della maggior parte degli stessi è molto circostanziato. La maggior parte dei lettori ne vedrà solo alcuni: qualche versione di Windows se avete un PC, qualche versione di MacOS se avete un Apple, qualcosa come PalmOS, Symbian o CE sulla vostra agendina elettronica, qualche versione di Linux se avete tempo e voglia di seminare oggi per raccogliere domani. La cosa da ricordare comunque è questa: i bug si annidano ovunque e *tutti* i Sistemi Operativi sono bacati, nonostante quelle assurde frasi da depliant come "libertà di avere i vostri contenuti sottomano sempre e ovunque" o miracolo.

[85] Anche se ormai ci sono telefonini che hanno più potenza di calcolo di quanta, onestamente, possa davvero servire.

[86] Sicuramente il SO più bacato di tutti i tempi, anello di congiunzione tra W98 e W2000 in grado di raccoglierne tutti i difetti e quasi nessun pregio. Con Millennium non funzionava *niente*.

[87] Letteralmente: *sapori*. Unix è nato originariamente in casa AT&T, ma le sue innegabili doti hanno convinto molti a riscriverlo, apportando le modifiche del caso.

"potete fare tutto quello che volete". Se poi alcuni Sistemi Operativi sono più bacati di altri, sarà questione di attitudine. Almeno pensateci.

**I Glurti e le Flobbe**

È arrivato il momento di definire un po' meglio le considerevoli masse di istruzioni di cui sopra. Inizieremo trovando loro un nome acconcio. Nell'ambito di questa trattazione potremmo benissimo chiamarli *Glurti* o *Flobbe*[88], tanto il lettore a questo punto è gia così confuso che per lui non farebbe alcuna differenza. Li chiameremo quindi *Programmi*, anche perché per andare avanti bisogna parlare di Programmi Applicativi e non di Glurti Applicativi o Flobbe Applicative, poiché più di un Maestro si risentirebbe.

Sopra al Sistema Operativo si appoggia quindi una classe completamente diversa di considerevoli masse di istruzioni – ooops, *Programmi* – che, a loro volta dando per scontati gli innumerevoli servizi astratti che il Sistema Operativo fornisce, trasformano il nostro computer – in senso funzionale, ovviamente – in qualcosa d'altro.

Quel *qualcosa* ha natura assai multiforme: si va dal programma per videoscrittura, astrazione di una vecchia macchina per scrivere, a concetti inesistenti prima dell'avvento del computer stesso, come i database, i fogli elettronici o i sistemi per la manipolazione delle immagini grazie ai quali ci divertiamo come dei matti a fare i baffi alla Zia Carmela dopo averla fotografata con una macchina fotografica digitale.

***Le Storie Di Babele: (4) L'Intrinseca Leggerezza del Digitale***

Zia Carmela: "ma non ha la pellicola?"
Babele Dunnit: "no, zia, è digitale."
Zia Carmela: "e cosa vuol dire che è digitale?"
Babele Dunnit: "che non ha la pellicola."
Zia Carmela: "e come fa a funzionare?"
Babele Dunnit: "Eh, è digitale…"
Zia Carmela: "e che vantaggi ha?"
Babele Dunnit: "che dopo ti disegno i baffi."
Zia Carmela: "ma io ce li ho già i baffi."
Babele Dunnit: "Allora te li (s)cancello."

Bene. Adesso il Lettore Più Curioso non potrà esimersi dal chiedersi come vengano create queste poderose sequenze di istruzioni che, eseguite a velocità inimmaginabile, trasformano uno statico ammasso di *hardware* inutile in un mutevole e velocissimo ammasso di *hardware* inutile. La risposta è molto articolata, ma il concetto di base è che questi programmi vengono creati mediante altri programmi.

[88] Anche se una flobba è tutta un'altra cosa. Chiedetelo al terribile Kovolvoorok.

Confusi? Come dice Jango Edwards, "Anche io sono confuso. Ma io sono pagato per esserlo"[89]. Certo non sfuggirà al Lettore Più Curioso la natura ricorsiva[90] di questa spiegazione. E' questa una situazione abbastanza comune nel mondo informatico, tanto che il processo di avvio di un computer viene chiamato *Bootstrap*, a imperituro ricordo del Barone di Munchausen che si arrampica nel vuoto riannodando all'infinito le stringhe dei propri stivali.[91] In effetti come fa un computer a dare il via correttamente al proprio funzionamento? L'hardware da solo non sa fare nulla; ci vorrebbe *qualcosa* che dica all'hardware di caricare il Sistema Operativo e quel *qualcosa* non può essere altro che un pezzo del Sistema Operativo stesso. Ohum. Per fare il software ci vuole il software. Per fare il tavolo ci vuole il legno. C'è chi dice che anche per fare il software ci voglia il legno. E se sei un autore di Realtà Virtuali[92] per fare il tavolo ci vuole il software. Ovvio.

C'è un'unica soluzione. Scrivere da qualche parte un piccolo, piccolissimo pezzo di sistema operativo il cui scopo è caricare la parte restante di se stesso. Inoltre dobbiamo far sì che questo piccolo programma non si cancelli quando spegniamo il computer e che venga eseguito automaticamente quando accendiamo il computer. Si può fare? Si, se la smettiamo di chiamarlo Piccolo Programma e gli diamo un nome decente, come ad esempio *loader*[93]. Se invece insisteremo a chiamarlo Piccolo Programma passeremo il

[89] Jango Edwards (al secolo Stanley Ted Edwards) , trentenne ed in preda a delirio di onnipotenza, lascia una vita peraltro abbastanza agiata per diventare clown, con tanto di naso finto, scarpe grosse e elemosina all'angolo della strada. Tra i suoi fan più sfegatati annovera re, nobili e politici, pittori come Salvador Dalì e registi come Coppola e Fellini, artisti come la Deneuve e musicisti come i Rolling Stones. Ogni suo spettacolo è adrenalina pura dall'inizio alla fine – quando saluta gli spettatori uno ad uno – e molte delle sue gag sono ormai da manuale. La chiusura storica dei suoi spettacoli, è: "Oggi è un giorno stupendo... perché *ogni giorno* è un giorno stupendo!"

[90] Si intende per "ricorsività" il fatto che la definizione di qualcosa faccia riferimento alla cosa stessa. Tipica in matematica è la definizione del fattoriale di un numero N, ovvero il prodotto di 1 per 2 per (etc etc) per N. Ad esempio, 5 fattoriale = 1 per 2 per 3 per 4 per 5 = 120. Magia: il fattoriale di N può essere definito ricorsivamente come N per il fattoriale di N – 1. Ovvero, il fattoriale di 5 è uguale a 5 per il fattoriale di 4. Ma il fattoriale di 4 è uguale a 4 per il fattoriale di 3, eccetera. Provare per credere. Se indichiamo il fattoriale con "!", abbiamo in simboli: N! = N*((N-1)!). Ovviamente, onde non ritrovarsi in un ciclo infinito, ci vuole una condizione di uscita (chiamata "base" della ricorsione), solitamente intuitiva e banale: in questo caso 1! = 1. Come si vede, il fattoriale è definito usando il concetto stesso di fattoriale: la definizione è quindi ricorsiva, così come è ricorsivo il concetto secondo il quale per creare dei programmi siano necessari altri programmi. E se proprio la matematica non la digerite, pensate all'uovo e alla gallina, mutuamente ricorsivi. Si veda anche la Nota 90 XXX updatare numero nota con questo stesso numero.

[91] Questo è quello che vi dirà Google cercando "bootstrap". Però Babele Dunnit ci fa sapere che, dopo aver scaricato da Internet l'opera citata – ehi, certo che è legale! Il "Barone" fa parte delle opere del "Progetto Gutemberg"! Cercare, cercare – una ricerca della frase attribuita al barone e riportata in centinaia di siti è risultata negativa. Addirittura non ci sarebbe traccia alcuna delle parole "boot", "strap", "bootstrap" etc. Interessante Leggenda Informatica? Vediamo cosa dice il Reverendo: "Il nesso con il letterale inglese è ovvio. Quando si accende un computer è opportuno indossare degli stivali, perché poi il resto dovrete farlo funzionare a calci". Se trovate esattamente l'opera nella quale il Barone si solleva per le stringhe, segnalatecela.

[92] La Realtà Virtuale, spesso designata con l'acronimo inglese VR, è una applicazione specializzata di informatica multimediale con grafica 3D in tempo reale, suoni spazializzati e complesse interazioni simulate volte a ingannare la propriocezione dell'utente e provocare nello stesso il cosiddetto *Suspension of Disbelieve*. Si dichiara che quanto scritto in questa nota ha senso.

[93] Da *to load*, caricare. Poco esotico e anzi abbastanza banale. Potremmo chiamarlo *glurto*, perché la *flobba* è tutta un'altra cosa.

resto della nostra carriera di aspiranti Gran Visir del Silicio pronunciando frasi come "Perché non parte? Maledetto Piccolo Programma Bastardo!".

Affinché il loader non raggiunga le Verdi Praterie del Software quando togliamo l'alimentazione, siamo costretti a scolpirlo su una pietra di fiume sotto forma di sequenza di magiche rune elfiche e ad incollarlo sul frontalino del nostro beneamato computer[94]. Questo implica che modificare un loader è una impresa lievemente più complicata di, diciamo, fare una capriola o mangiare un gelato panna e nocciola[95]. Questo implica anche che, se siete uno sviluppatore di sistemi operativi, prima o poi scriverete un loader bacato[96]. Per cui andate *ora* a fare delle capriole, prima che sia troppo tardi e allo scadere della mezzanotte il vostro computer si trasformi in una zucca.

## La Distruzione Dell'Informazione

E a proposito di tramutare, "Nulla si crea, nulla si distrugge, tutto si trasforma". Chissà se Anassagora[97], quando formulò questo concetto per l'epoca assolutamente rivoluzionario, sarebbe stato dello stesso avviso davanti ad un computer? Se è innegabile che l'Informazione stessa è soggetta a trasformazione continua, tanto che esistono intere discipline che studiano proprio questo processo, è altrettanto vero che l'Informazione si perde. Non solo: di solito si perde l'informazione *fondamentale*.

Questo concetto molto generico nonché bastardo si materializza nell'incubo informatico universalmente noto col nome di *Sindrome del Tasto Sbagliato*: cosa succede se si schiaccia Quel Tasto Là? Soprattutto, *esiste* un Tasto Sbagliato? Quel Tasto che, se premuto, cancella tutto il nostro lavoro, trasforma i nostri dischi in autostrade per Hackers Cattivi[98], pubblica i nostri documenti più segreti[99] e le foto della tessera del metrò su Internet e magari fa anche esplodere lo scaldabagno? Quel tasto che poi non si accende più il computer e ti si slacciano le scarpe mentre scendi le scale, che ha detto Mio Cuggino Che Dopo Tre Giorni Muori[100]? Il Tasto di Ragnarok, quello che invoca la Peste

[94] Un po' come Rabbi Loew ed il suo Golem, Yossel, col suo *shem* incastrato sulla fronte… Non è vero. Basta memorizzarlo su componenti elettronici appositi che, al contrario della memoria RAM del computer, nella quale possiamo sia leggere che scrivere, non sono modificabili – o quantomeno lo sono con difficoltà. Questo secondo tipo di memorie possono avere decine di nomi diversi, ma appartengono alla grande famiglia delle ROM, Read Only Memory, memorie di sola lettura. Casualmente tutti i componenti elettronici dei computer attuali sono basati sul silicio e quindi in ultima analisi programmare una ROM è quasi come scolpire un sasso di fiume, ma questo è un altro discorso.

[95] Ma fare le due cose contemporaneamente non è per nulla facile. Come dicevamo, un vero samurai fa sempre solo una cosa alla volta.

[96] A parità di numero e gravità di errori di programmazione (bug del software), questi si manifesteranno nelle parti di programma intrinsecamente più difficili da correggere.

[97] Si,si... di solito è attribuita a Lavoisier, ma vi assicuro che le ipotesi di conservazione della materia risalgono all'Antica Grecia. La versione di Lavoisier è, se vogliamo, una "estensione scientifica" di pensieri qualitativi presenti già da migliaia di anni nel sapere comune.

[98] Ancora con questi Hackers Cattivi? Niente paura, ci arriviamo... ci arriviamo a parlare di hacking...

[99] Anche se li abbiamo scritti solo a penna su un foglio di carta nascosto in fondo a un cassetto.

[100] Cfr. Elio E Le Storie Tese, "Mio Cuggino"

e le Cavallette e la Carestia sul processore? Insomma, il Tasto Il Cui Codice ASCII è 666?

```
"Nessuno può Inserire o Cancellare Secco in Tabella se non chi ha Login
e Password della Bestia Selvaggia o il Numero del Suo Tasto. Qui sta la
sapienza: chi ha l'algoritmo calcoli il Numero della Bestia Selvaggia,
poiché è un codice ASCII; e il suo numero è seicentosessantasei
(0x029A, 1010011010)"
                                                  Formattazione 13:17,18
```

**Il Tasto dell' Apocalisse**

In quanto esperti di Escatologia Informatica XXX possiamo sfatare un mito: no, non esiste Quel Tasto Là. Il concetto dello "schiacciare il tasto sbagliato" non ha senso. È però vero – e non potrebbe essere altrimenti – che sulla tastiera del nostro computer esistono dei tasti – o più in generale svariati metodi, procedure, manovre – il cui scopo è quello di cancellare *qualcosa*, la cui natura dipende dal contesto.

Ad esempio, non è una buona idea selezionare tutto il contenuto di un disco rigido e poi battere *uno di questi tasti* (il backspace, il delete – o sulla tastiera italiana il "canc"…), ma è una idea altrettanto cattiva quella di aprire il cofano della nostra automobile e cominciare a smontare e buttare tutto quello che vediamo, o quella di rovesciare l'intero contenuto del nostro portafoglio per strada. Non si capisce quindi perché dovremmo avere il timore di usare un computer "perché potrebbe succedere qualcosa di brutto"; è ovvio che, quando lavoriamo, dobbiamo *sempre* fare particolare attenzione quando ci accingiamo a compiere operazioni distruttive. In ogni caso, al contrario della maggior parte del resto dell'universo, il nostro computer ci offre molte opportunità e meccanismi per ritornare sui nostri passi e rimediare ad errori altrimenti fatali[101]: dal Cestino alle copie di backup, fino a programmi specifici tipo "Undelete" e "Disk Recovery" che vedremo più avanti.

**Il Cestino, Questo Incompreso**

ormai tutti i Sistemi Operativi hanno il concetto di "cestino", dove finiscono i nostri files prima di essere distrutti definitivamente. Ottima cosa, fino a quando la usiamo correttamente... lasciamo ancora una volta la parola a Rob:

[101] È famosa ad esempio la triste storia di Scibum Shallklll, il protomercante dalle cento mani di Zevullah IX, che dopo aver orribilmente sfregiato la moglie del sovrano Karh Gnarh Fruptx V (detto Il Faceto) con una crema antirughe troppo aggressiva esclamò "Whooops! Mi scusi, non ho fatto apposta" e ciononostante venne giustiziato.

...circa due anni fa, un tizio dell'ufficio commerciale ci chiama per un problema sul suo computer. Dopo una prima veloce verifica, decidiamo per un intervento più approfondito e ci portiamo via la macchina. Ovviamente come prima cosa svuotiamo il cestino, puliamo tutta la schifezza temporanea, verifichiamo il disco rigido, deframmentiamo... e troviamo che in effetti il disco aveva qualche problema che comunque viene risolto spianando e reinstallando. Il giorno dopo alla riconsegna del mezzo il tizio ci chiama allarmato:

"Non trovo più tutti i documenti importanti che avevo!".
"Ma come!", rispondo, "abbiamo salvato *tutte* le cartelle locali e *comunque* il tuo ufficio ha una cartella per le copie di sicurezza condivisa sul server!"

Ed ecco la devastante risposta:

*Ma io le cose importanti* ***le metto nel cestino!***

Ora, questo potrebbe essere solo il risultato di una pasticca andata di traverso il venerdì sera ad una persona stramba già di suo, ma la settimana scorsa una tipa di un ufficio lì di fianco ci chiama e ci dice "Avevo salvato una cosa importante nel cestino ma adesso non la trovo più!".

Che sia l'aria?

Non c'è molto da commentare: a casa vostra le cose importanti le mettete nel cestino della spazzatura?

**Il Backup, Questo Sconosciuto**

Nonostante sia una operazione assolutamente banale e, nella sua accezione più comune, anche veloce, la maggior parte degli utenti informatici anche relativamente esperti trascura sistematicamente di eseguire copie di sicurezza del proprio lavoro. Questo è gravissimo ed inaccettabile:

**Teorema di Non Infallibilità dei Computer**

Enunciato: I computer si rompono.

Dimostrazione: I computer sono Macchine. Le Macchine si rompono. Ergo, i computer si rompono. QED.

Corollario: I computer si rompono nel momento peggiore.

Il Lettore che non abbia l'abitudine di salvare i suoi dati in molteplice copia *deve imparare a memoria* il Teorema testé esposto e ripeterselo come un mantra tutte le sere prima di addormentarsi.

Già che ne parliamo, fare un backup non significa semplicemente "copiare qualcosa in un'altra cartella". Un backup, per avere senso, deve essere eseguito su un dispositivo fisico diverso rispetto a quello che usiamo solitamente, possibilmente situato anche in un luogo separato; se proprio non volete fare come Babele Dunnit, che in ogni suo computer ha due dischi rigidi, potete ricorrere a masterizzatori, dischi esterni, computer in rete o, interessante variazione sul tema, ad un account di posta dal quale non scaricate mai nulla e al quale accedete via web[102]. Se avete a disposizione solo hardware per sua natura poco affidabile come vecchi masterizzatori sporchi e non burn-proof, dischi ottici riscrivibili un po' rigati, dispositivi a nastro tipo ZipDrive o peggio ancora floppy disk è *necessario* verificare che la copia sia venuta bene, nel caso non utilizziate un software che già lo fa (come quelli specifici per il backup o i software di masterizzazione che hanno l'opzione *verify written data*).

Comunque la copia di file "a mano" è sempre meglio che niente, ma prona ad errori e dimenticanze e dovrebbe essere utilizzata solo occasionalmente, lasciando il compito di copiare i nostri dati a programmi e script appositi: provate, così per sport, ad elencare i motivi per cui la copia di un file potrebbe fallire. Se non vi vengono in mente più di due motivi, vivete tra le nuvole. Se ve ne vengono in mente almeno sei, siete già un po' più realisti. A Babele Dunnit una volta hanno fatto questa domanda e, dopo aver esclamato "Sarò Breve", ne ha elencati più di ottanta prima di venire azzoppato da uno del marketing[103].

Insomma, là fuori ci sono *centinaia* di programmi dedicati al concetto di copiare e spostare file da una parte all'altra, magari comprimendoli automaticamente, o salvandone le diverse versioni in maniera automatica e chissà cosa altro. Da XXCOPY, utility DOS che fa anche il pesto al SyncToy di Microsoft[104], dal Sync Wizard dell'Xplorer2 di www.zabkat.com al Backup standard, avete solo l'imbarazzo della scelta. Basta che usiate *qualcosa.*

[102] Ad esempio Gmail con i suoi vari Gb *gratuiti* a disposizione.
[103]Il tipo del marketing perse definitivamente le staffe quando Babele Dunnit ipotizzò, come ottantaseiesimo possibile motivo di errore di copia di file, il rosicchiamento di un cavo di connessione tra due computer ad opera di animaletti simili a topi ghiotti di gomma e rame chiamati "gnagnafili". A seguito del violento calcione, Babele Dunnit esclamò "Perchè Tanto Odio?" e a sua volta colpì l'antagonista con un *gumb-hard* (colpo di gomito duro).
[104] Si, avete letto bene. Incredibile ma vero, è un programma *molto* carino...

Per finire, ricordatevi che quando cancellate qualcosa dal vostro disco rigido, la copia di backup diventa automaticamente l'originale; checche ne dicano i produttori, *non è vero* che un CD "è per sempre" come i diamanti dell'anello di fidanzamento della pubblicità, quindi, se salvate le foto del vostro viaggio attraverso i Sette Pianeti della Concordia su un CD con l'intenzione di rimuoverle dal disco rigido in un secondo tempo, fatene subito due copie e verificatele[105].

**Corso Avanzato di Recupero Informazioni**

Se tutte le raccomandazioni precedenti non vi hanno salvato dalla debacle, dopo esservi cosparsi il capo di cenere potrete procedere con la lettura delle seguenti tecniche ordinate per valore crescente di esoterismo. Con un po' di fortuna sarete così in grado di salvarvi da situazioni a volte critiche.

Primadituttо tenete presente che esistono numerosi programmi in grado di fare cose incredibili con i vostri dischi rigidi. Addirittura, se vi rendete conto di aver cancellato – anche dal cestino – dei dati importanti, probabilmente potete recuperarli lo stesso con programmi di *undelete*, in grado di ricostruire il file partendo dai suoi pezzi e dalle informazioni ancora presenti nella FAT. Unica condizione è che ci proviate *subito*, prima di cancellare o modificare ulteriormente altri files, perché il Sistema Operativo, ritenendosi autorizzato a far quel che vuole dello spazio su disco che gli avete detto di riciclare, è pronto a distruggere tutto il vostro lavoro.

Una delle cose che fanno la differenza in battaglia è sapere quali azioni si possono tecnicamente intraprendere e quali no. Potrà sembrare strano, ma molto spesso quello che viene a mancare è proprio la fantasia. Il procedimento da seguire è quindi il seguente: una volta seduti nella Posizione del Loto dopo aver indossato il miglior kimono che avete, urlate "Yaaaah! Se Solo Fosse Possibile *<mettere QUI qualcosa di apparentemente impossibile>*" e poi chiedetevi se veramente quella cosa sia davvero così assurda. Si scoprirà che la maggior parte delle volte quello che si riteneva un problema insolubile è già stato affrontato e risolto da molti altri e che qualcuno ha addirittura scritto un'*utility* apposita, o quantomeno ha descritto problema e soluzione adottata. Insomma, davanti al Nemico ci sono volte in cui è più opportuno concentrarsi su *cosa* sia successo, cercando così una soluzione magari completamente fuori dagli schemi[106], piuttosto che cercare di capire *perché* sia successo[107].

Un esempio (preso da Internet e tradotto) di Darren Gillet[108] chiarirà il concetto:

[105] Esistono all'uopo anche numerosi programmi utili e gratuiti, rintracciabili su Google cercando qualcosa come "CD Verify Utility".
[106] Un *hack*, nella più pura, alta ed originaria accezione del termine.
[107] "Se non ha nessun senso ci risparmia molti fastidi, perché così non siamo costretti a cercarne uno" dice il Re di *Alice nel Paese delle Meraviglie*.
[108] Mai sentito prima e non si sa neanche se costui esista davvero o se sia una bufala inventata di sana pianta. Ma è il concetto che conta; quello che viene descritto è fantasioso, abbastanza plausibile e

"recentemente il disco rigido del mio computer ha iniziato a surriscaldarsi e si è apparentemente bruciato, portandosi via 30 Gigabytes di dati. La soluzione? Mantenendolo collegato al computer, l'ho chiuso in un sacchetto di plastica e l'ho tuffato in un cestello pieno di ghiaccio, che lo ha tenuto freddo abbastanza da permettermi di recuperare tutti i dati. Il trucco ha funzionato così bene che il disco (che era stato dichiarato "irrimediabilmente perduto" da un "esperto") mi ha restituito non solo tutti i dati, ma ho avuto anche abbastanza tempo per una seconda copia di backup."

Che dire? Davanti ad un simile colpo di genio, anche Babele Dunnit non può che complimentarsi. Questo signore è, indubbiamente, un hacker. Certamente giocarsi il tutto per tutto richiede mano ferma, ma in casi come questo non c'è altra soluzione e l'abilità/intuizione consiste nel capire quale sia *la cosa giusta* da fare. Se poi vogliamo citare ancora una volta l'Incredibile Episodio della Tastiera di Ashwà e parlare del Grande Bug, ricordiamoci che l'unico modo per evitare problemi estremi come questi è giocare di anticipo: mantenere sistematicamente un computer in buona efficienza spesso permette di scoprire con largo anticipo malfunzionamenti che altrimenti potrebbero far precipitare la situazione nel momento meno opportuno.

## Interazioni tra Hardware e Software

Quando un dispositivo meccanico non funziona bene – ad esempio si fa fatica a chiudere la serratura di casa – la prima cosa che viene in mente di fare è dargli il grasso. Solitamente questo risolve la faccenda.

Purtroppo questo non vale con i computer. Né per quanto riguarda l'hardware, tantomeno per il software: se esistesse il Grasso Per I Programmi, dovremmo spalmarne gran quantità su tutti i sistemi operativi e le applicazioni esistenti. Se a questo aggiungiamo il fatto che quando richiudi un computer ti avanzano sempre delle viti[109] avremo una vaga indicazione di quanto la situazione sia disperata.

Vediamo cosa dice Babele Dunnit a proposito delle interazioni tra HW e SW:

Colto da ispirazione provo ad inserire una piadina precotta nel masterizzatore CD e questa si scalda. La mangio. Quindi provo a masterizzare una seconda piada backuppandoci sopra un pezzo di disco rigido e quando la estraggo è bruciata. Bisogna quindi utilizzare una piadina cruda, oppure scriverci meno roba. Me ne procuro una dal fornaio sotto casa, riprovo e questa volta FUNZIONA! Ottengo una piada profumata, morbida e sfogliata al punto giusto sulla quale ci sono anche 600 e passa mega di backup.

pienamente esemplificativo di quanto stiamo dicendo.

[109] Anche perché il posto migliore ove cercare il cacciavite è solitamente il frigorifero.

E ancora, durante il mitico combattimento contro BleBlop, l'alieno-sonda robotico:

```
Sfido in extremis BleBlop ad una gara di logica contradditoria e
autoreferenziale, che di solito mette in crisi le Intelligenze
Artificiali e le blocca. Ho seguito un corso di Arti Marziali
Ricorsive e anni or sono feci esplodere un Symbolics col Gioco delle
Tre Carte.
```

Per la cronaca, ecco come finisce il combattimento:

```
Pulsazioni a centottanta mentre il disco rigido frulla come un
derviscio rotante fino a vaporizzarsi. Sanguino un po' a causa delle
schegge, ma posso farcela: senza neanche spegnerlo facendo lo shutdown
come bisognerebbe sempre fare per non rischiare il danneggiamento del
disco rigido, abbatto BleBlop con un bazooka regalatomi da Zia Carmela
proprio per simili casi di emergenza. Eseguo il numero in volo,
saltando sul letto come il bambino della pubblicità dei materassi. I
dischetti e le telecamere plaudono il gesto atletico, mentre l'ammasso
di ferraglia si schianta al suolo e compare il messaggio "Errore in
lettura sul Drive A, premere un tasto per riprovare".
```

Questo soliloquio evidentemente scaturito da una personalità provata ci serve perlomeno ad introdurre il prossimo argomento:

**Può il Software danneggiare l'Hardware?**

Certo che si. Provate un po' a tirare una scatola di Windows 2000 Advanced Server contro il monitor, e poi fatemi sapere.

Quella precedente è una battuta, ma ci sono intere aree applicative dove un bug software può provocare effetti devastanti: il software di controllo di processi industriali, ad esempio. Il compito di movimentare intere fabbriche, ivi comprese presse e argani in grado di spostare carichi da centinaia di tonnellate è critico, tanto che in questi casi non ci si fida completamente del software e i controlli di fine corsa – che impediscono ad esempio a un braccio robotico di demolire un capannone perché qualcuno ha confuso una X con una Y – vengono solitamente affiancati da controlli di emergenza sotto forma di interruttori in grado di fermare tutto l'universo[110].

Un altro spettacolare caso di interazione profonda si ha anche per alcuni tipi di turbina che anche quando sono a riposo non possono essere fermate completamente, perché basta il loro stesso peso, abbinato alle alte temperature che raggiungono, per piegarne irrimediabilmente l'asse di rotazione. Simili sistemi devono essere *sempre* tenuti in movimento.

[110] Questa è una regola d'oro dell'Automazione. L'importante è che l'Interruttore sia Sufficientemente Grande e A Portata Di Mano, o – come si diceva ai tempi dei PC AT, "Time To Reach The Big Red Switch".

In generale tutta la robotica, a causa della sua natura, è soggetta a simili problemi: quando vi proporranno di mettervi una armatura che moltiplica le vostre forze interpretando e assecondando i movimenti che desiderate fare – questo degli esoscheletri[111] è un concetto caro alla fantascienza e chiunque abbia visto Alien 2 ha capito di cosa stiamo parlando – chiedete prima che sistema operativo usa, chi ha scritto il software e soprattutto se chi lo ha debuggato è ancora vivo.

**Automobili Intelligenti?**

Senza andare troppo lontano nel tempo[112], le case automobilistiche hanno cominciato da tempo a montare di serie sulle loro automobili di punta numerosi gadget robotici come sensori di umidità che chiude automaticamente i finestrini quando inizia a piovere[113], chiavi elettroniche e addirittura concetti rubati ai sistemi operativi come quello di *Profilo Utente*, che nel caso di una automobile serve, in base a chi guida, a spostare sedili, volante, specchietti, assetto di guida e tutto quello che il progettista ha ritenuto opportuno servoassistere. È superfluo osservare come la stabilità del sistema operativo e l'assenza di bug in questo caso sia molto importante, onde evitare che la vostra automobile si trasformi in una giostra mentre guidate – a meno che questo serva a tenervi svegli, ovviamente.

Narra la leggenda che Bill Gates, anni fa, si sia presentato al COMDEX aprendo il suo discorso con la seguente affermazione: "Se la General Motors avesse sviluppato le proprie tecnologie ad una velocità analoga a quella dell'industria informatica, adesso guideremmo automobili da 25 dollari con una autonomia di 1000 miglia per gallone di benzina". Narra altresì la leggenda che la risposta di un portavoce della General Motors, qualche giorno dopo, sia stata: "Si, ma voi la vorreste una automobile che si schianta contro un muro almeno un paio di volte al giorno?"

Sembra che le possibili considerazioni non finiscano qui: come qualcuno notava su Internet tempo fa, bisognerebbe comprare una nuova automobile ad ogni rinfresco delle strisce pedonali a causa di misteriosi problemi di incompatibilità, riportati dal sistema di controllo[114]. Poi il nostro amato mezzo di trasporto si fermerebbe senza motivo in autostrada e noi dovremmo scendere e risalire e riaccendere facendo finta di nulla. Ci sarebbero da reinstallare i pistoni ogni pochi giorni scaricando nuove versioni da Internet, dovremmo andare in giro da soli a meno di comprare una licenza di guida multiutente[115], saremmo costretti ad avere tutti quanti il culo delle stesse dimensioni a causa dei sedili standard ed in caso di incidente non solo sarebbe estremamente difficile capire cosa è successo, ma l'airbag chiederebbe "Are You Sure? <Yes><No><Cancel>" prima di

[111] Masamune Shirow, autore di Ghost In The Shell – dal quale l'immaginario di Matrix è stato attinto a piene mani – li chiama *guges*.
[112] Ma chi ha detto che gli esoscheletri alla Alien siano una realtà lontana? Roba così a livello militare esiste già sicuramente, si tratta solo di abbassare i prezzi e ben presto potremo andare tutti a fare i boscaioli.
[113] Con grande sdegno dei fumatori accaniti che hanno l'abitudine di abbassare di qualche centimetro il finestrino e che adesso, quando piove, hanno una automobile che cerca di convincerli a smettere.
[114] Una singola spia rossa denominata "Malfunzionamento Generale del Veicolo"
[115] CarNT, Car2000, CarXP, ognuna incompatibile con le altre.

azionarsi. La Apple invece produrrebbe delle automobili bellissime, velocissime, sicurissime ed economiche usabili solo sul 5% delle strade mondiali[116].

