

3 績效評估

對於許多工作於科技巨頭或新創公司的軟體工程師來說，績效評估是一項至關重要，有時讓人緊張的任務。評估結果不僅決定了獎金多寡，還關係著是否能夠順利晉升。這是少數根據量化機制來評定個人表現的時刻。

然而，好消息是，你幾乎總是可以透過付出更多的努力，來獲得更好的績效評估結果！然而，在最後一刻所能採取的措施有限，提前準備非常重要。本章內容涵蓋以下重點：

1. 及早開始：蒐集脈絡並設定目標
2. 習慣的力量
3. 在績效評估之前
4. 績效評估

1. 及早開始：蒐集脈絡並設定目標

每家公司都是不同的。在你的公司中，哪些事情是重要的，哪些事情又是備受重視的呢？如何衡量影響力？關於這些問題，新創公司和科技巨頭可能有著截然不同的回應。

要從何處開始著手？最關鍵的一點是首先研究讓你獲得成功的脈絡。確定哪些目標對你、你的團隊和公司有幫助。

辨認和理解最重要的因素

你的團隊和雇主最在乎的東西是什麼？通常獲得優異績效評估、順利升職的人，他們能夠幫助業務取得成功，或是實現非常宏大的目標。你該如何成為這種人？首先，要理解哪些事情被重視，這意味著確定哪些工作是優先要務：

- **問問你的經理：**這是一個非常明顯的起點。問問團隊的目標是什麼，以及它們如何與公司的目標相關。問問你經理的個人目標，以及對他們來說哪些事情是重要的。你能給的幫助越多，你對你的經理和團隊就越有價值。
- **與團隊中的資深人員交談：**與經驗豐富的隊友交談，了解他們眼中哪些是重要的團隊和公司目標。
- **聽公司領導者的發言：**經營良好的公司在企業和組織層面都有著清晰的目標。找出最重要的優先事項通常很簡單，你只需參加全體會議、重新觀看會議紀錄或簡報影片，或閱讀來自領導層說明公司優先任務的電子郵件。在科技巨頭，你可能需要聆聽幾次全體會議，包括你所在組織、產品組織、母公司，以及技術組織等層級的會議。
- **問具有影響力的人：**與在業務中扮演關鍵角色且易於接觸的同事進行一對一面談。他們可能是你的產品經理、你的直屬主管、業務利益相關者或其他人。
- **了解績效評估系統的運作機制：**有一些常見的績效評估系統，然而，這些系統的具體運作方式也許存在很大差異，而好公司會根據其不斷演進的需求來完善評估機制。

有哪些績效評估系統？

以下是最常見的績效評估系統：

- **非結構化 / 臨時性：**經理會進行績效評估，但不是結構化的機制。偶爾，你的經理可能會在宣布加薪時，順帶給你一些工作上的回饋。這在規模較小的公司很常見，此處績效評估的重要性較低。優點是你不需要太擔心，也不會花費太多時間。缺點是你的評級基本上取決於經理對你的個人看法。
- **僅由經理提供意見和回饋：**這個形式更加結構化。例如，可能會有一份輕量級的職位期望文件。評估本身包括你的經理給予的回饋。在這個評估系統中，重要的是讓你的經理了解你的工作內容。與他們建立良好的關係，你也很可能獲得更好的評價。

- **基於同事回饋的績效評估：**團隊成員有定期的機會對同事提供回饋，而經理會審查這些回饋。他們會根據這個系統給你提供回饋和評級。在這個系統中，你與同事的關係對於獲得良好的評價很有影響。
- **正式、繁重的評估流程：**大部分科技巨頭和一些較成熟的新創公司會採用更為繁重的績效評估流程，以期排除常見偏誤。在典型機制中，你會為同事寫評價，準備一份自我評估，而你的經理則根據這些文件撰寫關於你個人的工作評價。這個流程的優點是你通常會得到具體的回饋，缺點是需要大量的時間和精力。

請蒐集關於你的績效評估的重要細節：

- **誰是做出最終決定的人？**答案幾乎肯定是你的經理，但明確掌握這一點也是值得的。
- **誰對你的績效評估有主要影響？**根據不同的評估機制，對你的績效評估有實質影響的人可能包含同事和利益相關者，或者只有你的經理。
- **績效評估於何時進行？**有哪些關鍵日期？在採行結構化流程的組織中，通常有一個必須在此之前繳交同儕評估和自我評估的截止日期、績效評估審核時程以及通知結果的日子。
- **確保公平的績效評估之建議。**與團隊內外的資深同事聊一聊。他們也許能與你分享一些有助或阻礙公平評估的建議。

與經理討論你的目標

在對背景脈絡有了更深一層理解之後，請據此設定目標。有些目標應該支持團隊和公司的目標，也許還包括你經理的目標。當然，這些目標也應該對你有益。

與你的經理分享你的目標，並尋求回饋。如果你在組織中有一位導師，也與這個人分享一下；了解他們的看法，他們對哪些目標設定感到滿意、有什麼缺失，以及他們有什麼建議？

與你的經理就目標達成共識

讓經理支持你的目標，是讓他們與你同一陣線的最佳方法之一。因此，請持續修改迭代你的目標，直到你的經理願意支持它們，並確保你的目標被記錄下來。

如果你實現了每一個目標，這可能是一個很好的機會，可以詢問你的經理會如何評價你的績效。然而，請注意，一位負責任的經理不會這麼說：「如果你做 X，你就能獲得超出平均的評估結果。」這種說法並不實際，因為績效評估是將你的工作表現與其他同事進行比較，並且通常採用一種校準機制。你的經理無法預測同事的表現，也無法預測校準的結果。

你可以向經理提出的一個較為公允的問題是，你的目標是否有助於達到期望，或者能夠在超越你目前職等的角色期望。你不會得到任何保證，但也許能獲得一些指引。

2. 習慣的力量

績效評估通常每年進行一次或兩次，但你的日常工作對一整年的表現來說都很重要。有一些簡單方法可以確保公平的績效評估，而這些方法只需要付出極少的時間精力，例如：

記錄你的成就

大多數績效評估都受到近期偏誤的影響。當績效評估時刻來臨，你和你的經理最容易記起近期的成功和成果。相反地，幾個月前的一些優秀工作可能就被忽視了。

為了避免這一點，確實記錄下你的成就，好比每週或每兩週。記錄你完成交付的專案，並將表揚你工作成果的電子郵件或聊天訊息截圖下來，以及其他能證明你表現的證據保存下來。

撰寫工作日誌

這是追蹤你的工作進度（不是只純粹紀錄勝利）的輕鬆方式。我觀察到，大多數採用這種方法的人都運用了一份保持持續更新的文件，在其中記錄值得注意的工作、相關頁面連結，並記錄他們工作的影響力。我們在第 2 章〈成為職涯的主宰〉中介紹了工作日誌的優點，並提供了一個參考範例。

與經理分享進度

每隔幾週與你的經理進行定期一對一會議，分享你所做的工作，討論勝利和挑戰，展示你正在高效工作。

作為一名經理，我經常驚訝於我的同事實際處理的工作量，而這些工作是不曾知道的。這一點對於那些攬下許多提案，更加資深的工程師尤其如此。如果他們沒有分享這些細節，我將對此一無所知；這不是因為我不在乎，而是因為我根本不知道。

因此，我建議在分享你的工作時盡可能地多多分享，不要假設你的經理知道一切。一個好辦法是不時分享一下你的工作日誌，並帶著你的經理瀏覽一遍你的所有工作內容。

把事情搞定

如果你被認為是一個只做「忙碌工作」（需要花時間處理但沒有真正價值的工作）或者只是玩政治手段而不是做有意義事情的人，那麼大多數建議都是毫無意義的。交付出色的工作成果非常重要，因為這能夠為你建立良好聲譽，讓人們將你視為一位可以確實「把事情搞定」的人，並幫助你在績效評估中獲得認可。

