

前言

多年來，我將任何跟人工智慧或機器學習有關的工作，都稱之為由機器人代勞完成的工作。因此，假設我問道，「這份清單是由人類還是機器人產生？」我的意思是問，這份清單是否以機器學習產生。之所以詢問「這份清單是由人類還是機器人產生？」，是因為我企圖從當今複雜的生態系統裡，萃取出其中最基本的元素。

我喜歡以標籤分類事物，因為這讓我更容易理解這個世界。

當年我推出第一本著作時，選擇的書名是《*Managing Humans*》，由於出版的時間較為久遠，雖然不若本書有抓到 2020 年代中期興起的 AI 熱潮，卻足以作為我偏好找出描述性標籤來分類事物的另一個例子。我認為這種做法非常方便，只要用少少的文字，就能表達出更多的想法。

本書最初出版時的書名是《*Being Geek* 晉身怪傑：軟體開發者職涯應變手冊》，但讀者應該有注意到現在已經不再採用這個書名。這是因為撰寫初版內容的時空背景，在 2010 年有許多熱門話題圍繞著電腦怪咖（nerd）和科技宅（geek）打轉，我也在許多公開演講上侃侃而談電腦怪咖和科技宅這兩類人的心態。從自己原本的專業立基——領導力改談其他面向，這種想法上的轉變，在當時確實令我感到興奮。

然而，我在撰寫初版內容的過程之中，定好的書名卻讓我越寫越無力。原因在於我撰寫的內容焦點主要是擺在領導力上，但我每次寫下一個新章節時，卻又硬要以某種方式將「科技宅」這個詞融入書中內容。全書完成之際，我已經完全回歸到自己的專業立基——領導力，於是，我發現自己其實是在寫另一本以領導力為主題的書。

即使書已經出版了那麼久，這種不協調的感覺還是一直卡在我心裡。相較於第一本著作《*Managing Humans*》，我對本書前一版的印象就是失敗之作，但事實並非如此，這本書的初版跟我的第一本著作一樣，至今仍然在市面上銷售，而且，我一直都有收到相關詢問。即便不是暢銷書，但確實有其底蘊，實為長銷之作。

就在我為第一本著作完成第四次改版時，「Rands Leadership Slack」社群裡的 Philip Sharp 傳來他的意見：「《*Being Geek* 晉身怪傑》要不要出第二版？」，我隨手抓起前一版，快速瀏覽了一下內容，於是，改版的念頭在我腦海中成形。哇，確實有一些寫得不錯的內容。

醞釀了幾個星期後，我向 O'Reilly 出版社提出簡報，說明改版的想法，並且和出版社一起討論我對初版內容的擔憂，結果，我們發現初版書名的副標：軟體開發者職涯應變手冊，正好適合作為新書的名稱。書名雖然唸起來有點長，卻更強而有力，更適合描述本書內容。

本書除了加入八個全新章節，另外也大幅刪去初版內容裡使用的詞彙「科技宅」。這遠比我預期中的容易，由於我原本主要就是想寫領導力方面的文章，所以只要將內容中用到的單字「科技宅」換成「工程師」，在某些情況下或是換成「領導者」即可，內容上不需要多做任何其他修改。本書編修過程中，還刪除了幾篇內容過時的章節，對於技術逐漸老化的章節，則更新至符合現代需求的內容。

本書最後仍舊保留先前的設計，帶讀者了解當前這份工作從開始到離職會面臨的各種考量，以及如何前往下一份工作。中間各個部分的章節內容依舊涵蓋許多有用的建議，協助讀者了解人性、領導者、文化和溝通。若仔細觀察我的職涯經歷，會注意到書中和領導力相關的章節，內容主軸都是在談初次擔任領導者的心態。這是因為本書前一版完成的時間點是在我嘗試擔任管理職之前，重新修訂本書內容時，我決定保留初衷。在快速成長的科技業裡，我認為這些來自初次擔任領導者的觀點，佔有重要的一席之地。

最後要說明的一點是，貫穿本書內容的敘事者 Rands，是我已經用了數十年的化名。利用 Rands 這個化名，我能自在地侃侃而談，從專業的距離來看我要介紹的主題。同樣地，出現在書中的人名以及描述的情況，幾乎每個都是虛構，一切都是為了我們精心設計的故事而刻意創造出來的。