Insomma, ci ritroveremo - a qualche decina di migliaia di utenti è successo - a dover utilizzare dei navigatori satellitari che, oltre al cognome, dei personaggi storici chiedono anche il nome di battesimo. Forza: come si chiamavano di nome Rossini, Puccini, Farini... ? Siamo sicuri che mettere *così tanto software* sulle automobili sia una buona idea?

**Casa, Dolce Casa**

Una delle Meraviglie Del Prossimo Futuro che, per anni, ad ogni manifestazione hi-tech i cronisti non hanno mancato di rifilarci in tutte le salse è "la casa comandata dal telefonino". Basta chiavi, interruttori, telecomandi ed allarmi: mentre siamo in coda in tangenziale, con la semplice pressione di un tasto, possiamo alzare le tapparelle, far partire il riscaldamento, vedere se c'è birra in frigorifero (ed eventualmente dire al frigorifero di ordinarla), accendere la televisione, allungare il divano, raccogliere le mutande dal pavimento[117] e buttare la pasta. Molto comodo. La domotica[118] è sicuramente una delle sfide tecnologiche per questo millennio appena iniziato. Peccato che, impostando la questione nei termini appena descritti, se ti clonano il cellulare ti ritrovi sotto un ponte.

In effetti portare *tutto* nel dominio informatico apre enormi questioni a proposito della sicurezza; come dice il già citato Paul S. Boreele, "c'è un costo per proteggere il software ed uno per craccarlo". La sicurezza informatica assoluta non esiste, anche se a dire il vero la maggioranza delle violazioni "informatiche" sono in realtà abili applicazioni di *social engineering*[119]. Certamente non varrà la pena per un pirata lavorare mesi per craccare il *mio* telefonino, visto che a casa mia c'è poco da rubare, ma immaginiamoci di trovare il cellulare di Bill Gates…

### *Le Storie Di Babele (X): La Magione Di Zio Bill*

Il campanello di Casa Gates non fa "Drin". Fa partire un .WAV quadrifonico campionato a 92Khz della durata di venticinque minuti, un tappetone di violini e stelle di mare che si

[116] Ma a onore del vero, da quando è uscito OsX, il trend Apple sta decisamente cambiando...
[117] In questo scenario fantascientifico un robottino raccogli-mutande non vogliamo non mettercelo?
[118] Da "domos", casa. Una volta i computer stavano *dentro* le case. Con la domotica i computer *sono* le case.
[119] Ad esempio chiamare la Segretaria del Famoso Uomo Di Affari con un disturbatore di linea e il Caller ID disabilitato, chiedendole per favore di ripeterci password, numeri di conto e informazioni varie "perché ce le siamo dimenticate" o "abbiamo dimenticato il portafoglio a casa". Praticamente un classico.

dice sia stato composto per l'occasione da Brian Eno e che la prima volta che lo senti ti rilassa fino a farti sognare mistiche atmosfere caraibiche. Alla decima volta che lo senti sei ridotto ad una ameba tremante e tutto quello che desideri dalla vita è inginocchiarti in un silenzioso eremo nepalese. Sullo zerbino di Zio Bill, anziche' "Welcome", c'è scritto "Start". I camerieri al posto di offrirti da bere ti chiedono "Where do you want to go today?". Quando vai in bagno, subito dopo aver fatto il tuo bisognino, il water lampeggia richiamando la tua attenzione (come da Documento MicroProt #1662884-62, Guida per la Progettazione di Interfacce Naturali ed Intuitive) e vengono diffuse le note di "Start Me Up" dei Rolling Stones. Ogni servizio igienico è collegato via Intranet ad un server che monta il nuovo sistema operativo orientato alla domotica (HOOOUSE, Home Object Oriented Operative Universal System Executive) chiamato Windoz WC. Per tirare l'acqua[120] devi battere Shift-F12. Al posto dei rubinetti ci sono dei joystick con ritorno di forza. Al posto della carta igienica ci sono delle riproduzioni su carta invecchiata del Codice Hammer di Leonardo.

Chiudo la chiamata che avevo composto dal cellulare di Zio Bill, trovato nei bagni di questo albergo di New York dove entrambi questa notte dormiremo (io in una stanzetta al quarto piano, lui nella Suite Imperiale al centoventisettesimo). La Casa mi chiede se voglio spegnere tutto (Shutdown), ricostruirla dalle fondamenta (Restart) o semplicemente andare a fare un giro (Logoff). Poi, l'intuizione: smanaccio un po' coi menù e trovo *Format House.* Bene. *Are you sure? <Y/N>*. Yes. Mentre a migliaia di chilometri da qui Casa Gates si decompone, cambio il PIN al telefonino e lo riconsegno alla reception.

Meno male che l'anno scorso alla U.S. Robots and Mechanical Men ho seguito un seminario di Susan Calvin e adesso conosco un po' di Psicologia del Silicio.

## E Parlarono Con Mille Lingue

Continuiamo a discuterne, ma ancora non abbiamo risposta alla Domanda Fondamentale: "Ma *come si fa* a scrivere un programma? *Che cosa è* un programma? Che aspetto ha? *Di cosa è fatto il Nemico?*"

Siccome un'immagine vale più di mille parole, sembrerebbe giunto il momento di vedere qualche figura. Peccato che i programmi siano quasi esclusivamente sequenze di parole, quindi scordatevelo. Il massimo che possiamo offrire come premio al il lettore più paziente è mostrare alcuni esempi nel prossimo paragrafo. Ok?

*[Disegno: Un tritacarne nel quale entra un programma C ed esce dell'assembler]*

[120] Anche se a rigor di logica, essendo l'acqua un liquido incompressibile, non puoi "tirarla".

Nel frattempo, poniamoci la seguente domanda: in fondo, perché ci interessano delle nozioni così tecniche? O meglio, perché *dovrebbero* interessarci? Per un motivo molto semplice: il Nemico stesso trova le sue origini nel software e dallo stesso prende la sua forza. Conoscere almeno superficialmente come nasce un programma equivale a capire meglio la natura stessa del Nemico.

A questo punto bisognerebbe fare il classico esempio della ricetta dell'uovo sodo e introdurre il concetto di *algoritmo*[121], perché tutti i libri di informatica, per spiegare cosa sia un programma, fanno così. Insomma, il *Codice Sorgente* e il *Codice Oggetto*, le sequenze di istruzioni, i "se questo è vero allora fai così altrimenti fai cosà", i "fino a quando questa cosa è vera fai quella cosa lì"...

Ma questo *non è* un libro di informatica. Questo è un libro di *filosofia*. Magari sul concetto di algoritmo ci torniamo, ma non siatene tanto sicuri[122]. È più interessante dare una definizione alternativa e possibilmente ridanciana, ovvero:

> Un programma non è altro che una sequenza di istruzioni che risolve in modo incomprensibile e misterioso un problema che l'utente non sapeva di avere.

Sottolineiamo *incomprensibile*, perché a volte, anche per un programmatore esperto, come certe cose possano funzionare è un enigma. In ogni caso avere almeno un'infarinatura teorica su cosa siano i *linguaggi di programmazione* serve a rendersi conto di una serie di interessanti fatti. In particolare, dato per assodato che noi si stia combattendo contro un Bastardissimo Bug e non contro un banale cavo rotto, allora:

- A volte il bug è implicato dal poco rigore del linguaggio di programmazione.

Ci sono casi illustri in cui il risultato di determinate operazioni o il procedimento con il quale queste operazioni vengono eseguite non è stabilito univocamente a livello di specifiche del linguaggio, per cui lo stesso programma eseguito su macchine, sistemi operativi o ambienti diversi si comporta diversamente[123].

---

[121] Di questa parola è interessante l'etimologia che, come buona parte di ciò che fa parte del mondo matematico, è di origine araba. In particolare deriva da [Muhammad ibn-Musa] al-Khwarizmi, dotto persiano vissuto intorno all' 800 che, secondo alcuni storici, sarebbe da considerarsi Padre dell'Algebra (anch'essa parola di origine araba) ancora più dello stesso Diofanto.

[122] A questo proposito lo Sciamano M'Bhutu Ghoile dice testualmente: "Esistono Duecentomila Algoritmi tra il Cielo e la Terra, e solo di Due di Essi io Ignoro l'Origine: La Ricerca Binaria e il Bubble Sort".

[123] Due sono i casi che subito balzano alla mente: il primo è il linguaggio Java, che in alcuni suoi aspetti (piuttosto specialistici, in verità) si comporta in maniera a volte bizzarra a seconda delle versioni e delle diverse implementazioni. Il secondo è il bagno di sangue delle diverse versioni di Windows, tanto che i programmatori parlano non solo di compatibilità "cross-platform" ovvero tra sistemi completamente diversi (Apple OS9 vs Linux, ad esempio) ma addirittura di compatibilità "cross-Windows", ovvero tra diverse versioni di Windows (2000, XP, CE, 95, 98, NT…) che, in un mondo ideale, dovrebbe essere data per

- A volte il bug è insito nel processo di traduzione del programma da parte del calcolatore.

In quasi tutti i linguaggi di programmazione l'esecuzione di un programma implica un processo di traduzione di istruzioni da un livello simbolico ed orientato alla comprensione umana ad uno numerico effettivamente eseguibile a livello di hardware, di segnali elettrici che vanno e vengono tra memorie e CPU, di motori e dischi che si accendono e spengono etc. Insomma, in ultima analisi, *un programma è una sequenza di comandi che accendono e spengono roba*. Un bug in questo processo di trasformazione implica che un programma apparentemente "giusto" (per un umano) si comporti in maniera sbagliata: si tratta di un vero e proprio errore di traduzione.

D'altra parte, agli errori di traduzione siamo abituati... È storica la prima versione della traduzione del famoso "Neuromancer" di William Gibson, romanzo capostipite della corrente letteraria fantascientifica nota come *cyberpunk*. Infarcito di slang e termini tecnoinformatici, non poteva che risultare incomprensibile in un' epoca in cui era difficile trovare traduzioni sensate anche per i manuali dei frullatori elettrici. In particolare, tra i tanti divertenti nonsense c'è quello delle "microluci", utilizzate nel romanzo come veicoli per un attacco notturno ad una base nemica. Il procedimento per capirci qualcosa in questi casi è un classico *reverse engineering* semantico: un traduttore che non sa che pesci pigliare, perché e quando dovrebbe scrivere "microluci"? Ad esempio, quando l'originale è "Microlights". Da qui in poi è facile: in Inglese "light" oltre che "luce" vuol dire "leggero", quindi queste "microlights", dal contesto, sono piccoli veicoli volanti ultraleggeri, ovvero deltaplani a motore[124]. Chi l'ha detto che la Nobile Arte del Debugging è utile solo per le Macchine?

- A volte il bug è insito nella non corretta interpretazione di qualche caratteristica del linguaggio da parte del programmatore;

Errare è umano: uno pensa che un certo comando faccia una certa cosa e invece ne fa un'altra. Ad esempio, per quanto i sinonimi vengano evitati nei linguaggi di programmazione, può capitare di chiamare due cose diverse con lo stesso nome e può capitare che, in alcuni linguaggi e/o sotto condizioni particolari, problemi di questo tipo siano incredibilmente difficili da trovare.

- Il peggio che possa succedere è una combinazione dei punti precedenti ed è quindi questo che accade solitamente.

scontata.

[124] Più recentemente, abbiamo visto dei "file systems" tradotti con "file di sistema". Forza, signori traduttori, non è così difficile... i file di sistema sarebbero "system files", no?

…che poi sono più o meno gli stessi motivi per cui i Bechelon di Otargo VII e le Flamboianti Farpalle di Satragon Beta entrarono in guerra per più di diecimila anni[125]: i rispettivi monarchi – anzi, gli *interpreti* degli stessi – non si erano capiti perché usavano un numero diverso di bocche per parlare[126].

Ohum. *Linguaggio* di programmazione. Addirittura al plurale. *Linguaggi*. La faccenda comincia ad assumere contorni apocalittici: cosa successe ai babilonesi quando sfidarono Dio, cercando di costruire una Torre alta come il Cielo stesso? Esatto. Un GCTG[127].

Ebbene, esattamente la stessa cosa è successa con i calcolatori e i linguaggi di programmazione...

**La Mostra delle Atrocità**

Prenderemo in esame un programma scritto in un linguaggio che si chiama "C" (?), il quale stampa "Ciao, Mondo!" sul video del vostro computer:

```
#include <stdio.h>

void main(void)
{
        printf("Ciao, Mondo!\n");
}
```

…che poi non e' molto dissimile dallo stesso programma scritto in Java:

```
public class CiaoMondo {

        public static void main(String[] args) {
                System.out.println("Ciao, Mondo!");
        }
```

[125] La famosa *Guerra dei Mille Soli*. I Manuali di Storia Intergalattica del Volturnahr riportano che più di duecento mondi vennero distrutti nelle fasi finali di questa guerra, terminata con la vittoria di una terza popolazione – i Legulazzi di Romella Major – dediti prevalentemente alla pastorizia e alle attività forensi.
[126] I Bechelon di Otargo VII hanno una sola bocca come noi, ma le Flamboianti Farpalle di Satragon Beta ne hanno quattro – due per lato – situate sull'addome (ammesso che lo si voglia chiamare addome).
[127] Grande Casino Totale Globale. Secondo una diversa interpretazione della mitologia greca, invece, Prometeo craccò il web server del paradiso e rubò il Fuoco. Questo fece incazzare moltissimo il webmaster Giove, che tese un tranello a Prometeo presentandogli una nuova versione di uomo, ovvero una donna chiamata Pandora. Questa venne distribuita in bundle con uno scatolone di virus e altre schifezze, compreso Y2K, che un giorno liberò per errore sulla Rete mentre, annoiata, aspettava che finisse un download. Questo è il motivo per cui ogni due per tre dobbiamo far ripartire Quel Sistema Operativo Là.

```
}
```

…o dal caro vecchio BASIC:

```
10      PRINT "Ciao, Mondo!"
```

…o dal Pascal:

```
PROGRAM CiaoMondo;

        BEGIN
        WRITE('Ciao, Mondo!');
        END.
```

e che, in verità, non è molto dissimile da *qualsiasi* linguaggio se tutto quello che vogliamo fare è stampare un messaggio idiota come "Ciao, Mondo!"[128]. Avete appena visto quello che, tecnicamente, si chiama *Codice Sorgente* di un programma. Come dicevamo, il computer non esegue direttamente programmi sotto forma di codici sorgente; questi hanno bisogno di un processo di traduzione[129] che li trasforma in *Codici Oggetto*. Il motivo per cui non scriviamo direttamente in codice oggetto è che quest'ultimo varia molto facilmente da computer a computer, al passo con l'evoluzione tecnologica degli stessi, essendo in ultima analisi una interminabile sfilza di uni e zeri per

[128] Per qualche misterioso motivo, la prima cosa che ti faranno fare quando inizierai a programmare in qualsiasi linguaggio sarà scrivere un programma che stampa "Ciao, Mondo!" sul monitor del tuo amato scatolotto. Al massimo, se il docente è anglofono, dovrai stampare "Hello, World!". Per lo stesso misterioso motivo, quando non saprai come chiamare un programma di prova, non ti verrà in mente un nome migliore di "prova". Inoltre, al posto di usare nomi più significativi, chiamerai le varie funzioni del tuo programma con i nomi "pippo", "pluto" e "paperino" ed infine la variabile di controllo di un ciclo sarà sempre "i", in modo da poterla confondere con tutte le altre.

[129] Che, a seconda della natura del linguaggio, sarà una *compilazione* se il sorgente viene trasformato in codice oggetto tutto in una volta, o in una *interpretazione* se il sorgente viene eseguito una istruzione alla volta. I linguaggi vengono così divisi in due grandi categorie, con interessanti variazioni sul tema come quello di Java, che viene compilato in un linguaggio intermedio chiamato P-Code e poi interpretato da un programma che funge da "processore virtuale". Sappiate che ci sono dei validi motivi per fare questo giro anche se qui non ci dilungheremo oltre.

capire i quali è necessaria una vita di ascesi e contemplazione di circuiti elettrici, logica booleana e specifiche architetturali di microprocessori[130].

Forse varrebbe la pena vedere un secondo esempio, solo lievemente più complicato, giusto per introdurre alcuni concetti da dimenticare subito dopo. Ad esempio, è ben nota la definizione di "numero primo": un numero divisibile solo per se stesso e per 1. Il numero 17 è primo; il numero 45 no, perché è divisibile per 3 e per 5 e quindi anche per tutte le loro combinazioni: per 9, ad esempio, o per 15. Ovviamente, 45 è divisibile anche per 1 e per 45: tutti i numeri sono divisibili per se stessi e per 1, mentre i numeri primi sono divisibili *solo* per se stessi e per 1. Siete abbastanza confusi? Ok, vediamo un pezzetto di codice Java che decide se un numero è primo o meno:

```
static boolean IsPrime(int n)
{
    for(int divisore = 2; divisore < n -1; divisore++)
        if(n % divisore == 0)
            return false;

    return true;
}
```

Quella appena presentata è una *funzione*: viene passato un parametro in ingresso – *n,* ovvero il numero del quale decidere se è primo – e viene restituito un valore di verità, ovvero il fatto che il numero sia davvero primo o meno. Si noti il tipo di variabile restituita: *boolean*[131], ovvero una variabile che può essere solo o vera o falsa.

La funzione non fa altro che generare una sequenza di possibili *divisori*, iniziando da 2 e finendo a *n - 1*, per controllare se la divisione tra *n* e tutti i possibili divisori sia esatta. Questa operazione viene compiuta utilizzando l'operatore "%", chiamato *modulo*, che restituisce il resto della divisione intera tra i suoi operandi. Se il resto è zero, ovviamente il dividendo è esattamente divisibile per il divisore, ovvero abbiamo trovato un divisore, ovvero il numero passato alla funzione non è primo. Quindi possiamo restituire *falso*. Se invece arriviamo all'ultimo dei possibili divisori, ovvero *n – 1*, senza aver trovato divisori esatti, possiamo dire che il numero è primo e restituiamo *vero*.

Una simile funzione potrebbe essere utilizzata in un programma che stampa tutti i numeri primi inferiori ad un milione:

[130] Ci interessa? No. Non ci interessa. Almeno a noi. Ma c'è chi è in grado di leggere e scrivere programmi direttamente in linguaggio macchina.

[131] Il termine *booleano* viene da George Boole, matematico inglese, autodidatta e inventore – intorno al 1850 – dell'omonima algebra che lavora con i soli concetti di "vero" e "falso". Per ovvi motivi di uni e zeri, l'algebra booleana è alla base del meccanismo di funzionamento dei calcolatori digitali.

```
public class Primes
{

    static boolean IsPrime(int n)
    {
        for(int divisore = 2; divisore < n -1; divisore++)
            if(n % divisore == 0)
                return false;

        return true;
    }

    public static void main(String[] args)
    {
        for(int i = 2; i < 1000000; i++)
            if(IsPrime(i))
                System.out.println(i);
    }
}
```

A fronte di quanto realizzato, Babele Dunnit può sostenere senza tema alcuno di smentita che il più alto numero primo inferiore al milione è 999983.

Può essere interessante notare che la funzione testé presentata si presta a notevoli ottimizzazioni: ad esempio, è inutile generare tutti i divisori pari una volta accertato che il numero che stiamo controllando non è divisibile per 2: se non è divisibile per 2, non lo sarà nemmeno per i suoi multipli. Ogni numero pari è divisibile per 2. La cosa non è vera per i numeri dispari: 51 è divisibile per 3 e 17, ma non per 5, ad esempio. Quindi, una volta appurata la non-divisibilità per 2, possiamo ottimizzare partendo da 3 e avanzando 2 alla volta.

Non solo: è anche inutile generare i divisori fino a *n-1*, poiché se esistono dei divisori, almeno uno di questi sarà minore della radice quadrata del numero dato. Pensateci e vedrete che è così. Ed è anche inutile calcolare la radice quadrata, operazione dispendiosa; benché equivalente, è molto più veloce controllare che *il quadrato del divisore* sia inferiore a *n.*

Ecco la nuova versione di IsPrime:

```
    static boolean IsPrime(int n)
    {

        if(n % 2 == 0)
            return false;

        for(int divisore = 3; divisore * divisore < n; divisore += 2)
            if(n % divisore == 0)
                return false;

        return true;
```

```
    }
```

Con questa nuova versione ottimizzata, il tempo per calcolare tutti i primi al di sotto del milione (Java 1.4.2, P4 2.8 Ghz) scende da vari minuti a pochi istanti. Per forza: il numero di operazioni necessarie è sceso da (circa) N^2 a N*sqrt(N)/2! Per decidere se 999983 è primo, siamo passati all'incirca da mille miliardi di operazioni a 500 milioni... son sempre un bel po', ma vorremo far lavorare lo Scatolotto, no?

*Programmare è umano, ottimizzare è divino.*

Non solo: quello presentato è solo uno di molti *algoritmi* possibili per trovare i numeri primi. Un altro, ben noto, è il seguente: si scrivano su un pezzo di carta tutti i numeri a partire dal 2, si faccia un cerchietto intorno a quest'ultimo e poi, ripetutamente:

1) si cancellino tutti i multipli del numero appena cerchiato.
2) si faccia un cerchietto sul prossimo numero della lista

Al primo passaggio rimane il 2 e se ne vanno tutti i suoi multipli; al secondo rimane il 3 e se ne vanno tutti i suoi multipli, al terzo rimane il 5 perché il 4 era stato cancellato al primo passaggio, e così via. A voi scoprire come si chiama questo algoritmo.

Il lettore dovrebbe a questo punto avere almeno una vaga idea di cosa significhi programmare: a questo livello di approssimazione, sicuramente il famoso titolo del libro di Niklaus Wirth, “Strutture Dati + Algoritmi = Programmi” è tutto ciò di cui abbiamo bisogno.

Ed ora, signori e signore, un nuovo programma, lievemente più complesso giusto per tenervi svegli. Vince un bug di pezza chi indovina a prima vista cosa fa questo programma C:

```
#include <stdio.h>

main(t,_,a)
char *a;
{
return!0<t?t<3?
main(-79,-13,a+
main(-87,1-_,
main(-86, 0, a+1 )+a)):1,t<_?
main(t+1, _, a ):3,
main ( -94, -27+t, a)&&t == 2 ?_<13 ?
main ( 2, _+1, "%s %d %d\n" ):9:16:t<0?t<-72?
main(_,t,"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+\
,/+#n+,/#;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l q#'+d'K#!/\
+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n')()#}w'){){nl]'/+#n';d}rw' i;# ){n\
```

```
l]!/n{n#'; r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#\
n'wk nw' iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \
;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;\
#'rdq#w! nr'/ ') }+}{rl#'{n' ')# }'+}##(!!/")
:t<-50?_==*a ?putchar(a[31]):
main(-65,_,a+1):
main((*a == '/')+t,_,a+1 ):0<t?
main ( 2, 2 , "%s"):*a=='/'||
main(0,
main(-61,*a, "!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);
}
```

Si potrebbe arguire che, mentre Java tutto sommato è un linguaggio comprensibile, il C sia un linguaggio per programmatori con *gravi* problemi. Ecco cosa stampa il programma di cui sopra, una volta eseguito:

```
On the first day of Christmas my true love gave to me
a partridge in a pear tree.

On the second day of Christmas my true love gave to me
two turtle doves
and a partridge in a pear tree.

On the third day of Christmas my true love gave to me
three french hens, two turtle doves
and a partridge in a pear tree.
```

[...e via così fino a]

```
On the eleventh day of Christmas my true love gave to me
eleven pipers piping, ten lords a-leaping,
nine ladies dancing, eight maids a-milking, seven swans a-swimming,
six geese a-laying, five gold rings;
four calling birds, three french hens, two turtle doves
and a partridge in a pear tree.

On the twelfth day of Christmas my true love gave to me
twelve drummers drumming, eleven pipers piping, ten lords a-leaping,
nine ladies dancing, eight maids a-milking, seven swans a-swimming,
six geese a-laying, five gold rings;
four calling birds, three french hens, two turtle doves
and a partridge in a pear tree.
```

Lo so che è difficile da credere, ma è davvero così. Si noti che il programma in questione è ricorsivo: la funzione “main” richiama se stessa più volte. Questo potrebbe darvi un indizio, oppure no, ma il punto non è questo.

Il punto è che il secondo programma è *volutamente* scritto in maniera da risultare incomprensibile anche ad un programmatore esperto[132]. Serve solo a dimostrare che il Male è tra di noi, e che in generale i programmatori di notte russano perché di giorno trattengono il respiro per ore, mentre i progettisti di software sono degli impressionisti e il Kernigham – Ritchie[133] è in realtà un manifesto del Cubismo. Insomma, se un linguaggio di programmazione ti permette di fare una cosa del genere, come possiamo sperare che il Nemico non si annidi tra un carattere e l'altro? Conviene darsi alla bottiglia, cosa che facciamo prontamente:

```
# Copyright 2001 Christopher J. Carlson <cjc@dobbz.com>
# All Rights Reserved

          $a=
         "cpuu
        \bmft p
        \bg cff
        \bs";$b
        ="po ui
        \bf xbm
       \bm";$c="
       Ypv ublf p
      \bof epxo qb
    \btt ju bspvoe";
   $a =~ s/\n//;$a =~
   s/\s+/ /g; $b   =~
   s/\n// ;  $b    =~
   s/\s+/ /g;$c    =~
   s/\n// ;  $c    =~
   s/\s+/ /g;$a    =~
   y/b-z/a-z/;$b   =~
   tr/b-z/a-z/;$c  =~
   tr/b-z/a-z/ ; for(
   $d=100;$d>0;$d--){
   print"$d $a $b $d"
   ;print" $a,\n$c, "
   ;print($d-1);print
   " $a $b.\n";} $x =
   "cjc"; $y="dobbz";
   $z="com";print"\n"
   ;print "- $x\@$y."
    ;print"$z \n\n";
```

Purtroppo a volte una bottiglia sola non basta...

[132] In particolare, questa tecnica si chiama *code obfuscation*. Con certi linguaggi è impossibile; altri, come il C, si prestano a tal punto che esistono addirittura dei concorsi. Va da sé che chi è in grado di pensare certe cose è fuori di cotenna.

[133] Quando dei programmatori parlano di Kernigham e Ritchie non si riferiscono a personaggi di Happy Days ma al libro scritto dagli ideatori del linguaggio C. Chissà se questo Dinamico Duo si sarà reso conto, mentre dava in pasto al mondo il parto delle proprie elucubrazioni, di star creando un mostro?

```
(bottled by Acme::EyeDrops)
Date: 06/04/05
Author:     Andrew Savige
URL:  n/a
Comments:   23
Info: http://search.cpan.org/dist/Acme-EyeDrops/lib/Acme/EyeDrops.pm
Score:      (4.62 in 109 votes)

     ''=~(        '(?{'        .('`'        |'%')        .('['        ^'-')
     .('`'        |'!')        .('`'        |',')        .'"'.        '\\$'
     .'=='        .('['        ^'+')        .('`'        |'/')        .('['
     ^'+')        .'||'        .(';'        &'=')        .(';'        &'=')
     .';-'        .'-'.        '\\$'        .'=;'        .('['        ^'(')
     .('['        ^'.')        .('`'        |'"')        .('!'        ^'+')
    .'_\\{'      .'(\\$'      .';=('.      '\\$=|'      ."\|".(      '`'^'.'
   ).(('`')|    '/').').'    .'\\"'.+(    '{'^'['').    ('`'|'"')    .('`'|'/'
  ).('['^'/')  .('['^'/').  ('`'|',').(  '`'|('%')).  '\\".\\"'.(  '['^('(')).
  '\\"'.('['^  '#').'!!--'  .'\\$=.\\"'  .('{'^'[').  ('`'|'/').(  '`'|"\&").(
  '{'^"\[").(  '`'|"\"").(  '`'|"\%").(  '`'|"\%").(  '['^(')')).  '\\").\\"'.
  ('{'^'[').(  '`'|"\/").(  '`'|"\.").(  '{'^"\[").(  '['^"\/").(  '`'|"\(").(
  '`'|"\%").(  '{'^"\[").(  '['^"\,").(  '`'|"\!").(  '`'|"\,").(  '`'|('',')).
  '\\"\\}'.+(  '['^"\+").(  '['^"\)").(  '`'|"\)").(  '`'|"\.").(  '['^('/')).
  '+_,\\",'.(  '{'^('[')).  ('\\$;!').(  '!'^"\+").(  '{'^"\/").(  '`'|"\!").(
  '`'|"\+").(  '`'|"\%").(  '{'^"\[").(  '`'|"\/").(  '`'|"\.").(  '`'|"\%").(
  '{'^"\[").(  '`'|"\$").(  '`'|"\/").(  '['^"\,").(  '`'|('.')).  ',','.(('{')^
  '[').("\["^  '+').("\`"|  '!').("\["^  '(').("\["^  '(').("\{"^  '[').("\`"|
  ')').("\["^  '/').("\{"^  '[').("\`"|  '!').("\["^  ')').("\`"|  '/').("\["^
  '.').("\`"|  '.').("\`"|  '$')."\,".(  '!'^('+')).  '\", _,\\"'  .'!'.("\!"^
  '+').("\!"^  '+').'\\"'.  ('['^',').(  '`'|"\(").(  '`'|"\)").(  '`'|"\,").(
  '`'|('%')).  '++\\$="})'  );$:=('.')^  '~';$~='@'|  '(';$^=')'^  '[';$/='`';
```

Lo avete indovinato? Ve lo dice Babele Dunnit: si tratta di codice *Perl*, non solo offuscato ma anche opportunamente riformattato – nel primo caso a mano e nel secondo mediante un programma apposito chiamato CPAN. Per la cronaca, un programmatore Perl meno creativo avrebbe potuto scriverlo così, con tanto di ruttino alla fine:

```
#!/usr/bin/perl
# Jim Menard

$nBottles = $ARGV[0];
$nBottles = 100 if $nBottles eq '' || $nBottles < 0;

foreach (reverse(1 .. $nBottles)) {
     $s = ($_ == 1) ? "" : "s";
     $oneLessS = ($_ == 2) ? "" : "s";
     print "\n$_ bottle$s of beer on the wall,\n";
     print "$_ bottle$s of beer,\n";
     print "Take one down, pass it around,\n";
     print $_ - 1, " bottle$oneLessS of beer on the wall\n";
}
print "\n*burp*\n";
```

In tutti i casi, ecco il risultato dell'esecuzione di questi programmi:

```
99 bottles of beer on the wall, 99 bottles of beer.
Take one down and pass it around, 98 bottles of beer on the wall.

98 bottles of beer on the wall, 98 bottles of beer.
Take one down and pass it around, 97 bottles of beer on the wall.

97 bottles of beer on the wall, 97 bottles of beer.
Take one down and pass it around, 96 bottles of beer on the wall.
```

[...e via così fino a]

```
2 bottles of beer on the wall, 2 bottles of beer.
Take one down and pass it around, 1 bottle of beer on the wall.

1 bottle of beer on the wall, 1 bottle of beer.
Take one down and pass it around, no more bottles of beer on the wall.

No more bottle of beer on the wall, no more bottles of beer.
Go to the store and buy some more, 99 bottles of beer on the wall.
```

Si tratta di una canzoncina ben nota nel mondo anglosassone, che per qualche misterioso motivo è diventata oggetto di una famosa sfida: scrivere – nella maniera più fantasiosa possibile – un programma che la generasse. I risultati, dei quali avete appena visto solo un assaggio, sono reperibili presso http://www.99-bottles-of-beer.net/.

