HIRING A DEVELOPER

THE ULTIMATE GUIDE

BY

Carter Thomas
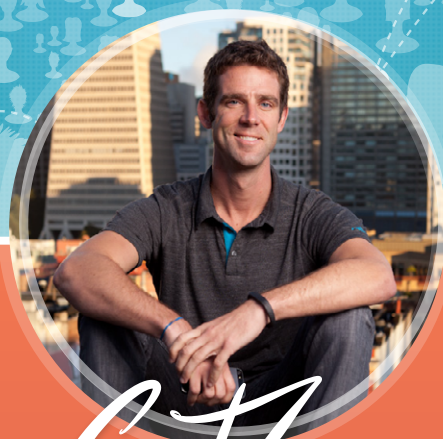
Welcome

Where it all Begins

The Intial Stage

Pre-Job Posting

Where to Start

Hiring a Developer

Next Steps

Carter

- Raised in New York and Maine, USA

- First app released October 19 ,2011

- Built and sold over 900 apps since then

- Consultant to some of the biggest apps and app companies in the world

Overview

Hello and thank you so much for joining! Hopefully by this time you've already got your app idea planned out, your wireframe ready and you're ready to find yourself the right person to do the job. If not you may want to consider reading Turning Apps into Winning Products

This ebook was written to show you specifically how to take on one of the most daunting tasks in this business - hiring a developer. The #1 goal is to show you that this is not as hard as you think and that there is a clear, systematic way to do this the right way.

About This Guide

This guide will lay out instructions, scripts, and tricks that you can learn from to find the best talent you can. Everything will be presented in a way that is easy to understand so that you walk away from this prepared and excited for your next move!

Before we dive in, I want to be very clear about something that I say at the beginning of all my ebooks. This is not a set-it-and-forget-it guide book. This will not magically land the developer of your dreams into your lap. Sorry :(

Nothing in here will ever refer to the idea of "getting rich quick" and I will rarely talk about hacks in the app store. I will **NEVER** talk about cheating.

The reason you've decided to read this book is because you're ready to take steps in your journey. In some capacity, you're ready to move forward. Not only with apps, but in life. **THAT** is awesome.

With that attitude comes great opportunity. When you use education and tools to fuel that ambition, amazing results happen. That's what this ebook is all about.

Have you signed the Bluecloud Ebook contract?

If yes, let's get going.

"This will not magically land the developer of your dreams... this is a guide... a tool."

1

WHERE IT ALL BEGINS

You Gotta Build Something!

When you look at any success in the app store, they all have one thing in common - apps. That may seem like a very obvious conclusion, but it's important to remember. You need to have **SOMETHING** to be in the app store. You can't just put an idea in there.

If you're like me, you can't program. I'm even good at web programming (building websites, etc), but the stuff that these guys can do for mobile apps is incredible. I remember watching a developer once who could code so fast it was like watching the Matrix.

Luckily, the app world has grown so much that developers are starting to come out of the woodworks to get jobs. When I first started, my biggest challenge was finding a developer at all. What's great about this for you is your biggest challenge is finding the **RIGHT** developer among many.

It's a buyer's market and that's a good position to be in for you, as the buyer, to be in.

Get Prepared

What we're going to talk about in this ebook is a clear plan on how to get started, sort through lots of potential developers, and hire one that will make your company take off.

Remember - this is a **PROCESS**, not an event. Hiring a developer is just like hiring anyone else - it can take time and have ups and downs. Luckily, I've already gone through so much of that, I can write ebooks like this for you to save you tons of time and headaches.

Let's start at the beginning.