那麼，我要開始說故事囉。

職場毒瘤悖論

每個人都擁有適應協調能力。跟任何人互動時，我們多少都會偏離自己的核心，變得跟對方有點像。這不代表我們背叛自己，而是在任意兩人之間定義中間地帶，設定一個妥協點，彼此才能相互溝通。

有些人很容易、也很自然就能到達這個境界，即使某些朋友跟你六個月沒見或毫無音訊，但你們雙方都知道彼此只需要 12 秒就能回到熟悉的境界，六個月的空白時光瞬間消弭。

有些人際關係則需要我們用心經營，這類的人需要一套規則，先設定雙方背景環境、轉換彼此對話以及謹慎聯繫。「嗨，我現在說的這些話跟你聽到的內容一樣嗎？你能理解我的話，太好了。」這套能力和溝通技巧要付出更多心力，需磨練多年才能完善技能。這也是經驗豐富的主管必須具備的技能之一，因為他們經常被丟到各種會議中跟陌生人開會。在這些會議上，他們必須快速且有效率地行動，盡快從「很高興認識你」這個階段，轉換到下一個階段「我們得開始一起工作了」。

我猜多數人際關係大多會落在這兩種類型：天生有緣型或用心經營型。我們和身邊周遭人們的互動關係，不論職場內外，多數都能加以管理。雖然我們無法跟所有人形成自然互動的關係，但只要我們願意投入心力去維繫，依舊可以跟這些人一起生活。

然而，我們還是會遇到一些無法應付的人。出於某些原因，我們可能永遠無法理解這些人為何會表現出這些行為，甚至永遠無法掌握、定義和喜歡他們這樣的行為。

因為，這些人本身就是毒瘤。

兩大毒瘤假設

本章接下來的內容要從兩大令人不安的假設前提談起。

在解釋毒瘤同事引發的嚴重問題之前，我必須先提醒讀者一點，當某個情況涉及兩人之間的隔閡，究竟誰才是毒瘤，總是會有兩個截然不同的故事版本。如果你聽到我說 Felix 這個人具有毒瘤性格，那你應該多花一些時間跟 Felix 相處，了解他對我的看法。雖然我可能已經謹慎地從各個角度對 Felix 的性格做了全面的調查，但你還是必須直接從他本人那裡蒐集資料。

宣稱某個人是毒瘤，這是一種個人的主觀判斷。有時來自一群人的集體定義，有時只是因為某個具有影響力的人無法跟某個人建立健康的人際關係，就將矛頭指向對方。不論情況如何，各位一定要自己親自調查，永遠不能相信他人貼上的毒瘤標籤。

再來，本章內容不是教你怎麼修理毒瘤人，而是談你對他們的看法，所以本章會先假設你已經做了各項嘗試，努力彌補自己與對方之間的鴻溝。讀者如果目前位居管理職，希望你是經過數月的謹慎協商、精密權衡和苦心溝通，最後才得到這個結果。

本章還會假設各位讀者工作的組織中，有一個完整的部門會針對如何改善團隊成員之間的互動關係，而且你也採用了他們所提供各種想法和技巧。

由於你已經越過這個階段，本書就不再提供處理毒瘤人的策略。雖然你知道這個毒瘤人正帶給團隊不好的影響，但你嚴重低估他們每天毒害團隊的程度。

因此，本章的目的是說服你，讓你相信該做出改變了。

團隊合作！

毒瘤人會扼殺團隊，我的意思是他們會將團隊之間的合作關係破壞殆盡。

團隊合作是令人痛苦的管理術語，如果在不適當的時機提出來作為激勵手段，就會變成笨拙的管理方式。「我們需要更好的團隊合作關係，以提升工作效率和生產力。」我知道你想表達的意思，但應該還有更好的說法。團隊合作是真的很神奇，透過這樣的關係，讓不同的團隊、不同的人實際一起工作。

事實上，團隊合作不是一種魔法，是我們透過多年練習而成。從小學開始，我們學習基本的團體生活規則：想說話時要舉手、對他人說「請」和「謝謝」，還有不要吃膠水。學校就是一個小型的現實世界，讓我們學習如何應付不同類型的個性，所以進入職場時，我們早就累積多年的社交經驗，知道如何與各種性格的人互動。