什麼才能算作「優異」的工作成果呢？其定義取決於你所處的環境。如果可能的話，確保你的工作在品質和速度取得良好平衡。在新創公司中，「優異」可能表示你能夠快速推出產品，而在規模較大的公司中，則可能意味著你的工作產出確實通過測試，程式碼乾淨俐落，或者是容易審查及維護的解決方案。

請找出並熟悉你所在的工程組織中對於出色工作的定義，並定期向同事和經理尋求回饋，以了解你的工作會被如何評價。當你交付優秀工作成果，累積良好信譽，那麼其他人就能更加信任你。

幫助他人

如果你只顧自己，埋頭專注於自己的工作，除非對你的績效評估或升職有幫助否則永遠不去幫助他人，你將無法走得很遠。大多數人的觀察力都足夠敏銳，能夠看出哪個同事只在乎自己的利益，他們可能開始討厭你，甚至可以合理聲稱你是一個不願合作的團隊成員。

根據你的情況，試著在對你有益的工作和對團隊和公司有益的工作之間取得平衡。

因此，一有機會就伸出援手；這可能是一些微不足道的事情，比如進行程式碼審查、提出結對程式設計的邀請、對專案和計劃文件進行回饋、協助進行研究、提出在你擅長領域的工作建議，或是幫忙排除其他人或團隊的瓶頸等等。

記錄你為他人所做的工作

花太多時間幫助他人的一個缺點是，他們得到了功勞與表揚，但你的經理（甚至你的同事）也許不會發現你的貢獻。請注意，當你花費大量時間幫助他人時，你可能會被誤以為不夠專注在自己的工作上，而你在其他地方的貢獻可能會被忽視。

首席軟體工程師 Tanya Reilly 將定期幫助你的團隊獲得成功的任務稱之為「黏合工作」（glue work）。她寫了一篇名叫〈Being Glue〉¹的文章，說明關於這類型工作的優缺點，內容相當引人深思。

我的建議是確保你所做的有意義的工作獲得關注，至少你的經理應該知道你在做什麼。這就是撰寫工作日誌的好處。此外，如果你注意到你花費大部分時間幫助他人，但很少時間「把事情搞定」，那麼你可能需要重新評估你的時間安排，或者你經理對你目前職及工作角色的期望。

¹ <https://noidea.dog/glue>

時常尋求具體回饋

績效評估出問題的經典方式之一是，經理就幾個月前的工作表現給出一些負面回饋。當工程師們聽到這個壞消息時，他們很可能會相當氣憤，並納悶：「為什麼不早點告訴我？」

我認為經理們有責任儘快向他們的下屬提供回饋。然而，你無法控制你的主管要如何行事。此外，在績效評估會議中，這可能不是指出他們未及時告知你的最佳時機。你可以掌控的是你自己的行為。

提早（早在正式的績效評估會議之前）就向你的經理和同事積極尋求回饋。儘早這樣做，以避免在評估會議時收到不愉快的驚喜。良好的回饋是及時的、具體的，而且切實可行的。為了獲得好回饋，請提前尋求回饋，專注於具體細節，並根據回饋內容判斷你自己可以採取哪些行動。

以下是一些獲得有用回饋的情況：

- **主持會議：**為你的團隊安排一個會議，在結束後，可以詢問團隊中更資深的工程師他們認為會議進展如何，還有什麼可以改進的地方。
- **解決故障：**你負責解決系統中的一個故障，你自認做得不錯。但事實真的如此嗎？不妨問問你的經理和團隊裡的工程師，了解他們的看法，以及詢問還有什麼可以改進的地方。
- **在大團體前進行簡報：**在全體會議上代表你的團隊進行簡報後，你可能會想知道這次表現是很成功，或者其實很糟糕。你可以去問問一兩位與會者的看法，他們認為你做得如何，還有什麼可以更清楚傳達的地方？你甚至可以錄下你的簡報，檢視自己的表現，找出需要改進的地方。
- **提出新的提案：**你向產品經理和工程經理提出了一個新專案，它被批准了。太好了！你還可以去請教一些參與過程的人，要如何才能使你的提案更具說服力呢？
- **領導一個專案：**在領導一個專案時，徵求團隊成員對專案的回饋，他們喜歡哪些部分，又有什麼地方令人困惑呢？在專案完成後，邀請人們進行意見分享，針對你作為專案負責人的表現、了解如何改進，同時進行事後回顧。

- **評論其他人的程式碼：**一位同事上傳了一份程式碼審查，裡面出現了很多問題。你對此提出了評論意見。但你的措辭是否過於嚴厲，或者太過溫和？請教有經驗的同事，請他們針對你的審查語氣、內容和風格表達看法。

你可以（也應該！）在第一次做某事時尋求回饋。這可能是接下新的任務，或者是與一個新的群體共事時。

認真對待回饋，但記住，這只是意見，並不是必須執行的指令。人們給你回饋，並不意味著你需要反射性地採取行動。傾聽他們的意見，然後想想這些看法是否合乎情理。如果是，哪些是你想要改變的地方？

找出哪些回饋是需要你採取行動，而哪些是多餘的，這是另一個話題。不過，話說回來，一切都從回饋開始，否則就會欠缺思考的燃料。

3. 在績效評估之前

當績效評估的日子越來越近時，值得花時間進行能夠確保公允評估的一些活動。評估過程越正式，提前越多時間準備就越明智。

你可以多大程度依賴你的經理作為支持者呢？

無論工作績效的回饋機制為何，你的經理在回饋過程中都扮演著關鍵角色。因此，請試著了解以下幾點：

- 你在經理眼中是什麼樣子？他們如何看待你？他們是否認為你達成了角色期望、表現不足，或者超乎預期？找出答案最簡單的方法就是主動詢問。
- 你在團隊和同事中的地位如何？績效評估有時是一種比較的過程。即使不是，你的經理也會知道誰「績效優異」、誰「表現穩健」，以及少數幾個「有待加強」的人。請對自己誠實；你認為自己屬於哪一組？不要指望任何待人得體的經理會直接告訴你答案。
- 你與經理的信任程度如何？你有多信任他們，他們對你又有多少信任？你們已經一起完成了多少次績效評估，過去是否出現過意外？

- 你的經理在組織中的地位如何？他們的任職時間有多長，在同儕團體中有多大的影響力？經理的影響力越大，任職時間越長，他們在校準會議中成功推動預期結果的能力就越強。

如果你有一位新經理，他的影響力不夠大，或任職時間不長，又或者你們之間的信任程度不高，那麼先做好心理準備，這些因素將對你的績效評估產生負面影響，並為此做好準備。你可能需要加倍努力彌足這些劣勢，讓你的經理難以給出不利的回饋。

找出重要的截止日期

完成績效評估文件的截止日期是什麼時候？決定評級的校準會議何時舉行？你的經理什麼時候可能完成他們的準備事項？提示：在接近會議日期之前。

掌握關鍵日期至關重要，因為在一切塵埃落定後提供額外的脈絡資訊信已經於事無補。在此之前完善關於你的貢獻的任何知識盲點。向你的經理提供一份摘要，供他們在績效校準會議之前參考。