Assuming you have your:
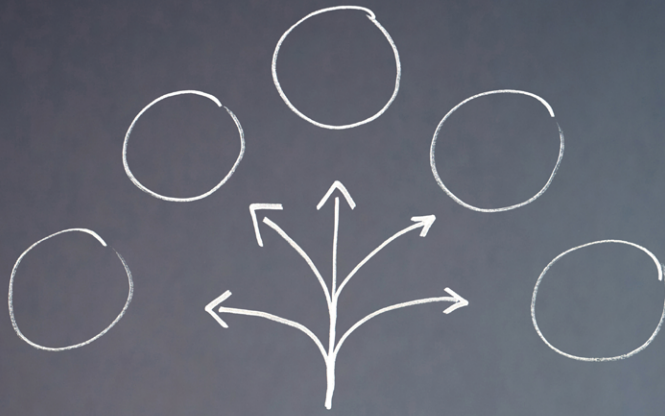
- ☐ idea
- ☐ wireframe
- ☐ goals set

...we're ready to make moves.

If you don't, consider reading the other **ebooks** about those very topics.

"this is a process... it can take time and have ups and downs."

2

THE INTIAL STAGE

Setting Yourself Up For Success

Before you begin your journey towards global app domination, it's important to do a little bit of planning (disappointing, I know). By doing this, you're going to save so much time and money down the road.

When you hire a programmer (or developer - they are the same thing, just different ways of describing them), you are looking for them to build you something. This can be anything - from scratch, updating an existing app, copying someone else's app, etc. I'll talk more about this in a minute.

What I want to reiterate is the idea that hiring a developer only one part of a much larger vision. You need to developer to help build you something, but it will not end here. That's why we're going to talk about planning and goals - to make sure you're on point every step of the way starting **NOW**.

Like any project, you need to be clear about your goals. In the case of hiring a programmer, ask yourself: *"What do I want this programmer to do?"*

Here are some examples of what type of work you will want the programmer to perform:

- **Build something from scratch.** You have an idea. You can't find it anywhere else so you want someone to turn that into a live app. This is usually a unique project that you have wireframed out and has never been done before. The relationship will be closer because you'll want to make sure details are accounted for, but you'll also be able to create something that's never been done.

- **Improve something already in the store.** You see an app in the store, but know you can make it better. You send it over as an example with instructions on how to make it better and unique (emulation). This is a powerful model and one that is used to help defray costs.

  This is not stealing by any means, it's about using an example to help the developer answer 80% of the questions they probably have.

  For example, if you want to build an alarm clock app, instead of going through the whole process of showing the exactly how to make the app, it's a lot more effective to find an app in the store that you like, then show the developer how you want to improve it ("we need to add a button that let's you choose how long your snooze is")

? ASK YOURSELF

•••

What type of work?

- Build from scratch

- Improve

- Customize

- Tweaks

- Bug Testing

- **Customize a template.** This is when you purchase source code and have a developer modify it accordingly. You may notice these types of source codes on the Bluecloud website. This is done in virtually every industry out there, including apps.

  Have you ever been to Nike's shoe customization **website**? You take your favorite shoe style, then you go through and customize it how you want. Blue color here, red shoelaces, white swoosh. Then you get them shipped to you.

  This is the same process you can do for apps. Change background to night time, turn the mouse into a penguin, make enemies dragons instead of eagles.

  Similarly, it doesn't have to be just graphics. You can have the developer take a template and add some functions to it.

- **Tweak a few pieces of code.** Maybe you already have an app but you want someone new to *"polish it up."* This is similar to the template customization request, but it's when you already have your own code.

  This happens when you've purchased an app from someone or if you built an app a few years ago and want to get back into it. Pretty much any situation where you have an app and want to update it in some way.

? ASK YOURSELF

...

What type of work?

- Build from scratch
- Improve
- Customize
- Tweaks
- Bug Testing

- **Bug testing.** Need a second opinion? This is done with larger apps and when people can focus on one or two apps as opposed to their portfolio. They might have a team that did a great job developing the app, but they want someone else to take a look as well.

  When you're working on a project for a few weeks, it's easy to miss details. This is true for you and for your developer. Sometimes it's a smart move to hire a second set of eyes to run through it.

  NOTE: this is done once you have some high level apps and have the time to spend doing this. Otherwise, you probably don't need to hire a separate team to do this.