然而，當我們面對毒瘤性格時，所有這些預先累積的輕鬆練習，不管是舉手還是不要吃膠水，幾乎都派不上用場。

回到本章一開始描述的人格特質類型：天生有緣型、用心經營型和毒瘤型。假設天生有緣型的成本是 $1x$ ，因為經營這種類型的人際關係不需要付出心力，也最簡單，所以本書將其當作經營人際關係成本的基本單位。用心經營型的成本是 $2x$ ，因為我們需要加倍努力才能彌補雙方在溝通和社會經驗方面的落差。經營這類的人際關係不難，只是需要投注心力。當我們累積更多經驗，對人們認識更深，就能降低經營成本，可是不管我們怎麼努力，這類人際關係永遠不可能跟天生有緣型完全一樣，現實生活就是如此。

毒瘤型則無法衡量經營成本，基於這類關係的本質，不管我們怎麼投入心力都無法經營。在這些人際關係中，我們永遠無法進入安心溝通的狀態。因為我們會一直處於社會常識不斷淪喪的情境，永遠無法預測這些人際關係的發展，當然也就無法帶來多高的生產力。某些時刻我們會有短暫的領悟，突然靈光一閃：她那個人之所以會有這樣的表現，是因為我說了那樣的話，原來這就是她為什麼老是會出現這種反應的原因……所以我不會再那樣做了，能找到原因，真是太好了！

然而，喘息的片刻非常短暫，出於某些永遠無法理解的原因，我們無法逆向分析毒瘤型人格特質或是我們對這類性格的反應模式，而且遲早會再次發現雙方之間基本上就存在著脫節的情況，於是，我們又回到先前的窘境，拼命掙扎著想理解這種難以捉摸的性格。

沒錯，這就是面對職場毒瘤時最糟糕的情況。

當我們看到一群人相處融洽，多半是因為大家都認同相似的文化。這些人當然各有其獨特之處，卻還是能處得來，全然是因為他們擁有類似的信仰體系，以及追求相同的目標。相似的信仰能帶來許多好處，但最大的好處是減少組織成員之間的摩擦。雖然成員之間還是會產生激烈的爭論，但是這些爭論是基於相似的信仰，只要有這些信仰存在，就表示有機會解決這些爭論。

現在請各位讀者想想自己平常面對毒瘤人時，會有哪些基本互動。假設你此刻身處在會議室裡，和毒瘤人面對面坐著，準備討論一個簡單的議題：目前我們正在討論，要對架構進行微幅的異動，由於你負責開發大部分的架構內容，所以我想聽聽你的意見。

你認為自己的表現合理、專業而且尊重對方。

你提的異動一點也不小，根本沒有深思熟慮，為什麼不早點跟我商量？早在 14 個月前，我就針對這個議題提出意見，當時大家都無視於我，事到如今你們卻說要認真考慮……？

一連串難以理解的毒言毒語，排山倒海而來。連珠砲似的話語正在數落每個人過去犯下的一串錯誤，可是，請回頭去看前一段的內容。這些內容是言之有物？或只是毒言毒語？

各位讀者好好想想吧，在這場會議裡，當大家都冷靜下來，你可能是最後的贏家，但會議前 30 分鐘你得先承受一連串的言語砲轟。這裡我要說個壞消息：團隊裡多數人都有過類似的經驗，跟毒瘤人互動過程中，多數人會窮盡心力思考怎麼避免讓毒瘤人發飆，而非實際上要怎麼完成工作。

這種情況持續一段時間後，甚至會產生更多有害的情況。大家在安排會議時，會避免跟這個人一起開會，不再前往這個人工作所在的辦公大樓。我要再次強調，這裡說的不只是一、兩個人會這麼做，而是指團隊中多數人都會出現這樣的行為。

我在定義毒瘤人時，我的想法不是基於能否跟這個人在職場上建立良好關係，反而是看一個團隊和公司文化實際上是否拒絕接納這個人。當每個人都在盡力迴避某個會造成危害的人，這種迴避性會持續從各個面向去影響團隊的生產力和士氣，進而變成每個人每天都要承擔這樣的情緒：挫折和士氣低落。