蒐集同事的回饋

如果已經有正式的流程讓同儕分享回饋，請確保你選擇的同事足夠了解你的工作脈絡，並願意給予回饋。告訴他們如果他們願意抽出時間給予回饋，你將不勝感激。

如果沒有這類正式流程，那麼你將會得到更少的回饋，這對你的專業成長或者讓你的經理清楚地了解你在團隊中的工作情況並不利。

我建議私下向同事尋求回饋；詢問你做得好的地方，以及有待改進的地方。我強烈建議將你聽到的內容寫下來。根據你與經理的信任程度，你可以向他們展示這些回饋。

撰寫自我評價

你可以透過提前提交一份自我評價，讓你的經理更輕鬆。他們需要寫一份關於你的工作摘要，並提供有關你表現的意見回饋，可能還要包括校

準以證明評級的合理性，以及比較不同下屬的最終表現分數。因此，請提供大量的證據來支持你的工作績效。

撰寫一份文件，並列出以下內容：

- **成果 (what)：** 包含過去的工作，按時間進展逐一交付的工作成果，幫助你的經理避免受近期偏誤影響。這就是工作日誌確實發揮效用的時刻。
- **方法 (how)：** 達成工作的方法，例如幫助他人、指導、追求卓越、做出務實的決定等等。
- **原始目標和成果：** 反思你先前設定的目標以及確實達成了哪些目標。
- **能力的例證：** 如果你的組織列出了職位角色應該具備的能力 (competencies)，以例子展示你如何勝任工作。
- **同事的回饋：** 如果沒有正式的同儕回饋機制，在此列出一些你蒐集到的意見。
- **讚美和正面回饋：** 蒐集你收到的書面或口頭表揚，因為這些互動很容易被忽視。蒐集這類正面回饋將有助於你和你的經理考慮到你在哪些地方表現優異。

有兩個問題需要考慮：其一，是否要在這份摘要文件中給自己評分，其次，你認為自己的工作表現是否達到、超出或是低於期望。這取決於你自己的方法和你與經理的關係；他們的工作是評分，而有些人可能認為為自己評分有點過猶不及。

另一方面，在我看來，誠實的自我評價是一件好事。這顯示了你對自己表現的真實看法。此外，無論如何，你的經理都會給出他們心中的評分。也許你自己的評價會顯示出你們的看法存在差異。我建議你寫下自己對自己表現的評價，除非有充分的理由不這麼做。

6 換工作

軟體工程師在同一家公司工作直至退休是非常罕見的情況。即便是像現任微軟執行長薩蒂亞納德拉（Satya Nadella），他在 1992 年以軟體工程師身分加入微軟，並於 2014 年接任微軟執行長，在這之前他也曾在昇陽電腦（Sun Microsystems）擔任過工程師。

關於何時該換工作，以及如何換工作的問題，有哪些明智的方式可以幫助我們知道何時該把握機會，到其他地方尋找新工作呢？本章涵蓋：

1. 探索新的機會
2. 等待升職 vs. 換工作
3. 準備技術面試
4. 降等
5. 升等
6. 入職新工作

1. 探索新的機會

人們出於各種各樣的原因去探索新的工作機會，通常可歸納為幾類：

積極求職和應徵

當你積極尋找工作時，最明顯的情況包括：

- 新畢業生尋找展開職涯的第一個機會。
- 失業並需要找到下一份工作。
- 私人生活變動，需要尋找更適合的工作：這可能包括搬家到新地點、需要更靈活的工作安排等。

- 在當前工作中感到不快樂：可能是因為與經理、團隊或工作環境存在問題，且無法透過內部調動解決。
- 意識到自己的薪資遠低於市場行情，希望透過換工作來獲得應有的薪資水準。

在積極求職時，你應該全力以赴投入這個過程，這表示你要積極搜尋職缺，至少也要回應那些主動聯絡你的訊息。你大概需要在 LinkedIn 上將求職意向狀態設定為「積極求職和應徵」，讓獵頭和招募人員清楚知道你正在尋找新的機會，並知道自己想要尋找哪類工作機會。

被動求職和應徵

在這種情況下，你對目前的工作還算滿意，但對潛在的有趣機會保持開放態度。你不會積極地搜尋職缺，但會閱讀來自獵頭或招募人員的訊息，並回應一些你有興趣的機會。如果某個工作機會引起了你的好奇心，你會與招募人員進行交談並參與招募流程。

在這種情況下，你對於面試的職位沒有太大的壓力，因為你在目前的工作中感到相對滿意。你對新工作機會的期望相對較高，通常會希望在薪酬和職等上都有所提升。你有本錢挑剔，因為你對當前的工作狀態還算滿意。

然而，如果你長時間處於這種被動參與面試的狀態，你可能會逐漸意識到自己已經準備好離開目前的工作，尋找一個不僅僅是「還行」的更好機會。

滿意目前工作

即使對目前的工作感到滿意，仍有幾個參與求職和面試的好處：

- 確定自己的市場價值：「我拿到的薪資是否對得起我的價值嗎？」是許多人常常自問的問題。尋找答案的一個有效方法是和招募人員聊一聊，對目前的市場薪資行情有大概認識。你甚至不需要走完整個面試流程；你只需要向獵頭或招募人員詢問某個職缺預期的薪資範圍。如果這個範圍遠高於你目前收入，你可能會決定爭取這個職缺。

- 了解自己的就業能力：如果久未練習，技術面試的相關技能容易生疏。在就職期間持續參與面試，是一種維持面試技巧的方法。
- 透過新工作提升自己：你也許想透過參與面試來爭取比現在更好的待遇，這意味著你不必在目前的公司等待升職才能獲得加薪。

探索新機會沒有單一的「最佳」方法。任何機會的出現時機和方式往往不受你掌控，你唯一能控制的是，要不要把握這些機會，以及要多認真對待它們。

以我個人來說，我曾在對工作滿意時放過潛在的大好機會，如果時間能重來，我希望當時能更仔細地考慮。2011年，我對一家剛剛募集到800萬美元種子基金的小型新創的徵才訊息視而不見。當時我並沒有太在意這家新創公司，有一位職稱是「QA測試」的人想要與我聊一聊開發他們公司的Windows Phone應用程式。幾年後，我才意識到這個人是WhatsApp的執行長Jan Koum。你大概有聽說過Whatsapp；Facebook在2014年以193億美元的歷史性高價收購了它，當時WhatsApp大約有50名工程師。我對當時的工作很滿意，但如果我能預見未來，我會選擇去面試那個職位。

當然，我也參與過一些最終證明是浪費時間的面試，因為該工作機會或待遇不夠吸引人。與那些我追求但最終放棄的機會相比，我更後悔沒有去參與那些可能很有趣的工作機會。

2. 等待升職 vs. 換工作

如果你認為自己有能力在更高職位上一展長才，這時候你可能會很想投向就業市場求職，而不是在目前公司等待漫長的升職流程。為什麼不嘗試外部面試呢？也許你能得到想要的職位和更高的薪水。這是一個理所當然的選項，特別是如2021-2022年那時火熱的就業市場。然而，在應徵心儀職位之前，你還有一些細節需要考慮。

升職通常基於過去表現

想要獲得升職，你需要在 6-12 個月內表現出超越目前職等的工作能力，最後讓升職結果證明你的確做到了。這也意味著你可能有機會跳到另一家公司，拿到更高職等並獲得更高薪酬。實際上，如果你能跳槽到一家在三峰分佈式薪酬模型中位階更高的公司（正如我們在第 1 章〈職涯發展路徑〉中所提到的）你的薪酬很可能大幅成長。

基於過去表現而定的升職結果也意味著，如果你能在外頭獲得更高階的職位，並且不需要在目前的公司等待升職，你在職涯初期能夠進展得更快。

職位越高，換新工作的風險也越大

在專家和首席工程師等資深職位尋求新的工作，與在初級職位換一份工作無法相提並論，因為想在這些資深角色中表現出色，你需要良好的政治手腕。你需要了解新組織的運作方式和不成文規則，與新的關鍵利益相關者建立信任，並知道如何影響影響更大群體的決策。

建立一個彼此信任的同事網路，並且搞懂組織如何運作，這需要相當長的時間。當你轉職到一個新職位，一切必須從頭開始。如果無法迅速適應，你可能會發現自己處於一種非常努力但難以滿足更高期望的壓力狀態。

隨著你的資歷增加，任職年限變得越來越重要。對於職業生涯初期的專業人士來說，經常更換工作是常見的選擇，但如果有任何公司停留的時間都不到 2-3 年，可能會在應徵資深及以上工程職位中吃虧。