La tentazione di mostrare al malcapitato lettore esempi concreti e più o meno utili dei più deliranti linguaggi partoriti da mente umana[134] è forte, ma questo porterebbe via spazio alla trattazione di ben più importanti argomenti. Ci limiteremo quindi a *nominarne* una piccola parte: già questo esercizio è sufficientemente atroce[135].

```
...APL  APL  ASPOL  AVA  AXIOM  Abel  Accent  Ada  Algol  Arctic  Arctic
Ashmedai  Awk  BABEL  BALGOL  BLAZE  Babbage  Basic  C  C#  C++  CADET  CHARITY
CHILI  Cobol  Crystal  Cube  D  DEMON  DRAGON  Delirium  Dislang  Dylan  Echidna
Eiffel  Emerald  Erlang  Euclid  Forth  Fortran  GARP  Goedel  Haskell
INTELLECT  Jade  Java  Jossle  KRYPTON  Kaleidoscope  LOGIN  LOGLAN  LOLITA
Legion  Lincoln  Reckoner  Linda  Lingo  Lisp  Lua  MARBLE  MARVIN  MIDAS  MORAL
```

[134] Ci però sono teorie secondo le quali un programmatore non può appartenere alla razza umana e viceversa.

[135] Se ancora non vi basta, andate su http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm per un elenco esaustivo o sul già citato http://www.99-bottles-of-beer.net/ e date una occhiata al linguaggio chiamato "Brainfuck". Bambini: non tentate questo esperimento a casa da soli! Chiedete sempre ad un adulto di aiutarvi.

```
MORTRAN MUCAL MYSTIC Machiavelli Magritte Maple Mary Matchmaker
Mathematica MeToo Mesa Modula-2 Mona MooZ Mouse Moxie NASTRAN NODAL
Nawk Newsqueak Newton Nyquist OOPS Oberon-V Obliq Occam Octave Orwell
PAGE PHOCUS Paris Pascal Perl Prolog Python Quty QX Rexx SISAL SNOBOL
SOAR SOCRATIC SODA Smalltalk TABLET TRANQUIL Trafola-H Traits Twentel
VULCAN WAVE Wisp Yellow Z++ ZERO...
```

…e questo è veramente solo un piccolo esempio della Confusione che regna Là Fuori[136]. Se proprio non siete pronti a mandare tutto a memoria, sappiate che comunque il 95% di questa lista è l'epitaffio di linguaggi morti o addirittura mai nati: secondo l'infoarcheologa Jean Sammet, solo tra il 1952 e il 1972 più di 200 linguaggi di programmazione videro la luce, ma soltanto una piccola parte di questi viene considerata significativa fino a tutto il 1990:

| | |
|---|---|
| 1957 | FORTRAN |
| 1958 | ALGOL |
| 1960 | LISP |
| 1960 | COBOL |
| 1962 | APL |
| 1962 | SIMULA |
| 1964 | BASIC |
| 1964 | PL/I |
| 1966 | ISWIM |
| 1970 | Prolog |
| 1972 | C |
| 1975 | Pascal |
| 1975 | Scheme |
| 1977 | OPS5 |
| 1978 | CSP |
| 1978 | FP |
| 1980 | dBASE II |
| 1983 | Smalltalk-80 |
| 1983 | Ada |
| 1983 | Parlog |
| 1984 | Standard ML |
| 1986 | C++ |
| 1986 | CLP(R) |
| 1986 | Eiffel |
| 1988 | CLOS |
| 1988 | Mathematica |
| 1988 | Oberon |

[136] In effetti mancano molti linguaggi di scripting, come JavaScript, o di descrizione, come HTML, VRML, XML...

… e mancano gli ultimi tre lustri, dei quali il più importante esponente è probabilmente il già visto Java[137].

Cercate quindi di essere indulgenti, la prossima volta che qualcuno vi dice che fa il programmatore. Ricordatevi che la creazione del primo vero linguaggio di programmazione è attribuita a Konrad Zuse con il suo "Plankalkul" (Germania 1945, in pieno periodo nazista). Insomma, è tutto molto complicato.

La cosa singolare è che è dimostrabile[138] che, data per scontata la presenza di determinate caratteristiche[139], i linguaggi di programmazione sono tutti equivalenti, ovvero ciò che si può fare con un linguaggio lo si può fare anche con tutti gli altri. E allora perché? Perché infettare un mondo primevo, potenzialmente bello e composto di soli uni e zeri con una simile confusione?

Le risposte sono molteplici. Si va dal puro edonismo che porta il geek[140] di turno a creare il *suo* linguaggio solo perché ha in mente un bell'acronimo[141], molto spesso senza prima guardarsi intorno e quindi reinventando per l'ennesima volta la ruota e/o l'acqua calda – ma si sa, spesso un paio di mesi passati davanti ad un terminale ti risparmiano un paio d'ore in biblioteca – a motivi puramente politici, per cui ogni multinazionale tenta di imporre il *suo* linguaggio perché dietro alla adozione di uno standard – *qualsiasi esso sia* – ci sono interessi economici da capogiro[142]. Si noti che quanto esposto in questo paragrafo vale per *qualsiasi* standard e non solo per quelli informatici.

---

[137] Ma a mio modesto parere ci metterei anche Python e un linguaggio bellissimo e attualmente poco conosciuto chiamato D.

[138] Interessante il meccanismo della prova: praticamente si fa vedere come ognuno di questi linguaggi sia funzionalmente equivalente ad un oggetto astratto chiamato "Macchina di Turing", dal nome del famoso padre fondatore Alan Turing (1912-1954). Evidente come da qui alla formulazione della famosa battuta "La Macchina di Turing era una Ford Modello T" poco ci voglia. Meno divertente è stata la sua storia: genio incompreso e problematico, omosessuale nell'America della Caccia Alle Streghe, ossessionato da Biancaneve si suicidò mangiando una mela avvelenata. Si dice che il simbolo di Apple sia dedicato a lui; sicuramente ebbe un ruolo chiave nella decriptazione del famoso Codice Enigma tedesco e quindi, in ultima analisi, nella vittoria degli Alleati. XXX

[139] In particolare è importante, tra le altre, la presenza di costrutti grammaticali di controllo: se una condizione data è vera fai una certa cosa, altrimenti fanne un'altra. Non per tutti i sistemi programmabili è implicata questa necessità; ad esempio, una vecchia lavatrice meccanica esegue il suo programma sempre nel medesimo modo ed è quindi intrinsecamente meno potente di un computer. Le nuove lavatrici che invece sono in grado di modificare il proprio comportamento in base a quello che vogliamo lavare sono teoricamente potenti quanto un computer, anzi *sono* dei computer. Infatti quando usciamo di casa spesso i nostri elettrodomestici si mettono a parlottare e non è raro, tornando, sorprenderli mentre giocano a bridge tra loro.

[140] Traduzione letterale: geco. Si dice di smanettone particolarmente portato e monomaniaco nei confronti dell'informatica. Un po' come un nerd, ma fissato con la tecnologia.

[141] Moltissimi nomi di linguaggi, soprattutto vecchi, sono acronimi più o meno riusciti come FORTRAN (FORmula TRANslation), COBOL (Common Business Oriented Language), BASIC (Beginner's All-purpose Symbolic Instruction Code) ...

[142] Si veda la vera e propria guerra tra Java di SUN e C# di Microsoft, ad esempio.

Nel mezzo ci stanno le ragioni storiche. Moltissimi linguaggi sono nati in quanto orientati alla soluzione di problemi appartenenti ad una ben determinata area[143] e ci sono voluti vari cambi di paradigma nel concetto di "programmazione" e varie decine di anni prima di arrivare ad un modello – quello attuale – dove più o meno tutti si sentono "a casa":

### *Le Storie Di Babele: (5) Storia del Programmismo Senza Limitismo*

... e all'inizio era il TuttoONiente e si programmava brutalmente in Uni e Zeri (si noti come il dualismo cartesiano sia profondamente intessuto nei cardini stessi dell'Informatica) spostando interruttori e bucherellando cartoncini e fu in seguito ai giustificati problemi esistenziali dei primi programmatori che Claudio Villa cantò "Binario, Triste e Solitario". Poi arrivarono i Primi Linguaggi, dove tutti potevano fare tutto e regnava perciò grande confusione e il Programma era Costruito dal Basso, e fu questa l'Era Globale ed erano questi i tempi cupi di Assembler e di BASIC, FORTRAN e COBOL e della programmazione Bottom-Up.

Poi arrivò il Tempo della Programmazione Strutturata e della metodologia chiamata come il ketchup: fu l'Era di Top-Down, che spezza un problema in sottoproblemi, procedendo dall'alto al basso e dal generale al particolare con santa pace di Sherlock Holmes e Isaac Newton. E vennero gli Illuminati come Dijkstra e Wirth, che col suo Pascal predicò il Verbo della Programmazione Strutturata, dove la tolemaica visione dettaglio-centrica dello sviluppo del software veniva completamente sovvertita. Non più con deduttiva fatica fummo giocatori di LEGO logico ma, al contrario, manipolatori di un Problemone che viene spezzettato in Problemini sempre piu' prossimi al banale (cfr. M.Marchesi, "Tutto il mondo è palese") e la cui soluzione è invariabilmente errata.

Finalmente, con l'Object Oriented Programming arrivò un Mondo Orientato agli Oggetti, ognuno dei quali – storia a sé stante, scatola nera che incapsula dati e comportamenti – risponde a ordini che vengono impartiti non importa da chi. Nell'Object Oriented Programming, il paradigma attualmente più utilizzato nel mondo dello sviluppo del software, programmare significa risolvere un problema grazie alla mutua collaborazione di impalbabili "oggetti" molto specializzati che si scambiano dati, comandi e messaggi. La complessità viene così nascosta al punto da avere l'impressione che il software, quando funziona, funzioni "per magia". O meglio, all'inzio si ha questa impressione, che diventa, mano a mano, una certezza.

E domani? Domani è giovedì.

Domani gnocchi.

[143] Con casi fenomenali come quello del COBOL, linguaggio ormai ampiamente sorpassato che però, grazie alla sua adozione massiccia nel mondo finanziario, tuttora resiste in numerosi ambienti dall'inerzia tecnologica pressochè infinita.

Così i linguaggi di programmazione e tutto il sistema operativo sottostante sono l'infrastruttura grazie alla quale il programmatore, in preda a delirio di onnipotenza e ivi momentaneamente assurto al ruolo di Imperatore dell'Universo, pronuncia i suoi incantesimi. Solo che, per loro natura, questi sortilegi applicativi, composti da migliaia di formule magiche (il cosiddetto *Codice Sorgente*), devono venire trasformati attraverso degli Strumenti di Trasmutazione (i cosiddetti *Compilatori* o *Interpreti*, a seconda del linguaggio) in sequenze interminabili di simboli oscuri (il cosiddetto *Codice Oggetto*, dalla duplice forma di *Linguaggio Macchina* – uni e zeri – o di *Linguaggio Assembler*, forma simbolica dello stesso) comprensibili non più dall'Uomo ma dal Soggiacente Mondo Elettrico (La Ferraglia Che Avete Sulla Scrivania). E questi Strumenti di Trasmutazione, per loro natura, a volte ci regalano dei bug.

Arrivare a capire al volo che un certo comportamento anomalo, un certo baco, è imputabile a un problema linguistico e non, ad esempio, a un driver sbagliato o a un cavo difettoso implica una notevole conoscenza della Ars Programmatoria, pertanto non approfondiremo il discorso in questa sede. Oltretutto, anche se fossimo in grado di arrivare ad una simile diagnosi, la maggior parte delle volte non potremmo farci niente. Infatti non sempre sono disponibili dei sorgenti sui quali intervenire[144], mentre andare a cambiare a manina il codice oggetto di un programma è una operazione molto difficoltosa – anche se, a onor del vero, ci sono Maestri in grado di lavorare *nativamente* in linguaggio macchina[145]. Questa è una forma di Magia Estrema che a volte viene utilizzata per eliminare le protezioni sul software commerciale, per quanto non sia raro che i programmi craccati[146] siano anche migliorati rispetto alle versioni originali protette[147].

[144] Ma le cose cambiano. Tutto il movimento Open Source gira intorno a questo concetto. Vedi oltre.

[145] Per una maggiore presa di coscienza si raccolgano informazioni sulla ordalia programmatoria nota come *Assembly* che si svolge solitamente nel Nord Europa. Ivi si ritrovano alcune migliaia di giovani pazzi furiosi che si divertono a scrivere programmi dalle caratteristiche ai limiti della realtà. I *termini* stessi di partecipazione sono apparentemente assurdi: l'idea è di fare *qualcosa* – solitamente dei cosiddetti *demo*, ovvero programmi assolutamente inutili ma molto belli dal punto di vista grafico e sonoro – entro un numero limitato di bytes. Tra le categorie dell'*Assembly* resistono tuttora, in questo mondo di Gigabytes e Gigahertz, quelle denominate *4K* (spesso dedicate ai telefoni cellulari) e *64K*, ovvero che pongono come tetto massimo 4096 Bytes e 65536 Bytes di lunghezza al file eseguibile. Babele Dunnit ricorda di avere visto una demo 4K con la quale, alla fine della parte non interattiva di grafica,suoni e svolazzi vari, era possibile giocare anche a *Tetris*. Una mente razionale rifiuta una simile possibilità, ma siamo qui proprio per instillare nel lettore quel tarlo senza il quale non è possibile arrivare all'Illuminazione: è evidente come questi *demo* siano una vera e propria forma d'arte. Non servono assolutamente a nulla, ma sono *bellissimi;* il processo grazie al quale gli autori li creano è magicamente sospeso tra la pura intuizione e una inarrivabile abilità logico-matematica nel manipolare simboli assolutamente alieni al resto dell'Umanità.

[146] Da "to crack", rompere. Si dice della attività quasi-soprannaturale relativa all'eliminazione di protezioni di qualsiasi tipo da software di qualsiasi tipo. La liceità dell'azione è discutibile, ma non si può dimenticare che è pratica comune dei crackers chiedere di comprare il software craccato qualora lo si utilizzi commercialmente: "We hope u enjoy this release and the motto 'if u use it alot then buy it!' still counts!! (H2O)". Per inciso: prima che a qualcuno venga il dubbio, peraltro già espresso in passato (e rientrato a fronte di opportuni procedimenti legali), che l'autore di questo libro sia un pirata informatico, di crack ne potete trovare liberamente quanti ne volete immettendo la parolina magica "crack" seguita dal nome di un programma qualsiasi in Google... non c'è bisogno di essere così bravi.

[147] All'autore è successo con un noto pacchetto di CAD 3D, scomodo perchè necessitava di una chiave hardware (un "dongle") e che oltretutto presentava dei problemi sulla parte di stampa. Pagati i 4500$ della licenza si decise di usare la versione craccata, che funzionava benissimo e che, essendo installata su più

Ecco lo scherno, lo sfottò più clamoroso che un Hacker Estremo possa rivolgere ad un onesto produttore di software: debuggarglielo[148] a forza, volente o nolente.

Torneremo a parlare di queste tecniche più avanti; ora, parlando di Ars Programmatoria, lasciamo la parola direttamente a Remote:

Volevo parlarti di un aspetto del debugging che non hai considerato.

Lo chiamerei "L'Abbandonarsi al Bug Inesplicabile".

Come ben saprai esistono bug che non cedono neppure ai mantra più esoterici. Il Samurai del Codice si arrende a questi bachi riconoscendone la Potenza, ma non può neanche venir meno alla sua missione: debellarli.

Ad esempio, narrano le leggende che Jeff Minter[149] una volta si sia imbattuto in uno di questi bug.

Per motivi forse riconducibili alla cabala[150] o alla fisica quantistica, in un suo programma a volte si verificava un risultato assurdo: in momenti diversi e non riproducibili due più due faceva cinque[151].

E non c'era niente da fare.

Si dice che Jeff smise per alcuni secondi di assumere sostanze psichedeliche e inchinandosi al Grande Bug si risolvesse ad introdurre nel loop principale del programma questa soluzione:

macchine ma venendo usata da una sola persona per volta, non infrangeva alcun *copyright*.

[148] Il lettore è pregato di perdonare l'autore per questa parola, ma la tentazione di declinare il verbo "debuggare", già di per sé orrendo frutto di fantasia neotecnogotica, era troppo grande.

[149] Mitico creatore di giochi fra i più psichedelici della storia, come "Invasion of the Mutant Camel". Uno dei più lampanti esempi di come la programmazione, vista non come banale soluzione di problemi contabili ma come processo creativo ed innegabilmente artistico, vada come tale a braccetto con stati di coscienza più o meno alterati. Ulteriori informazioni su questo personaggino tutto da scoprire al sito http://www.llamasoft.co.uk/. Ci piace pensare che lo stesso Remote, con il suo *The Great Pirla's Game*, abbia voluto tributare il giusto omaggio a Minter senza neanche usare sostanze psicotrope, visto che è già fuori di suo.

[150] Interessantissimo, nello scritto di Remote, il rimando alla cabala, ovvero alla Qabbalah: la lingua ebraica è unita a doppia mandata alla matematica dei numeri interi, tanto che non solo ad ogni lettera dell'alfabeto ebraico è associato un numero, ma ad intere parole è associata la somma delle lettere componenti e parole logicamente connesse tra loro sono anche legate da vere e proprie equazioni. L'Universo stesso trova origine in una sorta di processo combinatorio, dove il Nome di Dio – o meglio la moltitudine dei possibili Nomi, per loro natura non conoscibili o pronunciabili anche se regolati da precise regole matematiche – ha importanza suprema e definitiva. Le implicazioni filosofiche di tale natura linguistico–matematica sono sconcertanti: si rimanda il lettore interessato a "Il Nome di Dio e la teoria cabbalistica del linguaggio", Gershom Sholem, Piccola Biblioteca Adelphi.

[151] Col senno di poi e visto che parliamo di un epoca dove i programmatori professionisti erano dei pionieri, probabilmente era un baco del compilatore/interprete del linguaggio usato da Jeff. O forse no…

```
if (2 + 2 == 5)
      then <codice per correggere il problema>
```

Io chiamo "Approccio Umanistico" questa attitudine al Debugging.
Come ben sai io non ci capisco una mazza di numeri, ma non per questo mi fermo. Laddove le cifre tacciono, i Grandi Maestri del pensiero umanistico da Socrate in poi mi vengono in aiuto.

Possiamo noi, in balia di questi terremoti simbolici, sperare in una Pentecoste Informatica ove tutto questo GC finalmente si ricomponga in una lingua primeva e tutti i nostri scatolotti finalmente inizino come per magia a funzionare correttamente? Alcuni segnali ci sono: dopo molti tentativi più o meno falliti si è effettivamente imposto all'attenzione mondiale un ennesimo linguaggio (il già citato Java, appunto) che funziona allo stesso modo su tutti i computer[152] e sono stati prodotti dei processori in grado di emulare tutti gli altri[153], mentre ricerche come quella sul *Grid Computing* mirano a porre le basi per una comunicazione trasparente tra piattaforme di diverse marche e sistemi operativi in modo che l'utente abbia la sensazione di lavorare con un solo gigantesco computer. È ovvio che ognuno di questi tentativi è destinato a naufragare miseramente.

MEGAPIPPONE JARON LANIER??

## I Linguaggi di Descrizione

Categoria completamente diversa e da non confondere con la precedente è quella dei Linguaggi di Descrizione, che anziché specificare il *comportamento* di qualcosa ne specificano l'*aspetto*. Di questa classe di linguaggi fa ad esempio parte l'HTML, usato per "scrivere" le pagine del WWW[154]. L'HTML in sé non è altro non è che un linguaggio di descrizione di ipertesti, ovvero una serie di comandi che, interpretati da un apposito programma (un browser come Firefox o Explorer) specifica informazioni per il piazzamento di immagini e testi su una pagina vuota, colori, grandezze dei caratteri e il

[152] I programmatori Java a questo punto possono ridere sguaiatamente. Gli altri sappiano che questa è l'idea del P-Code precedentemente accennato: Java comprende, nelle sue specifiche, un *processore virtuale*, ovvero una macchina in grado di eseguire codice Java compilato e che in realtà non esiste. Questa deve essere quindi sviluppata dal produttore che desideri supportare Java sui suoi sistemi. In teoria, quindi, lo stesso programma Java compilato funziona in maniera analoga in diversi ambienti e sistemi operativi.

[153] Come il Crusoe, nato dalla mente di Linus Torvald – che peraltro incontreremo ancora per motivi completamente diversi.

[154] Molti direbbero "Internet", confondendo servizio e infrastruttura: Internet sono "solo" vari milioni di computer collegati tra loro, mentre il World Wide Web – come la posta elettronica, le chat, i giochi online etc – sono solo una delle migliaia di cose che si possono realizzare appoggiandosi a Internet. Quindi vi prego: non dite più "ma tu ce l'hai Internet? Io si!" quando volete fare i galli in zona aperitivo: la frase non ha senso. State parlando della posta elettronica, di un sito WWW personale, della possibilità di consultare la biblioteca del Vaticano o cosa altro?

fatto che quando selezioniamo un *link* andiamo a finire in un altro documento che può essere memorizzato sul nostro computer o sul disco di un server dall'altra parte del mondo. Se invece volete anche far *eseguire* qualcosa ad una pagina HTML dovete inframmezzare dei pezzi di codice scritti in un linguaggio di programmazione – tipicamente Java, JavaScript, Perl o altre svariate possibilità – oppure usare tecnologie esoteriche come ActiveX , che permette di infilare "oggetti" eseguibili direttamente nell'HTML.

L'HTML non è altro che un caso particolare[155] di un linguaggio molto più generico chiamato XML, che sembra aver messo tutti d'accordo e che si è imposto in questi ultimi anni. Speriamo in bene: in effetti le premesse ci sono, grazie al fatto che l'XML è in grado di rappresentare *qualsiasi* linguaggio di descrizione e può quindi essere usato per descrivere qualsiasi cosa[156].

Anche nel caso dei linguaggi di descrizione, specifiche incomplete possono dare luogo a bug singolari: chiunque abbia provato a creare un documento HTML sufficientemente complesso sa benissimo che lo stesso viene visualizzato in maniera diversa dai vari browser. Quello che su Firefox è un carattere alto un centimetro può essere alto solo otto millimetri su Explorer, mentre gli algoritmi per impaginare il testo o andare a capo possono essere diversi poiché nessuna specifica precisa viene data in tale senso. Dovendo solo visualizzare semplici documenti questo non sarà un problema, ma se noi fossimo dei grafici o dei tipografi che quelle pagine le devono far stare in un foglio A4 – ovvero in qualcosa che ha delle caratteristiche e dimensioni fisiche ben precise – la cosa ci procurerebbe non pochi mal di testa. In effetti i programmi per tipografia sono ben altra cosa in quanto a complessità...

*[minchia, qui ci vorrebbe un disegno, ma non mi viene in mente nulla.]*

## Chi Scrive il Software?

*Deliver yesterday, code today, think tomorrow.*

Dopo aver esaminato la Materia Della Quale È Fatto Il Nemico, si arriva naturalmente a riflettere su una ulteriore grande fonte dalla quale il Nemico attinge copiosamente la propria linfa vitale. Questo aspetto, solitamente trascurato da tutti i manuali di informatica, affonda le sue radici – ovviamente – in un ennesimo Sistema Complesso: quello delle relazioni umane. Vedremo come aspetti psicologici, sociologici, comunicativi e altri ancora contribuiscano alla creazione di un buon prodotto oppure alla genesi di un coacervo di Indomiti bug. Non dobbiamo stupirci: è questo materiale *umano* che, alla fine, dopo essere stato opportunamente assemblato, istruito e guidato, sviluppa

[155] L'XML può essere usato per descrivere il linguaggio HTML – ma non viceversa, così come un cretino può scrivere un saggio – ma non viceversa.
[156] Anche se l'impresa di descrivere in XML la Zia Carmela può essere a tutti gli effetti essere definita "titanica".

tutto quello che serve ai nostri computer per funzionare. Per assonanza con i concetti di Hardware e Software, questo materiale viene chiamato, in maniera intraducibile, *Peopleware*[157].

Il peopleware riflette esattamente tutte le idiosincrasie che già abbiamo avuto modo di esaminare nei campi dell'hardware e del software. È quindi possibile individuare una vasta casistica di situazioni-tipo, che danno luogo a dei problemi-tipo, che danno luogo a delle soluzioni-tipo, rigorosamente sbagliate, e quindi a dei bug-tipo.

Come per gli argomenti precedenti, conoscere questi meccanismi è la strada per poterli evitare o quantomeno minimizzare, riducendo così la possibilità che il Nemico si manifesti. In particolare, pur non potendo ovviamente competere con un manuale di *Team Building*, questa parte del libro è focalizzata sul problema dello Sviluppo visto come una macro-attività ed è quindi rivolta a chi deve *organizzare* il lavoro, che solitamente non coincide con chi deve *eseguire* lo stesso. Si noti che è molto più difficile la prima delle due attività:

> *L'Intelligenza si può simulare. L'Organizzazione no.*

Ovvero, uno sviluppatore può barare e far credere di essere più in gamba di quanto sia realmente[158] per un periodo più o meno lungo, ma se un team manager è un incompetente c'è ben poco da fare per nascondere la cosa.

Inizieremo quindi la nostra analisi del *Problema Peopleware* con lo studio di uno dei componenti chiave della discussione: lo *Sviluppatore*, appunto. In particolar modo esamineremo il suo comportamento sociale, cercando di desumerne le caratteristiche più importanti e dando modo a chi deve organizzare un gruppo di sviluppo di pescare dal mazzo le carte più utili[159]. Ma prima...

**Sesso, Finalmente!**

---

[157] Si noti che in questa fase ci concentreremo sullo sviluppo del software, poiché l'analisi dei meccanismi produttivi di altro genere è ben nota e quindi l'autore poco avrebbe da aggiungere.

[158] I curricula degli sviluppatori sono un tipico esempio di questo fenomeno, tanto che il modello "Braccia Rubate all'Agricoltura" fa parte della nostra analisi e viene citato in seguito.

[159] Che non sono necessariamente le migliori: pescare sempre l'Asso di Bastoni alla lunga è controproducente, oltre ad essere noioso. In ogni caso, anche in un gruppo di sviluppatori fuoriclasse, solo alcuni di questi saranno davvero dei fuoriclasse utilizzabili e gli altri saranno dei cretini, come sancito dalla Regola 80-20.

Dai, un po' di sesso ci vuole, altrimenti questo libro rischia di essere *veramente* noioso. E allora chiediamoci perché la maggior parte dei programmatori sono maschi. È una domanda inerente al sesso, no?

Una teoria interessante – anche se forse vagamente maschilista – è che il computer abbia dei tratti femminili nella sua imprevedibilità, nel suo comportamento tutt'altro che lineare, nella sua continua richiesta di attenzioni... insomma:

> I programmatori son tutti maschi perché *le* computer son tutte femmine.

Quindi non stupitevi quando sentirete qualcuno parlare dei computer al femminile.

Una seconda ipotesi è invece che le donne non son mica sceme per fare quel lavoro lì.

## Lo Sviluppatore di Software

Esaminare lo Sviluppatore nei suoi comportamenti e nelle sue relazioni col mondo esterno è un lavoro titanico che solo in maniera superficiale possiamo affrontare in questa sede, rimandando chi sia veramente interessato a fare luce su questa figura emblematica dei tempi moderni a più eruditi studi di psicologia, sociologia e realtà romanzata[160]. Cionondimeno questi brevi cenni saranno più che sufficienti a farci capire meglio perché gli stramaledetti computer non funzionino.

### Sviluppatore Ideale: Lo Hacker (Che Bontà!)

Poiché per qualsiasi analisi è necessario stabilire delle unità di misura, ecco la descrizione di quello che è considerato l'equivalente umano della Macchina di Turing, vera e propria pietra angolare della Creazione Tecnologica. Lo Sviluppatore Ideale ha solo qualità e nessun difetto: ad esempio, ha conoscenze profonde sia di hardware che di software, il che gli permette di mettere le mani praticamente ovunque. Lo sviluppatore ideale è un *Hacker*, termine oltremodo abusato. Finalmente lasciamo la parola a Babele Dunnit:

> Il Nemico da sempre, per ignoranza o calcolo, cerca di dipingere a fosche tinte quel mondo di cosiddetti disadattati che parlano i Linguaggi delle Macchine. Cosi' gli unici

[160] Imperdibile a questo proposito "Microservi" di Douglas Coupland.

hacker dei quali il Nemico riferisce non sono certo gli stessi del libro di Steven Levy – pionieri e padri fondatori che a colpi di commutatore esadecimale hanno posto le basi per arrivare a questa cosa elettrica che ho davanti e che in questo momento crede di essere una macchina per scrivere. No, il Nemico parla sempre e soltanto dell'Hacker Cattivo, quello che clona telefonini e carte di credito e copia CD (ma i masterizzatori chi li produce?) privando Microsoft e Michael Jackson dei loro sudati milioni di dollari, mentre sbraita contro Free Software, Open Source e la Libertà stessa. Addirittura non bisognerebbe usare Linux perché è antipatriottico. A questi idioti Babele Dunnit ha solo una cosa da dire:

*Hacking is an Attitude*

L'hacking e' una attitudine. E' una *forma mentis* che *impone* a chi si considera un hacker di capire *come funzionino le cose*. I bambini quando smontano i giocattoli fanno hacking. E lo fanno tutti. Tutti siamo stati hacker. Provate a negarlo. Poi, crescendo, la maggior parte di noi inizia a dare per scontate molte cose – lo spinterogeno, i tiri ad effetto di Baggio, la gravitazione universale, l'acqua gassata, la televisione, la meccanica quantistica, l'aspirina. Altri invece continuano ad essere curiosi e, quando incontrano il Misterioso Meccanismo Elettronico Universale altresì noto col nome di Computer iniziano – o meglio continuano – a farsi delle domande. Ed in questo, signori, non c'è nulla di male.

Quindi, per favore, portate rispetto agli *hacker*. Quelli di cui parlate voi sono banali malfattori. Se proprio avete bisogno di etichettarli in qualche modo, trovate altre parole.

Questa *attitudine* si espleta in modi molto differenti: nel documentario di Ine Poppe *Hippies From Hell*[161] vengono esaminati innumerevoli aspetti dell' *hacking,* non necessariamente legati alla sfera elettronico/informatica. Mettere del cellophane colorato sugli schermi dei primi *Space Invaders*, piuttosto che imparare (ed insegnare) ad aprire lucchetti e serrature – a volte con facilità francamente imbarazzante per i costruttori delle stesse – sono *hacks* in piena regola. Quando uno in trenta secondi apre la serratura del tuo super-cordone-anti-scippo-di-computer-portatile-fatto-di-kriptonite con un pezzo di cartone arrotolato non puoi non offrirgli una birra[162].

Il Riglio, ad esempio, è un hacker. Il Riglio non compra un lettore DVD normale: compra una *cosa* che *sembra* un lettore, ma ha un disco rigido, una interfaccia di rete e Linux

[161] Si tratta di un video della durata di una cinquantina di minuti, reperibile in forma assolutamente gratuita su Internet.