Of course, there are other random jobs you can hire a programmer for, but this list covers the big ones. Do you have one in mind? Which is most appealing to you?

Take a minute and write down the kind of job you're most likely to hire a programmer for.

x _____

Done? Awesome.

Now let's talk about the next step.

3

PRE-JOB POSTING

How to Prepare for a Successful Hire

We've walked through a few different options that you have when hiring a developer. Now we're going to talk about the pieces that most how-to books never touch on (but are incredibly important!!).

These are learnings that came from hiring over 20+ development teams in the last few years. In fact, I showed this list to a handful of other app entrepreneurs (appreneurs?) and they all agreed that this is huge for anyone who's getting started.

"I don't mean for your developer – I mean for your entire project."

Here is a list of the best ways to prepare for hiring a programmer before spending any money:

- **Set aside a budget.** I don't just mean for your developer - I mean for your entire project. Start to finish. Not only does this help you be disciplined with spending, you start to realize that money you spend here is money taken away somewhere else. If you treat each piece of your app project separately (development, design, marketing, etc) then it's much harder to cut costs.

  You're going to get a lot of different quotes and be tempted to sink more money into the project if you see something that looks promising. **DON'T**. Remember - this is a learning experience, **NOT** a transaction.

  One of the most common responses I get from blog reader is *"I wish I had spent less at the beginning and focused more on learning how this all works."* Spending less now will give you more to spend later on when you know **EXACTLY** what to spend it on.

- **Have a timeline in mind.** Be clear about how long you want this to take and be prepared to stick to it. Anyone in the corporate world knows about *"scope creep"* which is a fancy way of saying going over a deadline. Unfortunately, this happens a lot.

  The first thing you can do is to clearly define your project scope through wireframes, details, and communication.

  The second thing you can do is to set up milestones that force certain projects to be delivered at certain times or payment is not released (more on this later).

- **Ask around for references.** When you're starting the project, spend a few hours reaching out to people you know and asking if they have anyone they would recommend. You can send emails or do a phone call to friends/family and see if anyone has gone through this process already and can refer you or help you out.

  Other options are to go on social networks and forums to ask around. One of my favorites is to use a service like **Clarity** to find experts who can refer you to developers.

  Sometimes this will lead you to the perfect match in a fraction of the time you would have spent otherwise.

📖 **RESOURCE**

•••

Use a service like Clarity to find expert referals.

www.clarity.fm

- **Finalize the scope of work.** What's included? It's easy to forget to include things like icon, screenshots, or uploading to the store. You want everything to be clear.

  As a point of reference, the big things to ask about are:

  ☐ coding

  ☐ design

  ☐ icon

  ☐ screenshots

  ☐ uploading to app store

  ☐ updates

  ☐ installing ad networks

  You'll add or subtract based on your project, but at least you'll have the discussion early on.

- **Don't be afraid to say no!** No matter how strong you think you are, there are people that can get under your skin. Sometimes they make you feel guilty, *"If I don't get this job, my family will not eat"*, sometimes they offer you something that seems like an amazing offer.

   Do not be afraid to walk away. This is one of the biggest lessons I learned. Deep down I had a gut feeling that it wasn't right, but for some reason I didn't put my foot down.

   Especially at the beginning, if it's not what you want, end it.

- **Slow to hire. Fast to fire.** One of my favorite phrases. It's easy to *"give everyone a chance"* when hiring, but you'll burn through a lot of money and time that way. Be patient and only hire if everything looks right on.

   By the same token, if you hire someone and they're not working up to the quality you want, get rid of them.

If you have prepared all of this, you're at the starting line. We'll talk more about this as we go, but now it's time to be operational.

4

WHERE TO START

Outsourcing Websites

In the world of apps and app development, *almost 90% of indie developers go through outsourcing websites.* Unless you have a special circumstance like your friend is a programmer or you know someone personally, you're going to fall into this camp too.

Many of the agencies or domestic firms in the USA, while great, are hugely expensive and reserved for businesses and brands.