專家和首席工程師需要時間來建立人脈，並執行長期、有影響力的專案。必須透過了解背景脈絡、理解公司業務，並正確優先處理工作來推動這種影響力。在通常情況下，在大多數公司習得這些要領通常需要至少一年的時間。

任職年限也為領導者帶來了必要的視野。在同一團隊和同一公司任職多年，你過去做過的決策會回來困擾你和其他人。你會意識到當時沒有人

考慮到的一些事情可能會演變成大麻煩，或者更謹慎思考如何透過在早期多做一些工作，避免日後的大量工作。

在辭職之前，考慮一下你將放棄什麼。如果你在一家公司任職時間不長，這一點尤其重要。建立一個被廣泛採用的平台需要許多年。推出一個產品然後逐步迭代，經歷每個成長階段，也需要許多年。工程師和管理者在高成長新創工作多年之後，其價值越來越受追捧的一個原因是，他們在同一地點看到了幾個成長階段的發展，可以到其他公司加以應用這些寶貴的經驗。

在為這本書進行研究時，我與幾位 Uber 前同事聊了聊，他們現在是新創公司的專家工程師、技術長和工程副總。他們都在 Uber 工作了相當長的時間，而他們都告訴我同一件事：他們在 Uber 的那些年，經歷了其各個成長階段，這項經驗在今天給他們帶來了莫大優勢。他們能夠以信心制定策略性決策，因為他們知道在未來的 6 個月、12 個月或 24 個月內會發生什麼。怎麼做到的？因為他們在 Uber 待得足夠久，能夠見證決策在這些時間範圍內如何發酵。

別誤會，任職年限並不是你要奮力追求的目標。然而，如果你在這一行工作了 8 年以上，但沒有在任何一個地方待超過 2-3 年，那麼在下一份工作待久一些會是個好主意，獲得只有因任職年資增長才能獲得的視野。

升職難度隨著職位越高而增加

當你變得更資深，很容易發現內部升職機會變得更稀缺了。比方說你可能會遇到預算限制，在你想要爭取升職為專家，而此時組織沒有足夠預算可以養第二個專家工程師。對於管理職來說，沒有成長空間尤其常見，他們通常需要管理更大的團隊並實現更有影響力的成果才能爭取升職。

如果你遇到這種瓶頸，尋找外部機會是更明智的選擇，特別是當你發現學習的東西越來越少，或者你決心要在職業生涯中持續進步的時候。

在獲得一份工作邀請時，你不應該只根據薪資多寡來評估。相反地，你應該考慮其他重要因素，比如職業發展性、你將要合作的同事、面臨的挑戰、公司使命、靈活性等等。

最後，不要忘記，在一個升職週期中錯過升職機會可能看起來是件大事，但十年後你可能已經不會記得這件事了。將眼光放長遠，做好長期規劃，並考慮在這段時間內的整體策略。

3. 準備技術面試

如何為軟體工程師面試做準備？幸運的是，網路上有大量資源可以助你一臂之力。本節內容簡要概述資深工程師的典型面試流程。

中型和大型科技公司傾向採用類似的面試流程，這些流程有時被稱為「類 Google 面試」，因為它們與這家科技巨頭的面試流程非常相似。

面試類型	面試者
初次篩選	
履歷篩選	獵頭、招募人員、用人經理
初次電話	
技術篩選	
技術電話面談	軟體工程師或用人經理
程式設計作業	軟體工程師
現場面試	
程式設計面試	軟體工程師
系統設計面試(架構面試)	軟體工程師
經理面談(行為面試)	用人經理
抬桿者面試	專門的「抬桿者」面試官、用人經理

▲ 應徵資深工程師職位的經典面試流程

讓我們來看看不同類型的面試：

初篩階段

本階段目標是判斷候選人是否值得推進到第二輪的技術篩選，根據以下幾個標準進行評估：

- 符合該職位的基本條件。比方說，如果某個職缺需要候選人擁有 5 年以上行業經驗（包括後端領域）的專家級角色，那麼缺乏這些經驗的候選人將不會進入下一階段。
- 具備相關軟實力。這一點將在與招募人員或用人經理的初次對話中得到確認。
- 對參與面試流程表現出積極性。某位候選人從履歷上看來相當出色，但他 / 她對面試流程有足夠的參與意願嗎？有些候選人純粹出於好奇心而應徵，而有些人並不願意積極參與面試。在這種情況下，讓這些不適合的人選進到面試下一階段將會是浪費時間。

初篩通常由負責填補職缺的技術獵頭、用人經理或招募專員負責。

「招募專員」（Inbound sourcer）是一個專門延攬人才的人資角色，常見於大公司。由於每個職缺都可能收到成千上百份應徵信，而每一份申請都需要人工審查，因此這些公司設有專門負責審查職位申請的職位。招募專員主要負責處理那些主動申請職位的候選人之履歷。他們的工作是篩選、評估並將合適的候選人推薦給經理或面試官。

招募專員角色通常不太受歡迎，因為這項工作本身很少帶來刺激感。但這些人能夠非常高效地審查履歷和評估候選人。當我在 Uber 擔任用人經理時，我曾與一些招募專員合作，他們每天都會瀏覽無數份履歷。

初篩分為兩個階段：履歷篩選，以及與經理的初次通話。

履歷篩選：此階段對應徵者的履歷進行審查，評估應徵者是否符合該職位角色的基本條件。光是如何打造一份亮眼的軟體工程師履歷，就有說不完的內容。我的另一本書《The Tech Resume Inside Out》針對履歷撰寫提供許多詳盡建議。

這裡分享一些寫出吸睛履歷的要訣：

- 記住，你的目標是打動篩選人員，讓他 / 她覺得與你進行初次通話是值得的。這裡的目標不是詳細描述你的職涯軌跡。
- 從你想申請的職位為出發點，講述你希望招募人員或經理聽到的故事。連結相關經驗，呈現你做的這些事與申請職位的相關性。
- 客製化修改履歷。不要用同一份履歷申請所有職位，你可以準備好一份「主版本」，在遞出每一份申請前進行細部調整。
- 使用方便閱讀的排版。越容易被快速閱讀的履歷格式，更能讓招募人員或經理留下印象。避免使用花哨的格式和雙欄式排版。請參考〈The Pragmatic Engineer's Resume Template〉¹ 的免費範本。
- 強調成果、影響和數字。盡可能具體展現你的成就，用數字定義你的影響力。
- 一份有力的推薦信通常比亮眼的履歷更有幫助。請從你的人脈中尋找推薦人。

更多其他建議，請參考《The Tech Resume Inside Out》²。

與招募人員或經理的**初次通話**：通過履歷篩選後，下一關是電話訪談。這次通話的目的是蒐集更多關於應徵者的資訊，讓招募人員了解他們的情況、尋找什麼職位，並為他們提供更多關於某職位的背景資訊。有時，這會變成一個「推銷」通話，由招募人員告訴應徵者為什麼某個職缺是個吸引人的機會。

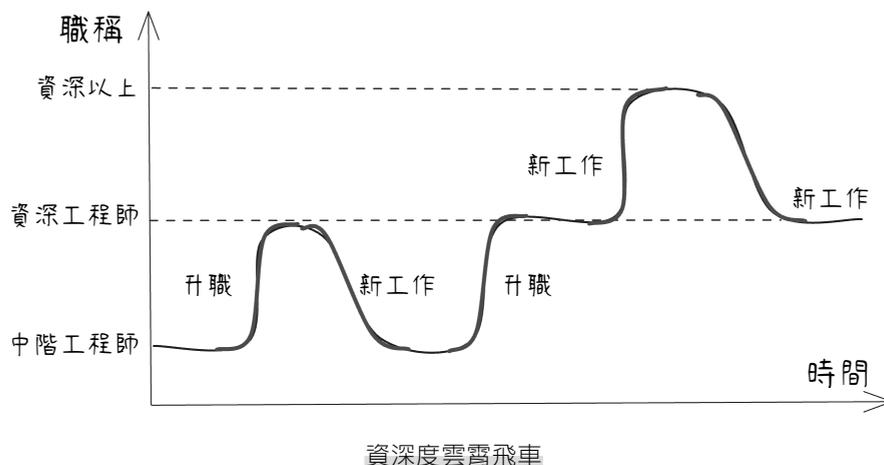
如果是與招募人員進行的通話，他們通常會蒐集一些「軟」信號，比如職位候選人的溝通能力是否清晰、有條理，以及他們對該職位有多大興趣。