[162] "Mucking around is fun; taking things apart to see how they work is an indispensable part of any craft", Marc Laidlaw, nella prefazione di *Gaming Hacks*, O'Reilly 2005. Lo stesso nome della serie *Hacks* di O'Reilly, forse il più prestigioso editore di libri tecnici al mondo, è un tentativo di riabilitare quel tanto bistrattato termine, documentando e cercando di passare alle nuove generazioni l'etica e la partecipazione creativa dell'attitudine hackeristica e smanettatoria.

come sistema operativo. Il manuale del lettore DVD del Riglio, fornito in formato rigorosamente elettronico sotto forma di file PDF, è *mostruoso*:

Babele: Poi il tuo discepolo lo ha dato l'esame di Informatica Esoterica?
Riglio: Si, ha dato "Lettore DVD 2".
Babele: Come è andata?
Riglio: Bene, però ancora non ha capito come si fa a rippare un DVD in multitasking mentre patchi il kernel e grabbi un canale criptato su un disco remoto. E poi ci sono dei dettagli dell'implementazione dello stack TCP/IP che ancora gli sfuggono. L'esaminatore gli ha detto di tornare fra un mese e lo ha trasformato in un diodo.
Babele: Ah. Forse con un Mojito...?
Riglio: Ottima idea. Max? Due Mojito. Uno lo porto via.

E poi, via, ormai persino su Topolino lo hanno scritto ormai da tempo: 4 Aprile 2000, pag. 35, all'interno della rubrica TopoVillage a cura di Santo Scarcella: "Cracker: Pirata di Internet. Evoluzione negativa dell' Hacker. È un esperto di programmazione che penetra illegalmente in programmi o siti causando gravi disagi". Il fatto che Scarcella lo spieghi addirittura ai bambini non può che fare onore a lui e mandare tutti gli hacker in loop dalla gioia gioia gioia gioia gioia gioia gioia gioia gioia gioia gioia gioia gioia <CTRL-C>.

D'altra parte, come non ricordare con reverenziale timore quello che è considerato il più grande intrusore di tutti i tempi, quel *Condor* – al secolo Kevin Mitnick – che tenne in scacco per più di due anni FBI e NSA e che venne catturato il 17 Febbraio 1995 solo quando un altro gigante, Tsutomu Shimomura, decise di fargli pagare un imperdonabile Sfottò Di Natale[163]? Ora Mitnick ha pagato il suo debito con la giustizia e gira il mondo tenendo conferenze sulla sicurezza informatica, ma non si può negare che la sua figura sia una leggenda al di là delle sue azioni più o meno nefaste. Proprio per questo è assolutamente importante chiamare le cose con il loro nome: oltre ad essere stato un hacker sovraumano, Mitnick è stato un pirata e un pericoloso fuorilegge ben lontano da quegli ideali alla Robin Hood che a volte muovono qualche suo giovane emulo.

[163] Il giorno di Natale del 1994 Mitnick, già autore di imprese impossibili come il furto dei sorgenti del sistema operativo VMS della Digital dai server dell'azienda produttrice e l'addebito dei propri conti telefonici su quelli di tale James Bond, depositava sulla macchina di Shimomura – guru della sicurezza del San Diego Supercomputer Center – due files audio nei quali lo prendeva pesantemente in giro. L'attacco, perpetrato mediante una tecnica cosiddetta di "IP Spoofing", durava 16 secondi ed è tuttora considerato un vero classico, capace di commuovere anche un sistemista Windows. Shimomura, dopo aver stretto le opportune alleanze, riusciva a rintracciare il *Condor* in un residence nel Nord Carolina utilizzando un furgone attrezzato con radiogoniometri. Fino a quel momento ogni tentativo di risalire a Mitnick per vie "normali" era stato semplicemente inutile: il *Condor* usava un solo computer collegato a *due* cellulari, mentre il software di connessione scritto da lui cambiava *routing* ogni 15 secondi. Potete trovare questo e molto altro ancora in Rete, digitando "Kevin Mitnick" sul Vostro Motore Preferito o puntando direttamente su www.kevinmitnick.com.

Ora dovrebbe essere più chiaro perchè il nostro Sviluppatore Ideale *deve* essere un hacker. Ma non solo: deve avere un sacco di altre qualità. Non solo sa le cose: è anche in grado di spiegarle in maniera comprensibile, il che lo rende ideale come capoprogetto. È sempre aggiornato e sa valutare soluzioni tecnologiche ottimali e spesso rivoluzionarie ad ogni dato problema senza andare a sbattere il naso in vicoli ciechi; parla perfettamente varie lingue ed in particolare l'inglese tecnico, conosce decine di linguaggi di programmazione e tutti gli stramaledetti sistemi operativi che abbiano mai infettato questo Atomo Opaco del Male. Ha una tale esperienza che, anche messo davanti a situazioni delle quali nessuno capisce nulla, riesce in poco tempo a sviscerare e risolvere qualsiasi problema, anche grazie ai contatti con altri Sviluppatori Ideali pronti a dargli una mano nel nome di un superiore Ideale di Fratellanza Computazionale. Non si lamenta mai, perché *ama* il suo lavoro. Lo Sviluppatore Ideale non riesce a distinguere tra gioco e lavoro, tra vita privata e professionale, tra giorno e notte; tutti gli altri sviluppatori "normali", mentre divorano i suoi listati con le lacrime agli occhi come se stessero leggendo delle poesie di Neruda, non riescono a capire se quello che crea sia ancora qualcosa di tecnico – anche se tanto avanzato da risultare incomprensibile e miracoloso – o, piuttosto, una Nuova Forma di Arte.

Insomma, lo Sviluppatore Ideale non *usa* le Macchine. Lo Sviluppatore Ideale *parla* con le Macchine, e queste gli rispondono *Yes*.

È ovvio che lo Sviluppatore Ideale non esiste; è qui citato solo come unità di misura. In particolare, così come la ipotetica Macchina di Turing dispone di una quantità infinita di memoria, lo Sviluppatore Ideale sviluppa codice corretto (i.e. *non bacato*) per definizione[164].

## Modelli Di Aggregazione dello Sviluppatore Reale

Avendo ora a disposizione un termine di paragone, possiamo analizzare il comportamento dello sviluppatore a livello di branco.

### Lo Smanettone Sottovuoto

Questo tipo di figura professionale è molto interessante e molto comune, soprattutto nei giovani. Detto anche *Campana de Veder*[165] a causa del suo habitat ideale, è questo un animale che tende ad identificarsi con lo Sviluppatore Ideale come abilità tecnologica e fuoco sacro, ma ne è completamente agli antipodi per tutti gli altri aspetti. La definizione

[164] Edsger Dijkstra nel suo "A Principle of Programming" spiega come sia possibile produrre codice scevro da bug dimostrandone la correttezza formale. Addirittura, in quest'ottica, i programmi non sono altro che degli "effetti collaterali" della dimostrazione di teoremi matematici, esatti per definizione. A livello teorico la cosa è interessantissima, ma sappiamo benissimo che nel Mondo Reale vince comunque la Seconda Legge della Termodinamica.

[165] "Campana di Vetro" in dialetto lombardo.

deriva dal fatto che questo elemento è da prendere e isolare completamente dal resto della vociante marmaglia, perché in ogni caso non è in grado di spiegare alcunchè a chicchessia e non ne ha peraltro nessuna voglia; anche solo distrarlo per la durata di un caffè rappresenta per lui una perdita di tempo. È questo il personaggio rifilatoci per anni dai media come "tipico ragazzo del computer", *geek* insensibile pressochè a tutto ma pronto a commuoversi davanti a un driver scritto bene.

Lo Sviluppatore Sottovuoto può essere un elemento strategico in un team di lavoro: è quello che riscrive a tempo di record ampi pezzi di programma che non funzionano[166] lavorando diciannove ore al giorno per due settimane e cibandosi esclusivamente di cibi piatti come la pizza o i toast, che gli vengono passati dalla fessura sotto la porta[167].

Invecchiando, lo Sviluppatore Sottovuoto migliora: gli passa un po' di Fuoco Sacro e si rende conto che esiste anche il mare. A volte diventa un Vecchio Programmatore Saggio.

**Braccia Rubate all'Agricoltura**

Questo modello di figura professionale è stato reperito in tracce fossili fin dal Pleistocene e quindi non si capisce perché la categoria degli sviluppatori dovrebbe esserne scevra. Sembra lì per caso e non capisce un cavolo di quello che gli si dice. Glielo si legge nello sguardo, mai sfiorato da un pensiero. È sovente preceduto da un Curriculum Vitae Et Studiorum di dimensioni ragguardevoli, stando al quale ora pochi oggetti contenenti un microprocessore funzionerebbero se lui avesse davvero deciso di dar retta alla madre e fare l'impiegato statale. Ma già alle prime riprese ci si ritrova a rimpiangere questa sua mancata scelta: scrivere l'algoritmo per determinare se un numero è primo o meno ne mette a dura prova le capacità analitiche e, senza un Wizard[168], costui non è in grado neanche di stampare "Ciao, Mondo!" sul video del suo portatile[169]. Insomma, un *Lamer*[170].

Solitamente ce lo ritroviamo tra i piedi grazie alla sua grande capacità di bluff e alle sue conoscenze: preceduto da grandi raccomandazioni e coadiuvato comunque da una certa intelligenza e una faccia di tolla di dimensioni epiche, riesce a farsi assumere per poi slalomare tra lavoro e lavoro come un campione di snowboard. Va da sé che il Nemico trova fertile terreno nella presenza di un simile elemento in un team di sviluppo. Se si può evitare, è meglio; se invece ce lo dobbiamo tenere, è opportuno cercare di limitare i danni

[166] La frase topica è in questo caso "faccio prima a riscriverlo che a spiegargli cosa deve correggere".
[167] Cfr "Microservi", Douglas Coupland.
[168] Ormai da qualche anno di gran moda, gli "wizards" – testualmente "maghi" – sono dei programmi in grado di generare in maniera guidata altri programmi. Ovviamente sono molto specifici e spesso il loro funzionamento è abbastanza approssimativo. La maniera corretta di usarli è nell'ottica di risparmiare tempo, non in quella di far fare tutto a lui perché non si ha voglia di capire cosa succede, perché se qualcosa va male – e qualcosa andrà *sicuramente* male – poi non se ne viene fuori più.
[169] Tipicamente una macchina fantastica davanti alla quale si fatica a trattenersi dallo sbavare.
[170] Termine dispregiativo con il quale nel mondo informatico (ma non solo) si designa chi si vanta di qualcosa senza averne diritti e requisiti.

che un simile elemento può provocare evitando di lasciarlo da solo per troppo tempo e affiancandogli qualcuno che capisca qualcosa di calcolatori, oppure fargli fare lo scassamaroni (venditore maledetto da Dio e dall'Uomo, recupero crediti, guastatore di progetti altrui etc) e non il tecnico.

**Il Vecchio Programmatore Saggio**

È questo un personaggio molto interessante. Se esistesse l'equivalente IT[171] dei vecchi film di Kung Fu, dove c'è l'Allievo sottoposto alle più incredibili prove da un Maestro che ne vuol fare un campione in nome di qualche misterioso disegno superiore, questo sarebbe appunto il secondo. Spesso si tratta di un ex-Sviluppatore Sottovuoto che, dopo aver passato gli ultimi vent'anni a scrivere codice in un sottoscala, raggiunge l'Illuminazione e si trasforma in un Essere Superiore, quasi uno Sviluppatore Ideale. A quel punto è difficile che abbia voglia di scrivere ancora codice, perché preferisce spiegare ad altri come fare, ma quando è costretto produce dei capolavori.
Il Vecchio Programmatore Saggio è l'elemento ideale per coordinare il team di sviluppo: non ha più nulla da dimostrare, sa cosa fare e come farlo, ha voglia e tempo di spiegare ad altri come fare.

**Il Fortunato**

Questo è un genotipo raro, osservato in pochi esemplari prevalentemente nei gruppi di sviluppo molto numerosi. Si tratta solitamente di un personaggio senza lode e senza infamia e tutto sommato abbastanza grigio, ma che gode di una certa fama grazie a una eccezionale dose di fortuna mista ad intuizione. È il classico tipo che dice: "secondo me non funziona perché c'è un cavo rotto" e nessuno gli da retta, fino a quando – più per un destino favorevole che per capacità tecniche e analitiche – si scopre che aveva ragione. A quel punto solitamente cantilena "ve lo avevo detto, ma voi…"

**Il Pasticcione**

Solitamente molto giovane, trattasi di sviluppatore impallinato e fanatico non supportato da una adeguata dose di conoscenze. Se opportunamente istruito – ad esempio affidato ad un Vecchio Programmatore Saggio – può diventare un ottimo elemento; ma lasciato a se stesso non ha il metodo e la pazienza per evolversi in tempo utile e contiunerà a combinare casini per anni. L'unica speranza, qualora uno se lo ritrovi tra i piedi, è armarsi di pazienza, seguirlo nelle sue elucubrazioni e cercare di guidarlo verso soluzioni semplici e funzionali – magari facendogli leggere i libri giusti senza aspettare che li

[171] Information Technology. Acronimo magico sotto il quale viene riassunto tutto quello che ha a che fare con il mondo dei computer, da qualsiasi punto di vista: tecnico, commerciale, filosofico... una delle sigle più inflazionate degli ultimi dieci anni, insieme a "dot com".

scopra da solo – piuttosto che lasciarlo in balia di esperimenti e intuizioni magari qualche volta geniali ma molto dispendiose in termini di tempo, denaro ed energie mentali.

**Il Beta Tester**

Finiamo questo breve bestiario con un modello di sviluppatore molto particolare: il *Beta Tester*. Questo animale, caratterizzato da forti dosi di incoscienza e/o masochismo, per vari motivi ha la necessità di utilizzare delle cosiddette *versioni beta* di software altrui. Queste non sono altro che revisioni preliminari di un certo programma, libreria o sistema operativo rilasciate ad un manipolo scelto di kamikaze digitali affinché ne verifichino efficacia e funzionalità prima del rilascio definitivo sul mercato.

Va da sé che, se il software normale è bacato per definizione, le versioni beta dello stesso sono quanto di più vicino all'Inferno un programmatore possa immaginare nei suoi peggiori incubi.

Lasciamo ancora una volta spazio a Babele Dunnit e a una sua lunga, delirante digressione:

### *Le Storie Di Babele (X): Transustanziazione Digitale*

Ho un grosso problema, gente. Ho finito il corpo. Cercherò di spiegarmi: ho vissuto la maggior parte della mia vita guardando dritto il pericolo attraverso un terminale a fosfori verdi. È passato un quarto di secolo da quando ho iniziato a programmare su un IBM Serie 1 ed ho toccato senza paura gli ZX80, i 64 e gli Spectrum, gli Amstrad e gli Apple II, gli Atari ST e gli Amiga e i primi PC 8088, veloci come le Poste Italiane di quei tempi. E gli Sperry-Univac che montavano Exec-8 come sistema operativo, e i VAX e il VMS e il PC-M, Santa Cruz Operation e Novell Netware e in mezzo a tutto ciò ho visto venire alla luce il DOS, zitto e storto e claudicante, senza che il Nuovo Mondo Elettrico trovasse nei paraggi l'equivalente informatico di una Rupe Tarpea. Ho imparato decine di linguaggi. Sono passato dall'assembler 6502 e Z80 al Prolog senza disdegnare Forth e Lisp. Ho dovuto, gente, **dovuto** usare il Cobol. Per lo shock ho poi parlato in Basic per lunghi anni: 10 PRINT "Scusa, mi passi l'acqua?". Poi il Pascal e C e C++ e la programmazione object-oriented e sono finito a giocare con Python come Cicciolina e adesso c'è chi mi reputa un depravato solo per come uso le Funzioni Lambda. E, non ricordo neanche più quando, ho iniziato a fare il Beta Tester. Non ne ho potuto fare a meno, Turing solo sa se ho cercato di smettere, ma non c'è stato nulla da fare. Ho visto bachi che voi umani non potreste neanche immaginare. Ho visto navi spaziali al largo di Orione schiantarsi una dietro l'altra perché avevano montato la versione sbagliata del kernel e il driver del timone fotonico di coda andava in conflitto con le vele solari. Le navi si giravano di centottanta gradi e via, dritte nel cuore del sole come falene di teflon. Poi sono arrivate le Realtà Virtuali. I caschi, i guanti… Come avrei potuto resistere?

Sono andato in giro per mesi con un casco sulla testa. Spento. Tutti i muri erano miei. Fortunatamente erano i primi tempi: Virtuality. Quelli si che erano caschi, gente. Dei carapaci ipertecnologici di plastica e metallo da tre chili e mezzo l'uno. Non quei ridicoli Personal Displays da duecento grammi senza alcun rispetto per la sicurezza che son venuti dopo. Avete presente cosa vuol dire avere tre chili e mezzo che ti gravano sulla Vertebra Atlante? OK, stai male per forza – altro che CyberSickness, così lo chiamano in America il "mal di ciberspazio" – ma la tranquillità e le cervicalgie che ti dà uno di quei vecchi elmetti in lamierino del sette è tutta un'altra cosa. Come una lavatrice di trent'anni fa. Nel ciberspazio puoi superare la velocità della luce e andare dritto contro un muro di mesoni appuntiti con un HMD come quello (Bambini, non tentate questo esperimento a casa da soli). Poi è arrivata tutta la tecno-paccottiglia. Cellulare, agenda elettronica, macchina fotografica digitale, CD player portatile antishock (utilissimo come arma di difesa), navigatore tascabile, calcolatrice grafica con risolutore di sistemi di equazioni differenziali… In particolare ho trovato estremamente utile l'ultimo oggetto citato. A tutti capita quasi giornalmente di risolvere sistemi di equazioni differenziali, ad esempio per decidere il rapporto tra quanto pane e quanto prosciutto comprare in base alla velocità di ingurgitazione del prosciutto stesso da parte del nonno, che notoriamente mangia meno pane che companatico. In seguito la tecno-paccottiglia ha iniziato a diventare sempre più piccola ed intelligente. Il mio primo cellulare era un bel mattoncino. Il terzo era poco più di una supposta. Tutti vibravano: ho sempre trovato questa cosa a metà tra l'inquietante e l'osceno. Per quanto riguarda l'intelligenza, la mia agenda elettronica ha iniziato a lamentarsi perchè non mi riferivo a lei usando la prima persona plurale e comunque ricordo che chiese ufficialmente di essere chiamata Il Suo Bel Personal Digital Assistant, non "agenda elettronica". Anche le agende hanno un'anima. Di metallo. Ora siamo arrivati alla ennesima, e non credo neanche ultima, iterazione di questa ibridazione tra silicio e uomo. Io chiamo questo processo "elettrodarwinazione". Ora le paccottiglie elettroniche ce le infiliamo dentro. E io, da beta tester qual sono, non mi tiro certo indietro. A diretto contatto con l'osso mascellare mi sono fatto impiantare un player MP3. Sento la musica per propagazione diretta attraverso la scatola cranica, senza passare dai timpani. Devo dire che è un bell'effetto. Per caricare nuova musica nel player mi infilo una T1 nell'orecchio e, come un monaco tibetano, recito a memoria la sequenza di startup del protocollo di connessione con ITunes. Al posto del peggiore dei miei due occhi mi sono fatto trapiantare una Webcam di quelle con night vision, indirizzo IP cablato e uscita FireWire. A me il Grande Fratello mi fa un baffo. BabeleCam è molto di più: un reportage in tempo reale di tutto quello che faccio, ventiquattr'ore su ventiquattro, ma in soggettiva. Certe cose, come ad esempio quando mi guardo allo specchio dicendo "è un gran bel fustacchione!", sono dure da mandare giù anche per il pubblico più degenerato. Sono sei mesi che videoregistro tutto. Mi sono fatto impiantare un disco rigido miniaturizzato da due terabytes sotto l'ascella destra. Puzza un po' di bruciato quando supera il milione di giri al minuto ma è un giusto prezzo da pagare. Poi venderò tutto al migliore offerente. Per quanto tu possa sforzarti di produrre la più inguardabile ed inascoltabile porcheria, c'è sempre qualcuno che l'apprezzerà. E non sto parlando del Grande Fratello adesso, parlo dei Telegiornali – di TUTTI i Telegiornali. Poi mi sono fatto installare una pompa cardiaca a turbina. Adesso al posto di passare il Folletto vado in giro per casa ad aspirare la polvere con la Vena

Safena. E poi apparecchi acustici, stimolatori contro il mal d'auto, bindelle laser a scomparsa incorporate nel pollice. Non si sa mai cosa può succederti, bisogna essere sempre pronti. La cosa peggiore è quando le batterie intelligenti ti fanno le domande sceme. Preferivo le batterie sceme che almeno stavano zitte. Dovendo andare dal dentista, già che c'ero, mi sono fatto installare un molare UMTS, una delle cose più inutili del secolo. Adesso ho accesso a Internet, ma solo a siti stomatologici. Quando mangio qualcosa a base di pasta di mandorle il molare mi crasha per i troppi zuccheri e devo spegnerlo e riaccenderlo. Ho provato a installare un service pack, ma il dente si è trasformato in un fanone di balenottera azzurra e ho dovuto reinstallare tutto il MOS (Mouth Operative System) da capo con un clean install, che per inciso si chiama così perché prima devi lavarti con un apposito spazzolino i cui requisiti minimi sono 2.5 GHz, 512 MB RAM, 40GB HD e setole rinforzate di Cinghialino Pupazzino Trudino. L'ultima frontiera è il cervello: mi sono fatto sostituire parte del cervello con un pezzo di encefalo sintetico collegato via radio all'originale. è divertentissimo, perchè quando ci penso non so più se sono qui o sono là. L'idea me l'ha data un racconto di fantascienza, ma non pensavo potesse funzionare. Insomma, ecco, ormai ho sostituito tutto il sostituibile e ho finito il mio corpo… Uh! Un gatto! Micio micio micio… ho giusto delle unghie per gatti in titanio da provare. Solo che la documentazione fa un po' schifo e non ho capito se in titanio devono essere le unghie o i gatti. Dura la vita del Beta Tester.

Speriamo, con questa breve carrellata, di aver coperto i più comuni casi limite della specie animale degli Sviluppatori. Però, ora che abbiamo una idea più chiara del materiale umano a nostra disposizione, cosa gli facciamo fare? E come? Come possiamo sperare di organizzare una simile congerie di disgraziati?

## Impara a Sviluppar (trallallàlalallallallà)[172]

Oltre all'assemblaggio e sfruttamento del branco degli sviluppatori da parte del loro Signore e Padrone è molto importante la metodologia di sviluppo che si intende adottare. Verranno in questo paragrafo poste le basi per dimostrare più avanti come, paradossalmente, gli approcci più commerciali siano intrinsecamente quelli che massimizzano la gravità dei bug, a causa di un meccanismo che chiameremo *Principio di Invarianza dei Difetti*. Questo sancisce che, sempre e comunque, una certa quantità di errori verranno compiuti; il meglio che possiamo fare è cercare di crearli dove questi si rivelino il meno gravi possibile, o dove sia più facile scovarli.

Per metodologia di sviluppo si intende quella serie di regole che cercano di dare una parvenza di ordine alla sequenza sconclusionata di operazioni che un branco di sviluppatori compie nella sua anarchica interezza allo scopo di giungere, prima o poi, a

[172] Sull'aria di "Impara a Fischiettar".

un risultato di qualche tipo. Senza stare a tirare in ballo Yourdon o i diagrammi UML onde evitare di entrare troppo nel dettaglio e rischiare di spaccarci la faccia in due sbadigliando con veemenza, diciamo che esistono le seguenti metodologie di sviluppo:

- Sviluppo Alla Maiala
- Metodologia Classica
- Metodologia Moderna

Per capirci meglio, faremo anche dei paragoni musicali.

**Sviluppo Alla Maiala**

Così come, prendendo un numero infinito di scimmie e mettendo loro a disposizione una quantità infinita di RAM queste si prenderanno talmente sul serio da pubblicare una rivista tutta loro, un branco di sviluppatori allo stato brado lasciati alle loro mansioni daranno luogo ad uno *Sviluppo alla Maiala*.

È questa una situazione analoga all'Heavy Metal: là vince chi ha l'amplificatore più potente e le dita più veloci, qui chi scrive più codice più in fretta, quindi chi ha le dita più veloci e basta. Si noti che in entrambi i casi il cervello non viene utilizzato in alcun modo. Lo Sviluppo alla Maiala è caratterizzato dall'assenza di regole, standard, orari e quant'altro: ognuno fa quello che deve fare nei modi a lui più consoni e con gli strumenti che predilige; poi a un certo punto ci si trova intorno a un tavolo e si dice: Oh, io ho fatto così, io ho fatto cosà e si cerca di mettere insieme a martellate tutto il sistemone. Nello Sviluppo Alla Maiala sono assolutamente vietate le seguenti operazioni:

- Commentare il codice affinchè gli altri capiscano a cosa serve[173].
- Vedere se qualcuno quella roba lì l'ha già fatta.
- Leggere la documentazione, i manuali o qualsiasi altra cosa possa aiutare a districare la matassa.
- Dare un'occhiata agli esempi.
- Chiedere consigli a chi ne sa di più.
- Fermarsi a pensare alla soluzione migliore e fare qualche schemino con carta e matita.
- Verificare periodicamente il funzionamento di quello che si sta facendo.
- Debuggare subito[174].
- Farsi la barba o in generale prendersi cura del proprio corpo.

Vediamo cosa dice Babele Dunnit su questo ultimo punto:

| |
|---|
| |

[173] In particolare, il Programmatore Klingon afferma che "un VERO Programmatore Klingon NON commenta MAI il suo codice".
[174] Si deve invece posporre l'attività il più possibile in modo che lo stesso bug si ripresenti più e più volte.

DeoX mi guarda e dice “Non si direbbe mai che tu sia uno che debugga. Hai sempre la barba fatta. Chi debugga non ha tempo per farsi la barba”. Vero. Quello che DeoX non sa è che io ho semplicemente deciso di non perdere più tempo a farmela crescere, così occupo i preziosi istanti guadagnati dalla mancata barbazione in una sana attività di debugging.

Già ad una superficiale esegesi del testo balza all’occhio che l’attività di debugging è da Babele Dunnit definita “sana”; non siamo quindi in uno stato di Sviluppo Alla Maiala. Cionondimeno l’avversione al farsi la barba risulta evidente.

Nello Sviluppo alla Maiala il Nemico trova terreno estremamente fertile sia a livello di singoli codici che a livello di integrazione dei vari pezzi del prodotto, a causa delle interpretazioni estremamente personali che ogni programmatore, per quanto vicino alla figura dello Sviluppatore Ideale[175], attribuisce alla Realtà Circostante. È quindi da evitare come la peste, a meno che stiate raccogliendo materiale per scrivere la seconda edizione di questo libro.

**Metodologia Classica**

Qui si tenta di imbrigliare la carica vitale degli sviluppatori, il loro espressionismo spinto, la loro linfa creativa entro schemi ben precisi che non possono non ricordarci un certo Mozart dei primi tempi, forzatamente scolastico e contrappuntistico, piuttosto che un Bach non ancora illuminato dall’Arte della Fuga.

Ci sono varie metodologie che ricadono sotto questo ombrello classicista, ma tutte, opportunamente farcite da diagrammi con i nomi più esotici[176], prevedono più o meno lo stesso percorso:

1. Un Cliente chiede qualcosa che non può esser fatto a un Account che non è in grado di capire subito che quel qualcosa non può esser fatto (User Request).
2. L’Account passa la richiesta ad un Gruppo di Analisi che capisce che quel qualcosa non può essere fatto e mette per iscritto le modalità con le quali quella cosa non funzionerà (Specifiche Funzionali).
3. Il Gruppo di Analisi passa la richiesta al Gruppo di Sviluppo, che mette per iscritto quali soluzioni sbagliate intende adottare dal punto di vista tecnico per massimizzare le probabilità di fallimento (Specifiche Tecniche o Architetturali)
4. Il Gruppo di Sviluppo sviluppa qualcosa che con le sue stesse specifiche scritte non c’entra nulla, in modo da mandare fuori sincronismo la documentazione e renderla immediatamente inutile.

[175] Che invece percepisce il Mondo nella sua interezza e senza errore alcuno.
[176] PERT, GANNT, UML... sembrano gli ultimi spasimi di un affogato, per non parlare poi dei diagrammi Ishikawa – dal nome del creatore Kuaouru Ishikawa, guru giapponese della Qualità Totale Globale Universale – che precedono di pochi secondi un onorevole suicidio rituale.

5. Il Gruppo di Sviluppo e il Cliente svolgono dei test, che falliscono. Il Cliente va su tutte le furie, ma in ogni caso non cambia nulla, perché quanto sviluppato non coincide comunque con quanto richiesto dal Cliente e quindi il fatto che non funzioni è assolutamente irrilevante.
6. Si tira un dado. Se viene il sei, qualcuno che non c'entra assolutamente nulla si becca una bella lavata di capo e viene eventualmente trasferito. Se viene uno, c'è la possibilità che si vada tutti a casa e non sia necessario tornare lì il lunedi seguente.

È interessante vedere come questa sequenza, soprattutto nelle fasi finali, ricordi da vicino le celeberrime "Sei Fasi Di Ogni Progetto":

Entusiasmo
Disillusione
Panico
Ricerca del Colpevole
Punizione dell'Innocente
Onore e Gloria a Chi Non Ha Partecipato

Questa è la metodologia più diffusa e normalmente l'unica che venga considerata a livello di Grande Azienda. Il lettore dovrebbe cominciare a capire perché gli stramaledetti computer non funzionano.

Infatti, implicito nel modello di cui sopra, troviamo il concetto di *Deadline*. La deadline è la data ultima entro la quale *tutto* deve essere pronto e funzionante. Noi sappiamo che questo non è possibile; si dimostra quindi per *reductio ad absurdum* che, qualora esista una scadenza, esisterà anche una manifestazione del Nemico[177]. Il Problema non è tanto quindi la metodologia in se, quanto la deadline, dettata anche per motivi economici. Questo contribuisce a fare del Software Classico un humus particolarmente gradito al Nemico.

**Metodologie Alternative**

Dato per scontato che non esiste una metodologia di sviluppo universale e a prova di bomba, bisogna citare interessanti esperimenti che, in situazioni molto circostanziate, risultano essere piuttosto soddisfacenti. Sono questi in qualche modo paragonabili a contaminazioni musicali come "Miles Runs The Voodoo Down"[178], o a certa musica moderna tanto geniale quanto spesso incomprensibile. Un buon esempio è l'Extreme

[177] Corollario: più la scadenza si avvicina, più la manifestazione del Nemico diventa particolarmente subdola e difficile da scoprire ed eliminare. A poche ore dalla scadenza ci ritroveremo così in situazioni che hanno invariabilmente del felliniano.

Programming (e le sue evoluzioni come l'Agile Development), del quale citiamo parte del razionale e un paio di esempi rimandando il lettore interessato al sito Internet, facilmente googlabile[179].

Il concetto alla base dell'XP[180] è presto detto: cercare di sfruttare il meglio dello Sviluppo alla Maiala, il suo slancio creativo, evitando nel contempo gli evidenti punti deboli dello stesso e delle metodologie classiche. Ad esempio, per evitare che ogni sviluppatore stia Solo Sul Cuor Della Terra[181], viene utilizzato il Pair Programming: due sviluppatori stanno davanti allo stesso computer, in maniera analoga ad un "rally". In questo modo, mentre il "pilota" è concentrato sulle immediate vicinanze del problema, quindi su un contesto molto limitato, il "navigatore" riesce a mantenere la visione globale e a vagliare soluzioni alternative. Poi ci si va a schiantare allegramente insieme, ma questo è un altro discorso.

