

borntoraisedesign

Born To Raise Design

Official Contract

Table of Contents

<u>Introduction</u>	1
<u>Scope</u>	2
<u>1.01 About the Project Scope</u>	2
<u>1.02 My Thoughts on Scope</u>	2
<u>1.03 Making Changes to the Scope</u>	2
<u>Payment</u>	3
<u>2.01 How Payment Works</u>	3
<u>2.02 Refunds and Payment Edge-Cases</u>	3
<u>Open Source</u>	4
<u>3.01 What is Meant by Open Source</u>	4
<u>3.02 What that Means for Your Project</u>	4

Introduction

Hi! I'm excited to be doing business together! If I weren't so adamantly against wasting champagne, I'd say let's smash a bottle to commemorate the maiden voyage of our working together. I'm a good developer; my work is professional, my rates are reasonable, and my demeanor is friendly. I am certain you will be a good client as well.

I can read and write a couple different languages, but legalese is not one of them. I'm sure you are in the same boat. And lawyers are really expensive. So my purpose behind writing this contract is so that you, the client, and I, the developer, can communicate our expectations and requirements in plain language, so that we're crystal clear about who gets and does what when.

It's a bit like the "sign it in blood" sentiment only—yeah, put the knife down—ink will do just fine.

Scope

1.01 About the Project Scope

Before I begin development on your project, I'll want to, over the course over several conversations, develop a scope for the project. If you are unfamiliar, a project scope is simply a document that details exactly the work I will be doing for you.

It's kind of like a blueprint for a house: you don't want to build a house off the cuff, making decisions as you go. You will likely end up with a house that is structurally unsound, or worse. The same applies here: before we begin the project, I need to know exactly what I'm building so that I can plan it out properly.

The project scope (and adherence to it) protect both of us. It is the document that we can look to if there is any dispute over whether a certain feature is or isn't in the budget.

1.02 My Thoughts on Scope

I like to adhere to a more Agile definition of project scope. That is to say, I do not fear changes to the scope. In fact, I expect that, as we start building, your ideas may change; that's sort of how people work. And rather than choosing to ignore the fact that this happens often in software development, I prefer to embrace the fact and be upfront about it. So you, the client, must understand that changes to the scope can (and usually will) affect the timeline and budget of the project.

If you would like to change something in the scope of the project, just let me know and I will (usually) gladly accommodate.

1.03 Making Changes to the Scope

Easy as pie. Simply contact me and we can discuss it. Please do keep in mind, as I mentioned above, that scope changes will/may affect the budget or timeline of your project.

Payment

2.01 How Payment Works

Once we reach an agreeable scope and cost thereof, the next issue to discuss is payment. It's pretty simple, actually, and my reasoning behind this is actually both to minimize the risk for myself, the developer, and to give you, the client, some peace of mind.

You, the client, will pay me at three (3) different points. Firstly, I'll need one-third of the total cost upfront. Then, I'll need the second third after I show you the work I've done and you've approved it. Finally, I'll need the last third when I actually push the work I've done to your production server (i.e. "made it live"). I do my best to be accommodating, but please understand that I am very serious about payment (as I'm sure you are too).

Unless we arrange otherwise, I'd like to be paid by check, or cash if you're local and want to do it that way.

2.02 Refunds and Payment Edge-Cases

Both parties (me, the developer, and you, the client) may, at any point in the term of the project, withdraw our involvement for any reason. This may occur, for instance, if the scope of the project changes to the degree that I am no longer confident in my ability to do the work in a way that makes sense for me financially. And this rare event will undoubtedly happen after several conversations and an unwillingness on the part of the client to accommodate the financial consequences of a change in scope. In short, be nice to me and I'll happily do the job you pay me to do.

If this happens, I'm confident that we can work out some sort of deal by which I get paid for a portion of the work completed, and you receive the portion of my work that you paid for. Or, perhaps, we both just decide to call it even.

I will under no circumstances release the source code of my work to you, the client, without receiving some sort of compensation. To me, this just makes sense; you don't get the goods without payment. Neither of us is unreasonable, so, again, be nice to me and I'll do everything I can to make sure you are a happy client.

There is one exception, however: time billed as "Project Management" (i.e. client meetings and phone calls, preparing documents, communicating via online tools) is non-refundable. If we both agree to part ways, for whatever reason, you must still pay me for project management time. Sorry, but I've been burned in the past by clients who spend a lot of time planning only to pull the plug at the last minute. You know the type.

Open Source

3.01 What is Meant by Open Source

The discussion about Open Source may have little bearing on your life if you are not a developer; it tends to be a bit technical, so I'll try to keep the language plain and the points relevant to our work together.

I am very committed to the ideals of the Open Source movement. By working with me, you are already receiving, for no additional charge, the benefits of other developers' open source efforts: software like Drupal and all of its contributed modules, and so on. I believe strongly in giving back to the community, and the way that most developers are able to do that is by abstracting work that clients pay them to do so that the work is useful to other people.

3.02 What that Means for Your Project

Alright, the previous section was a bit vague, but here's what it comes down to. While you will own the source code for whatever I build for you, I reserve the right to abstract it out and release it to the community. Don't worry: I won't be giving out any data that is sensitive to your company/business (passwords, payment information, company secrets, etc.). Rather, I'll be taking the functionality of the thing I built for you and, if possible, making it more broad so that it's not so specific to your business model or industry.

If I am able to and choose to do this, you can count on three things. Firstly, I will notify you that I am doing so; secondly, I will give you attribution in some capacity for having financed the project in part; and thirdly, I will, unless we arrange otherwise, do all abstracting and releasing on my own time. Open source is something I believe in and I do not expect you to foot the bill if that's not your thing.

Most likely, this will not affect you whatsoever. But if you have any questions about open source, your project, or my intentions, feel free to ask.