Programmers in the USA (and most other Tier 1 countries like UK, Australia, and Canada) are usually good, but expensive as well. Later on and for bigger projects, they might be the right fit, but for what you're most likely doing, you want to keep costs down.

Remember - the goal is to **LEARN first, INVEST later**.

Don't spend money until you know exactly what to spend it on based on its potential return.

That being said, there are a few outsourcing websites that are worth talking about, specifically Elance and Odesk.

It might be worth saying somewhere that if you're reading this book I'm assuming that either you don't personally know a developer that can do the job for you, or that they are too expensive right now.

Just so that you know, you may move on to your own networks and connections to get programmers, but to start, this is where it all happens. It's a powerful exercise for learning the entire process.

These websites aggregate people from all over the world who have services and skills to offer. Not just development, but copywriting, virtual assistance, even accounting! They create a profile and then bid on jobs created by people like you. It's a wonderful system and has done great things for the global economy.

📖 **RESOURCE**

•••

www.elance.com

www.odesk.com

I'm going to describe each of these sites and then show you exactly how to create a winning job posting that will put you on a path to successful hiring.

1. **Elance** I tend to use this one more, but that's only because it's where I started early on. Both are great (in fact, they merged a while back).

   The best thing for you to do is to sign up via the link above and get your profile setup. They have a great wizard to get you started and show you around.

   You'll be able to browse through different contractors/developers and see what sort of talent is out there. You'll see how prices differ based on geographic location and quality. It's actually pretty amazing to see how many people are available to create anything you want!

   One other great exercise is to browse currently listed jobs. This is a running list of jobs that were created by people like you looking for work. I created a **link** so that you can see what I'm talking about for mobile application jobs, just click below to see it (You have to have an account and sign up to view).

📢 **ACTION**

•••

· Sign up

· Setup Profile

· Browse
  Developers

· Explore Current
  Job Listings

2. **Odesk** Same as above, a standard outsourcing site. Sign up for free and take a quick walkthrough their interface.

   In the meantime, you can browse through some mobile app developers **here**:

   Odesk also does a great job walking you through the process so when you're ready to post your first job, it's easy!

ODESK.COM

•••

• One of the most popular freelancing websites

• Use this to find people to help with development, design, marketing, and virtual assistance

• A great place to find job ideas and learn hiring skills

Security

One important piece to note is the security these sites have. Both of them take that very seriously, specifically about your payments. Almost everything is done through escrow so that the work you receive can be verified before you release any payments.

In other words, they have safeguards in place to protect you and the developer you choose to work with. This is great for building trust early on because everyone can feel secure that they are going to get what they expect from the job.