一個已然形成的文化幾乎不會因為某個人而改變，所以當毒瘤人出現時，文化會以拒絕接納對方的做法來保護自身。

職場毒瘤悖論

Rands，毒瘤人只是沒有遇到和他處得來的人。

話不能這麼說。

我們身處的環境是職場，不是高中校園，所以我談的不是學生時代的小圈子，我談的是職場文化。小圈子很難避免，通常是一小群喜歡彼此外貌和聲音的人聚集在一起；文化則是由一大群人共同形成一個大的基礎架構，其中包括信仰、價值觀和目標。一個健全的文化具有包容性，會尋求新成員加入，推動自身進化成某個更好的新文化；以有趣的方式不斷發展，因為文化建立的根本就是人。

發現自己身處於不被接納的文化之中，任何人都不會感到高興，但這種拒絕接納的情況並不是基於個人喜好，也不是因為某個人不喜歡另一個人所造成的，純然是因為雙方缺乏共同的核心信仰。當一群人有共同的目標時，即使個性截然不同也能相處融洽。

沒錯，人就是會變得小心眼，似乎會因為一些無聊的理由就討厭對方，所以主管和人資的工作就是要想方設法，怎麼在這些人之間建構具有建設性的工作關係，但這不是我現在要說明的情況。我要說的是，當你試著想把一個方形的毒栓子硬塞到一個圓形的文化孔裡，這根本行不通。你當然可以使出洪荒之力，持續推進，但……還是不會發生你想要的結果。

當毒瘤人對著你大吼大叫時，你很難想起這一點，他們真正想吼叫的對象其實不是你，是這個拒絕接納他們的文化。他們生氣的原因，是因為他們心裡清楚知道自己的信仰結構跟其他人不一致，他們自知……永遠無法勝過這個論點。同時，他們也知道，為了勝過這項論點，就必須重新塑造公司文化，相較於這方面需要投入的艱鉅努力，公司重組似乎簡單多了。

最糟的一點是：毒瘤人的論點可能是對的。

矽谷發展史上充滿著毒瘤人的故事……最終也證明他們當時的想法是對的。在當時的時空背景下，現有的文化制度難以接納這些人提出的議題和想法，導致他們被各自的公司革職，但對那些公司來說，其實這些才是他們應該採取的正確策略。

悖論在於，公司需要毒瘤人的存在。這些以自我為中心的混蛋完全漠視文化常規，把事情弄得一團亂，搞得面目全非。這些人不需要社交禮儀，也不需要個人魅力，即使兩者對他們都有助益，但他們強烈相信自我價值在於自身信奉的文化。

雖然公司和團隊需要這些傢伙，但我們不能以犧牲團隊士氣和生產力作為代價。確實，毒瘤人自恃的信仰或許與公司的核心文化相違背，卻是公司未來發展的關鍵，我們可以採用，但要謹慎評估風險。如果我們將毒瘤人的想法整合進來，代價卻是團隊裡有一半的人因為無法與他們共事而揚言退出，這樣也能算是一個可行的解決方案嗎？

不是？或者是？

是否要將毒瘤資產驅逐出境，做出判斷的基本想法是：對現況妥協很容易，要實際做出改變卻很困難。

每個人都擁有適應協調能力

「我們在工作上解決了哪些困難的問題？解決這些問題時，是誰和我們一起並肩作戰？」這兩個任務能幫助我們定義職業生涯。我認為遇到越困難的問題、遇見越棘手的人，越有助於發展職涯。因為我相信每一個不可能的挑戰之中，都存在著寶貴的經驗。

當我們確信某個人的本質真的是毒瘤人，他們的存在對職場不只是一種干擾，更是一場災難。衡量這一類人對團隊造成的損失和精神痛苦程度時，我們不該問自己「接下來要做什麼？」，而是要問「我能多快解決這個情況？」

以鍾情原則找到 令人狂推的工具

我有一位姊妹的老公住在郊區，他之前需要砍掉五棵樹，每棵不到 10 到 15 英尺高，6 英寸粗的樹幹，他覺得根本不是問題。

我則是住在北加州的紅衫樹林邊緣，那裡生長了結實的橡樹、活潑的楓樹、可愛的脫皮樹、長得像雜草般的月桂樹，當然還有巨大的紅衫樹。住在森林雖然愉快，但也要付出代價。任何規模大小明顯的森林，永遠都會發生樹木倒下和死亡的情況。