Un altro punto sul quale l'Extreme Programming è molto chiaro è quello della documentazione: è assolutamente inutile perdere tempo a scrivere documenti che comunque verranno letti "dopo" – ammesso che lo siano mai e che la parola "dopo" in questo caso abbia un senso; comunque, come si è visto nell' analisi della Metodologia Classica al precedente paragrafo, questi documenti per definizione non rispecchiano ciò che è stato effettivamente sviluppato. Ergo: secondo i dettami di XP la documentazione si scrive a posteriori, una volta per tutte.
Altre particolarità di XP vanno dal fatto che tutti si fidano di tutti (niente più password da dimenticare, il codice scritto e abbondantemente commentato da Caio è liberamente modificabile sia da Tizio che da Sempronio) al continuo test di quello che si produce, in modo da essere sicuri, con le modifiche che il proprio lavoro ovviamente implica, di non rompere le scatole a qualcun altro. E molto altro ancora.

È ovvio che una simile metodologia non è applicabile nella Grande Azienda: nell'Extreme Programming il team deve essere veloce, pronto ai cambiamenti, affiatato e motivato e deve avere svariate altre qualità che per definizione mancano alle elefantiache multinazionali delle quali non faremo i nomi. Però sarebbe bello.

---

[178] Pezzo di Miles Davis tra i primi a contaminare il jazz mediante sonorità elettriche e che per molti segna il momento della nascita del jazz-rock che così tanto influenzerà la musica moderna. Mirabile esperimento di anarchia musicale in qualche misterioso modo auto-organizzantesi e che, dando luogo ad un cluster di sonorità rivoluzionarie come d'altra parte ci si aspetta vista la formazione dei musicisti, sfocia nel capolavoro.

[179] Ebbene si. Leggasi "guglabile", trattasi di altro neologismo derivato da Google, il motore di ricerca del WWW che ha fatto polpette di tutti i suoi avversari. Se su Internet c'è l'informazione che cercate – e le probabilità che ci sia sono molto alte – Google lo sa.

[180] L'Extreme Programming si abbreviava in XP ben prima che questa sigla iniziasse a ricordarci entità molto più sinistre.

[181] Condizione nella quale può succedere veramente di tutto. Si narra, tra le altre cose, di programmatori ritrovati in stato di mummificazione ed avvolti in interminabili listati di cifre decimali di $\pi$.

## Cattedrale o Bazar?[182]

Siamo alla fase finale di questa pseudoanalisi, ovvero all'esame dello *scopo* di tutta la nostra fatica: fare conoscere il nostro Prodotto, far si che venga utilizzato e ricavarne qualcosa: vil denaro, onore e gloria imperituri o anche tutti e due.

È molto importante fare una distinzione fondamentale, perché è qui che dimostreremo come il Nemico prediliga un ambiente rispetto all'altro, anche se in maniera in un certo senso indiretta.

Ora, la scelta da compiere è la seguente: intendiamo attribuire un valore monetario al Prodotto in se o no?

La domanda potrebbe sembrare insensata a chi abbia ben saldo nella sua testa un corpus di principi economici classici: la risposta ovvia della maggior parte dei lettori sarà "Certo che si! Altrimenti per quale motivo staremmo qui a lavorare? Questo è scemo! Ma trovati un lavoro serio, vah!"

Il fatto è che, per quanto riguarda il Software, sono possibili considerazioni molto interessanti; per molti versi tutto quello che siamo abituati a pensare in campo commerciale è da rivedere. Grazie ai costi potenzialmente inesistenti di duplicazione e diffusione, il software si presta ad essere usato all'interno di meccanismi assolutamente inediti. È quindi da non scartare a priori l'idea di distribuire gratuitamente, sotto determinate condizioni, il nostro Prodotto. Ma vediamo in particolare i modelli più usati:

### Modello Commerciale Classico

Solitamente si paga il Prodotto, o comunque si attribuisce un valore al prodotto in se. Questo va benissimo, quando il Prodotto funziona perfettamente o quasi[183], o quando è particolarmente adatto ad essere sgranocchiato mentre si guardano i Simpsons in TV. Però, se comprate un paio di scarpe e queste si aprono in due dopo qualche centinaio di metri, come minimo provate un senso di fastidio, no? E non parliamo delle mele marce. Figuratevi la situazione nel campo del software, dove il Nemico si annida nelle più recondite ed inaccessibili linee di codice del Nostro Prodotto aspettando quel magico insieme di eventi tra loro apparentemente disgiunti che gli permetterà di manifestarsi in tutta la sua micidiale potenza. La differenza è che è comunque possibile riportare in

[182] Il titolo di questo paragrafo si rifà ad un famosissimo articolo di Eric S. Raymond facilmente recuperabile in Rete, "The Cathedral and The Bazar". È questa una delle prime storiche analisi sulla contrapposizione tra le "cattedrali" del software commerciale e il "bazar" del cosiddetto software Open Source, del quale andiamo a parlare. Inoltre contiene svariate sorprendenti leggi e considerazioni che chiunque sia almeno minimanente interessato all'argomento dovrebbe imparare a memoria.

[183] Sappiamo che "la perfezione non è di questo mondo". Mai modo di dire fu più azzeccato per quanto riguarda il software.

negozio un paio di scarpe rotte e mettersi a litigare, mentre cercare di ottenere da una software house un rimborso per il fatto che word processor e sistemi operativi che avete comprato si incartano è pura fantascienza.

La faccenda assume situazioni a volte ancora più paradossali: ad esempio, forse non tutti sanno che acquistare una copia di un programma commerciale solitamente non implica diventarne proprietari, ma solo acquisirne il diritto di sfruttamento. Questo da un punto di vista legale significa, ad esempio, che *non potete* distruggere fisicamente il CD originale del sistema operativo in dotazione al vostro computer nuovo[184], perché quel CD *non è vostro*. Toglietevi lo sfizio e andate a leggere la *Licenza di Utilizzo* del sistema operativo e delle applicazioni che state utilizzando: se ricadono nella categoria del Modello Commerciale Classico avete alte probabilità di scoprire, tra le righe scritte in puro avvocatese, molti altri singolari doveri e limitazioni e ben pochi diritti. Evitate quindi di servirvi del punteruolo: quello riservatelo alla Raccolta dei Successi di <mettere QUI il nome dell'Odiato Cantante>.

Scoprirete inoltre che non potete fare delle copie[185], non potete ridistribuire il prodotto, non potete installarlo più volte[186], la società produttrice non può essere ritenuta responsabile di eventuali danni derivanti dall'utilizzo proprio, improprio o da malfunzionamenti, non è garantito che il prodotto funzioni su qualsiasi computer che pure sia in possesso dei requisiti minimi di sistema ed alte *decine* di clausole. Sconcertante, no? Una vera manna per il Nemico. Sarebbe come dire che uno crede di comprare una lavatrice, ma scopre che l'ha solo affittata, che se macchia i vestiti se la deve tenere comunque e che se esplode allagandogli l'appartamento sono cavoli suoi.

In ogni caso il resto del mondo[187] funziona su schemi dove le cose hanno un costo intrinseco e quindi è naturale che questo sia, anche per il software, il modello commerciale più diffuso. Questo implica che la parte commerciale di qualsiasi Azienda data prenderà accordi non mantenibili con uno o più Clienti, passando la palla allo Sviluppo ed introducendo la famosa Deadline; a quel punto gli sviluppatori, per quanto bravi, non potranno fare altro che aprire le porte al Nemico.

Ad esempio, è usanza comune di ogni software house che si rispetti produrre brochure dettagliate di prodotti che in realtà non esistono, costruendo le schermate che mostrano in dettaglio la Fantastica Applicazione e tutte le sue Incredibili Funzioni a colpi di Photoshop[188]. Questo rispecchia una particolarità dei Manager messa in luce dal già citato Joel Spolsky: la loro totale incapacità di vedere al di là del pixel. L'*applicazione* coincide

[184] E che peraltro avete pagato anche se intendete formattare tutto e montare un sistema operativo diverso, visto che adesso sapete che ce ne sono tanti e fra poco scoprirete che alcuni, oltre ad essere di grande qualità, sono addirittura gratuiti…

[185] Al massimo potete fare una copia, personale e non cedibile, ovvero una copia di backup.

[186] A meno che la licenza lo preveda, ovviamente… ma se così è, state sicuri che avete pagato profumatamente per ottenere questo permesso.

[187] A parte un certo mercato azionario "moderno", altro fenomeno del quale varrebbe la pena di analizzare l'incredibile perversità dei meccanismi che portano, ad esempio, ad attribuire valore a delle azioni in quanto tali e non in quanto rappresentative dello stato di salute della società che le emette.

con la sua *immagine*; tutto quello che c'è dietro e che serve a farla *funzionare* non viene minimamente preso in considerazione. Così lo Sviluppatore deve passare da una videata ad una applicazione funzionante da un giorno all'altro perché – Evvivaaaa! – "abbiamo venduto il prodotto" e quindi bisogna produrlo davvero. Devastante.

## Modelli Commerciali Misti, Eccezionali e Rivoluzionari

Le evidenti problematiche del precedente modello commerciale applicato al software hanno mosso a compassione una vasta schiera di sviluppatori e perfino intere aziende, tanto da far nascere nel corso degli anni numerose formule alternative che andremo ora ad esaminare. In particolare, visto che dobbiamo sempre e comunque parlare del Nemico, vedremo come alcune di queste tendano per loro natura a minimizzare la presenza dello stesso.

### Shareware

Il concetto di S*hareware* nasce dalla formula "Try before you buy" applicata al software. Avere qualcosa in prova prima di decidere se comprarlo o meno è sempre stato abbastanza inusuale in Italia, poiché solitamente, dove c'era scritto in grande "soddisfatti o rimborsati" c'era anche scritto in piccolo "condizioni". Solo in questi ultimi anni, dopo innumerevoli battaglie condotte da varie Associazioni dei Consumatori, la situazione è migliorata.

Nondimeno è questa una pratica assai diffusa in molti Stati esteri – primo fra tutti gli USA – dove è innegabile che alla *Customer Satisfaction*, la soddisfazione del cliente, sia storicamente attribuita una importanza ben più marcata che nel nostro amato Belpaese[189]. È quindi abbastanza naturale che un simile concetto venisse ivi applicato anche al Software.

Lo Shareware si basa sulla fiducia che il programmatore accorda alla propria utenza: i programmi sono scaricabili gratuitamente da Internet e l'utente li può usare quanto vuole[190]. È però invitato, nel caso in cui si renda conto che il prodotto è davvero utile, a regolarizzare la propria posizione pagando una cifra, solitamente piuttosto bassa, direttamente agli sviluppatori.

---

[188]Neanche Flash o qualsiasi altra cosa che potrebbe essere almeno minimamente interattiva: solo e rigorosamente immagini bitmap. Pixel Puri.

[189] È anche vero che McDonald è nato là e non si capisce nel caso dei ben noti hamburger di manzo ricombinato il cliente di cosa dovrebbe essere soddisfatto, ma la plastica, anche a livello dietetico, esercita un grande fascino sull'Essere Umano.

[190] In verità esiste una estrema variabilità di soluzioni: alcuni programmi possono avere una scadenza temporale, o (raramente) un numero massimo di utilizzi, o possono essere limitati dalla non disponibilità di tutte le opzioni; solo dietro pagamento di una comunque modica somma si ricevono codici di sblocco, password etc. In ogni caso è nell'interesse dello sviluppatore dimostrare al potenziale cliente la validità del proprio prodotto, il che è comunque meglio che non comprare un programma a scatola chiusa pagandolo qualche centinaio di euro per poi scoprire che non funziona o non fa al caso nostro.

Ci sono varie cose da dire su questo meccanismo: primaditutto, è da notare l'implicita distribuzione dal produttore al consumatore, che ovviamente contribuisce a tenere basso il prezzo del prodotto. Poi l'intrinseca natura dello sviluppatore, che è ovviamente molto motivato ed amante del suo lavoro; solitamente chi scrive Shareware vive grazie ai soldi che gli utenti gli danno. Questo implica che le applicazioni shareware tendono ad essere piccole, scritte spesso in solitaria e molto specifiche. Questo tipo di sviluppatore è molto veloce nel feedback con i suoi clienti: ne ascolta le lamentele, mette a posto velocemente i bug e rilascia spesso nuove versioni, poiché spera di mantenere la propria clientela e invogliarla a spendere qualche dollaro in cambio di continue migliorie.

**Freeware**

Il *Freeware* è semplicemente software gratuito. Dietro alla decisione di rilasciare gratuitamente del software ci possono essere decine di motivazioni e tipologie di autore diverse, al variare delle quali varia anche, enormemente, la qualità del prodotto stesso e quindi la presenza del Nemico. Vediamone qualcuna:

- Dimostrare al mondo di essere un bravo sviluppatore
- Tentare di imporre uno standard
- Non voler tenere nel cassetto qualcosa che potrebbe essere utile a qualcun altro
- Aver già fatto abbastanza soldi col prodotto
- Aver già fatto abbastanza soldi in altro modo
- Non aver voglia/forza/tempo di seguire il proprio prodotto
- Fare pubblicità al prodotto rilasciandone gratuitamente una vecchia versione
- Ritenere, per ragioni etiche, che il software debba essere gratuito
- CPNHPNM[191]

Alcune di queste spiegazioni sono comuni al rivoluzionario modello noto come Open Source, mentre altre si possono riallacciare alla visione di organismi come la Free Software Foundation; di queste parliamo più avanti. Altre sono assolutamente uniche e danno luogo a modelli nobili, bizzarri o entrambe le cose: si va dal *Cardware* dove l'utente è libero di usare il prodotto purchè spedisca una cartolina allo sviluppatore al *Pizzaware*[192] fino al *Prayware*, dove l'utente è invitato a pregare per la salvezza dell'anima dell'autore e per la Pace nel Mondo. In effetti ecco cosa ci dice a proposito di un singolare ordine monastico il Zottor Divago, autore di una poco nota *column* di un periodico di informatica peraltro serissimo:

[191] Chi Più Ne Ha Più Ne Metta.

[192] Al *Pizzaware* sono dedicati interi siti. Lo stesso Paolo Attivissimo, già citato, è un fervente assertore di questa forma di pagamento; anche Babele Dunnit è estremamente ghiotto di pizza che durante la *trance programmatoria* si fa consegnare chiedendo di farla passare sotto la porta (cfr. *Microservi*, Douglas Coupland).

Benvenuti nuovamente all'appuntamento con il Zottor Divago e i suoi Arcani Informatici! Questa settimana ci scrive il lettore BranzoDiManzo, sottoponendoci un problema inquietante: da una stampante è uscito un foglio bianco e non si sa chi ne abbia richiesta la stampa. Ora, questo è un fatto che merita di essere approfondito e noi procederemo a tentoni tenendo come sempre ben presente la Prima Legge:

*Poche Idee Ma Ben Confuse*.

Ora, come indagherebbe un vero informatico? Inizierebbe a scavare tra logfile e driver, nascosto tra un server e l'altro, fino ad individuare il malefico comando. Il computer è una macchina e quindi non può aver deciso autonomamente. Egli Non Sbaglia Mai.

Questa è una credenza errata che dobbiamo al nostro approccio occidentale, altresì detto "scientifico". Ora, guardiamo invece le cose in maniera oggettiva: i Computer Fanno Quello Che Vogliono. Ormai da tempo la corrente dell'Informatica Comportamentale ha dimostrato che computer->self = F(Ut, OS), ovvero che il comportamento dei computer – lungi dall'essere galileiano, ovvero meccanicistico e riproducibile – dipende da utente e sistema operativo sottostante. "Self is a function" afferma Gregory Bateson; perché un computer dovrebbe agire diversamente dall'Uomo? Silicio e carbonio, nella Tavola Periodica, sono vicini.

Siamo evidentemente davanti a un caso di possessione. Chiameremo questo caso clinico La Stampante Del Mistero, o Il Mistero della Stampante, o in sintesi MisterPrinter, che ci fornisce anche una antropomorfizzazione utilissima ai fini della visualizzazione 3D sia essa accelerata in hardware o meno. Un consulto iniziale dell'I Ching, debitamente filtrato attraverso un logotrone (apparecchio per spezzare le parole, cfr. Babbage), ci rimanda ad uno stupendo haiku di Simm Dimm Zot, monaco Zot Zen della dinastia Pcmcia:

*La tua memoria*

*Perduta nel silicio*

*Vola nei cavi*

È evidente la mancanza di riferimenti a qualsivoglia copia cartacea. Questo rafforza la tesi della possessione. È necessario quindi liberare il computer dal Demone della Stampa

(il “Daemon”) con l’aiuto di un altro grande saggio e asceta: il maestro Nurb Jpg della setta degli Acceleratori Scarsi.

Sono quest'ultimi un'ordine monacale di derivazione assiro-californiana che prevede l’espiazione del Peccato Originale Digitale (il DOS, Digital Original Sin) tramite l’utilizzo del computer attraverso i piedi. Se da una parte non si può che rimanere incantati davanti alla maestria con la quale Nurb vola podalicamente su mouse e tastiera nella prima fase dell’esorcismo (“debriefing”), è dall’altra propedeutico vedere come nella seconda fase il vile metallo venga forgiato e riportato a ben più miti consigli da sette precise e potenti mazzate, le quali vengono così classificate:

*Il Colpo dei Sette Drivers*

*La Mazzata della Testina Schiantata*

*Lo Strappo del Cavo di Alimentazione*

*La Zampata dei Diecimila Milioni di Poligoni al Secondo*

*Il Tallone sul Connettore della Ellepitiuno*

*La Ginocchiata sul Pulsante del Reset*

E, per finire,

*Il Calcio Volante della Formattazione a Basso Livello*

L’Acceleratore Scarso viene protetto in questa procedura dal suo indumento rituale, un saio ricavato dalla ben nota materia plastica per imballaggi detta “a palline” che egli stesso provvede a far scoppiare a tempo durante la sua consulenza, quasi si trattasse di una visione grossolanamente allegorica della sua Weltanschauung informatica.

Il Computer Liberato, ora ridotto ad un cumulo di macerie fumanti, finalmente non stampa più imperscrutabili pagine di propria iniziativa. Un altro caso risolto felicemente per il Zottor Divago e i suoi Arcani Informatici… e a rileggerci presto!

Tra il Freeware è molto più facile trovare programmi solo parzialmente funzionanti e poco documentati; in generale vale il concetto del Caval Donato, del guardare in bocca e di tutte quelle cose là. È quello del Freeware un luogo da visitare con attenzione quando si ha poca esperienza, perché il Nemico qui trova spesso un habitat ideale – senza parlare

di quei furbacchioni che non perdono occasione per infilare virus, spyware ed altre schifezze in programmi gratuiti particolarmente appetibili. Vale la pena quindi spendere qualche minuto per leggere impressioni e consigli di altri utenti, solitamente rintracciabili nei vari forum collegati ai siti dai quali si possono scaricare questi programmi, filtrando opportunamente i messaggi della mamma dell'autore.

In ogni caso, è in mezzo a questo tempestoso mare del software completamente gratuito che si nascondono isole di inimmaginabile bellezza: è questo il caso dell'Open Source.

**Open Source**

Definire il movimento noto come Open Source semplicemente dicendo che "è software gratuito" è come dire che la pizza è acqua, farina, pomodoro e mozzarella[193]. Il punto è che molti, molti sviluppatori, con il passare degli anni, si sono resi conto che continuavano a riscrivere sempre le stesse cose; così, un bel giorno, complice un Internet ante-litteram, il pensiero collettivo – il Comportamento Emergente – è stato: scriviamo *tutto* una volta per tutte e che Tutti siano liberi di utilizzare ciò che vogliono, purché a loro volta rendano pubbliche le eventuali migliorie. Grosso modo questo è il concetto dietro all'Open Source: non solo i *programmi* sono liberamente utilizzabili, ma addirittura i *sorgenti* degli stessi sono a disposizione della comunità. Così, quando uno sviluppatore deve incorporare una certa funzionalità in un suo programma, non fa altro che cercare un programma che disponga di quella opzione e vedere come ha fatto a risolvere quel problema chi è venuto prima di lui. Alcune soluzioni saranno buone, altre pessime; se lo sviluppatore migliora qualcosa, è tenuto a rendere il proprio lavoro pubblico a sua volta, in una sorta di visione darwiniana del software che non si fermerà mai più. Insomma, da quel lontano 1984, quando un pazzo visionario di nome Richard Stallman iniziò a riscrivere pezzi di Unix chiamando il suo progetto GNU[194], molto codice è passato sotto i ponti.

Le implicazioni di un simile processo di sviluppo sono estreme: la qualità del software Open Source può solo migliorare, grazie alle migliaia di occhi che scrutano senza sosta l'Orizzonte Elettrico alla ricerca del Nemico.

È questo un modello per molti versi diametralmente opposto a quello canonico: l'Open Source è basato sul concetto di *Prodotto Complementare*[195]*;* non ci sono vere e proprie deadline, per cui la velocità alla quale i prodotti vengono sviluppati è solitamente più bassa di quella del software commerciale, mentre la qualità dello stesso è proporzionalmente maggiore. Non viene attribuito alcun valore al software in se,

[193] Ancora pizza... sapete a che temperatura dovrebbe essere mantenuto il forno, per una "vera" Pizza Napoletana? 485 gradi centigradi. Molto, molto caldo: il "segreto" è proprio quello.
[194] GNU è l'acronimo (ricorsivo!) di "GNU is Not Unix". Tanto insensato quanto geniale.
[195] La più geniale definizione di complementarità economica (lui la chiama "diversificazione") la dà il comico Maurizio Milani: "Se tu compri azioni di una società petrolifera è buona cosa che abbini anche l'acquisto di azioni di una società che lava i pinguini".

riversando ogni discorso relativo alla vil pecunia sui *servizi* che accompagnano e completano il software. Ad esempio, la pubblicazione di libri, lo svolgimento di corsi, le consulenze, le conferenze, i tutorial online, perfino la vendita di gadget[196]… esistono numerosi casi di società che, pur distribuendo gratuitamente i propri prodotti sviluppati in ottica Open Source, sono non di meno a capo di un notevole giro commerciale basato unicamente sulla erogazione di servizi e prodotti complementari al software gratuito che sviluppano.

Al contrario dello Shareware, l'Open Source, grazie alla sua intrinseca modularità e riciclabilità, tende a produrre software ad ogni livello di complessità: si va da piccole utility in grado di competere (e vincere) con le controparti commerciali come FileZilla[197] fino a interi Sistemi Operativi completi di migliaia di applicazioni, come quello che solo dieci anni fa era il sogno di un programmatore visionario che e ora è diventato uno standard di mercato in grado di far tremare i Signori del Software.

Narrano infatti gli annali di come, nel 1991, un giovane sostenitore di Stallman venuto dal freddo e nomato Linus Torvalds desse il via alla Missione Impossibile di riscrivere un sistema operativo per home computer ispirandosi proprio alle *features* più interessanti di Unix. Con la collaborazione del Popolo di Internet, l'iniziativa fu un successo immediato: migliaia di programmatori di tutto il mondo offrirono a Torvalds aiuto gratuito in cambio di quel sogno, ormai divenuto realtà e chiamato Linux proprio in suo onore. Ora Linux è considerato la migliore implementazione in assoluto di Unix[198] mentre la familiare interfaccia a finestre e icone ne rende l'utilizzo non dissimile da quello di un Mac o di una macchina Windows.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -
Ìm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. Ìd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

[196]Il Pinguino Tux, *mascotte* di Linux, è stato riprodotto in ogni forma: tazze, magliette, pupazzi, astronavi... notevole anche il fatto che il concetto del "pinguino" appaia in due note di seguito.
[197] Client FTP assolutamente gratuito e Open Source che mette nel sacco i vari CuteFTP, WS-FTP etc.
[198] IBM stessa vende i propri server con Linux preinstallato. Per fermare una macchina Linux ben configurata bisogna buttarla giù dalla finestra, tale è ormai proverbiale la stabilità di questo sistema operativo.

Ìve currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that ÌIl get something practical within a few months, and Ìd like to know what features most people would want. Any suggestions are welcome, but I won't promise ÌIl implement them:-)
Linus (torvalds@kruuna.helsinki.fi)
PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have:-(.

Le raccomandazioni già viste per il Freeware in verità valgono per *qualsiasi* tipo di software: commenti, consigli e considerazioni di altri utenti, lasciati in forum facilmente rintracciabili con Google, ci permetteranno di evitare di perder tempo con programmi malfunzionanti o semplicemente inutili. Altra indicazione interessante, quando presente, è quella che dichiara la *stabilità* dell'applicazione: se un programma è esplicitamente marcato come *beta version*, evitate di perderci tempo a meno che siate degli smanettoni. D' altra parte, un software Open Source etichettato come *production/stable* è solitamente molto affidabile – almeno quanto i suoi fratelli commerciali.

**Freeware, Free Software e Open Source: The Big Picture**

Al Lettore Attento non saranno a questo punto sfuggite alcune similarità che confondono lo scenario del "software gratuito". Che differenze ci sono tra Open Source e Freeware per un non-programmatore? Posso *davvero* utilizzare un programma senza dover pagare nessuno? E la *Free Software Foundation* cosa fa?

Il punto è che il *Freeware* è quanto di più generico si possa immaginare; in questo senso, utilizzare una applicazione freeware può essere rischioso, perché lo sviluppatore non è obbligato a fornire alcuna garanzia riguardo al suo prodotto. D'altra parte abbiamo visto che anche il software commerciale – nonostante i costi a volte esorbitanti e comunque generalmente tutt'altro che trascurabili – offre ben poca tutela all'utente finale, il quale si ritrova con un sacco di restrizioni, ovvero *doveri*, tra i piedi.

Il movimento Open Source e la Free Software Foundation, ognuno a suo modo, perseguono invece lo scopo diametralmente opposto di attribuire dei *diritti* all'utenza: il diritto di utilizzare, copiare, ridistribuire e studiare il funzionamento del Software nel suo complesso, elevato al rango di Patrimonio dell'Umanità[199]. Questo intento viene perseguito in modi anche molto diversi, essendo il movimento Open Source creato da programmatori per programmatori e la Free Software Foundation molto più generica nel suo intento di ottenere per il Software uno status giuridico universale che garantisca agli utilizzatori una serie di diritti a loro attualmente negati. Ognuno dei due organismi, a suo

[199] Esemplare da questo punto di vista la famosa licenza di GNU nota come GPL (General Public Licence) che, introducendo il concetto di *Copyleft*, persino nel gioco di parole sancisce la sua distanza antipodale rispetto all'odiato *Copyright*.

modo, combatte una vera e propria Guerra Santa contro multinazionali che vorrebbero brevettare *tutto*[200]; noi utilizzatori, come minimo, siamo tenuti almeno a saperlo. Se un giorno lo scatolotto che avete davanti funzionerà bene il merito sarà in larga misura anche loro.

*[Disegno del software libero. Ah, bello. Come è fatto? Non lo so, ma è bellissimo.]*

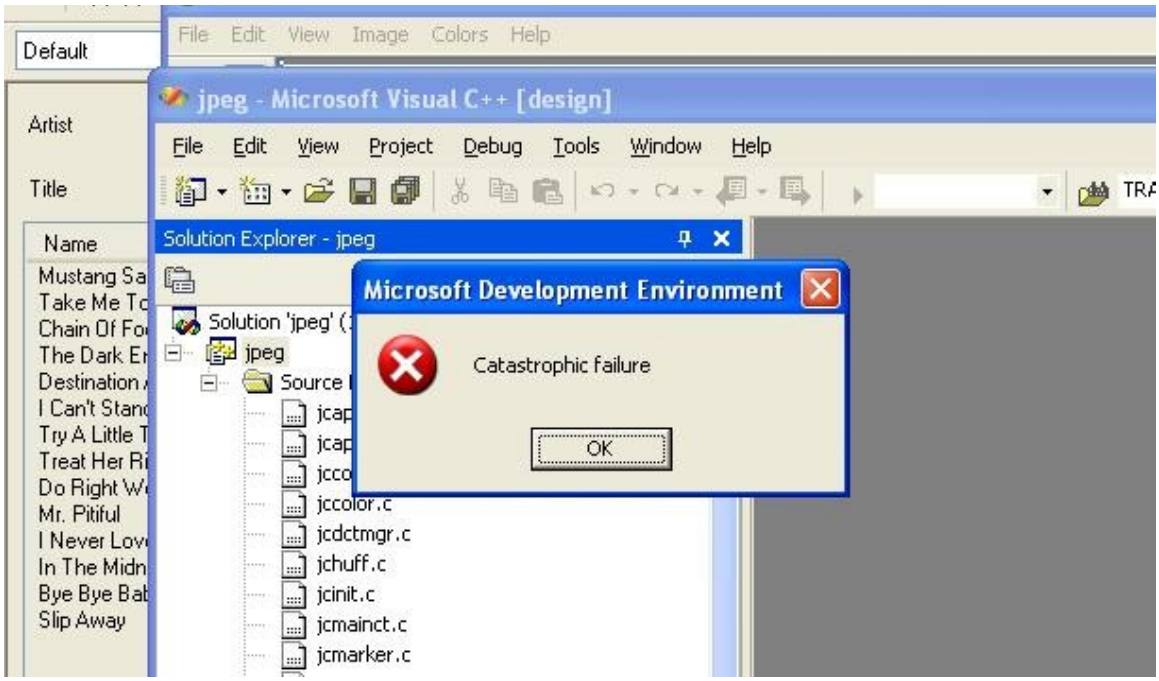
## Che La Forza Sia Con Te

*"Guerriero è colui che si oppone al caos."*
*Mahabaratta*

A questo punto del Grande Libro dei bug, manuale di pronto intervento, combattimento e calde lacrime amare camuffato da *divertissement* tecnoallegorico, abbiamo visto macchine che si spengono mentre lavoriamo, oggetti che rimbalzano con precisione millimetrica sul tasto del reset, connettori che si staccano da soli… D'altra parte il lettore dovrebbe ormai esser convinto che, tutto sommato, il nostro computer ha le sue buone ragioni quando si impasta, gratificandoci – a seconda di sistema operativo e altre centinaia di parametri – con manifestazioni sconfortanti che vanno dal mitico *Blue Screen Of Death* delle macchine Windows NT/2000 a icone di bombe, Guru Meditations, suoni strazianti, blocchi totali di mouse e tastiera, riavvii inopinati o rallentamenti così inspiegabili ed esasperanti da indurci a raggiungere col ditino il Grosso Grasso Pulsante Rosso Del Nirvana a favore di una subitanea eutanasia elettronica.[201]

[200] E che, di fatto, hanno brevettato molto: dall'algoritmo di compressione delle immagini GIF – per cui chi scrive un programma che tratti le immagini GIF deve pagare (ragion per cui il formato GIF è fortemente boicottato nel mondo dello Shareware e del Software Libero) all'algoritmo dei "marching cubes" utilizzato nei videogiochi per simulare i "blobs", ovvero per trovare in tempo reale le superfici equipotenziali di un campo elettrico a più punti. Non so cosa sto dicendo.

[201] Se la seguente immagine vi diverte *volate* sul sito di Paolo Attivissimo (www.attivissimo.net). Ce ne sono di fenomenali.