You will do all billing (and often communication) straight through their platform which gives you control and peace of mind.
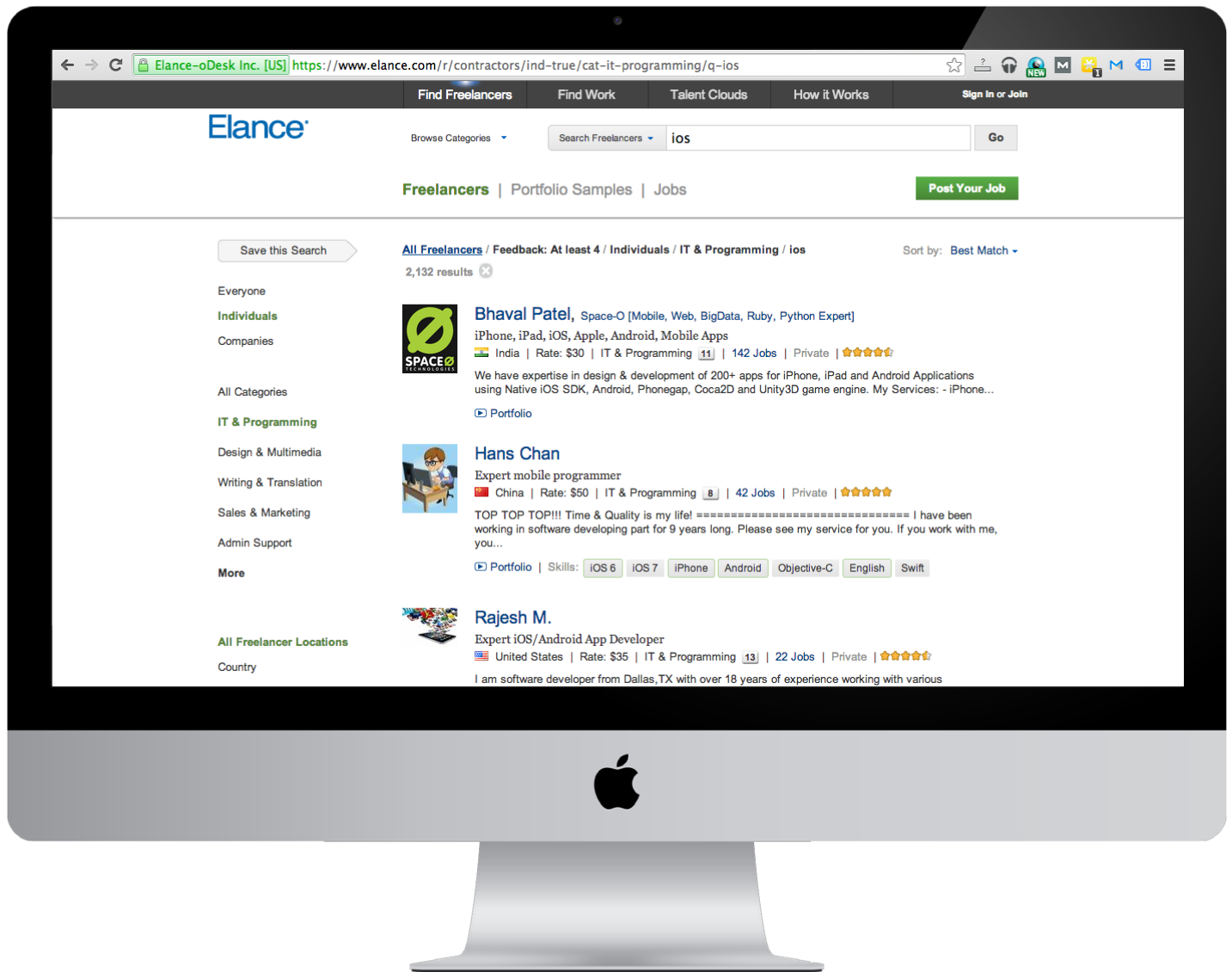
Now, let's get something out there :)

5

HIRING A DEVELOPER

Writing The Job Description

Writing a job description can do wonders for your hiring process. For a long time I didn't spend too much time worrying about it - I just figured I could narrow it down once I saw the developers start applying for the job.

What I realized, however, is that developers want to work with great people too. If they think you're high caliber, they are more willing to apply for you job. Great effort attracts great people.

When you open your job posting screen, it may be a little overwhelming. Someone once told me the best and worst projects are the ones where the teacher hands you a blank canvas and says *"make something."* I agree with that. But - we are hardly in a position where you have to come up with everything.

This has been done so many times, now you can follow best practices so that you hit the ground with your best foot forward.

When you are crafting your job posting, here are the best points to focus on to get great candidates:

- **Start with a a clear definition of what you're looking for.** Developers want to know exactly what you want so that they can deliver it to you. This does not have to be complicated, but there needs to be a place where the programmer can see what the project will be about without any question.

  For example, a clear definition would be *"I am looking to create a photography app that is very similar to XYZ app (link) but will add the following features (list). I am looking to hire someone to build, design, and upload this to the store. Please message me for details."*

  A non-clear definition would be: *"I'm looking to create a photography app that has awesome filters like instagram that I can share with friends."*

  You see my point.

