

弦而時習之

Aotokitsuruya

The Web is attracting me, so I start learning PHP, HTML, CSS and JavaScript. When I know about Ruby, I fall in love with it. Now, I working with my friend to design games using C++ and still learning web about Golang, Ruby and JavaScript. In the other side, I also be a designer, the reason why the web is attracted me.

Feb.20

學生接案的報價技巧

會寫這篇文章是因為昨晚學弟（非系上）問我他現在接了一個案子，該怎麼報價。
然後我就開始問他一些問題，來當作報價的建議依據。

同時也發現到，其實很多再接再案用於「報價」的自我判斷，學弟也沒有什麼概念。我想應該也有不少人會有這方面的問題，所以就打算來寫一篇文章分享一下我的報價技巧。

不過我已經有一年左右沒有接案了，另外就是這些「技巧」會依照行業、經營方式有所不同，這只是我個人的計算方式。

當然，裡面仍有許多「基本要素」是學生再接再案的時候會忽略的，這篇文章另一方面是要提醒各位同學這個問題。

順帶一提，網路上討論這個主題的人似乎不多，難怪學弟會抓不太到方向。

報價本身是談判

基本上，我認為報價跟談判差不了多少。簡單說就是一手交錢一手交貨，一方不滿意就各自走人。不過，絕對不會有「固定值」這就跟國中、高中學的市場經濟一樣，接案之所以有議價空間就是雙方要去找出「都可以接受」的價位。

當然，對方是社會人士，手上的籌碼一定比你多。這時候能不能運用自己的優勢逆轉勝，就要看你的談判技巧。

了解你的成本

很多人在第一次的報價會犯一個錯誤，那就是沒有計算「成本」這就像是你花了一百塊買了一個 DIY 玩偶組，等你做出成品後，因為市面上的玩偶平均只有六十元，你就用六十元售出一樣，從頭到尾你根本是虧本的。
（但是手工玩偶絕對會超過一百元以上）

所以，我們必須清楚了解到，成本跟技術價值是分開的東西。
你必須吃飯、睡覺、出門與客戶討論，你也會需要電腦、軟體等，這些就是你的成本。

假設你一個月生活費要花上五千元，那麼你在製作需要一個月的案子上，就必須超過這個價位，不然就是等著餓死。
而軟硬體部分，假設兩年買一次新的 Mac 那你就得把一台 Mac 的價格均分到每個月裡面（如果分期付款有利息之類的，也要一起算）

簡單來說，維持你的基本生活、工作水準的費用，要請讓你工作的人支付。

技術價值

當你的成本確定下來後，就可以確定在你結束接案生活之前，你的生活保證可以過下去。
但是，你也不過就是每個月把錢用完而已，就「賺錢」來說，這不能算是賺錢，只能說是「維持生活」

所以你必須搞清楚自己手上所持有的「專業技術」在市場上值多少錢，這是個人的「尺度」要是你覺得你一件作品是一萬，那就是一萬。沒有人可以改變這個尺度，只有你自己可以。

以學弟為例，他接下的是一個「網站前端」的工作，但是當我問到「你覺得你的經驗、程度到哪裡？」的時候，他一直無法回答我正確的答案。

我一開始得到的就是「我會 HTML、CSS……我學了 N 年」這類的回答，但這絕對不應該是答案。首先，我們去工作老闆絕對不會以「你會幾種東西」幫你一項項一項的加薪，很明顯的，多會一種東西不代表你擁有這份「專業」

專業勢必是依靠「時間」去累積的，也許你時間很多可以練出複數的專業。但是只說明「技能」是沒有意義的，專業必須是一種「對某件技能深入了解，而且能夠純熟運用」的狀態。

也因此，所謂的「經驗」就是一種指標，而「程度」就是你對這個「技能」投資了多少時間，深入了多少「程度」註：像是網站開發者部分，今天剛好就有一篇文章在討論可能具備的「特質」這也是一種指標的參考。

至於這個部分，最簡單的區分方式就是「找出你認識最強的人」以及「找出你朋友平均的素質」然後把自己放到這個範圍的其中一個點，做出相對的位置。

不過，要注意的是像是網站開發這種領域，變動快速的情況下。你還必須考慮到你的技能是否與時代相符，如果已經完全跟不上時代（像是你還在用 FrontPage 之類的……）那你的技術價值可能要先被扣分（如此古老的東西做出來真的能在現代用嗎？）

以網站開發來說，除了對某種語言的專業外，使用工具也是不錯的指標。像是 MVC Framework、VCS、寫單元測試等。程式碼的品質也可以做為指標，這些綜合起來的一個「尺度」就是你的技術價值。

（像是使用 MVC Framework 可能加快你的開發速度，但是也因為你擁有這份較為複雜的技巧，即使時間縮短了，但是技術價值提高了，所以不會因為時間變短而降低你的報酬）

製作時間

這部分就很明確而且直觀。

首先要知道的是「有多少時間」然後評估「實際上要花多少時間」然後來決定報價的形態。

客戶通常會希望在某個時間點「完成」但是這不代表製作所花費的「時間」所以在報價的時候，應該要以「實際會花費的時間」來計算，而非「從接下案子到交件」的時間來計算。

（總不可能客戶要你一天做完，但是大家都要花七天的狀況下，你只收一天的費用，這也太划算了吧……）

基本薪資

再來談談這個問題，假設你現在剛上大學，只上過幾門課。但是卻有人找你接案，但是你的技術價值算出來幾乎等於零，沒有生活壓力，難道要報價免費嗎？

以目前規定的最低薪資來說，每小時一百一十五元。即使你再沒有技術，也該要有這個程度的薪水，畢竟你還是被雇用了（不過這是借用時薪計算來輔助確認報價的適當性）

那麼，假設你判斷你的技術價值大於一般大學畢業生。那麼現在一畢業基本上就可以拿到兩萬二的薪水，那你的報價在花費一個月時間的案子，也就不該低於這個價位。（一旦低於了，你的技術價值也同時被當作低於這個價格）