如果這次通話是與經理進行訪談，他們通常會更進一步，蒐集更多信號並進行更「嚴格」的篩選。我認識的一些用人經理會深挖候選人做過的

¹ <https://blog.pragmaticengineer.com/the-pragmatic-engineers-resume-template>

² <https://thetechresume.com>

我稱這種現象為「資深度雲霄飛車」。



職稱下降而薪資增加可能有幾個原因，例如：

- 移動到更高「等級」。一個常見的原因是跳槽到更高等級的公司會帶來更高的期望，同時也帶來更多的薪資。你可能會看到你的職稱被下調，但薪資數字增加了。
- 不同公司的職稱隱含不同的期望和薪資。例如，科技巨頭對資深工程師的要求比在小型開發代理商、非科技為主的公司或新創公司更高。
- 面試表現。候選人在面試中的表現對於定級有著直接影響，也是被降等的最常見原因。面試官在對候選人有疑慮的情況下，通常會提供比他們面試的職位低一級的職位。
- 換公司和技術堆疊，代價是更低職等的工作邀約。那些以特定技術經驗年數來評估候選人的非科技優先公司和開發代理商，可能會給一個「僅」有一年 Java 經驗的人提供更低的職稱，即便這位應徵者已經是有十年資歷的軟體工程師。出於這個原因的降等常見於以特定技術為面試重點的職缺。

特別是首次加入大型科技公司時，被降等尤為常見，因為這些公司通常設定高期望，入職訓練時間比在小公司要長，並且公司經常旨在聘請能夠在特定職等「提升標準」的人。

5. 升等

儘管降級更常見，但相反的情況也可能發生。你可能會得到一個比你預期更高、或是認為自己已準備好勝任的下一職等。

當我在微軟工作時，我是一名 L62 軟體工程師，比 L63 資深軟體工程師低一個職等。然而，我從 Skyscanner 收到了一個首席工程師職等的工作邀請（比資深工程師還高一等）。究竟發生了什麼？

在我的例子中，我在微軟的職等可能並未真正反映我在職業生涯中的位置，我可能被（微軟）降等了。與此同時，我的經驗對 Skyscanner 來說是有需求的；我在面試中表現出色並且與經理建立了良好聯繫。我並沒有打算離開微軟，除非有一個非常誘人的工作邀請。因此，Skyscanner 給了我一份升等的工作邀請，才讓一切拍板定案。

想獲得升等的工作邀請，需要達成幾個天時地利人和的條件：

- 在面試中表現出色。說服招募小組你是一個傑出的候選人。
- 擁有獲得更高職等的槓桿。如果你得到另一份更高職等的工作邀請，或者你願意拒絕一個沒有幫你升等的工作邀請，那麼你就有了槓桿。
- 在公司內部有人為你「爭取」。成功爭取升等通常要歸功於面試小組中的一個或多個人（或者是推薦你的人）說服用人經理或招募委員會。

小提醒，一旦你獲得更高職等，要意識到從入職第一天起，期望值就會提高。確保你自己釐清這些期望的要求，努力在你的新角色中達成這些期望。

6. 入職新工作

你接受了一份新工作，恭喜你！你值得花點時間來慶祝成就。但不要太過放鬆，要成功地融入你的新公司，迅速推進工作。

在新公司最初幾個月的表現，所形成的印象往往會持續許多年。這表示，如果你的入職進度良好且你的貢獻超出了基本期望，你很容易被視為一個優秀人才。相反，如果你適應公司的速度被認為進展緩慢或有些困難，你可能會被認為效率不高；這樣的印象可能需要你花費比入職時間更長的時間才能消除。

作為一位新加入者，你可以（也應該！）超前部署，主動承擔你的入職責任，而不只是希望被迎接進一個完美的入職流程。這麼做表示你將更快地融入，並顯得像一個能迅速開始工作的人。這樣的印象應該會使你在職場生活更加輕鬆，並持續到未來很長一段時間。

以下是在第一天之前就主動適應的方法：

在所有公司和所有職等

有效方法包括：

- 如果在簽署工作合約後有較長的通知期，可以考慮在你開始工作前主動聯絡未來的經理。詢問關於如何做好準備，迅速開始工作的建議。
- 閱讀麥克瓦金斯（Michael D. Watkins）的《從新主管到頂尖主管：哈佛商學院教授教你 90 天掌握精純策略、達成關鍵目標（The First 90 Days）》。這本書分享了一些在新地方開始工作時如何創造良好形象的最佳建議。
- 從一開始就紀錄工作日記。這將幫助你在入職期間進行反思，並為與經理的一對一會議、績效評估和升職提供完善的參考資料。
- 寫下每週日記。記下你在最初幾個月中學到的三件事，以及三件你不理解的事。
- 與經理溝通，確定你在第 1 個月、接下來 3 個月和前 6 個月的目標。

在小公司的入職

在新創公司和小型組織中，可以做以下一些實用活動：

- 在加入前，溝通並釐清你將要做的工作。
- 及早與最資深的人建立聯繫。他們通常在職時間較長。

- 不要預設事情的運作方式；要主動發問。從一家大公司來到新環境時，這一點尤其重要。如果你提議一些對新公司來說沒有多大意義的意見，可能會顯得你很無知。
- 努力在第 1 週內交付某些成果。在小公司，你更加責無旁貸。

在大公司的入職

大型組織更為複雜，因此從第一天起就要適應並掌控這種複雜性：

- 即使沒被指派，也要找到一位「入職夥伴」（onboarding buddy）。在理想情況下，這應該是你團隊中的某位同事，你可以向他們提問，他們也願意在被打斷的情況下幫助你快速熟悉事物。
- 用一份「速查表」（cheat sheet），將所有新事物記錄下來。這可能包括各式縮寫及其含義、構建指令、關鍵資源的超連結等等。這份文件將加速你的學習過程，也可以看作入職文件的補充資料。
- 熟悉公司的技術堆疊。學習你將要使用的新語言或框架。
- 接受許多事情一開始可能看來沒有道理，特別是如果你來自一家較小的公司。別擔心，這是正常的！

資深或以上職位的入職

身為一名經驗豐富的工程師，請為你的入職增加結構性：

- 與一位資深或以上職等的同事建立聯繫。了解他們如何組織和衡量工作。
- 優先在團隊的程式碼庫上提高生產力，親自動手（寫程式碼）。
- 與你的經理溝通，確定 3 個月、6 個月和 12 個月的期望目標。
- 了解當前和未來的專案以及團隊的優先事項。團隊正在做什麼，為什麼這些工作很重要，這些工作如何與公司策略和優先要務相契合？
- 尋找並瀏覽與你的團隊和領域相關的事故報告文件，以便了解問題領域，並在問題重新出現時不至於束手無策。
- 寫下值得了解的工程團隊清單。當你聽到有人提到某個新團隊時，將團隊名字加到此這份清單中。然後，向該團隊的一名成員介紹自己，

了解他們的工作內容，以及你的團隊與他們的團隊如何產生關聯。身為一位新人，主動聯繫非常容易，因為你有向人介紹自己的藉口——因為你是新人！你可以透過這種方式了解其他團隊，認識並建立將來可能有用的聯繫對象。