  NOTE: Don't worry about sharing your conceptual idea. You can rest assured that you're probably not the first person that's ever thought of it but you, unlike all the others have already done the hard work designing and developing it to the point where the idea is already a plan. People are going to appreciate the detail, but they aren't going to want to go and do all the hard work you've already done. They're as intimidated by that part of the process as you were before you read the first ebook and realized just how easy it could be.

POINTS OF FOCUS

●●●

- Clear Definition

- Hire for Skills & Deliverables

- Use Triggers

- Previous Work

- References

- **List off 5-10 specifics you are looking for.** This can be a bullet list, but it's always good to list off anything so that they know where you're coming from.

  This can include things like features, examples, skills, additions, or even language requirements. The important part is making it highly specific.

  This is going to help not only create a good picture for them, it's going to help get a much better cost because there is little gray area.

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

10. _____

- **Hire skills along with deliverables.** This is a big one for me. Often people will create a list of functions and deliverables they want, but they fail to list what types of people they are looking for. Remember that you are hiring **someONE** and not **someTHING** to do this. The human element is important.

  Examples of deliverables are: icon, 9 filters in the app, export to instagram

  Examples of skills are: fluent English, detail oriented, works weekends

- **Use triggers that weed out programmers looking to blanket reply everything.** Unfortunately there are some people on these sites that will just take a templated response and paste it into as many job postings as they can to get more work.

  One way to get around this is to add trigger words deep into your job posting.

  Example: (in the 4th paragraph) Please reply with the word **LUMBERJACK** at the top of your response so I know you read this.
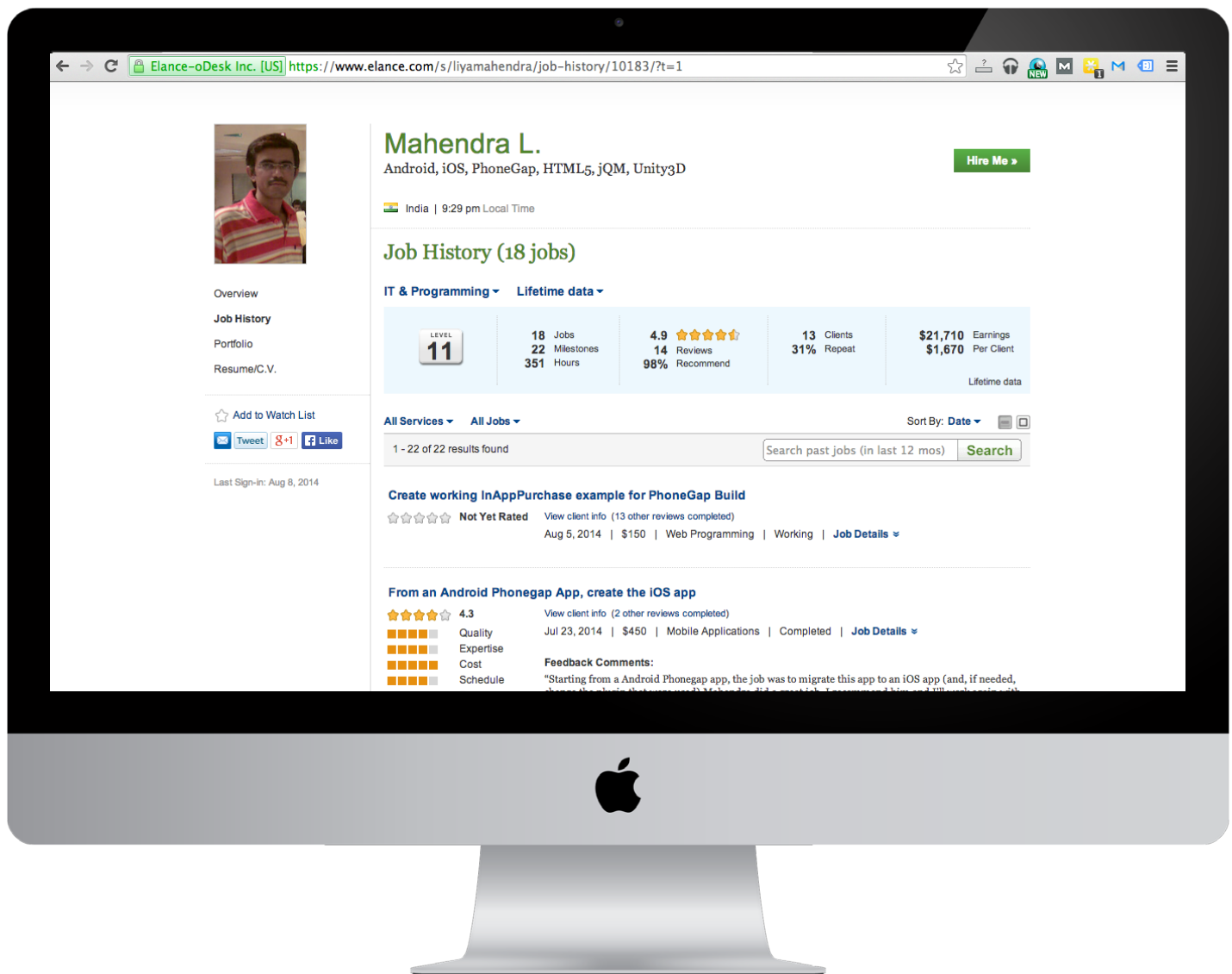
  If anyone doesn't put that in, automatic delete.