你如果住在森林，就需要電鋸。以我個人為例，我需要三把。電鋸一號是 Junior（如圖 29-1 所示），非常擅長處理小型工作，很輕，適合站在梯子上使用。



圖 29-1 電鋸一號：Junior

再來是電鋸二號：Marty（如圖 29-2 所示），這款電鋸大小適中，日常生活中的任何大小工作幾乎都能處理，非常適合居住在郊區的人使用。



圖 29-2 電鋸二號：Marty

最後一把是電鋸三號：Rocket（如圖 29-3 所示），任何一棵樹都是它的對手。



圖 29-3 電鋸三號：Rocket

各位就算沒用過電鋸，或許也用過手鋸，這是消耗體力又折磨人的苦力，大概只有前三分鐘會覺得有趣，接下來你會開始懷疑人生：我真的有進度嗎？我姊妹的老公興沖沖拿了手鋸，自己鋸了其中一棵樹。三分鐘後，他只鋸了……一根樹枝。

當我帶著電鋸二號：Marty 出現後，我們在一小時內鋸掉了全部的五棵樹，砍掉樹幹和樹枝，弄成一堆準備丟棄。

從這次的經驗，我學到一個教訓：使用正確的工具，生產力會有如指數般激升。

本章用了很長的引言來說明一個淺顯的道理，不過，我要再延伸這個引言，花點時間，帶各位一起走進我這位姊妹的老公的內心世界。我想砍掉幾棵樹……來看看我的車庫裡有什麼？兩隻槌子、一整罐油漆桶的釘子、一些剩下的木頭，還有……一把鋸子。太完美了，我有一把鋸子。

一個人的背景環境會塑造他的觀點，根據車庫的物品組合，他知道車庫宇宙裡沒有電鋸這個工具存在。雖然他聽過電鋸，也猜想使用電鋸會比這把費力、令人汗流浹背又折磨人的鋸子來得更快，但車庫裡沒有電鋸，他只好憂喜參半地用這把鋸子慢慢鋸。對我這種擁有一批電鋸的人來說，看著他如此浪費時間的糟糕行為，簡直是荒謬可笑。

我要再說一次，這次的經驗讓我學到一個教訓：使用正確的工具，會讓生產力有如指數般激升。

鍾情原則

身為工程師的你，必定能列出少數幾個會讓你狂推的工具，沒錯，你為這些工具瘋狂，而且是愛不釋手。

這份簡表上都是一些常見而且你以前就聽過的工具，重點在於你需要關注這些常用的工具，具備對這些工具知之甚詳的能力，例如，撰寫程式碼時，為什麼採用黑底綠字才是正確的方式。各位要成為工具的狂熱使用者，而狂熱始於工具適性。

我曾經是資料庫開發人員，後來轉職到瀏覽器領域，然後成為網路應用程式開發人員。這幾個專業領域各自搭配一套光鮮亮麗的工具，但這些工具是什麼並不重要，就連工具內的某某具體功能也沒那麼有趣。我認為各位處於像 Xcode 這樣功能豐富的開發環境，跟你使用 Visual Studio 搭配一堆視窗，兩者的生產力應該差不多。重點不在於你使用什麼工具，而是工具的使用介面、體驗和功能，是否能配合你的視覺、操作習慣和工作方式。

這些是令我鍾情於一項工具的原則，可能跟各位的清單差異很大。這樣很好，我的開發經驗跟各位不同，我開始使用電腦的時候，滑鼠還沒問世，也就是說我的使用習慣更依賴鍵盤。我開始從事開發工作的時候，搭配整合