專家或以上職位的入職

Staff+ 職等的角色在不同公司間有所不同。弄清楚公司 / 團隊對你的期望是什麼，然後建立你的人脈網路，這樣你就可以解決棘手的問題並實現這一層次所期望的重大成果：

- 提早與你的團隊見面，給人留下友好的印象。
- 也要提早與同級的 Staff+ 工程師見面。你將在很大程度上仰賴他們了解事情的運作方式，特別是在大公司。嘗試結交一些新朋友。
- Staff+ 角色因公司而異，所以要弄清楚你的角色中哪些事情什麼被期望的，哪些不是。
- 與你的經理以及在理想情況下與你的跳級經理（經理的經理）討論 3 個月、6 個月和 12 個月的期望目標。團隊的主要優先事項和需要解決的最大痛點是什麼？
- 花時間與產品人員交流，了解他們目前的關注點和計劃。

重點整理

科技產業變化迅速，大多數開發者傾向於每隔幾年就換一次工作。同時，我們大多數在科技領域工作的人可以期望自己的職業生涯持續數十年。在〈A forty-year career（職業生涯四十年）〉⁴一文中，首席技術長兼作家威爾拉森（Will Larson）思及科技領域之外的典型職業生涯：

我的父親幾年前退休了，他在北卡羅來納大學艾塞維亞分校擔任教授，這份工作幾乎貫穿了他成年後的大部分生活。自從那時起，我花了比預期更多時間在思考他退休之前的職業生涯。特別是當我發

⁴ <https://lethain.com/forty-year-career>

現，自己的職業生涯是可以在四十年的時間跨度上有意識地發展的觀念。不是四次首次公開募股，不是十四次兩年的任期，而是四十年。

我竟然忽略了「職業生涯四十年」這一視角實在奇怪，畢竟在很長一段時間裡，這是我唯一知道的模式。（中略……）如果你有幸開始一份長期職業生涯，盡你所能避免絆倒，並搭乘這條「職業生涯輸送帶」邁向成功、擁有自己的房子，最終退休。

請記住，開發者的職業生涯通常持續相似的時間長度，為你考慮接受的工作機會制定規劃很有幫助，無論是不同類型的公司，還是如何應對績效評估、升職或轉換工作。本章簡要地涵蓋了上述職涯發展的所有面向及規劃方法。

但如果你從本章中只能學到一件事，那它應該是：

沒有人能像你自己那樣關心你的職業生涯。無論是你的經理還是你的同事都不會。

因此，了解你有多想拓展自己的職涯，並為之付出相應的努力。

關於本節內容的延伸閱讀，歡迎參考第一部的網路版補充章節：



Working at a Startup vs. in Big Tech

pragmaticurl.com/bonus-1

22 協作

Staff+ 工程師的大部分工作都涉及與其他工程師、經理、產品團隊、業務相關人士及其他同事的協作。在許多情況下協作並非由你主動發起，而是人們主動來尋求你的協助。

最具挑戰性的專案，其難度往往不在於所需編寫的程式碼。通常，主要的痛點在於與他人合作，這常常讓人感覺像是在「牧貓」——個個特立獨行、難以控制，簡直就像一場不可能的任務。

在與同事協作的過程中，你難免會在某種程度上捲入內部政治，或被認為參與其中。為什麼會這樣呢？因為根據古希臘的一則智慧，人類本質上就是「政治動物」。協作牽涉到在你人際網路中的人們，而你的影響力能為團隊中的工程師創造機會。這是打造成功職涯的一個關鍵因素，同時也是我們著重討論這個話題的動機。本章將涵蓋以下內容：

1. 內部政治
2. 影響他人
3. 與主管協作
4. 與其他 Staff+ 同儕協作
5. 擴展你的人際網路
6. 幫助他人

1. 內部政治

內部政治，或者說辦公室政治，在許多軟體工程師心中觀感不佳。如果一位個人貢獻者（IC）或主管被認為「有政治手腕」或「善於搞政治」，幾乎總是帶有負面含義。這通常意味著某人幾乎不做技術貢獻，而是利用他人來達成自己的目的，有時甚至運用操縱或算計的手段來實現個人目標。

那麼，被視為「有影響力」又是什麼樣子呢？這與「善於搞政治」有很大區別嗎？影響力通常用來描述一位具有強大技術能力的同事，他們同時也擅長為有利於團隊或組織（而非出於私利）的倡議爭取支持。

實際上，政治手腕和影響力常常相伴相隨，儘管我們對「有政治手腕」的同事和有影響力的同事有截然不同的評價。影響力是一種「良性的」辦公室政治形式。這就是為什麼如果你想支持你的團隊並在軟體工程師的職涯中取得進展，被視為有影響力而非善於搞政治的人會更有幫助。

「不當」的政治手腕

「政治手腕」這種標籤之所以名聲不佳，是因為它常常描述被視為自私自利的行為，為了某個人或小團體的利益而犧牲他人。我們應該避免給同事留下這樣的印象：為了自己或「小圈子」的利益而使他人的工作變得更加困難。

當成功主要依賴於非正式的軟實力和人脈關係時，這種情況就顯得格外不妥。我認識的大多數開發人員（包括我自己）都認為軟體工程應該是中立的，根據想法的優劣來客觀評判，這正是許多工程師厭惡辦公室政治的原因。

然而，**觀感確實很重要**。你可能做了一件無私的事，但如果同事們缺乏完整的背景資訊，你的動機反而可能會被誤解是自私自利。

讓我們以一位專家工程師為例。他參與了升職委員會，要決定其團隊中某位資深工程師的薪酬方案，而提案在這位專家工程師未能支持後被拒絕。

人們可能會問：這位專家工程師是否因為沒有提攜團隊成員而顯得自私？他們的動機是什麼？是否有政治考量？誰從中受益？顯然不是那位沒有獲得加薪的資深工程師。那麼，這是否暗示著這位專家工程師間接受益？這類難以回答的問題，如果不在決策現場，很容易導致人們認為他們的同事在搞政治算計。

實際上，委員會成員很可能需要在利益衝突的情況下迴避投票。這位專家工程師很可能就是如此，無法參與關於其團隊成員的升職決策。

然而，人們最容易做出的假設是這位專家工程師出於某種原因破壞了提案，沒有投票支持。這種假設在這個案例中大大誤解了最為關鍵的動機：這位專家工程師藉由拒絕參與他們有直接利益關係的決策，事實上是為了維護更高層次的公正性。

這個例子生動地體現出為什麼在缺乏完整背景資訊的情況下很難做出準確判斷。在實際工作中，人們常常用猜想來填補知識空白。這正是觀感和背景資訊如此重要的原因！

有問題的觀感

自私自利：如果某人被認為只一心在乎自己的專案和工作，那麼他們會因為看起來只專注於個人升職與否而難以結交善緣。誰願意與一個不懂得回報且利用他人來實現個人抱負的人合作或幫助他們呢？

排擠同事：比赤裸裸的自私更糟糕的是，被視為為了升職不惜一切排擠他人的人。例如，一個獨占專案並阻礙他人貢獻以獲得所有功勞的工程師，不應該對團隊成員對自己的負面看法感到驚訝。

固執己見：在討論提案時表現出毫無彈性，一味堅持自己觀點的工程師，可能被視為在追求私人目的。特別是當他們使用權威而非理性論據時，說些類似「我是專家工程師，我說了算」的話。

兩面派：對不同人說不同或矛盾的話，會給人操弄算計和追求私利的的印象。一旦有人被認為是這樣行事，同事對他們的信任通常會急劇下降。

強行推行己見：亞馬遜常用的一個領導原則是「有主見；敢承擔」。這是鼓勵同事對於無法苟同的決策表達不同意見，但當做出決定之後，就要放下分歧，勇往直前。這種「有異議但承諾」的心態在許多公司都存在。然而，當有人在缺乏共識的情況下卻一意孤行時，這種方法很容易被當成武器。在短期內或「戰時」模式下強行推動倡議可能很有效率，但這種方法很少能贏得朋友。

