

Subcontracting a developer: what to agree first.

For an agency bringing in a white-label web developer. The decisions to make before the scope, and the lines to put in the contract, so nobody is working it out halfway through a client project.

This is general information, not legal advice. A solicitor should see the contract itself, especially the non-solicit and the IP clause.

CHOOSE THE MODEL, BEFORE THE SCOPE

- ☐ **Behind the scenes: the developer never appears**
Simplest for you. Every question takes two hops, so it suits defined, self-contained work rather than discovery.
- ☐ **Fronted under your brand: they join client calls as part of your team**
Fast, and worth more than a rate difference on a project with real uncertainty. It needs trust, in writing.
- ☐ **Named subcontractor: the client knows who the specialist is**
Good for migrations and technical work, where the client wants the risk explained by the person who assessed it.
- ☐ **The model is written down, and everyone involved knows which one it is**
Most of the friction in subcontracting comes from two people assuming different models.

WHO ANSWERS THE CLIENT

- ☐ **One named person answers client questions**
Otherwise the developer waits for the account manager and the account manager waits for the developer.
- ☐ **A time limit on answers, even indirect ones**
"Questions go to [name], who answers directly, or briefs [name] within a working day" is the whole rule.
- ☐ **Which email address and which tools the developer uses**
Your inbox, your project tool, or theirs. Decide before the first client call, not during it.
- ☐ **What happens when the developer is away**
One person can only take on so much. Knowing the limit up front beats finding it in week three.

MONEY

- ☐ **The rate, and what counts as an evening or weekend**
Agree the multiplier before anyone needs it, not on the Friday it comes up.
- ☐ **Fixed scope or day rate, and which parts are which**
A fixed price needs a scope tight enough to price. Say which parts of the job are not there yet.

☐ **Payment terms, and what happens if the end client pays you late**
Clients sometimes pay late. Agreeing this up front keeps everyone on good terms when they do.

☐ **How a change gets quoted, and who approves it**
The alternative is a change being done as a favour, then argued about at the invoice.

IN THE CONTRACT

☐ **IP ownership passes on final payment, stated plainly**
The client's contract with you usually promises this, so your contract with the developer has to match it.

☐ **A mutual non-solicit, for a reasonable period**
Reasonable to ask for, and a good developer will not mind. Mutual is fairer and easier to sign.

☐ **Confidentiality that covers what they see of your client's business**
They will see more of it than you expect: analytics, inboxes, customer data.

☐ **Insurance, if your client's contract requires it**
Check what level your own client contract demands before you promise it.

☐ **How either side ends it, and with what notice**
Rare, but much cheaper to write down while everyone is getting on.

THE END OF THE PROJECT

☐ **Handover is a deliverable with a date, not a favour**
If it is not in the scope it slips, and support you never priced for quietly carries on.

☐ **Who holds the credentials, and who the client rings in March**
Name the person and the route, or it lands on whoever answers first.

☐ **The code, the accounts and the documentation are transferred**
Transferred to the client or to you, deliberately, not left in the developer's accounts.

☐ **Case study and credit: what each side may say publicly**
White-label work carries someone else's name, so agree in advance what can be shown and what stays private.

Take it, change it, use it with anyone you subcontract. No need to credit me. adamroe.co.uk/blog/white-label-web-developer