



Next Level Freelance

By Jevin Maltais



CREATING AN

ACTIONABLE

ROADMAP



CHAPTER 6



Phew! If you've done all those exercises (thanks for

hanging in there), you know where you want to go, and where you are now. You should now have a critical view of:

- Your current important numbers in your business
- An understanding of your previous clients and your ideal clients going forward
- Your expertise through the eyes of your clients

In this chapter, you're going to take that information, combine it with your freelancing goals going forward and come up with a plan for this next stage of your business.



EXERCISE 5: EXTRACTING YOUR CURRENT EXPERTISE

Maybe you want to make the same amount of money but work half the billable and non-billable hours. Maybe you are willing to work the same amount of time, but want to make 50% more in the next 12 months. You will have to do some calculations to figure out how much higher your rate must be. Maybe you want to live in Panama for a period of time, but you don't know what the cost of living there is. You might have to do some research to give you an idea of what those numbers need to be.

In light of your foreseeable desires going forward, pull out "Exercise 6: The Important Numbers" from earlier. This time, fill in the "Future" column on the right side with real numbers.

DEVELOP YOUR FREELANCE BUSINESS OR BUILD PRODUCTS?

It is likely that for you, the new "hard numbers" in the "future" column are vastly different from the "current" column. Perhaps depressingly so... AND THAT'S FINE! In the same way that calculating your budget made you aware of realities and you at least had a concrete framework to work from, now you know where you need to go to achieve your dreams in your career.

Now we'll spend some time looking at some different ways to achieve those hard numbers. such as:

- The strategies that are best for you
- Case studies from others that helped them in their businesses
- Pointers to get you started
- Resources that have tactics to get you there

Sometimes, the different directions you can take your business is overwhelming. You already established your core priorities. So with that in mind, let's look at the different options for right now.

If you're not booked solid, you have even more options than someone who IS booked solid. Namely because you have time on your hands. The question you should be asking yourself is: Should I work on getting more clients or build products?

Priority #1 would likely be to stabilize your income, right? If that's the case, what's the best way to do so based on where you are right now? More than likely, having a steady stream of clients asking you for work to be completed is probably going to bring you a more predictable, steady stream of income. A software product can take a long time to develop real income, an info product offers only a short term cash injection, and raising your rates or hiring more people isn't really viable if you aren't

fully booked yourself yet. Once your income is stable, then you can spend the time doing those other things.

Of course all situations are different, so if you feel your income is stable enough then by all means, try one of the strategies listed in the last chapter.

From here on out, I will lay out some ideas that have helped me that I picked up from others and successfully used. You may want to jot down things that resonate with you and incorporate these strategies into your business based on the hard numbers you want to realize in the exercise you completed earlier in this chapter.

DEVELOPING YOUR CONSULTING BUSINESS

Let's get tactical and dive into strategies to help you on your consulting

Positioning

There are a ton of developers out there who can sling code and build stuff at a mind blowingly low rate. Don't believe me?

1. Visit oDesk.com

2. Search for your language of choice:

- For ".NET", 63,741 come up and 40,384 are under \$15/hr
- For "PHP", 104,421 are listed and 68,448 cost under \$15/hr
- For "Rails, 6,480 are listed and 2,186 cost under \$15/hr

How the heck are you going to compete with that? Most clients don't understand the difference between a high and low quality developer. They might grill you with questions like:

- Why should I pay \$150/hr vs. \$15/hr for the same code?

- Why would you spend extra time writing automated tests?
That costs me extra money!
- Why do you want to understand more about my business problems? The cheaper developer doesn't ask these extra questions.
- This less expensive developer says they can build the application in Java in the same amount of time as the expensive Ruby on Rails. Why shouldn't I go with him?

If you're reading this book, you probably have a certain amount of self-respect and a desire to develop software in the best way you know how. For the most part, non-technical clients who ask these kinds of questions can come over to your side if you explain to them why some approaches are better than others. In response to the above questions, here is how you might answer to demonstrate the difference between you and a less valuable resource:

- There has been research to suggest that there can be a 10x difference in productivity between 2 programmers. In other words, a feature that takes a developer who charges \$150/hr one hour to do, will also cost the client \$150 if a \$15/hr developer worked on it (assuming of course that they are both charging based on their relative ability levels). So at times, the price is justified.
- Automated tests have been shown to greatly reduce bugs and the time taken to modify code bases on anything but the smallest sized programs. While it does take extra time, the payback is well worth it as the project grows. In my opinion, not writing tests is a lazy and short sighted approach.
- When a developer is educated on the background of the business problem, this makes it far easier for them to incorporate technical design decisions that make sense. If they don't understand the purpose, they will simply do as per the specification and leave it as that.
- In most cases, the time it takes to setup a production Java system is far longer than Ruby on Rails. Rails is designed with rapid development methodologies in mind and is superior if you want to iterate quickly. For almost any web application, Rails is far better than Java.

But what is really underlying these questions that a potential client might ask you is one overarching question: "Can this

software developer get the project done on time and on budget?" Read that again and burn it into your mind. When they are comparing two developers to each other, that's really the only thing they are asking themselves.

Developing Trust

If the client ultimately cares if the project gets completed, how can you convince them that you can get it done? If they explain how they want things to work and you want to convince them you understand exactly what they want, why not write out some high level features and build some clickable mockups with Balsamiq and post them online for them to click through? I recently wrote the features and built mockups for a potential client and when I sent them off to her she exclaimed, "Wow, this is exactly what we had envisioned." Needless to say, the other company she was considering didn't do this so it really showed that I understood what she wanted to build. It took 3 hours to prepare, but it made the difference between landing and not landing the job.

**Buy the book for the rest of this
chapter at: nextlevelfreelance.com**