不信任其他工程師：一些經驗豐富的工程師會將工程工作委託給經驗較少的工程師。這麼做非常好！然而，如果他們發現經驗較少的工程師做事方式不符合慣例，就會收回這些工作。這會造成對同事缺乏信任的觀

感，並可能產生負面印象，導致經驗較少的工程師可能會迴避資深同事尋求建議和指導。

光說不做：一種特殊的政治動物是那些從不寫程式的老鳥工程師。這是一個棘手的觀感問題，因為隨著工程師升職到 Staff+ 職等後，他們被期望做的寫程式工作會大大減少。由於需要處理其他優先事項，他們根本沒有時間親自動手。

儘管如此，那些從不接觸程式碼庫、從不參與值班，並強行推動對工程師產生他們不認同的程式碼變更的 Staff+ 工程師，通常會給人負面觀感，被視為在決策中過於疏離和事不關己。

對不良的政治手腕提供回饋

回饋的重要性不容忽視，因為缺乏回饋通常會使情況每下愈況。「我今天要表現得自私自利，像匹脫離團隊方向的脫韁野馬，並且算計我的同事。」幾乎沒有軟體工程師會帶著這樣的念頭上班。然而，有些人確實給人留下這樣的印象。那麼，究竟是什麼導致了這種情況？

缺乏回饋是有時候工程師被認為在「搞政治」的一個常見原因。而這些工程師往往渾然不覺自己給人這樣的印象！

提供這種回饋確實具有挑戰性。但我認為，作為團隊成員，我們不僅可以而且應該採取行動，幫助他人了解他們的行為如何被其他人看待：

- **如果你與對方同級：**可以直接向他們提供回饋，或向你的主管反映情況。你的選擇將取決於你與當事人的關係。建議在可能的情況下直接給予回饋，但如果彼此缺乏信任，這可能會很棘手。
- **如果你比對方資深：**應直接向他們提供回饋。努力描述具體的行為或事件，並耐心聆聽他們的解釋。如果他們的意圖與行為給人的觀感有所不同，可以建議他們如何調整行為。
- **如果你比對方資淺：**最好向你的主管提供回饋，因為直接給予回饋可能不合時宜。但請記住，聽取你的觀察並決定如何處理，是你主管的職責。

2. 影響他人

聰明的做法是避免給人留下過於熱衷於搞政治的印象。儘管如此，影響工程師和主管的能力通常很重要，同時這本身也是一種「良性」的辦公室政治。以下是一些影響力可以發揮作用的情境：

- **爭取提案通過**：你有一個新系統的提案，相較現有系統有諸多優勢。你深信若其他人能看到這點，此提案將為組織帶來巨大效益。
- **抵制不利於組織的決策**：上層下達了轉移至新系統的指示。然而，你發現新系統存在諸多缺陷，意味著你的團隊要麼必須投入大量額外工作來彌補這些缺陷，要麼客戶將失去某些功能。兩種選擇都不可接受，因此你必須向決策者反映情況。
- **為團隊成員的提案辯護**：你的團隊成員提出了一個很好的提案，但未能獲得支持，包括你主管的支持。你認為這個提案有價值，考慮到其對業務的積極影響，應該在團隊層面討論。你可以運用你的影響力為同事的想法爭取支持。
- **參與重要專案**：你得知另一個團隊啟動了一個新專案，你的專業知識可以幫助他們加快進展。對組織而言，正確的做法是你投入時間在這個新專案上。然而，你無法同時完成當前工作並在新專案上投入足夠時間。因此，有必要說服你的主管，轉移你的工作重心對組織來說是正確的決定。

在組織中擁有強大的人脈以及足以影響人們的影響力，兩者密不可分。一般而言，人們聽從你是因為他們信任你，這意味著你已經付出努力並建立了這種信任關係。但你要如何贏得信任，讓人們願意聽從你呢？在第 12 章〈協作和團隊合作〉中，我們介紹了對資深工程師有幫助的方法。接下來，我們將討論適用於 Staff+ 職等的方法。

累積「信任資本」

許多事情的成敗都取決於「信任資本」。在你的組織中，這種資產的來源可能包括：

- **頭銜 / 權威**：人們會把注意力留給那些擁有專業或權威頭銜的同事，例如首席工程師、傑出工程師、總監、工程副總裁等等。
- **任職年資**：如果某人在組織中工作了很長時間，並且被認為擁有深刻洞察，那麼即使沒有特殊頭銜或權威，人們也會認真對待他們的意見。
- **專業知識**：如果有一位精通 React Native 的工程師加入組織，即使他們缺乏長期任職經驗或權威，人們也很可能會找他們解決 React Native 的相關問題。這一點適用於任何技術領域。
- **業績紀錄**：即使任職時間較短且權威較小的人，如果他們績效卓越，擁有良好的完成任務紀錄，也可能擁有超乎預期的影響力。
- **工作的能見度**：如果你做得很好但沒人知道，那算不算做得好？軟體工程師常犯的錯誤是預設了他們的工作成果會為自己發聲。但現實往往並非如此。殘酷的事實是，當你的主管和團隊成員不清楚你做了什麼、如何做的，以及它的影響時，好的工作就失去了應有的重要性。

想建立信任資本，最行之有效的方式是在長時間內持續完成任務，並在這個過程中建立良好的業績紀錄和資歷。以下是一些可能有助於加速建立信任過程的方法。

另一種有助於理解信任的方式來自前 eBay 產品總監 Anne Raimondi。她將信任定義為可信度、可靠性和真實性的總和，然後除以自私自利的觀感。她在〈Use this equation to determine, diagnose, and repair trust〉（使用這個等式來確定、診斷和修復信任）¹ 這篇文章中分享了更多增加信任的建議。

提問並積極傾聽

邀請同事分享他們的觀點和專業知識，藉此向他們學習。這在你缺乏資訊或專業知識時特別有用，比如剛加入一家新公司時。

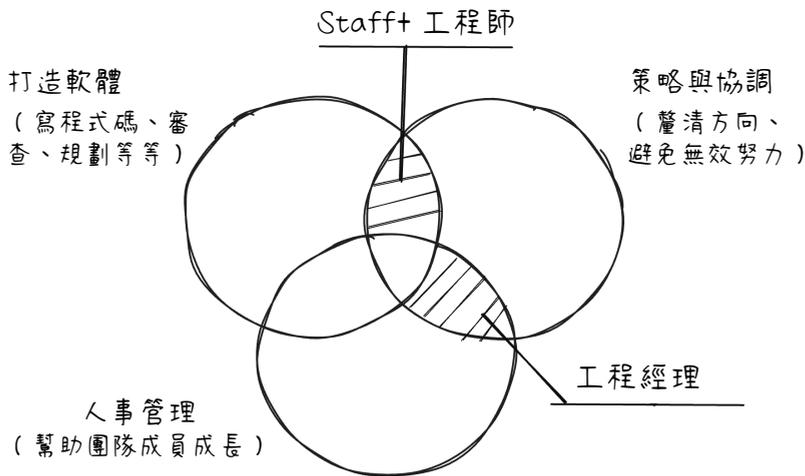
我在 Skyscanner 工作時認識的一位資深工程副總裁就是用這種方式融入團隊並進入工作狀態。Bryan Dove 坦白地告訴同事他是目前懂的最少的

¹ <https://review.firstround.com/use-this-equation-to-determine-diagnose-and-repair-trust>

3. 與主管協作

Staff+ 工程師與工程經理之間往往存在獨特的互動關係。這是因為兩者通常有相似的影響範圍，但關注點略有不同。下圖是 Staff+ 工程師和工程經理通常如何分配時間的視覺化呈現：