開發環境的偵錯器才剛推出，所以我偏好用命令列偵錯。再跟各位強調一次，重點在於對自己使用的工具愛不釋手，因為情有獨鍾會激發出你的最佳潛力。

以下是我對工具的鍾情原則。

使用看似簡單的工具

Visual Studio (<https://visualstudio.microsoft.com>)、MacOS 終端機、Transmit (<http://www.panic.com/transmit>)、LaunchBar (<https://oreil.ly/Ivq36>) 和 Dropbox (<https://www.dropbox.com>)，這些工具的平均設定時間只要幾分鐘，所以半小時內，我就能在新機器上搞定開發環境。這段話背後暗示了幾個便利性。我使用的工具都很容易取得，而且輕量；作業系統以外的所有工具都能在短時間內下載，並且完成安裝；設定和配置這些工具同樣不需要花什麼時間。

各位或許以為我期待自己的電腦隨時會爆掉，怎麼可能。這些工具其實一點也不簡單，都經過完善校調。Visual Studio 的使用者很清楚這套應用程式就像洋蔥一樣，一層層剝開就能發現新功能，加速開發體驗。MacOS 終端機和 LaunchBar 也是一樣，不只提供基本功能的運作需求，若使用者渴望開發特定需求，這些工具也能滿足。

工具不會影響我的工作環境

各位曾經歷過幾次這樣的情況嗎？在本機上快速寫好一段程式腳本，做了某個聰明的功能，精心調整程式腳本，然後更新到伺服器上，卻再次發現這項規則：本機環境與實際營運環境完全不同。

任何不能讓我在實際營運環境下直接進行開發的工具，都會拖慢我的工作進度。因此，當有人展示給我看，如何設定 Transmit 就可以編輯遠端機器上的檔案，我頓時豁然開朗，以前要花大量時間在營運環境下解決各種疑難雜症的問題都消失了。

當然啦，在本機上編輯檔案確實很快，尤其是當你住在紅衫樹林邊緣，網路延遲明顯是個問題，但是不能讓我透過網路開發程式的工具，對我來說根本稱不上工具，只是拖累我的阻礙。

Rands，你剛說要直接編輯檔案？在營運環境下？你是瘋了嗎？

不，我沒瘋。此處要提到「背景不相干」原則，意思是：你如果不清楚營運環境下要做哪些事，就不應該待在那個環境裡。

從這個原則可以推論出一個必然的結果：不論我在哪個環境下工作都沒有影響。以不需要實際營運環境的工作為例，例如寫書，我就不會費心去想最新的工作版本存放在哪裡。沒錯，我說的就是版本控制，但是，小聲點（噓），我們在這裡不講版本控制，就直接說是「Dropbox」。我只要有提供網路連線的環境下工作，這套工具就會神奇地更新每一台機器上設定的共享目錄。我都想不起來上次擔心自己正在使用哪個版本的文件，是哪時候的事了；也就是說，使用這套工具後，我投入更多時間在工作上，不用再擔心版本控制的問題。

工具的設計宗旨是移除重複的操作

當年上電腦科學這門課的第二堂課，我們正在學排序演算法，那是我第一次想要「讚爆」一個演算法。那堂課的教授先優雅地帶我們完整看過一遍各種不同演算法的建立方式，詳細解釋優缺點，然後才教了「快速排序法」（Quicksort）¹，我當下的心情是，哇靠！還有這招！

快速排序法讓我想要讚爆，不只是因為優雅和簡單遞迴，更因為它讓我發現，利用個人想像力可以找到效率極高的方法而且更簡單。不管各位是不是受過正式訓練的電腦科學高手，都深知效率的價值，也就是讓你採取的每個行動都具有某種意義。各位應該很清楚，當你更有效率，就會有更多時間去做你熱愛的事。

因此，任何讓我依賴的工具，都必須滿足一個簡單的條件：完整支援鍵盤。我會借助滑鼠操作只進行一次的行動，但如果我知道有任何行動會重複執行，就會自問：如何降低這個操作的成本？

請各位試著思考這個情況。如果我告訴你每次要儲存檔案時，你必須先站起來，爬到椅子上，然後站在椅子上上下跳躍，並且大叫：「我現在要儲存檔案內容！」第一次必須這麼做時，你會覺得還蠻有趣的，但在那之後，每次都這麼做會把你徹底搞瘋。每次我伸手拿滑鼠，就有類似這種感覺。我覺得自

1 讀者若想了解「快速排序法」（Quicksort）的相關資訊，請前往這個演算法在維基百科上的資料頁面：<http://en.wikipedia.org/wiki/Quicksort>。