不過這就是輔助用，實際上用這種方式計算不管什麼情況都會比你實際拿到的還低。

風險與信用

這差不多是最後的項目了，不過這部分也是在「接案」的時候必須要有的重要觀念。

前面一個段落有提到「自己完全沒有技術價值」的情況，如果你選擇接案，那就等於你在浪費你的「信用」這很重要。自己的技術能否順利完成任務，是決定是否接案的「先決條件」

當你順利完成一定數量的案子後，你的信用也會不斷的提升。而這個將會成為之後你談判中的重要籌碼，假設你每次都順利完成而且品質優良，那麼客戶在「願意提供」的價格範圍上，也會相對提高，甚至不需要談判就給予你理想的價格。

這就是「信用」的價值，雖然沒有實質的加成。但是卻會讓你接案的過程順利，而且雙方更容易達成共識。

（但是要注意，這比累積還更容易失去，很有可能失敗一次就完全歸零。）

至於風險，我們要搞清楚「接案」的風險是比「正職」員工還高的，因為你永遠不知道這個案子結案後是否還會有新的案子。在這種可能被「斷糧」的前提下，收費本身就會「相對」的提高。

（至於外包為什麼比較便宜，我也不知道，就這個層面來看應該不會比較省才對啊～～）

我都還沒到學校就收到網友的建議，這邊來補充一下：

接案本身還會有收不到款的風險，所以也要注意簽約（至少也要寫上權力跟義務，還有著作權等問題）

而收款方式則是分階段的訂金，先收一到三成，然後中間驗收收七成，最後驗收才把最後的三成收下。

接案時選擇也要注意，像是要避免金流這類高風險的類型，以免不小心犯錯時以學身身份無法負擔賠償問題。

– Muka、TonyQ

售後服務（2/26 追加）

報價上也需考慮到後續維護問題。

以及後續維護時所需的維護費用。

– henryjuan

關於維護問題，如果是製作比較簡單的像是 Plugin、Theme 之類的，可以依照修改次數修費。
至於如果是一套完整的系統，可以考慮持續收取維護費的模式（如果有在運作，而且需要負擔維護責任）
不過我個人還是不建議一開始就接手太大的案子，建議從小工具等開始做起。
（如果是一套完整的系統，則會建議有和同學合作的經驗後，和幾個可以信任的朋友協力完成會比較好。）

那麼，就先拿學弟當範例計算一下。

學弟最後給我的技術價值是 2.8/5（分五級）至少我認為 0~1 都不太可能出現，畢竟當新手很難，反而是入門完之後突破重重難關到最高級的時候最花時間。
（中間我有提問和要一些程式碼來看，也在我接受的範圍內）

那麼假設他認識的人之中，最高的等級是月薪五萬左右。
那麼以這個方式計算，請他工作每小時要花一百七十五元。
他這次的案子只有三天，但是它實際工作得花上五天左右。

於是得到這樣的公式： $175 \text{ (成本 + 技術價值)} \times 8 \text{ (每日工作時間)} \times 5 \text{ (天數)} = 7000$
不過三天要做完五天的事情，勢必會壓縮到休息時間，可能影響之後的工作。

因此建議 1.5 來減少風險，所以應該會得到 $7000 \times 1.5 = 10,500$ 的最終報價。
至於對他來說是否太高，或者太低，還有真的合不合理之類的，那就要看最後與業主的談判結果。

至於報價的是你自己，你有權決定該怎麼調整（例如業主是老客戶，付費也很乾脆，所以不加收急件的費用之類的）

那麼，我們來整理一下重點。

- 報價是一種談判，自己要斟酌出牌時機
- 不要忘記成本
- 技術價值是真正賺錢的部分
- 薪水的行情可以用來輔助計算
- 注意信用的經營
- 適當的調整價格

不過還有很多技巧沒有寫到（像是價格算出來後，其實會偏高，因為對方通常會先以壓低價格為目標，這是一種防禦措施）

當然，我想文章中也有不妥的部分，如果有前輩之類的可以給建議，也希望能夠提供一下讓大家知道該怎麼更加準確地拿捏。

另外，技術價值其實很難算，不過如果你確實在某個領域認真過，從你認識的人、接觸過的事物，應該是比較能夠抓到自己的位置（但是絕對不能只活在某一群人的小圈圈中，除非那已經是最頂尖的社群了……）

相關資料：

[Less Everything](#) - 裡面有一本書討論接案
[Planscope.io/](#)
– XDite



留言

0 Comments 弦而時習之**1 Login** ▾ **Recommend**  **Share**

從最好的優先排列 ▾



Start the discussion...

Be the first to comment.

ALSO ON 弦而時習之

Rails Girls 4 活動感想

1 comment • 3年前 •



WM — 那不是特殊能力，是你興趣廣泛...

mruby on Web

3 comments • 2年前 •



WM — 他們並沒有把這個實驗的原始碼放出來。

Container Summit 2014：未來的雲端




2 comments • 2年前 •



William Yeh — 是 Reactive System。詳細內涵請搜尋 Reactive Manifesto 宣言。BTW，這和最近很紅的 React.js 無關。

RubyConf TW 2014 會後心得

6 comments • 3年前 •

良葛格 — 【簡報檔與講稿】Understanding Typing. Understanding Ruby. <http://www.codedata.com.tw/...> **Subscribe**  加入 **Disqus** 到你的網站 **Add Disqus Add**  隱私©2013 [蒼時弦也](#). All right reversed.