你通常在這三項活動中投入多少時間？



Staff+ 工程師和工程經理通常在哪些領域投入大多數時間

Staff+ 工程師和工程經理都在策略與協調上投入大量時間。因此，與工程經理協作配合這件事無庸置疑，尤其是那些你支援其團隊的主管！

向工程經理明確表達你們站在同一邊。你支援其團隊的主管（包括你自己的主管在內）應該清楚地認識到，你是在與他們合作，支持他們的團隊。因此，請花時間與他們溝通，了解他們的工作方式，並找出你們如何能夠更有效協作。

避免與工程經理產生衝突。Staff+ 工程師和工程經理都在協助團隊調整方向，保持一致，你可能會遇到你的決策凌駕於另一個主管之上，或感覺某個主管總是凌駕於你之上的情況。如果發生這種情況，就像對待同級的 Staff+ 工程師一樣處理：私下討論，探討你們如何能「同舟共濟」。明智的做法是避免這些分歧在資淺工程師面前表現出來。

與其他主管建立互信關係。你的目標應該是讓其他主管視你為夥伴，反之亦然。實現這一點需要建立信任，並證明你是可靠的。

從與你自己的主管建立信任開始。坦誠地討論責任分配，找出你可以分擔他們工作的領域，無論是協調工程事務、領導專案，還是解決團隊面臨的棘手依賴項問題。向你的主管表明，你的目標是在工程相關事務上成為他們真正的盟友。

4. 與其他 Staff+ 同儕協作

要成為一位高效的 Staff+ 工程師，你不僅需要與團隊、業務相關人士和其他主管有良好合作，同時也要與你的同儕（其他 Staff+ 工程師）建立良好的協作關係。

面對面認識 Staff+ 的同事。如果有機會親自介紹自己，千萬別錯過這個機會！了解他們的工作內容，分享你自己的情況，並討論如何互相協助。這種個人連結可能會產生深遠的影響。如果你是遠距工作，也要努力透過視訊形式建立這樣的關係。

加入或建立 Staff+ 社群。一些公司設有 Staff+ 社群，讓你可以定期與同儕交流。例如，亞馬遜以其強大的首席工程師社群聞名。在早期，所有首席工程師每年都會參加一次外部活動，每週共進午餐，並舉辦全公司範圍的技術講座。

如果你的公司還沒有這樣的社群，至少考慮為你的直接同儕組織一個，而形式可以簡單如定期舉行討論分享會。這樣的社群對所有 Staff+ 工程師都有幫助，因為擔任這個職位角色的你或組織中的其他人都會提出許多問題。你可以參考亞馬遜的做法⁶，獲得組織活動的靈感。

要注意，與不同的 Staff+ 角色原型互動時，有著不同的協作方式。工程總監暨作家 Will Larson 在他的著作《Staff Engineer》和《An Elegant Puzzle》中歸納出四種專家工程師的角色原型⁷：

⁶ <https://pragmaticurl.com/amazon-principal-engineers>

⁷ <https://staffeng.com/guides/staff-archetypes>

- 技術負責人（Tech Lead）：指導特定團隊的方法和執行
- 架構師（Architect）：負責關鍵領域的方向、品質和方法
- 解決者（Solver）：深入複雜問題並找出前進路徑的人
- 右手（Right hand）：借用其工程總監的範疇和權限在高度複雜的組織中運作

與領導專案的技術負責人合作，其協作方式完全不同於與架構師或右手型專家工程師。因此，首先要釐清自己可能屬於哪種 Staff+ 角色原型，以及你可以與哪些其他 Staff+ 工程師保持一致。

5. 擴展你的人際網路

建立一個相互信任的人際網路，對於找到盟友和發揮影響力至關重要。以下是一些擴展人脈的方法。

為自己尋找組織內的導師

即使身為 Staff+ 工程師，你仍應尋找可以學習的對象。這些人可以在你需要時成為顧問或盟友。他們不一定是工程師，也可以是工程領袖，或者來自產品部門，甚至像首席技術長這樣的領導高層。

指導可以是非正式的，例如與資深同事一起合作專案。有些組織也會提供正式的指導計劃。

參與跨團隊專案

建立人脈最有效的方法不是「刻意社交」，而是與其他團隊的人長期共事。這種機會在共同專案中自然產生。對 Staff+ 工程師來說，這類專案通常是工作的一部分。如果沒有，就主動尋找參與的機會！

與他人合作時，努力抽空更深入地了解他們。你可能會建立起比任何專案都更長久的聯繫。

參加內部訓練

在大公司，內部訓練是一種被低估的建立人脈方式，尤其是主管的管理訓練和實體課程等活動。

我在 Uber 時，托訓練課程的福認識了許多非技術領域的主管。與他們交流讓我了解其他業務部門的運作方式。我還建立了一些人脈，之後可以與他們保持聯絡，交流想法。整體而言，內部訓練能夠將興趣相近的人聚在一起，提供共同經驗，為進一步交流奠定基礎。

認識團隊外的人

在公司活動和異地活動中主動與同事交流，如果你在辦公室工作，不妨把握午餐時間多認識其他人。善用員工資源小組，參與跨部門的計劃。你認識的團隊外的人越多，就越有可能建立持久的人際關係。隨著你在職場上不斷升職，這些關係會變得越來越重要。

人脈與影響力相輔相成

想要發揮影響力，首先要贏得人們的信任。這需要透過完成任務和幫助同事來實現。

你的人脈網路對職涯發展的助力遠遠超過眼前的工作範圍。你信任的人未來可能在其他公司為你提供介紹和引薦，或幫你獲得未公開職位的面試機會，甚至跳過初步面試。

建立強大的人脈需要多年時間，這是長期積累善意、信任和投入專業工作的成果。從現在開始建立強大網路的最佳方式是幫助他人、做好本職工作，並打造一個能夠完成任務、樂於助人，並善用影響力的良好聲譽。

6. 助人為樂

身為 Staff+ 工程師，你是組織中經驗最豐富的工程師之一。你擁有知識和影響力，能夠切實幫助同事。因此，請善用這些優勢！

指導他人

指導的核心在於引導他人、分享知識，並協助他們成長。作為 Staff+ 工程師，你擁有的豐富經驗可以加速他人的成長。指導不必是正式的，它可以非常簡單，比如主動協助新人快速融入工作。

養成幫助他人成長的習慣，並且不求回報，你將提升自己的教學和表達能力。長期持續指導和幫助他人，你能夠建立起良好口碑，成為資深及以上工程師尋求建議的對象。

第 12 章〈協作與團隊合作〉中，以及在〈Mentoring software engineers〉（指導軟體工程師）⁸一文中詳細討論了指導的相關內容。

提攜後進

提攜比指導更進一步；它意味著為某人發聲，並運用你的地位促成他們的職涯發展。

作為 Staff+ 工程師，你的影響力觸及涉及主管和業務相關人士。你不僅可以，而且應該運用這種影響力來提攜那些你能協助職涯發展的工程師。這可能包括以下幾個方面：

- **表揚傑出貢獻**：當你發現某位工程師表現卓越，遠超職責要求，但其主管似乎未察覺時，你可以選擇提攜這位同事，提升他們工作的能見度。例如，邀請他們在團隊會議上分享專案經驗。
- **力薦升職**：若你認為你提攜的人已具備更高層級的能力，卻未被考慮提拔，你可以與其主管及其他相關主管溝通，強調這位同事已準備好迎接下一階段挑戰，並表達你的支持。
- **創造參與機會**：當你參與複雜專案的規劃時，你可能有能力獨自處理，但若意識到這對你提攜的同事而言是難得的學習機會，不妨邀請他們加入，讓他們能參與關鍵決策並有所貢獻。

⁸ <https://blog.pragmaticengineer.com/developers-mentoring-other-developers>