已沒有必要去做一定會浪費時間的工作，因為使用滑鼠有時會發生失誤，不斷地失誤會大幅浪費時間。

在理想情況下，四次按鍵就可以找到任何檔案或應用程式。Command 加空白鍵（啟動 LaunchBar）、輸入第一個字母、第二個字母，再按下「Return」鍵。幸運一點的話，有時只要三次就能找到，每次發生這種幸運時刻，微笑就會掛在我的臉上。

工具只會做我交代的事

時間快轉回到 Adobe 剛推出 Dreamweaver (<https://oreil.ly/88vBX>) 的時候，我曾經想愛上這套工具。那時我正好對開發 HTML 網頁時不斷重複的操作感到厭倦，正好有工具以視覺化方式處理這項費力的流程，這樣的想法很吸引我。問題是……Dreamweaver 會在未經詢問的情況下，直接改掉我的程式碼。

蛤？

我知道 Dreamweaver 嘗試想要提供協助，但是當我發現程式碼被重新格式化的當下，不禁勃然大怒。Dreamweaver 動了我的程式碼！至今我還無法擺脫那次糟糕的第一印象，到現在還是很氣。

我的印象和看法是，強大的整合開發環境要能大幅協助使用者想像目前究竟是發生什麼情況。Borland 過去曾針對程式開發，推出一些相當出色的整合環境，但我發現自己依舊使用極為原始的開發環境：在 Visual Studio 修改程式碼，以及在多個終端視窗內進行偵錯。

我很清楚整合偵錯環境所有厲害之處，但我從過去多年的開發經驗中發現，擁抱酷炫的工具表示你要額外投入時間校調工具，才能讓工具依照你想要的方式運作。

根據這項原則可以推論出一個必然的結果：我使用的工具不會搭配大量的移動元件。像 Dreamweaver 這種等級的工具因為有搭載可靠的開發工具，很少會發生程式碼攻擊的情況，但酷炫的工具會伴隨使用成本。各位或許十分樂意而且熱衷於付出這樣的代價，但我不是很願意投入這方面的成本。

我比各位有效率嗎？或許。我清楚自己與工具之間的定位關係嗎？是的。當那些開發優雅工具的人追求更優雅的境界時，我會重新學習開發流程嗎？這點就免了。

個人化工具

選擇某個事物，讓它變成你的東西。所謂選擇，是個人將適合自己的邏輯、迷信、固執和體驗，綜合而成的獨特結果。

各位沒有看錯，我認為撰寫程式碼的環境要採用黑底綠字，現在就來說明箇中原因。我是那種喜歡用 DOS 這種老派工具的人，當年我第一次使用的文書處理工具是 WordStar，這個文書處理程式帶我進入類似神馳的狀態，發揮極限生產力：絕對專注領域。這就是為何我在目前偏好使用的 Visual Studio 編輯器上，依舊愛用黑底搭配彩色文字的原因。這種色調會讓我聯想到原先使用的可靠環境，讓工具在這個環境下發揮作用，更能幫助我擺脫眼前的困境，激發出更高的生產力。

這就是為什麼身為工程師的我們，會堅持使用適合自己的東西。也是一些上古神獸級工具，像是 vi 和 eMacs 至今依舊蓬勃發展的原因。工程師會從眾多工具中找出適合自己的工具，一旦選擇之後就會持續使用，進而對這項工具愛不釋手。

不斷進化的工具令人愛不釋手

我這位姊妹的老公其實不需要電鋸。我幫他砍掉五棵樹之後，就已經消滅了他原本擁有的一半樹木。電鋸雖然是聲音、力量和鋸下來的屑末的美妙組合，但我這位姊妹的老公選擇居住的房子周遭，沒有那種具有攻擊性的高大樹木，所以他不需要電鋸這種防禦性武器。

他確實需要知道這個宇宙間存在電鋸這種工具，因為時間一分一秒都很珍貴。然而，我們與猴子之間的區別，並非我們能針對適合自己的工作挑選適當的工具，而是挑選出最佳工具。

我們永遠不會停止尋找新工具，這是最後一項鍾情原則如此重要的原因：會讓我狂推的工具，一定也會為自身的生存而戰鬥。