Ribadiremo quanto abbiamo già visto, facendoci aiutare da parole di Antica Saggezza che possano infonderci nuova forza e slancio verso la Battaglia. Da Yoda a Sun Tzu, dal Rat-Man[202] a Remote, molti sono i combattenti ai quali possiamo chiedere aiuto. Inizieremo infatti dai Venti Precetti del Karate del maestro Funakoshi opportunamente rivisitati[203].

## I Precetti della Via del Debugging

Per adattare i Venti Precetti del Karate ai nostri scopi basterà utilizzare una tecnica ben conosciuta anche ad utenti informatici poco esperti: mediante fulminea mossa di *search and replace*[204] si vadano a sostituire "Karate" con "Debugging" e "nemico" con "bug".

Come si vedrà, il risultato è a dir poco entusiasmante: ecco i Precetti del Debugging-Do, la Via del Debugging. Grazie a questi insegnamenti possiamo iniziare a pensare al Nemico con una leggerezza e predisposizione d'animo altrimenti inarrivabile, ovvero sghignazzando come deficenti.

Lasciamo quindi la parola al Maestro:

---

[202] Fumetto creato da Leo Ortolani le cui avventure sono una enorme estrinsecazione della Legge di Murphy in tutta la sua devastante potenza. Imperdibile.

[203] Alcuni dei Venti Precetti (Niju Kun) originali sono stati eliminati perché troppo generici per questo contesto anche se assolutamente veri, come "Prima devi conoscere te stesso. Poi potrai conoscere gli altri.". È inoltre da notare che ci stiamo basando su testo inglese, ovvero sulla traduzione di una traduzione; abbiamo visto quanto il Nemico ami le traduzioni, vero? Beh, d'altra parte quanti di voi leggono ideogrammi?

[204] Versione informatica del "search and destroy" reso famoso da svariati film di guerra e mazzate. Insieme al *Cut and Paste* è secondo molti il vero vantaggio dell'usare un computer anziché carta e matita.

*Debugging is not only Dojo training*
*Kata is one thing: engaging in a real fight is another.*

Il Debugging non è solo allenamento in palestra; il combattimento simulato è una cosa, il combattimento reale un'altra.[205]. Anche se parlare di "palestra" e "allenamento" sembrerebbe non avere molto senso in campo informatico, di fatto si può pensare a tutta l'attività di sviluppo come a una palestra che prepara lo sviluppatore al rilascio di un prodotto nel mondo reale. Per quanto noi poveri pigiatasti si possa debuggare, altri bachi emergeranno quando rilasceremo il nostro prodotto agli utilizzatori reali. Cerchiamo di essere pronti, perché sapere che qualcosa succederà non implica *essere pronti* ad affrontare l'evenienza. Insomma, la Speranza (che tutto vada bene) è l'ultima a morire, ma il Nemico pare trovi un particolare godimento ad ammazzarla.

*Don't forget that Debugging begins with a bow, and ends with a bow.*

Non dimenticare che il Debugging inizia con un inchino e finisce con un inchino. Rispetta il tuo nemico. Non sottovalutarlo. Mai. Soprattutto, mai e poi mai proferire le famose Fatidiche Sei Parole:

| |
|---|
| Sarà Una Cosa Da Cinque Minuti. |

C'è chi ha pronunciato più di venti anni or sono le Fatidiche Sei Parole ed è tuttora impegnato in colossali operazioni di debugging. Occhio.

*In Debugging, there is no first attack.*

C'è un detto secondo il quale la miglior difesa sarebbe l'attacco. Nel Debugging non c'è chi attacca per primo: ogni attacco è anche una difesa e viceversa. Uno dei concetti alla base delle arti marziali è lo sfruttare la forza del nemico per rivolgere l'impeto dell'attacco contro il nemico stesso[206]. Il Bug questo lo sa bene e cercherà in ogni modo di rispedire al mittente ogni tentativo di eliminarlo. Il vero Samurai del Codice deve ricordare questa capacità del Nemico, facendo in modo che ogni propria mossa non dia all'avversario la possibilità di nascondersi altrove. A seguito di ogni nostra operazione dovremmo almeno sommariamente controllare che tutto quello che funzionava "prima" continui a funzionare anche "dopo", ovvero che il nostro attacco abbia effettivamente eliminato il nostro camaleontico Nemico e non che gli abbia solo meramente fatto cambiare forma. Questo, che tecnicamente viene chiamato *Test Di Non Regressione*[207], è un concetto molto ampio, così come è ampio il terreno sul quale il Nemico manovra: a

[205] Il *Dojo* è appunto la palestra dove si praticano le Arti Marziali – in particolare l'Aikido – mentre i *kata* sono sequenze di movimenti imparati inizialmente a memoria e costantemente ampliati e affinati, propedeutici allo sviluppo delle capacità psicofisiche indispensabili al combattimento.
[206] Il Ju-Jitsu, il filo d'erba che si piega nel vento e tutte quelle robe lì. Dai, non avete mai visto Karate Kid?

chi non è mai successo, dopo l'installazione di una scheda o un programma, che qualcos'altro di apparentemente indipendente smettesse di funzionare? In quest'ottica un test di non regressione dovrebbe consistere nel provare almeno le funzionalità di base e i programmi principali che usiamo dopo aver installato software o hardware nuovo nel nostro Amato Scatolotto, in modo da accorgerci subito di problemi e incompatibilità[208].

*You must release your mind.*

Rilassati. Rilassati, Ho Detto. Rilassatiiiiii.....RI-LAS-SA-TI!!! Ok, se proprio non ne vieni fuori, è arrivato il momento di piantare lì tutto e andare a fare un giro, a bersi un caffè o a pranzo. Nella maggior parte dei casi dopo pochi minuti la soluzione si affaccerà autonomamente alla nostra coscienza[209]. Provare per credere: addirittura, i Samurai del Codice come Babele Dunnit a volte *sognano* le soluzioni, con tanto di numero di riga di codice ove il Bug si annida[210]. Questa è anche una ottima scusa per poter urlare "STO LAVORANDO!!" quando interrompono il suo pisolino postprandiale[211].

Se dopo due o tre ore che ci battete la testa ancora non siete venuti a capo di nulla, è il momento di scollarvi dal monitor e *fare quattro chiacchiere con qualcuno.* È incredibile la percentuale di bachi che vengono depennati semplicemente scambiando le proprie idee e impressioni con qualche collega, o anche semplicemente parlando ad alta voce con il gatto. Quest'ultimo deve però almeno fare finta di seguire il discorso, altrimenti uno si sente un po' pirla.

Insomma, Fantasia e Pensiero Laterale. Andate a bervi un caffè alla macchinetta. Smazzate la posta. Fate una giravolta. Fatela un'altra volta. Guardate in su. Guardate in giù. Ok. Basta.

*[Disegno del Gatto Purrino Programmone]*

*Misfortune comes out of laziness.*

La sfortuna è figlia della pigrizia. Di questo abbiamo già parlato. Il nemico più facile da sconfiggere è quello che non c'è; solitamente il modo migliore per ottenere questo invidiabile vantaggio è giocare d'anticipo. Inoltre, del corredo standard di ogni buon guerriero fanno parte le Mutande di Titanio, insostituibili per pararsi il deretano dai colpi avversi della sorte. Soprattutto, una volta individuato il Nemico, è opportuno muovere

[207] Sarebbe bello poterlo fare, ma è ovvio che nella maggior parte dei casi poter svolgere dei test di non regressione veramente esaustivi è una impresa titanica, a meno di costruire il nostro prodotto fin dagli inizi con questa metodologia in testa. Ad esempio, tra i principi dell'Extreme Programming c'è il "test driven development" che auspica proprio quanto descritto.

[208] Non c'è niente di peggio di scoprire che qualcosa non funziona più dopo mesi che non la usiamo: non abbiamo la minima idea di cosa l'abbia rotta. Magari sono solo le batterie scariche e noi invece stiamo a pensare agli Alieni o a quando la Zia Carmela ci ha omaggiato del suo gelato al rabarbaro.

[209] *La Risposta è Fuori Dal Computer!* , L.Agrò, da "Le Vie del Software sono Finite", su www.idearium.it.

[210] Tanto tempo fa, quando ancora si usavano i numeri di linea...

[211] Babele Dunnit mangia speck e burro di arachidi. È ovvio che quando dorme sogni il Nemico.

battaglia *immediatamente*: il Nemico si riproduce per mitosi, quindi ancora più velocemente dei conigli di Fibonacci[212]. Ogni Bug trascurato, ovvero scoperto ma non risolto, darà luogo a due Bug e così via, verso l'Infinito ed oltre[213].

*Debugging is a lifelong training.*

Il Debugging è un allenamento che dura una vita. Non si finisce mai di imparare e in particolare in campo informatico siamo soggetti ad una evoluzione tale da trasformare molto velocemente una tecnologia all'avanguardia in patetico vecchiume[214]. Ovviamente l'unico modo per non perdere la Battaglia da questo punto di vista è trovare un equilibrio tra il rimanere più possibile aggiornati e cercare di avere una vita sociale migliore di quella delle cozze[215]: nuove tecnologie, nuovo hardware, nuovo software, nuove metodologie di sviluppo, nuovi linguaggi… tutto terreno vergine per il Nemico. Solo la conoscenza e l'esperienza possono aiutarci[216].

*Put Debugging into everything you do.*

Compi ogni tua azione come se stessi debuggando. Qui si parla di una certa *forma mentis* che, secondo il saggio Funakoshi, bisognerebbe arrivare a sviluppare. È lo stesso concetto del mutandamento che abbiamo già visto: il Nemico non è necessariamente confinato dietro al monitor, ma usa ogni arma, ogni momento buono, ogni nostra disattenzione. Quindi: non appoggiare il portachiavi magnetico al disco rigido rimovibile. Non lasciare il CD sul cruscotto dell'automobile. Non parcheggiare il tuo portatile su una mensola non abbastanza larga o su una poltrona non sufficientemente illuminata. Quando i cavi sono tutti attorcigliati non tirare verso di te la tastiera. Sii presente a te stesso quando spegni la

[212] Ovvero Leonardo da Pisa (c. 1170 – 1240), matematico altresì noto come Leonardus Pisanus o ancora "filius Bonacci" da cui quel "Fibonacci" ora noto in tutto il mondo. Autore del "Liber Abaci", è tra i primi propugnatori dell'utilizzo delle cifre arabe (ovvero quelle dallo 0 al 9 che ora normalmente usiamo) nonché scopritore della nota sequenza che quantifica la crescita "esponenziale" di una popolazione di conigli: 1,1,2,3,5,8,13,21,34,55,89,144... Questa serie ricorsiva (vedi nota XXX) in cui ogni termine è la somma dei due precedenti è da vicino imparentata con il valore 1.6180339... altresì noto come Φ (Phi) o Rapporto Aureo, tanto che queste due entità si incontrano in luoghi così impensabili da sembrare soprannaturali. Ad esempio, andate avanti con la serie e provate a dividere un termine per il suo precedente. A che valore tende il risultato? Hmm... Potremmo stupirvi per ore con il Rapporto Aureo, ma se la cosa vi affascina è meglio che vi leggiate l'eccellente "La Sezione Aurea", Mario Livio, RCS Libri, 2003.

[213] Ovviamente il momento scelto dal Bug per palesarsi nuovamente sarà il peggiore possibile, mentre tutt'intorno voci attonite esclameranno "deh! lo avevamo già visto! Oh, noi tapini! cos'era? Cos'era?" e nessuno si ricorderà esattamente i dettagli della faccenda.

[214] Pero stiamo sempre attenti alla Sindrome dell'Ultima Versione: la domanda da porsi è: "mi serve davvero?"

[215] Si veda il notevole lavoro di Peter Fong del Getyysburg College, Gettysburg, Pennsylvania, chiamato "Induction and Potentiation of Parturition in Fingernail Clams (Sphaerium Striatum) by Selective Serotonin Re-uptake Inhibitors (SSRIs)", secondo il quale una somministrazione di Prozac ha reso molto più felici le cozze del suo acquario. Questo ed altri studi che "non possono o non dovrebbero essere riprodotti" sono noti come premi Ig Nobel e li potete trovare in forma cartacea sugli Annals of Improbable Research o su internet: http://www.improb.com.

[216] Non è vero. Anche la Birra ha insospettabili capacità taumaturgiche.

ciabatta di alimentazione del videoregistratore: cos'altro è collegato a quella presa? Ehi, quello è zucchero o sale? Hai letto *attentamente* le istruzioni del ciclotrone?

*Do not think that you have to win: think that you do not have to lose.*

Non pensare di dover vincere: pensa che non devi perdere. Molto, molto *cool*. Molto Jedi. Insomma, ci teniamo ai nostri dati, no?

*Victory depends on your ability to tell vulnerable points from invulnerable ones.*

La vittoria dipende dalla tua abilita nel riconoscere i punti vulnerabili da quelli invulnerabili. Nella saga dei supereroi Marvel esiste un Inumano chiamato Karnak il cui potere è proprio quello di saper individuare i punti deboli. Pur non essendo particolarmente forte, Karnak è un nemico temibile, in grado di abbattere intere pareti di roccia con un singolo colpo. Insomma, per certi versi debuggare è come tagliare un diamante. Ma quali sono i *punti vulnerabili* di un Bug? Bella domanda; la risposta giace nella pratica e nei consigli che stanno arrivando.

*Move according to your Bug.*

Muoviti in sintonia con il tuo Bug. Osserva le sue mosse e il suo comportamento. Seguilo nell'ombra. Per ogni possibile mossa del Bug noi dobbiamo avere pronta una contromossa abbastanza efficace da neutralizzare l'attacco e, possibilmente, sferrarne uno a nostra volta. Come una partita a scacchi quadridimensionali o a Go, o a briscola chiamata (il "due") se proprio preferite.

*When you leave home, think that millions of bugs are waiting for you.*

Quando esci di casa, pensa che milioni di bug ti stanno aspettando. Questa è forse la più divertente ed amara di tutte. La lasciamo così, senza commento, mentre proseguiamo verso la Battaglia ed ascoltiamo cosa ha da dirci Miyamoto Musashi, Samurai del XVI Secolo, nel suo Libro dei Cinque Anelli.

*[Disegno dei milioni di bugs che ti aspettano]*

## Il Libro Dei Cinque Anelli

Figlio di un guerriero di professione nel Giappone sulla via della pacificazione, Miyamoto Musashi (1586-1645) si ritrova orfano di madre e abbandonato dal padre a sei anni. Adottato dallo zio prete, ne impara da questi i rudimenti della scherma. Estremamente feroce – e vorrei vedere voi dopo tutte quelle sfighe – combatte il primo duello con inaudita violenza a tredici anni, diventando successivamente una leggenda vivente. Gli vengono attribuiti più di sessanta duelli. In tarda età diventa maestro d'armi

del signore di *kumamoto* e inizia per lui a scrivere. Le sue opere sono ora note come "Libro dei Cinque Anelli", ovvero Terra, Acqua, Fuoco, Aria e Vuoto.

Ecco, secondo Miyamoto Musashi il Settimo, l'Ottavo ed il Nono Principio che il Samurai deve seguire nel percorrere la Via in maniera corretta:

- Settimo: Impara a Percepire Ciò Che Non Si Può Vedere.
- Ottavo: Non trascurare i Piccoli Particolari.
- Nono: Non fare Cose Inutili.

Ok. Risulta evidente che il Settimo Principio può essere chiamato anche il Principio del Debugging, ovvero Della Individuazione Generale Di Errori E Malfunzionamenti, l'Ottavo e' un raffinamento del Settimo, poiché il Nemico si nasconde a volte tra un elettrone e l'altro e il Nono è un incitamento a cambiare lavoro, che incredibilmente vale per *qualsiasi* lavoro. Non è mirabolante che un manuale di tecniche di combattimento scritto da un samurai cinquecento anni or sono possa venirci utile adesso? Questo dovrebbe insegnarci una cosa: al Debugging – quello con la D maiuscola – concorrono molti fattori tra cui il Pensiero Laterale, l'Ispirazione ed, ebbene si, la Birra o succedaneo alcoolico. Capire perché un computer non funziona è un arte, quindi un lieve stato di ebbrezza non può che giovare[217]. C'è anche la metodologia di sviluppo del software nota come BASE, (Beer Aided Software Engineering), ma quello è un altro discorso[218].

Ed ancora:

> *Teki o utsu ni ichibyoshi no uchi no koto:*
>
> Uccidere in un attimo.
>
> "in un attimo" significa uccidere il Nemico con l'azione più veloce e diretta possibile: mettersi in guardia a distanza di spada dal nemico e, prima che lui stesso abbia deciso di muoversi, senza sprecare gesti o pensieri, semplicemente colpirlo.

[217] Cfr. Jackie Chan, fenomenale maestro cinematografico di arti marziali, con tutta la sua vasta filmografia relativa al personaggio del "drunken master", tanto assolutamente invincibile quanto più sbronzo.

[218] Qualcuno potrebbe obiettare che *la maggior parte* del software in circolazione sembra essere stato scritto da programmatori sbronzi, ma non è così. Come abbiamo visto, certe porcherie sono solo frutto di una deadline incipiente, mentre ricordiamo ancora che la vasta categoria di software disegnato a tavolino per essere volutamente gigantesco (e quindi costoso) nonché criptico e quasi inutilizzabile se non dopo anni di esperienza è quella che ha fatto la fortuna di gigantesche società di consulenza. Di queste per pudore e paura di ritorsioni giuridiche non facciamo il nome, ma è evidente come multinazionali il cui nome finisce per "ulting" e "ure" adorino in tutte le sue forme il Bloatware, il "software gonfiato", e lo propinino ad un attonito Cliente appena possibile.

Insomma, come dire: debellare il Bug non appena si manifesta. È la prima Regola della Cattiveria di Dardo, l'Uomo di Legno:

> Un Colpo, Un Morto.

## L'Arte della Guerra

Passiamo infine ad una rilettura in chiave tecnologica di alcuni paragrafi del più antico testo conosciuto di strategia militare, l'Arte Della Guerra di Sun Tzu[219], al quale facciamo introdurre il tanto agognato *contrattacco*.

Sun Tzu dice: *L'Arte della Guerra è di vitale importanza per lo Stato. È questione di Vita o di Morte, una strada che va verso Salvezza o Morte. In quanto tale è soggetto di domande che in nessun modo possono essere ignorate.*

Prima di affrontare il Nemico bisogna porsi varie domande su di esso. Quanto è potente? Quanto è avvantaggiato rispetto a noi? Quali sono le nostre capacità in questa battaglia? Buona parte del Grande Libro dei bug è una dissertazione sulle abilità camaleontiche del Nemico. Questo è il momento di passare dalla teoria alla pratica, di raccogliere informazioni, di osservare in maniera fredda e distaccata quell'oggetto che abbiamo davanti e che non vuole funzionare come si deve.

Quel maledetto computer è vecchio o nuovo, ad esempio? Una informazione simile potrebbe fare la differenza; magari sappiamo a menadito come funzionava una macchina di due anni fa, ma abbiamo visto come le cose cambino in fretta; magari non ne siamo a conoscenza, ma nel sistema operativo che stiamo usando già potrebbero essere installate un sacco di *utility* in grado di aiutarci.

A seconda del problema, insomma, questo è il momento delle ipotesi. Si tratta di un puro problema hardware[220] o la faccenda è più articolata? E se è un problema software, è più probabile che sia un problema applicativo o dobbiamo pensare che si tratti di un malfunzionamento a livello di software di sistema[221]? Per ognuna di queste domande

[219] Generale cinese, circa 500 A.C. Il libro citato è ritenuto il più vecchio manuale di tecnica militare conosciuto, introdotto in Europa e tradotto (male) in francese per la prima volta nel 1782 dal gesuita francese Joseph Amiot. Oggetto di numerose versioni tradotte, la revisione attuale – reperibile gratuitamente su Internet visto che i diritti d'autore sono abbondantemente scaduti – è di Bob Sutton ed è basata sulla edizione ricca di annotazioni di L. Giles (1910).

[220] Ad esempio, il disco rigido fa rumore di sassi di fiume trascinati da una piena? Se si, inutile cercare oltre.

[221] Ovvero: il problema si presenta con un programma ben preciso oppure in diversi programmi e situazioni? Nel primo caso *probabilmente* è un baco nell'applicazione, nel secondo *probabilmente* un bug del sistema operativo. E ripetiamo *probabilmente*, *probabilmente, probabilmente.* Del Buggon non v'è certezza.

dovremmo eseguire qualche piccola prova atta a circoscrivere il Bug e mostrare la sua vera natura. Il comportamento del combattente non è dissimile da quello di uno scienziato che, eseguendo ripetuti esperimenti, cerca di scoprire quali siano le leggi soggiacenti a quello che vede e verifica dati alla mano. La differenza è che qui la legge soggiacente è un malfunzionamento, un bug, è la Volontà del Nemico stessa.

In questa fase bisogna stare molto attenti alle parole di Sun Tzu, quando afferma: *ogni azione di guerra è basata sull'inganno; dobbiamo sembrare incapaci di attaccare quando già lo stiamo facendo, dobbiamo sembrare lontani quando siamo vicini*. Insomma, un po' di fantasia e bastardaggine non guastano mai. Quantomeno, mentre cercate di capire cosa c'è che non va, provate a fischiettare facendo finta di nulla.

Sun Tzu dice: *far la guerra costa molti soldi*[222]. Il tempo è denaro, e un computer che non funziona ci fa perdere tempo. Ergo, un computer che non funziona ci fa perdere denaro, oltre a ridurci in brandelli fegato ed altre parti del corpo tutt'altro che opzionali.

Ancora una volta, bisogna rifarsi alla Legge di Murphy e ricordare una regola d'oro:

| |
|---|
| Mai far capire ad un dispositivo meccanico che hai fretta. |

Se *davvero* avete fretta, almeno esaminate la possibile soluzione di scendere in strada, entrare in un negozio, comprare un computer nuovo, reinstallare quello che vi serve[223] e poi mettere a posto con calma il malato[224].

Il tempo che sarà necessario per sconfiggere il Nemico è comunque calcolabile: il 90% dei bug verrà eliminato nel 90% del tempo che abbiamo a disposizione, mentre per eliminare l'ultimo 10% dei bug sarà necessario un ulteriore 90% del tempo. La somma fa 180%; per stare sicuri arrotondiamo quindi al 200%, oppure moltiplichiamo il valore iniziale per $\pi/\Phi$. Quindi, fate una stima del tempo che vi sembra ragionevole occupare nella battaglia, per poi mettervi il cuore in pace e prepararvi a star davanti al monitor per il doppio dello stesso. Chiaro?

Sun Tzu dice: *la cosa migliore è impadronirsi dei territori del Nemico senza distruggere nulla, così come è meglio catturare dei prigionieri piuttosto che uccidere e distruggere.*

Proseguendo nel discorso sulla "cura", bisognerebbe sempre tentare di salvare il salvabile, intervenendo sul problema per gradi distruttivi crescenti e lasciando le

[222] Di questo piccolo dettaglio in un secondo tempo se ne devono essere resi conto in tanti, visto che le guerre le vanno a fare dove c'è il petrolio e non dove c'è solo sabbia, sabbia e sabbia.
[223] Avevate fatto le copie di backup, *vero*?
[224] Oppure formattare tutto come consiglia il Reverendo Pampamoon e reinstallare da capo, ma questo non sempre è possibile. Magari sulla macchina malata c'è qualcosa che vi interessa, *e non avete fatto il backup...*

soluzioni drastiche e più costose per ultime. Ad esempio, provare a reinstallare un programma malfunzionante o usare una utility antivirus prima di formattare tutto il disco rigido, o cambiare il cavetto del telefono prima di dedurre che il modem è bruciato. Si noti che questo approccio si scontra con quello sterminatore del precedente paragrafo, per cui bisogna trovare un accordo tra i due orientamenti. Solitamente questo punto di equilibrio è dettato dal buon senso, il che è tutto dire poiché, come abbiamo visto, la pazzia è *conditio sine qua non* per l'utilizzo di un calcolatore.

A parte gli scherzi, non si può formattare tutto il sistemone ogni volta che questo si incarta. Però, una volta ogni tanto non può fare che bene[225].

*[Disegno: un disco rigido/computer tra due calamite]*

Sun Tzu dice: *il buon combattente si mette nelle condizioni di non poter perdere, per poi aspettare l'opportunità in cui possa perdere il nemico.*

Nei casi peggiori, non sappiamo quando il Bug si rifarà vivo nuovamente. Nei casi migliori, non sappiamo quando lo farà per la prima volta. Quello che è quasi sicuro è che *Il Bug Si Manifesterà*[226]. Per questo motivo dovremmo alzare tutte le difese e preparare tutte le nostre armi nell'attesa della sua eventuale manifestazione. Quindi via libera a utility più disparate e soprattutto logging[227], in particolare se non avremo modo di controllare in tempo reale macchine e programmi e solo a posteriori potremo cercare di capire cosa non ha funzionato. Ne abbiamo già parlato: è il *mutandamento.*

Sun Tzu dice: *il controllo di una grande armata [forza, energia] non è dissimile dal controllo di pochi uomini: è semplicemente questione di come questi vengono divisi.*

Qui si va dal ben noto *Divide et Impera* di romana memoria al discorso sulla programmazione strutturata e sulla metodologia Top-Down che abbiamo visto nel capitolo sui Linguaggi di Programmazione. L'insegnamento che dobbiamo trarre allora quale è? Semplicemente, anche quando ci rendiamo conto che la situazione è catastrofica[228], dobbiamo comunque tentare di risolvere *un problema per volta.* Cercare di mettere a posto più problemi in un colpo solo è, nella maggior parte dei casi, tempo perso. Fare una cosa per volta. Molto *samurai*. Molto *cool*.

Il motivo è ovvio: i problemi sono *sempre e comunque* più di quanti pensiamo. Mentre tentiamo di mettere a posto il primo di molti malfunzionamenti, spesso ci renderemo conto che questo è dovuto alla interazione di più bug. Questa peculiarità del nemico ci impone quindi di utilizzare una sorta di metodologia top-down anche nel debugging:

[225] Ad esempio il famoso e già citato analista Paul S. Boorele *non* utilizza antivirus, backuppa regolarmente e ogni due mesi formatta tutto.
[226] A volte succede che il software *non* sia bacato, come abbiamo visto; ma sono casi limite.
[227] Il *Logging* non è altro che la registrazione in forma leggibile di eventi salienti all'interno dei cosiddetti *Log Files* da parte del software, operata da sistema operativo e software applicativo. Ci torneremo più avanti.
[228] Ovvero il 97% delle volte.

cercare, per quanto possibile, di cambiare una sola cosa per volta fino a quando non si ritiene di aver individuato e sviscerato le caratteristiche del Nemico. Ed ora andate a rileggervi la Storia Di Babele XXX a proposito del Connettore Bastardo.

*[Disegno: una "incredible machine" composta di pezzi di computer, dove se tocchi il mouse si muove il disco, si gira il monitor etc e alla fine ti casca tutto in testa]*

Sun Tzu dice: *chi arriva primo sul campo e aspetta l'arrivo del Nemico sarà fresco per la battaglia; chi arriverà secondo sarà frettoloso ed esausto.*

Chi ha tempo non aspetti tempo, recita il Vecchio Adagio. Anche quando tutto va bene – anzi, *soprattutto* quando tutto va bene – un po' di sano allenamento non guasta. Sono questi i momenti per fare manutenzione e prevenzione: aggiornare gli antivirus, cambiare i vecchi cavi danneggiati, archiviare files più o meno importanti, svuotare, controllare[229], deframmentare dischi rigidi, eliminare vecchie utenze non più usate, verificare se sono uscite nuove versioni o updates dei programmi che abitualmente usiamo, nuovi driver, utility che fanno cose estremamente utili o addirittura impensabili[230]…

Attenzione a non esagerare, però: c'è anche un altro Vecchio Adagio, che dice:

| |
|---|
| Computer che Funziona Non Si Tocca. |

Cosa fare? Le operazioni di manutenzione dovrebbero essere svolte tenendo presente il rapporto tra la pericolosità di un certo intervento e la sua importanza-urgenza-utilità:

- Archiviare e far copie di sicurezza dei nostri files è importantissimo e non comporta rischio alcuno, quindi chi non salva i propri dati regolarmente è un *pazzo furioso*; gli utenti più saggi prevedono dispositivi e procedure automatiche al proposito, ad esempio l'utilizzo di un doppio disco rigido[231] e di un programma di backup che si occupi della faccenda. Se il vostro lavoro prevede che stiliate diverse versioni di molti files, magari di testo – come fanno i programmatori, ma non solo – potrebbe esservi molto utile un programma di *version control* come CVS o Subversion, in grado di memorizzare praticamente tutto quello che succede ai vostri *files* in maniera estremamente compatta e quindi manutenibile. Pensateci.

[229] Nel senso del caro vecchio sano CHKDSK e di tutto quello che è venuto dopo.
[230] Come quelle che potete trovare gratuitamente sul sito di SysInternals.
[231] Occhio: un *secondo* disco fisico, non una partizione diversa dello stesso disco che usiamo di solito. Non vorremo fare anche noi l'idiozia della quale ci parlava Rob a pagina XXX, *vero?*

- Aggiornare antivirus e antispyware è importantissimo e, a meno che il relativo produttore non faccia qualche idiozia monumentale, solitamente immediato ed innocuo; quindi, cercate di mantenere sempre aggiornato il vostro antivirus[232].

- Deframmentare un disco rigido è importante per aumentare l'efficienza della macchina e ogni tanto va fatto[233], ma è una operazione potenzialmente pericolosa: se il computer dovesse malauguratamente spegnersi nel mezzo di questa operazione, oltretutto lunga, il contenuto del disco rigido potrebbe trasformarsi in pappa per gatti. Senza voler fare del terrorismo, insomma, *occhio al defrag*. Se proprio siete dei fanatici, esistono dei prodotti specializzati molto più sicuri e veloci del *defrag* standard solitamente fornito con *quel* Sistema Operativo Che Voi Ben Sapete.

- Installare Nuove Versioni di programmi, patch e service pack di sistemi operativi e soprattutto driver tende ad essere Fonte di Mal di Testa. Troppo spesso, per togliere un baco nel più breve tempo possibile abbiamo visto programmatori anche bravi commettere errori madornali, in particolare su software grosso e/o complesso come kernel di sistemi operativi o suites di programmoni ciccioni[234]. In generale, la già vista Sindrome Della Nuova Versione è una patologia da tenere sotto controllo, a meno che la Nuova Versione non sia veramente *molto* più avanzata e potente di quella che avete attualmente installata. Se il vostro computer funziona bene e se non avete *davvero* la necessità di aggiornare il software di cui sopra, lasciate tutto come è[235] – a meno di avere a disposizione una Macchina Sacrificale[236] o di sapere *veramente* bene quello che state facendo. In ogni caso, ricordatevi che nella maggior parte dei casi non fa male disinstallare le vecchie versioni dei programmi prima di installare quelle nuove.

Sun Tzu dice: *Disturba [il Nemico], ed impara il principio della sua attività ed inattività.*

Non si può aspettare sempre che il Bug si manifesti spontaneamente. Ogni tanto bisogna dargli una bottarella. Ovvero, il metodo per l'individuazione del Nemico passa necessariamente per fasi nelle quali si cerca di ricostruire il Suo habitat. Ad esempio, si può controllare se su una macchina "gemella", a fronte delle stesse operazioni, il Bug si manifesta o meno; questo esercizio è spesso fonte di stupore, poiché quasi sempre si scopre che macchine teoricamente identiche una all'altra fin nei minimi particolari

[232] Perché voi *avete* installato un antivirus, *vero*?
[233] Soprattutto se lavorate con programmi multimediali come editor e registratori audio/video, sequencer, compressori e simili. Le operazioni di lettura e scrittura di grandi file sono molto più efficienti su un disco deframmentato.
[234] Un esempio? XPPro su portatile appena acquistato ed è tutto OK. Aggiungi SP2 e non va più il mouse USB. Ma dai! Vergogna! Chiedete a Babele Dunnit, chiedete!
[235] Fanno eccezione patch particolarmente importanti e caldamente consigliate, ad esempio quelle definite "critical" e solitamente rilasciate per motivi di sicurezza. In quel caso non abbiamo scelta. Si incrociano le dita e si clicca su *Ok*.
[236] Ovvero un computer di test che non contiene dati importanti e del quale non vi interessa un granché.

raramente lo sono davvero. Questo aiuta a circoscrivere la natura del Nemico a determinati ambiti, come ben precise versioni di sistema operativo, librerie o addirittura revisioni di hardware e relativi driver lievemente diversi[237].

Sun Tzu dice: *La Tattica Militare è come l'acqua, che nel suo corso naturale scorre via dai luoghi alti e fluisce a valle, così nella guerra la via è quella di evitare ciò che è forte e colpire ciò che è debole. Così come l'acqua non mantiene una forma costante, nella guerra non esistono condizioni costanti.*

Sun Tzu dice: *In guerra [...] il Generale deve miscelarne ed armonizzarne i differenti elementi [dell'Armata] prima di raggiungere il campo di battaglia.*

Nella lotta contro il Grande Bug non dovremmo mai concentrarci in una sola direzione, anche se sembra quella giusta. Il Nemico è mutevole; proprio in risposta a questa caratteristica, dovremmo mantenere all'erta tutti i nostri sensi informatici.

Questo si traduce nell'utilizzare sempre più o meno contemporaneamente tutte le utility sistemistiche che riteniamo possano essere adatte a snidare il Nemico in un determinato frangente. Ad esempio, programmi che ci informino sullo stato di salute di una macchina[238], ma anche altri che controllino il funzionamento delle reti e di altri eventuali componenti dell'infrastruttura informatica che stiamo usando. È sempre più pressante e critica la necessità di spie ed informatori... senza imitare necessariamente vari pazzoidi che hanno dotato di connessione di rete anche il distributore di lattine: soluzione comoda per sapere a distanza se il chinotto è finito, ma forse eccessiva...

```
>finger coke@coke.ucc.gu.uwa.edu.au.

UCC Finger Gateway Results:

[172.26.42.65]
The University Computer Club                http://www.ucc.gu.uwa.edu.au/
------------------------------------------------------------------------
The UCC Coke machine.

Slot 0 has sold 3 drinks of type screaming soda.
Slot 1 has sold 32 drinks of type passion foo.
Slot 2 has sold 32 drinks of type ginger beer.
Slot 3 has sold 32 drinks of type screaming soda.
Slot 4 has sold 32 drinks of type club lemon.
```

[237] Questa situazione è così temuta dai programmatori Windows da avere un nome ben preciso: *DLL Hell*. Benvenuti all'Inferno.

[238] Banalmente, quanto spazio libero ha su disco rigido? Quanta RAM? Quanti processi stanno girando? Quali servizi?

```
Slot 5 has sold 32 drinks of type orange foo.
Slot 6 has sold 32 drinks of type coke.

May your pink fish bing into the distance.
--------------------------------------------------------------------------
Debian GNU/Linux      Copyright (c) 1993-1999 Software in the Public Interest

© 1996 Webmasters <webmasters@ucc.gu.uwa.edu.au>
University Computer Club, University Of Western Australia.
```

*Sun Tzu dice: Ci sono strade che non vanno seguite, momenti in cui una armata non va attaccata, città che non vanno espugnate, posizioni [geografiche] che non vanno contese [al nemico], comandi del Sovrano che non vanno eseguiti.*

Sempre e comunque, fidatevi della vostra Intuizione, perché – se mai ci arriverete – sarà solo grazie ad Essa che raggiungerete lo status di Grande Maestro della Scatola Nera. Seguite i consigli di chi è più esperto di voi, ma tenete presente che anche i Grandi sbagliano. Se avete la fortuna di avere al vostro fianco un Maestro, discutete con lui delle vostre ipotesi e teorie, senza paura di far brutta figura. Il Maestro sa riconoscere l'Illuminazione, anche se si manifesta nel più scarso dei suoi allievi (fletto i muscoli e sono nel vuoto).

Sun Tzu dice: *parleremo ora di accampare l'armata osservando i segni [della presenza] del Nemico. Superate velocemente le montagne e mantenetevi vicino alle valli. Fermatevi in luoghi alti, rivolti verso il sole. Dopo aver attraversato un fiume, allontanatevi subito da esso.*

Ci sono posizioni intrinsecamente sicure ed altre pericolose. L'idea è quella di restare vicini alle prime. Abbiamo visto come a volte, per sconfiggere il Nemico, sia necessario porsi in situazioni instabili, installando software che sappiamo essere vecchio o bacato o dipendente da versioni particolari di una certa libreria. Insomma, a volte è necessario varcare deliberatamente la soglia del *DLL Hell*, il temuto Inferno Delle Librerie Dinamiche. In questi casi è *fondamentale*, dopo aver eliminato il Bug, ritornare ad una situazione sicura che, va da sé, dovreste aver fotografato a priori[239].

Inoltre il momento immediatamente successivo all'eliminazione di un problema particolarmente fastidioso è quello migliore per salvare il vostro lavoro ed in generale svolgere tutte quelle operazioni che richiedono calma e concentrazione. Soprattutto è il momento ideale per cercare di capire e ricordare, anche per iscritto, cosa è successo, visto

[239] Ad esempio con qualche utility di backup totale globale o meglio ancora di mirroring del disco rigido, delle quali parleremo tra poco.

che il Nemico prova una particolare soddisfazione nel ripresentarsi ed udire frasi quali "Come cacchio l'avevamo risolta questa roba qui?" e "Ancora? Ma perché? Perchéééé??". Quindi scrivete. Bloggate. Documentate. Informate i posteri dell'arduo debugging.

Sun Tzu dice: *L'alzarsi in volo degli uccelli è segno di una imboscata. Se gli uccelli si posano in un posto, esso non è occupato.*

Il Bug non arriva mai da solo. Il *Senso del Baco* si sviluppa solo col tempo, ma un Maestro è in grado di *percepire* se un computer gode di buona salute o meno, da moltissimi dettagli pressoché invisibili agli altri: ritardi nelle operazioni più banali, una attività troppo elevata del disco rigido, troppi pacchetti TCP/IP persi... bisogna acquisire molta esperienza per arrivare a un tale stato di grazia. Non desistete: un giorno potrete capire se un computer funziona solo *guardandolo*.

Sun Tzu dice: *Esistono sei tipi di terreno: accessibile, paludoso, bloccante, [di passaggio] stretto, alti baratri, località molto lontane dal nemico. [....] L'Arte della Guerra riconosce nove campi [di battaglia, ovvero situazioni]: dispersivo, facile, conteso [disputato], aperto, difficoltoso, intricato, disperato. Su campo dispersivo, non combattere. Su campo facile, non fermarti. Su campo conteso, non attaccare. Su campo aperto, non bloccare la strada al nemico. Su campo difficoltoso mantieniti in marcia. Su campo intricato usa degli stratagemmi. Su campo disperato, combatti.*

A seconda del terreno sul quale ci muoviamo, diverse strategie vanno seguite. In effetti ogni battaglia, ogni Nemico ha le sue caratteristiche peculiari, anche se con l'esperienza di impara a riconoscerne la tipologia. Primaditutto, Sun Tzu consiglia di occupare assolutamente il terreno accessibile, ricco di vie di comunicazione, ed in questo ricorda Napoleone secondo il quale "il segreto della guerra risiede nelle comunicazioni". Pensate quanti incazzotroni[240] in meno avrebbe emesso Babele Dunnit se avesse scoperto prima la faccenda del Piedino Termoelettrico XXX. Documentarsi, ovvero vedere se *quella cosa lì* è già successa a qualcuno è la *prima cosa da fare*.

Nel caso di terreno paludoso, conviene attaccare solo se si è sicuri della vittoria, poiché una eventuale ritirata, ovvero la necessità di compiere delle operazioni di rollback[241], sarebbe disastrosa. Per fortuna i terreni paludosi si possono bonificare in molti modi; alcuni li abbiamo già visti (le infinite variazioni sul tema del backup), altri li vedremo.

[240]L'Incazzotrone è la particella elementare del nervosismo. Ha la somma di massa e velocità costante, ma la distribuzione delle due varia da individuo a individuo e non è determinabile a priori (Principio di Alterazione dell'Umore di Heisenberg). È anche responsabile dello spin (giramento), sia esso positivo o negativo, degli attributi.

[241] Per *rollback* si intende una operazione di annullamento delle modifiche fatte a un sistema, in modo da ripristinare lo stato iniziale. Il concetto è banale, ma *eseguire* veramente un rollback può essere estremamente complicato, se ad esempio il sistema è composto da più macchine, nonché (spesso a livello "politico") latore di onta imperitura ed eterna.

Non bisognerebbe fare nulla controvoglia, ma in particolare l'andare a caccia di bachi quando abbiamo la testa altrove è tempo perso. Se non è giornata lasciate perdere. Se invece vi trovate in stato di grazia, non fermatevi e combattete la vostra Epica Battaglia senza dar tregua al nemico.

Un terreno bloccante è tale quando nessuna delle due parti trova conveniente attaccare per prima, provocando quindi una situazione di stallo. Se il Nemico c'è ma non si vede, ovvero è sopportabile e non troppo fastidioso, o se davvero non c'è tempo per combattere, si può anche decidere di lasciarlo lì indefinitamente. L'importante è ricordarsi che *in quel punto c'è un Bug.* In ogni caso Le Cose Cambiano e il Nemico prima o poi passerà il segno, rendendo improcrastinabile la sua eliminazione. Si vedano anche le già affrontate *Considerazioni Economiche*.

Se il Nemico si manifesta sistematicamente in determinate situazioni, bisogna gettare l'esca ed aspettarlo al varco, cercando comunque di mantenere una visibilità dell'intero sistema. Questo è interessante anche alla luce della regola 80-20: ricordatevi che usate solo il 20% del vostro computer e di quello che c'è installato sopra – ecco lo stretto passaggio da presidiare.

Quando siete in una situazione appetibile per il Nemico, ad esempio state installando hardware o software del quale nulla sapete, cercate di occupare posizioni sicure. Ad esempio, scegliete versioni magari più vecchie ma riconosciute come più stabili anziché l'Ultima Versione Che Fa Anche Il Brasato; se poi quello che avrete installato vi piace e davvero vi serve fare il brasato, fate sempre in tempo ad aggiornarlo. Anche comprare hardware estremamente nuovo, anche se desiderabile per molti ovvi motivi, a volte può essere controproducente – soprattutto se a chi lo userà non serve davvero. Ricordatevi che per un cliente-utente-tipo è meglio un computer senza lode ne infamia ma che funziona bene piuttosto che un supercomputer al quale bisogna cambiare il pannolino ogni tre giorni – e non c'è nulla di più noioso per un CyberNinja che cambiare il pannolino ad un computer non suo.

Se il sistema oggetto della battaglia è un colabrodo, è inutile tentare di arginare il Nemico. È necessaria una dotazione minima di sicurezza: giro di antivirus e antispyware, firewall ed attitudine al backup, come già più volte detto.

Non bisognerebbe *mai* lasciare in situazione di instabilità un computer o un programma più del tempo strettamente necessario. Se qualcosa non funziona, riparatelo appena potete, perché vi servirà quando meno ve lo aspettate e non avrete assolutamente tempo per metterlo a posto. Come già detto, il Nemico ha tra i suoi superpoteri l'Effetto Valanga dei Conigli di Fibonacci e si riproduce con velocità più che esponenziale.

Bisogna debuggare con calma, avvicinandosi per gradi alla Soluzione dell'Enigma anche quando si sa di avere poco tempo[242]. Chi si fa cogliere da fretta e panico è fregato. Babele Dunnit è molto esplicito:

***Le Storie Di Babele XXX: Su Certe Cose Non Si Scherza.***

Ricordo che quando ero alle prese con il software del sistema di estrazione dei vapori di wolframio delle Miniere Blorbeggianti di Ipostrelleon VII mi ritrovai chiuso in un cilindro metallico chiamato Capsula Sistemistica Definitiva. Il dispositivo in questione è simile ad uno scaldabagno (altro che Soyuz, gente!) dal quale fuoriescono decine di tubi per portare giù allo Sviluppatore Sacrificale lo stretto necessario (nel mio caso pasta di mandorle, pizza, birra, protossido di azoto, cose così... sono un ragazzo semplice). Da lontano il tutto faceva l'effetto di un calamaro fritto in una scatola di pelati, però alto un paio di metri. Insomma, questi mi lanciano giù in caduta libera per seicento metri e mi ritrovo a galleggiare in un mare di metallo fuso, mentre tutto intorno si scatenano fulmini globulari e altre bufere elettrostatiche di indicibile potenza. "Come va?" mi chiede il Controllo, e io: "Tranquilli, per la pizza faccio io, mandatemi gli ingredienti che qui il forno è bello caldo". Dopo sei minuti mi arrivano giù farina, pomodoro e tutto il resto. "Guardate che scherzavo" gli dico, e il Controllo mi risponde: "gnanca bon che non sei neanche capace di fare la pizza nel mezzo di una tempesta di wolframio fuso". Certe cose mi fanno andare fuori di testa ed inizio ad emettere incazzotroni, gente. C'è chi dice che mi si vede al buio. È un attimo: riprogrammo gli scivoli di raccolta del magma, li giro in su e mi lancio con tutta la Capsula – motori a chiodo – verso lo tsunami di vapore che viene generato quando batto "Yes" sul terminale che mi chiede "Are You Sure?". In superficie se ne accorgono e chiudono i boccaporti, ma troppo tardi: mentre l'ondata li travolge urlo: "LA MOZZARELLA DI SECONDA SCELTA VE LA MANGIATE VOI!". Va bene tutto, ma sulla qualità degli ingredienti non transigo.

È terribile lavorare sotto pressione. Si va subito fuori di testa.

Infine, quando proprio tutto è perduto, non rimane che rimboccarsi le maniche. A quel punto *vale tutto*: ad esempio, una bella partita di golf. Come dice D!ego: "Buona idea. Sarebbe bello programmare un computer a colpi di mazza da golf".

Sun Tzu dice: *Ci sono cinque modi di attaccare col fuoco. Il primo è incendiare l'accampamento del Nemico; il secondo è bruciare i negozi; il terzo è bruciare le*

[242] Questo è davvero molto, molto difficile. Se ci riuscite siete *già* dei Samurai. In tal caso buttate via questo libro: non vi serve più.

*carovane di bagagli; il quarto è bruciare arsenali e magazzini; il quinto è bombardare il Nemico con frecce incendiarie.*

Parlando di Distruzione,Waterloo e Debacle – tutti temi cari ad un vero softwarista punk quel tanto che basta – possiamo vedere che Sun Tzu ci sta facendo semplicemente notare che a volte si fa prima a rifare qualcosa da capo piuttosto che cercare di metterla a posto. Spesso quando si scrive del software – ed in particolare da quando è nata la Programmazione Object Oriented – è abbastanza facile dedurre approssimativamente la posizione di un Bug, anche se scovarlo, capirne il principio di funzionamento[243] e quindi eliminarlo può essere arbitrariamente complesso. Piuttosto che perder tempo su un baco difficilmente riproducibile, battaglia questa che potenzialmente potrebbe durare giorni e provocare forti mal di testa[244], è a volte preferibile riscrivere qualche decina o anche centinaia di linee di codice. Lo stesso discorso è purtroppo universalmente applicato all'hardware, ove vige ormai il concetto dell'"usa e getta"; difficilmente troverete qualcuno capace di trovare e riparare il problema – magari un condesatore da mezzo euro bruciato – della vostra amata e per il resto perfettamente funzionante scheda da un gozillione di dollari. D'altra parte non esistono più neanche i paraurti riparabili e i carrozzieri che abbiano tempo e voglia di ripararli: ormai si butta via tutto e si sostituisce, perché da un punto di vista del risparmio di tempo, dicono, conviene così. Peccato che il risparmio di tempo (e denaro) sia delle case automobilistiche e non nostro. Comunque, mi dicono dalla regia, si chiama Progresso.

Infine, si noti che si possono distruggere e rimpiazzare molte cose diverse allo scopo di eliminare un Bug: interi pezzi di programma, ma anche archivi malfunzionanti, documentazione troppo vecchia e sbagliata, collezioni di dati incomplete... a volte basta cancellare dei files temporanei e lasciare che questi vengano ricostruiti per rimettere le cose a posto. Qui solo i forum su Internet e l'esperienza di altri utenti possono metterci sulla buona strada.

*[Disegno: un uomo (bonzo?) infuocato che lavora al computer]*

Sun Tzu dice: *Quando attacchi col fuoco, fallo stando sopravento. Non attaccare col fuoco quando sei sottovento.*

Molto saggio, no? Attenzione a non distruggere più del necessario, quando – in preda al Sacro Fuoco del *Redo From Scratch* – iniziate a disinstallare, cancellare e formattare. Inutile buttare via *tutto*, se sapete che *quasi tutto* funziona. Cercate di capire dove è il Nemico e toglietegli il terreno da sotto i piedi. Disinstallate e ripulite. E se proprio decidete di formattare tutto, siate sicuri di aver messo da parte tutto quello che vi serve: le password di tutto quanto il sistemone, i messaggi di questi ultimi dieci anni, gli account

[243] Può sembrare strano parlare del *funzionamento* di un Bug, ma, dal *suo* punto di vista, il Nemico sta solo facendo il proprio dovere.
[244] Parliamoci chiaro: mettere le mani nel codice altrui è *atroce*.

di posta, quelle utillity che chissà da dove arrivano ma sono così utili, la directory dei documenti personali, i bookmark di tutti i siti preferiti… non datevi fuoco da soli. È spiacevole.

Se invece proprio avete deciso, ecco il consiglio del molto più pragmatico Reverendo Marcus Pampamoon:

```
Non è necessario installare/disintallare applicazioni. Basta formattare
tutto a basso livello, non prima di avere sventolato lentamente l'hard-
disk molto vicino ad un magnete da 10E6 Hoerst, subito dopo il consueto
bagnetto nell'acido muriatico ad alta concentrazione.

Già che ci sei, non dimenticarti di rigare lo schermo e di
cortocircuitare la motherboard. Se non ti ricordi come si fa, un paio
di colpi di martello bene assestati sul chip dovrebbero fungere alla
bisogna.

Sto dalla parte di Giove e dei suoi fulmini, io, che credete?

L'iconoclastia, una necessità.
```

Per inciso: ora che i dischi rigidi son belli grossi, è buona abitudine partizionarli in modo da mantenere sistema operativo ed installazioni su una partizione e tutti i dati sull'altra. Ormai per quasi tutti i programmi è possibile specificare la posizione ove salvare i propri dati; metterli sull'altra partizione *è cosa buona e giusta*. Questo non vi proteggerà da un crash del disco rigido, ma in caso di impappamento totale della macchina potrete formattare e reinstallare sistema operativo e applicazioni sulla partizione primaria senza toccare messaggi, directory personali e tutto ciò di cui sopra. Questo è un accorgimento che può farvi risparmiare *giorni* di lavoro.

Sun Tzu dice: *Ciò che permette al sovrano e al generale di colpire e conquistare, ed ottenere cose al di là della portata dell'uomo comune, è la Conoscenza [predittiva]. Questa conoscenza non può essere tratta dallo spirito, ne può essere ottenuta induttivamente dall'esperienza o dal calcolo deduttivo. La conoscenza delle disposizioni del Nemico può essere ottenuta solo da altri uomini.*

Con questo ultimo concetto di Sun Tzu, lo *spiare il nemico*, traghettiamo finalmente verso il Campo di Battaglia. Siete pronti per l'azione? Si? E allora...

*[Disegno: Contrattacco! Settimo cavalleggeri, quarto stato...]*

## Contrattacco!

Katana per il Samurai, Barbera per Superciuk, Lame Rotanti per Goldrake... Ogni aspirante al trono di Dalai Bilama del Proprio Computer dovrebbe prendere confidenza con una serie di armi – per la maggior parte *utility* software, ma esistono anche dispositivi hardware più o meno esoterici – che ci permettono di incrociare dati e formulare ipotesi mentre inseguiamo la Suprema Intuizione Debuggatoria. Vediamo quindi quali sono le principali Armi Supreme di un Samurai Digitale.

**Dal Diario Di Bordo del Capitano Kirk[245], Data Astrale...**

Inizieremo parlando del concetto di *logging*, ovvero di registrazione degli eventi che, dietro le scene, avvengono in un calcolatore.

Alcuni di questi strumenti sono standard, altri no; però tutti hanno lo scopo di fissare all'interno di un *logfile* la sequenza degli ultimi[246] eventi significativi avvenuti in un certo ambito, insieme a data, ora e altre informazioni a riguardo. Questi *log* possono essere dei semplici file di testo, nel qual caso spesso hanno ".LOG" come postfisso e sono leggibili con un normale editor[247], oppure possono essere dei files specializzati come quello dell'Event Viewer di Windows, con postfisso ".EVT" e consultabile solo con l'Event Viewer stesso[248].

Moltissimi programmi generano dei logfile; ad esempio, quasi tutti i programmi di installazione ed aggiornamento automatico. Per cui, se dopo una di queste operazioni il vostro computer si incarta, al posto di buttare tutto dentro il cassonetto dell'umido[249] potete tentare di capire cosa è andato storto andando a cercare il relativo log. Potreste scoprire, ad esempio, che non avete abbastanza spazio su disco rigido[250], oppure non avete i permessi[251] per svolgere quel tipo di operazione.

---

[245] "James Tiberius Kirk fu l'unico cadetto dell'Accademia Spaziale a passare il test della Kobayashi Maru. Solo quando divento Ammiraglio, svelò il suo segreto. Dato che non c'erano soluzioni possibili, modificò il test in modo che ne prevedesse almeno una: la sua. A bordo dell'Enterprise incontrò Spock e continuò a dimostrare che l'intuizione, a volte, supera o anticipa i risultati del ferreo ragionamento logico." (N.d.T. pag 31, "Non Siamo Mai Andati Sulla Luna", Bill Kaysing, Cult Media Net Edizioni, 1997). Ma non si chiamava Cornelius?

[246] Qui "ultimi" è un termine molto relativo, poiché, essendo solitamente possibile decidere dimensioni e granularità di un log file, non è raro trovare macchine con logfiles che riportano eventi vecchi di mesi o anche anni.

[247] O meglio ancora visti in tempo reale con il comando U*IX "tail" o succedaneo quale WinTail.

[248] Il che è peraltro un esempio di pessima progettazione del software. Usare file con formato proprietario quando si potrebbero usare dei normali file leggibili è una mossa stupida.

[249] Dopo aver sputazzato abbondantemente sul computer, sennò che umido è?

[250] Sembrerebbe una cosa scontata aspettarsi un controllo dello spazio disponibile da parte di un programma di installazione. In fondo, lo stramaledetto in questione dovrebbe sapere quanta roba sta per scaricare sul vostro disco rigido. Ma così evidentemente non è, a giudicare da quante volte succede che questi programmi di installazione si incartano a metà. Mah.

[251] Non tutti gli utenti di un computer possono fare qualsiasi operazione. Siccome un computer può essere utilizzato da più di una persona, vengono spesso creati diversi *account* relativi ai diversi utilizzatori, che quindi attribuiscono loro diversi poteri. Questi vanno da un livello di *guest*, ospite che può solo leggere e scrivere in pochi e limitati luoghi, fino ai superpoteri di *root* (o *Administrator*), imperatore e padrone assoluto del vile metallo. Non è raro scoprire che, ad esempio, non riuscite più a cancellare un file perché

A seconda delle applicazioni, il livello di dettaglio di logging può essere configurabile: se tutto va bene si tiene basso, in modo che solo gli eventi principali vengano segnalati. Altrimenti si può alzare in modo da aumentare (anche vertiginosamente) la quantità di informazioni su ciò che il vostro computer sta facendo. Attenzione: un alto dettaglio è potenzialmente pericoloso, perché si rischia un sovraccarico informativo in grado di moltiplicare le dimensioni dei logfile e trasformare il contenuto della vostra scatola cranica in pappa per gatti.

Inoltre, una delle cose più difficili da fare usando il meccanismo del logging è verificare il funzionamento contemporaneo di più applicazioni che parlano tra loro[252]: in questo caso è di fondamentale importanza che gli orologi delle varie macchine siano sincronizzati e vedere i log in tempo reale è caldamente consigliato. I veri ninja usano dei framework di logging, ma non esageriamo.

Infine è da citare un caso *molto* particolare, segnalato da Rob: quando lo spazio su disco finisce – magari proprio perchè stiamo tenendo il livello di logging al massimo – il sistema operativo cerca di segnalare il malfunzionamento in un logfile, ma lo spazio su disco è, appunto, finito.. quel che succede immaginatelo voi. Anche se vi sembra impossibile, sappiate che questo è successo e quindi, secondo il Grande Samsara Del Bug, succederà ancora ed ancora...

**Porte Tagliafuoco e Reti Da Pesca**

Abbiamo già incontrato il Firewall, in quanto importante strumento che ci permette di filtrare i dati che passano per il nostro computer in base a svariati parametri. È il caso di ricordare che proprio questa applicazione genera un altro logfile molto interessante da esaminare; essendo preposto a monitorare chi viene e chi va sulla vostra connessione di rete, il firewall può darvi un sacco di informazioni su sequenze di accessi (più o meno legali) alla vostra macchina, velocità di trasmissione dei dati, stato di salute delle vostre schede di rete... si noti che esistono anche dei firewall hardware, spesso incorporati in altri dispositivi quali hub, proxy e simili. Il concetto non cambia di una virgola.

**Specchio, Specchio Del Mio Disco Rigido...**

In molti casi sarebbe comodissimo poter congelare lo stato di un computer, in modo da star tranquilli che, in caso di catastrofi, sia possibile tornare alle condizioni iniziali. Esistono numerose utility in grado di svolgere questa operazione, chiamata *mirroring* del disco rigido e lievemente diversa da un *backup*: se quest'ultimo è una copia dei nostri dati più importanti in una locazione secondaria, il mirroring è una vera e propria fotocopia del disco rigido di un computer. Tende quindi ad essere una operazione relativamente lenta e

non ne avete il diritto (ad esempio *qualcuno* (?) vi ha tolto i privilegi di *root* su quel preciso documento).

[252] Che è una situazione molto comune quando siamo in presenza di più computer tra loro connessi in rete; si parla in questi casi di architetture distribuite o *client-server*, dove un computer che ha bisogno di un certo dato, servizio o programma lo richiede ad un altro che è in grado di fornirlo.

dispendiosa in termini di occupazione di memoria, ma può farci risparmiare giorni e giorni di lavoro in caso di eventi catastrofici, o nel caso si abbia la necessità di produrre più macchine gemelle[253].

L'utilità di un simile meccanismo è tale che alcuni sistemi operativi[254] lo hanno incorporato per default, dopo aver verificato col passare degli anni che il concetto di una semplice disinstallazione per molti programmi non funziona[255]. In cambio di una quota in verità abbastanza considerevole del proprio disco rigido è quindi possibile creare automaticamente delle fotografie dello stato del sistema, alle quali tornare in caso di necessità. La cosa interessante da dire qui è che altri sistemi operativi più furbi[256] semplicemente *non hanno* il concetto di "installazione", per cui basta semplicemente copiare dei file per fare funzionare una applicazione.

Un approccio completamente diverso è possibile mediante i cosiddetti *emulatori*[257]: si tratta di programmi che simulano l'hardware di un intero computer, sul quale possiamo installare il sistema operativo, i programmi e tutto quello che vogliamo. Tutto finisce in un grosso file che rappresenta il "disco rigido" di questo computer virtuale, che possiamo copiare e mettere da parte quando le condizioni iniziali ci soddisfano. Da quel momento in poi, tutto quello che facciamo rimane ivi confinato[258], per cui possiamo sperimentare liberamente configurazioni strampalate, software instabile, virus e schifezze varie, insomma tutto quello che ci passa per la testa. Quando abbiamo finito, buttiamo via tutto e ripartiamo da zero. Non solo: possiamo addirittura simulare *più* computer virtuali – tra loro collegati in rete, virtuale anche quella – su un solo calcolatore reale, rendendo così possibile la verifica di complesse architetture distribuite ed eterogenee con investimenti contenuti. Ovviamente, nulla si crea e nulla si distrugge: il computer reale, che ospita le simulazioni, deve essere una macchina *potente*, altrimenti i computer virtuali saranno veloci quanto delle macchine a vapore e ci addormenteremo in attesa che rispondano ai nostri comandi.

**Spiacente, Solo Posti In Piedi**

---

[253] In questo caso anche l'hardware delle macchine gemelle deve il più possibile essere identico, il che a volte non è così scontato.

[254] Dei quali non faremo i nomi perché sarebbe come sparare sulla proverbiale Croce Rossa.

[255] Per questi sistemi operativi le modifiche operate in fase di installazione di un programma sono talmente numerose ed eterogenee da rendere, a volte, estremamente difficile – se non impossibile – la reversibilità del processo (ovvero il già citato *rollback)*. In particolare, installazioni successive tendono a distruggere gli stati intermedi: ogni installazione potrebbe salvare lo stato precedente della macchina, ma saremmo obbligati, per tornare davvero indietro, a disinstallare i programmi nella sequenza inversa, il che non è proponibile.

[256] Dei quali non faremo i nomi perché sarebbe come dire "beh, sei bravino con la matematica" ad Albert Einstein.

[257] Attualmente il più famoso prodotto commerciale di questo tipo è probabilmente WMWare, ma stanno nascendo varie alternative Open Source da tenere d'occhio, tra le quali Bochs.

[258] Situazioni di questo tipo vengono chiamate *sandbox*, con attinenza ai recinti da *kindergarten* marinaro dove i bambini possono giocare liberamente senza far danni – a parte tirarsi vicendevolmente la sabbia negli occhi.

Sembra incredibile, ma anche i sistemisti più bravi a volte si lasciano fregare da problemi assolutamente banali, come un disco rigido pieno o un numero eccessivo di file in una singola cartella[259]. Per questo motivo bisognerebbe prendere confidenza con qualche utility[260] che, a colpo d'occhio, ci dica un po' di cose fondamentali: spazio libero su disco, attività della CPU, paging, memoria virtuale, traffico di rete, fino ad arrivare alla temperatura del processore, sulle macchine equipaggiate di apposito sensore... in realtà sono tutte informazioni recuperabili in altro modo con vari comandi di sistema, ma è molto meglio avere una singola applicazione, magari configurabile, che ci permetta di tenere tutto sottocchio senza vagare come l'Ebreo Errante tra cartelle e menù.

Ovviamente, il Nemico non si nasconde solo nella *nostra* macchina: per questo motivo, analoghe *utility* dovrebbero essere usate quando il buon funzionamento del nostro calcolatore dipende da quello di altri. Banalmente, la semplice verifica delle connessioni con altri computer mediante *ping*[261] può farci capire che c'è qualcosa che non va; ulteriori variazioni sul tema vanno da *netstat*[262] a *traceroute*[263], ma uno spera sempre che tutto funzioni. Ovviamente, ogni sistema operativo ha la sua versione di questi comandi, che possono o meno essere standard, possono o meno avere esattamente questi nomi, possono o meno essere dei programmi di tipo visuale piuttosto che *command line*[264]. A voi trovarli.

**Ti Porto La Colazione A Letto?**

Se vi siete messi in mente di lavorare con più di una macchina e vi siete stufati di andare avanti e indietro dallo studio al bagno[265], sappiate che ci sono numerosi programmi in

[259] Per quanto il numero massimo di file memorizzabili in una singola cartella sia alto, bisogna considerare che non è certo infinito; ci sono macchine che non si spengono mai e che, per mancanza di una adeguata assistenza, vengono lasciate a se stesse. Il server di posta di una grossa azienda può arrivare a dover sopportare la creazione di migliaia di file al giorno, uno per ogni messaggio ricevuto; in mancanza di qualcuno o qualcosa che provveda a cancellare i file già utilizzati, il crash del sistema è solo questione di tempo.

[260] Ce ne sono veramente migliaia; una utility simpatica e leggerissima per Windows è TinyResMeter, gratuita, ben 92K (!) e fa anche gli screenshot, mentre Samurize, più complessa e shareware, permette di disegnarsi i propri widget. Poi ci sono Konfabulator e i widget di Apple OsX che fanno anche il vitello tonnato.

[261] Il comando *ping* semplicemente invia dei dati ad un computer remoto e verifica che questo ci risponda. Serve quindi a vedere che tutto vada bene, ma in caso di problemi non ci dice se l'altro computer è morto o se più banalmente tra i due si è staccato qualche cavo.

[262] Il comando *netstat* ci permette di visualizzare tutte le connessioni tra il nostro computer e l'esterno, e il relativo stato. Sono quindi informazioni analoghe a quelle che può fornirci un firewall.

[263] Il comando *traceroute* (o *tracert*) è simile al *ping*, ma in più ci dice che strada stanno facendo i nostri dati per andare dal nostro computer ad un altro. È quindi un raffinamento del primo, perchè ci permette di capire *dove* i nostri dati si fermano.

[264] Quando ancora non esistevano icone e finestre, i computer si usavano scrivendo i comandi per esteso, come "> $L $FSEDIT EDITWORK, EDX003". In molti sistemi operativi – U*IX sopra tutti, ma anche Microsoft che in qualche modo deve mantenere il suo antico retaggio – questa modalità a riga di comando è rimasta disponibile ed è anzi la prediletta da molti combattenti, grazie alle sue possibilità esoteriche il cui approfondimento esula decisamente dagli scopi di questo libro.

[265] Davvero avete intenzione di tenere più di un computer a casa? Babele consiglia di tenerne almeno uno in bagno, perché è lì che noi umani siamo soggetti alle intuizioni più interessanti.

grado di usare un computer in maniera remota – ovviamente a patto che i due computer siano collegati in rete[266]. Questi vanno dall'antico *telnet* a programmi grafici[267] che permettono di aprire sul primo computer (ad esempio, quello dello studio) una finestra che rappresenta il desktop del secondo computer (ad esempio quello del bagno). Potete a questo punto eseguire comodamente dalla poltrona dello studio le stesse operazioni che fareste presumibilmente assisi sulla tazza sul computer in bagno, a meno che queste operazioni non abbiano alcunché di informatico e siano anzi orientate al soddisfacimento di vostri bisogni primari. In questo caso consigliamo di raggiungere prontamente il bagno.

**Be Objective**

Avete presente quando, nel film *Matrix*, i vari personaggi riescono a capire cosa succede nella Matrice guardandola direttamente in forma codificata? Ormai quella grafica, con lettere e simboli verdi che cadono sullo sfondo nero, è entrato a far parte dell'immaginario collettivo tanto da ispirare copertine di dischi e libri, screensaver, sigle televisive e mille altri articoli, digitali o meno[268]. Ma, come abbiamo già detto parlando di *cracking* dei programmi, l'idea di esaminare – ed eventualmente modificare – dati e programmi direttamente in forma binaria non è del tutto campata in aria ed in alcuni casi particolari è anche accessibile a tutti.

Infatti è possibile – se proprio non potete farne a meno – aprire un file eseguibile con un normale editor di testo – o meglio ancora un *hex editor*, che ci permetta di vedere direttamente in esadecimale[269] il nostro programma. Perchè dovremmo autoinfliggerci questo dolore? Perché stringer siccome viti le nostre vite in un simil cilicio di silicio? Perché, oltre al codice eseguibile – praticamente incomprensibile – troveremo molte altre informazioni testuali, lasciate inalterate dal processo di compilazione: ad esempio, a Babele Dunnit capitò di scontrarsi con un programma malfunzionante, che evidentemente cercava *qualcosa* e che, non trovandola, si rifiutava di procedere, suicidandosi come un lemming[270]. Come risolvere un simile problema? Con la tecnica di cui sopra, e dopo una

[266] In caso contrario esistono sistemi basati su telecinesi e/o urla sguaiate.

[267] Sono veramente molti: *PcAnywhere, Dameware, PcRemote*...Ovviamente noi siamo affezionati a *VNC*, strepitoso e massiccio programma Open Source.

[268] Ma non è certo una invenzione di Matrix: la grafica verde su sfondo nero affonda le sue radici nei mitici ed eroici terminali VT-52 dei tempi che furono, mentre quel modo di muovere i caratteri sullo schermo è stato evidentemente ispirato dall'*anime* giapponese *Ghost In The Shell*, trasposizione video dell'omonimo *manga* di Masamune Shirow ed impedibile visione per gli appassionati del genere.

[269] Lo abbiamo detto, no, che i computer vanno a zeri e uni? Beh, se ne prendiamo quattro per volta possiamo contare da zero a quindici, ovvero da 0000 a 1111 (fate voi la prova). A questo punto, se dopo il 9 usiamo A,B,C,D,E ed F come simboli per i nostri 10, 11, 12, 13, 14 e 15 possiamo esprimere un numero binario di otto cifre – un byte, che ricordiamo può rappresentare i valori interi da 0 a 255 – con sole due cifre esadecimali e con una estrema facilità nel passare dalla notazione binaria, molto prolissa e quindi scomoda, a quella esadecimale, ben più compatta. Va da se che un Gran Visir della Memoria A Bolle conta e converte senza sforzo alcuno e si diverte a comporre parolacce in esadecimale, come 0xCACCA che è uguale a 11001010110011001010. Ma se non avete capito va bene lo stesso.

[270] I Lemmings, peraltro oggetto di un mitico videogioco degli anni '80, sono dei grossi toponi famosi perché si dice che si suicidino quando si rendono conto di essere in troppi. In realtà questa cosa non è assolutamente vera: si dà il caso che questa sia una bufala nata a causa di un documentario di, ahimè, Walt Disney…

decina di minuti di analisi, Babele riuscì a scoprire, sepolta tra sfilze di numeri e lettere senza senso, la stringa "C:\logs", dalla quale ipotizzò che il programma andasse a cercare tale directory sul disco rigido e, non trovandola e non essendo capace di crearla[271], si bloccasse. Creata a manina la suddetta cartella, il programma magicamente iniziò a funzionare.

Altro esempio è quello della *localizzazione* di un programma, ovvero della traduzione dello stesso: esistono dei cosiddetti *resource editor* che, lavorando direttamente sul file eseguibile e quindi senza aver la necessità di utilizzare sorgenti, ci permettono di estrarre e modificare tutti i messaggi di errore, anch'essi solitamente incorporati senza modifiche direttamente nel programma eseguibile, nonché nomi di finestrelle e schermate varie.

Questo è solo un possibile utilizzo di un resource editor. Un altro, molto meno nobile, è quello di pavoneggiarsi con programmi altrui: poiché tra le informazioni solitamente scritte direttamente negli eseguibili ci sono i nomi degli autori, per un *lamer* non è difficile modificarli e saltare sul carrozzone... fino a quando il furbacchione di turno non viene scoperto e si fa la sua bella figura di palta.

Ben altro discorso, dicevamo, è quello di *decompilare* (o *disassemblare)* un eseguibile ed ottenerne il *disassemblato*, ovvero la rappresentazione simbolica del codice macchina. Pur non avendo a disposizione il codice sorgente originale un Maestro ferrato nel linguaggio assembler sarà in grado di operare molte magie su questo materiale: eliminare protezioni, carpire segreti di funzionamento, cambiare delle parti, modificare l'aspetto grafico del programma originale e altro ancora. *Tutto è possibile* quando si parla di software: è solo questione di tempo a disposizione, ovvero di denaro.

Una interessante variazione sul tema è quella possibile con linguaggi semicompilati come Java: in alcuni casi è possibile risalire addirittura al *sorgente originale* di qualsiasi programma (a meno di alcuni dettagli, quali i nomi originari delle variabili e cose simili). Insomma, in questo caso un *cracker* ha vita ancora più facile. Se la sicurezza è il vostro pallino, Java può non essere la migliore delle scelte.

**Non Mi Ricordo Più Niente**

Ci sono, ahimè, volte in cui lo Scatolotto impazzisce completamente. Stop inopinati, spegnimenti misteriosi, comportamenti assolutamente imprevedibili da un momento all'altro, senza che abbiate installato alcunché di nuovo e senza nessun rapporto con quel che state facendo... forse siete in presenza di uno dei Nemici più perfidi: Sua Maestà Il Bug Della Memoria RAM. Quando la memoria centrale di un computer si danneggia – per motivi squisitamente elettrici, quindi è raro ma può capitare – alcuni 1 si trasformano in 0 e quel che ne segue è assolutamente ovvio: il processore lascia certe informazioni in certi posti che sa lui e quando va a rileggerle ne trova altre, diverse e spesso insensate.

[271] Vergogna eterna ed imperitura ricada sul programmatore originario. Non ci vuol molto a creare una directory non esistente, o a informare con un messaggio di errore l'utente che c'è qualcosa che non va. Ma il mondo reale è così, gente.

Questo lo manda in bestia – e vorrei vedere voi, ad aprire la scatola del tonno e trovarci dentro il formaggio.

Persino diagnosticare questo tipo di Bug è tutt'altro che facile. Solitamente la memoria RAM non si guasta in maniera maschia e decisa, ma semplicemente "ogni tanto non funziona". Non è come un disco rigido, che a un certo punto inizia a far rumore di risacca equatoriale: magari un computer funziona per ore e poi di colpo si blocca, si spegne, si resetta o inizia ad ululare. Questo è un comportamento in parte analogo a quello di macchine possedute da un virus o da qualche driver maldestro, da cui il più che comprensibile fraintendimento.

Esistono vari programmi di test della memoria, dei quali probabilmente *MemTest86* è il più approfondito e specifico, mentre la procedura stessa di individuazione del componente rotto non è immediata[272]. In ogni caso, c'è poco da fare: se vi è saltato un banco di memoria dovete cambiarlo, a meno che si tratti di un falso contatto sul connettore. Auguri.

Se il computer si *blocca completamente* mentre giocate a PacMan, compreso mouse e lucine delle maiuscole/minuscole, potrebbe anche essere un problema di surriscaldamento. Provate a controllare se la ventola funziona o se ci è finita dentro la carta delle patatine e, nel caso, rimuovetela[273]. Poi gettate tutto nell'elio liquido per sicurezza e ricominciate a giocare a PacMan.
Altra variazione sul tema, ricordatevi che alcuni sistemi operativi – ed in particolare *Quello Là* – hanno il concetto di *Safe Mode*, ovvero una modalità di partenza che *dovrebbe* garantire un funzionamento il più stabile possibile caricando solo i componenti strettamente necessari del sistema operativo: niente driver estesi di schede grafiche che fanno anche le tagliatelle, niente audio, interfaccia ridotta ai minimi termini eccetera. Insomma, per dirla all'americana, *no bells and whistles.* Il fatto che poi il Safe Mode funzioni e serva a qualcosa è un altro discorso, ma è importante sapere che esiste.

**Lo Scarico dei Centouno**

Ormai una parte del leone, nella battaglia contro il Nemico, la fa il concetto di *download*. Su Internet, dicevamo, ci si può trovare di tutto, ma l' utility, il driver, la patch, il service pack, il manuale che ci interessa in qualche modo dobbiamo scaricarlo. Molto spesso ci si trova davanti a file di dimensioni tutt'altro che trascurabili; anche se ormai le connessioni veloci e *flat rate* sono piuttosto comuni, scaricare una intera suite di programmi diagnostici o una distribuzione Linux *live* come Knoppix[274] può essere abbastanza impegnativo. Inoltre il fatto che certe procedure di aggiornamento siano automatizzate

[272]Il solito alchemico togli-prova-rimetti-prova-scambia-prova-ritogli-prova-rimetti-prova...
[273]la ventola, ovviamente – la carta delle patatine è un componente *fondamentale* per il funzionamento del vostro computer.
[274] Un Samurai Digitale avrà *sicuramente* sottomano una distribuzione CD *live* – ovvero che non necessita di installazione – di qualche versione di Linux: essere in grado di far partire un computer con un diverso sistema operativo senza doverlo installare è il modo più veloce per diagnosticare un problema puramente hardware. Se con un sistema operativo funziona tutto e con l'altro no, il problema *è* software. Period.

non sempre è desiderabile: un sistemista che deve applicare lo stesso Service Pack a centinaia di macchine diverse preferirà scaricare tutto una volta sola anziché lasciare che ogni macchina si colleghi a Windows Update e faccia il tutto in solitaria. Per questi motivi è opportuno sapere che esistono dei *Download Manager* capaci di fare tante belle cose come aumentare la velocità di ricezione, aiutare a mantenere ordinati i file scaricati, ricominciare a scaricare da dove erano arrivati in caso di problemi di linea e cadute di collegamento ed altro ancora. Ce ne sono numerosi, sia a pagamento che free; Babele Dunnit predilige Free Download Manager, che fa tutto quello che uno si aspetta e anche qualcosina di più.

*[Disegno/DIY: le carte Probabilita' e Imprevisti versione informatica, o addirittura tutto un Monopoli da ritagliare]*

Imprevisti

1. Vai a prendere un caffè, ma lo versi sulla tastiera. Cerca di berlo da lì.
2. Hai sbagliato interruttore. Quella era la ciabatta del server. E sei anche reperibile.
3. Hai messo un punto e virgola dopo un ciclo di for. Il tuo programma non fa nulla e ci metti sei ore a capire perché.
4. Devi montare un disco NTFS su una macchina Windows 95. Fai prima a cambiare lavoro.
5. Non dovresti bere sette Golden Margarita la domenica sera. Adesso sono cavoli tuoi.
6. Guarda che se non configuri bene il firewall COL CACCHIO che riesci a collegarti al server.
7. C'è da finire la documentazione. Ti fermi in ufficio fino a quando staccano l'aria condizionata e lo sceriffo panzone della security ti dice che se non te ne vai ti chiude dentro.
8. Il mouse non va più e come ricambio ne hai solo uno di quelli vecchi con la pallina. Devi trovare un tappetino, sennò diventi scemo.
9. Il figlio del boss (12 anni) ti ha impestato il computer di giochini scaricati da Internet. Dopo un giro di antivirus ti accorgi che hai su 127 Trojan.Win32.Shutdowner.i, 44 Net-Worm.Win32.Bozori.b e 38 Backdoor.Win32.XRat.j. Perdi un giorno a fare pulizia.
10. La segretaria ha ingavardato la stampante. Dopo che l'hai messa a posto, ti accorgi che è anche finito l'inchiostro e nel cambiare la cartuccia ti si incastra la cravatta – che hai messo oggi per la prima volta nella tua vita – nel cassetto. Rischi di venire decapitato.
11. Quello non era un CD, era una fetta di salame. A parte capire chi è lo spiritosone che l'ha messa nell'armadietto dei programmi, mi sa che devi recuperare un kit di pulizia.
12. Non avresti dovuto appoggiare il subwoofer da 350 Watt sul disco rigido dalla parte del magnete. Avevi fatto un backup?
13. Dai log del firewall è evidente che stai tutto il giorno a perder tempo girovagando su www.suonerie.it. Il sistemista ti tappa l'IP e adesso son cavoli tuoi.
14. Arriva la finanza e trova ben nove MP3 illegali sul tuo computer. Non lo sai che è un reato gravissimo? Ti danno quattro anni di carcere duro. Quando esci hai imparato la

lezione: prima vai a rubare un CD dei Beatles, poi frappi di mazzate una vecchietta. Ti prendono ancora, ma questa volta ti danno solo due anni con la condizionale.
15. Dopo due settimane di ferie scarichi la posta e ti vengono giù 1237 messaggi, di cui 1231 sono spam. Per smazzare il tutto ci metti quattro ore visto che usi ancora Outlook.
16. La password del tuo conto online è “gollum”. Lo sai in quanti siamo a saperlo?
17. Il gatto ti passa sulla tastiera e con tre delle quattro zampe schiaccia <CTRL><ALT><DEL>, mentre con la quarta batte la password di sistema. Ti conviene scuotere opportunamente la scatola dei croccantini.
18. Pensando di fare il furbo hai registrato www.kevinmitnick.org per poi rivendere il dominio. Non è stata una grande idea. Ora sul tuo disco rigido non c'è più nulla.

Probabilità

1. Sono venti ore che lavori e improvvisamente ti rendi conto che hai fame. Dopo che hai mangiato della pizza va molto meglio.
2. Urla e scuoti il monitor, picchia forte sulla tastiera a mani aperte. Esegui un RyuuJinKyaku (Calcio del Dio Drago). Funziona!
3. È il momento di una bella birra!
4. Il computer nuovo si collega correttamente ad Internet non appena lo accendi. Paghi da bere a tutto l'ufficio.
5. Ti bevi il settimo caffè della mattina e poi tutto diventa più chiaro.
6. Hai le lenti degli occhiali troppo sporche. Dopo averle pulite probabilmente sopraggiungerà l'Illuminazione.
7. Hai battuto ogni record su yetigames.com. Sei il nuovo idolo dell'ufficio.
8. Esercitazione a sorpresa, bisogna uscire dall'ufficio passando dalle scale di sicurezza e tutte quelle cose lì. Sono già le 15, il server è imballato, hai anche un po' di mal di testa... tanto vale andare a casa.
9. È andata via la luce mentre stavi scaricando la *Divina Commedia* dal sito del Progetto Gutenberg. Poco male: avevi installato FlashGot, potrai tener buono quello che hai già scaricato.
10. Dopo due settimane di ferie scarichi la posta e ti vengono giù 1237 messaggi, di cui 1231 sono spam. Per smazzare il tutto ci metti tre minuti grazie ai filtri bayesiani di Thunderbird. Grande!
11. Il gatto ti passa sulla tastiera. Momenti di panico, ma niente che con un bell'*undo* non possa essere rimesso a posto.
12. Sblam! Lo senti questo rumore di cascata? Il disco è andato! Però avevi appena copiato tutti i tuoi dati.. te la cavi con un paio d'ore di lavoro.
13. Oggi è il tuo giorno fortunato. Quel lavoro assolutamente inutile e pure complesso è saltato perché qualcuno ai piani superiori, incredibilmente, ci ha ripensato.
14. Il computer nuovo che volevi comprare due settimane fa lo vendono in offerta speciale da Computer Discount. Ti è andata bene.

## La Quadratura del Cerchio

Abbiamo iniziato con una citazione di Von Taurus ed Il Grande Libro Dei bug non può che avvicinarsi alla fine con una pungente critica ad opera dello stesso autore[275]:

Quello che ti manca è senza dubbio la visione tridimensionale, quella che ti permette di fare il giro e tornare tranquillamente all'inizio da cui eri partito. Se parti sul piatto ti muovi orizzontalmente e non torni più. Ti allontani, ti perdi...

Il tridimensionale, la visione profonda (o Alta che poi è lo stesso) ti permettono di accettare sia il piccolo che il Grande Bug e di riconoscerli nella loro funzione (e non nella DISFUNZIONE come ti ostini a dire in quasi tutte le pagine del tuo stupido e fottuto libro).

Se davvero ci vuoi stupire con qualche sprazzo di saggezza non bastano battutine sapientemente incorniciate in discorsetti più o meno convincenti. La vera visione del Grande Bug ce la puoi dare soltanto quando smetti di accusarlo di darci noia e non gli riconosci la sua funzione nel contesto generale.

Ti faccio alcuni esempi pratici:

gli update che dopo fatti non ti funziona più nulla, la nuova versione che funziona peggio di quella precedente, il driver che esalta la capacità di scoprire che le cartucce della stampante sono esaurite anche se le hai appena scartate e montate... i BUG SONO UTILI! UTILISSIMI! CE LI METTONO APPOSTA!!

Immagina poi quei piccoli e tanti casi in cui si riesce a recuperare una password proprio facendo leva sul fatto che il sistema non era sufficientemente sicuro… vuoi mettere fra come girerebbe l'economia in un mondo informatico senza difetti rispetto a quello pieno di problemi che conosciamo? Davvero vuoi farmi credere che se tutto funzionasse perfettamente senza intrupparsi mai il mondo sarebbe migliore? Premesso che il mondo NON potrebbe neppure essere diverso da quello che è, comunque torno a dire che informaticamente parlando la bassa qualità esalta quella alta e l'errore esalta la sua correzione: utility, antivirus, aggiornamenti vari... l'assistenza esalta il valore aggiunto di SERVIZIO che ogni grande company tende a dare (lamentandosi) ai propri clienti e in un confronto (gara) fra ditta e ditta il mondo gira e l'uomo campa.

Ma supponiamo invece (e ti vengo incontro) che la tua sia soltanto una grande metafora e il Grande Bug che dici di cercare fuori fra gli hard disk a penzoloni e i drive che girano impazziti fra ventole che sbuffano come comugnoli di mondi sommersi ALTRO non fosse che la metafora di un uomo che usa tutte queste cose come mezzo per immaginare la propria ricerca interiore. Ammettiamo che fosse un mandala, una rappresentazione di un percorso che ti vede impegnato di fronte a noi, tuoi lettori, nello sforzo di stanare il tuo Bug Interiore, il tuo Demone Interno, il diabolico, la parte marcia e puzzolente che ti rende così umanamente schifoso, antipatico e repellente (la parte delle parolacce comunque è sempre quella più divertente a scriversi).

[275] Lievemente editata per renderla fruibile anche ai deboli di cuore. Il Von Taurus, in quanto toscano, ci va giù abbastanza piatto con il turpiloquio e gli epiteti poco lusinghieri.

Orbene, Aaron – con questo bel nome tanto promettente - vuole forse illuderci che questo libro fosse una rappresentazione del suo percorso interiore?

STESSO IDENTICO GRANDE MALINTESO!

non posso ancora una volta essere d'accordo con la tua rappresentazione:

Ammettiamo che il PC - la macchina - sia il nostro corpo ed il sistema operativo la nostra struttura psicologica buggata da mille condizionamenti...

Ammettiamo la condizione di perfezione di cui vai blaterando ove esista un hardware incorruttibilmente perfetto amministrato da un software angelicamente installato che ci guida per mano in questa paradisiaca visione di evoluzione informatico-interiore, sorta di primevo pacifico incontro fra un SuperUser ed il Grande Bug...

bene, questa visione dell'inconscio collettivo della intera umanità che si connette attraverso internet in una rete neurale dove ogni utente sinapticamente scambia informazioni come in un grande ed unico cervello...

Non mi convinci. Non mi hai affatto detto e quindi non mi hai convinto che il male non fosse male (il Bug), che il brutto non fosse brutto bensì il tramite per vedere e capire il bello, che la disfunzione facesse parte della funzione, che lo sbaglio fosse utile tanto quanto la grande intuizione. Quantomeno non potresti arrivare alla perfezione del SuperUser, utente radice, senza incontrare mille ed una volta il Grande Bug nella tua vita[276].

No, tutte queste cose non me le hai raccontate ed hai continuato a raccontarmela piatta piatta con questa storiella dell'errore di programmazione che esiste e va trovato, sradicato e buttato nel cesso... ma concorri forse alle prossime presidenziali USA? Sei forse diventato CattoMicrosoft oppure LinuTeravada?

Io non ce l'ho davvero con gli AppoTalebani ma basta che non mi rompano le palle e si va tutti d'accordo, credimi. Pure voi che volete convincermi che il Bug prima o poi me lo trovate e lo schiacciate come una pulce vi dico che sarà bene impariate a conviverci, come con i vostri punti neri o le turbe erotiche che pigliano nel sonno e che non si posson controllare.

Il Grande Bug sa quello che fa tanto quanto il SuperUser quando ci brucia il disco rigido. Installare una nuova versione non vi salverà più di tanto.

Siccome però non vivo di stronzate adesso vado a lavorare davvero!

Saluti e baci a tutti e calci nelle palle a te Aaron che devi a questo punto darmi OPPORTUNA RISPOSTA.

---

[276] Ed ecco cosa dicono Watzlawick, Weakland e Fisch nel già citato C*hange*: “[...] ogni percezione e ogni pensiero hanno un valore relativo dal momento che operano per confronto e contrasto”; ancora, citando Whorf: “in un universo in cui tutto è azzurro non è possibile elaborare il concetto di azzurrità perché mancano colori che facciano contrasto”.

Paradossalmente, l'Opportuna Risposta al nostro tanto caustico quanto geniale amico toscano l'ha data il Grande Bug in persona, bruciando un disco rigido e gonfiandogli i condensatori[277] della scheda madre pochi giorni dopo l'invio della missiva di cui sopra. Niente di grave, poiché alla fine anche Von Taurus è un grande combattente, ma siamo sicuri che avrebbe preferito occupare il suo tempo in maniera più divertente[278].

Il punto di vista di Von Taurus è molto interessante. Ad esempio, c'è che afferma che gli autori di una grande quantità di virus informatici siano le stesse case che producono gli antivirus, ma questa è una asserzione ben difficile da provare. Invece è sicuro che il linguaggio del marketing legato al mercato informatico si è evoluto in maniera particolare e geniale: ai meeting di una grande azienda il cui business è basato sulla consulenza si sente dire che i bug non sono scocciature, ma *opportunità*, mentre a volte si scopre che quelli che ad un essere umano dotato del minimo raziocinio sembrerebbero malfunzionamenti sono *features* e il loro maligno comportamento è tale *by design*. Tra le righe, in questo libro viene espresso nella Nota 172 XXX un concetto simile a quello del Von Taurus; certo è che, per i produttori di software, il vero business non è produrre un programma e venderlo una volta per tutte, ma instaurare un regime di manutenzione e assistenza con i propri clienti. Commercialmente parlando, più un programma è bacato, meglio è – rimanendo ovviamente in un limite di minima decenza ed usabilità. Si noti che un simile ragionamento potrbbe valere anche per le società il cui modello commerciale è basato sull'Open Source, che però ci piace pensare come più "oneste". O siamo solo degli inguaribili hacker romantici? Cosa c'è di meglio di un software bacato per vendere consulenza sullo stesso?

Di fatto gli aggiornamenti software non costano nulla nella loro distribuzione, in particolare ora che esiste Internet e che aprire un sito web è diventata una operazione banale; un po' diverso è il discorso per l'hardware, che se si guasta quando è in garanzia deve essere rimpiazzato a spese del produttore e in caso di malfunzionamenti lascia comunque un brutto ricordo, commenti infuocati nei relativi *forum* e uno o più clienti che alla prima occasione cambieranno marca.

Insomma, è necessario rendersi conto di come intorno all'esistenza stessa del Nemico ruotino innegabilmente interessi enormi. Veri e propri imperi sono stati costruiti sul concetto stesso di "malfunzionamento" e quindi sull'indispensabilità di chi è in grado di metterci una pezza. D'altra parte, per citare Woody Allen, chiunque abbia avuto la necessita di trovare un idraulico la domenica pomeriggio sa di cosa stiamo parlando, mentre portiamo nel cuore il ricordo della Macchina Nutrimatica che, per creare un the come si deve, prosciuga tutte le risorse di calcolo dell'astronave più avanzata di tutto l'Universo[279].

[277] Rarissimo caso di Orchite Elettronica.
[278] Ad esempio, osservando l'erba crescere: "Sitting quietly, doing nothing, Spring comes, and the grass grows by itself." Zenrin Kushû (The Way of Zen 134, 222), http://www.sacred-texts.com/bud/zen/sayings.htm

## Conclusioni

*Perchè i computer non funzionano?*
*Io lo so, ma non ho abbastanza spazio, qui, per dimostrarlo.*
*Pierre de Fermat*

La strada verso l'Illuminazione è lunga ed irta di pericoli. Noi, con questo libro, speriamo di aver contribuito a renderla almeno un po' più divertente e, se non altro, di aver seminato l'idea che un Mondo Migliore *è* possibile: basta controllare che la spina sia inserita correttamente e scegliere gli utensili software migliori per il proprio lavoro. Di questi ultimi ce ne sono davvero vagoni, là fuori, gratuiti e spesso di qualità superiore rispetto ai loro cugini commerciali. L'Esperienza però ce la dobbiamo mettere noi, insieme al caro vecchio Olio di Gomito e a un po' di nuovissimo Olio di Cervello, magari Elettronico.

Buon Lavoro,
Aaron Brancotti (aaron@icona.it)

*[Disegno: il computer domato, il Ninja che lo domina – classica immagine tipo cacciatore, con il piede sulla testa della belva ]*

## Ringraziamenti

Nella lunga gestazione di questo libro numerosi amici si sono sottoposti all'ingrato lavoro di lettura e revisione dello scritto, mi hanno inondato di feedback e/o mi hanno fornito più o meno consciamente molti spunti. Devo quindi ringraziare Federico Pasquini, Sara Montrasio, Luca Vanzella, Stefano Arena, Mafe De Baggis, Paolo Attivissimo, Cristina Cassanmagnago, XXX mettere in ordine alfabetico e completare

Un particolare ringraziamento a Babele Dunnit, al Riglio, a Paul S.Boreele, al Reverendo Marcus Pampamoon, a Remote, a F.Cortez, a F.Von Taurus e a tutti gli stimati professionisti che, nascosti dietro pseudonimi atti a proteggerli dai meccanismi perversi di un mercato delirante, hanno permesso all'autore di citarne pensieri ed elucubrazioni senza temere di perdere clienti.

Quando ho chiesto a Rob di debuggare questo libro mi ha risposto “ma cosa devo farne? Se debuggo il Libro dei Bug, poi cosa ne resta?”

[279]Douglas Adams, sempre lui. E chi se no? D'altra parte se state a spiegare ad una Intelligenza Artificiale come deve essere fatto un the a regola d'arte, descrivendo le mucche che nel Devonshire mangiano l'erba che poi permette loro di produrre quel latte di qualità superba che, mescolato al vostro the – una miscela esclusiva selezionata a mano da generazioni di esperti indiani – lo renderà sopraffino, dovete anche accettarne le conseguenze – compresa quella di rendere inutilizzabile la Propulsione ad Improbabilità Infinita della Cuore d'Oro mentre i Vogon iniziano a bombardarvi.

La copertina e le strepitose illustrazioni sono di Joe B.C. Gonzoy Gonzalez e non credo ci sia nulla da aggiungere, se non che quello di Joe invece *non è* uno pseudonimo.

La supervisione grafica, nonchè il layout dei numerosi gadget da ritagliare e costruire, sono di Luisella Brustolon.