  This is just like on a dating site (if you've ever been on one). Personal is always better!

- **Ask for previous work.** You'll usually be able to see this in their Portfolio section, but it can be good to ask as well. I've worked with a lot of developers who have portfolios they don't keep on Elance or Odesk that have been good to see.

  This will give you a sense of their style and the projects they've worked on. This is important because it can work better if you hire someone who's familiar with your category (Game developer probably doesn't know much about Weather apps).

- **Ask for references.** This is not mandatory and I only usually do this if I'm going to spend a lot of money on something, but it can be a good way to double check.

  It also shows you how open the developer is to share his previous clients - if they weren't proud of their work, this could be a problem for them, which is a red flag.

  One big advantage of using a system like Elance is that it has it's own built in reference system, as seen below:

I wanted to create something you can use immediately. This is a variation on some of the best performing copy I've used to attract top talent.

Feel free to use this and replace the relevant pieces for your own job.

TITLE

Looking for Rockstar Developer to Create **[CATEGORY]** *App*

DESCRIPTION

Hi!

I am looking to build a **[FILL IN SIZE]** *sized* **[FILL IN CATEGORY]** *app for the* **[FILL IN WHICH PLATFORM]** *app store. This app will be very similar to this app* **[PASTE LINK FOR APP IN STORE]** *with a few improvements which I will explain via private message. For your proposal, the biggest improvements will be* **[FILL IN IMPROVEMENT #1]** *and* **[FILL IN IMPROVEMENT #2]**.

*This job will be paid as* **[FILL IN "HOURLY POSITION" OR "COMPLETED JOB"]**.

What I am looking for:

- **[FILL IN "INDIVIDUAL" OR "TEAM"]**
- *Perfect English*
- *Ability to problem solve*
- *Amazing coding skills for* **[ADD PROGRAMMING LANGUAGE OR PLATFORM]**
- *Detail oriented*
- *Provide daily reports that are ready each morning in* **[FILL IN TIME ZONE]**
- *Can work quickly, but is very conscious of bugs and code errors*
- *Available to work* **[FILL IN NUMBER]** *hours a week for the next* **[FILL IN NUMBER]** *month to complete this job, but will probably turn into a much longer relationship once this project is complete.*

Please send me a portfolio of apps you have worked on in **[FILL IN CATEGORY]** *so that I can preview your skills. Also please be prepared to supply me with a reference so that I can ask about your work.*

*When you respond write "***[FILL IN YOUR WORD]***" at the top of your response so I know you read this. Also feel free to add a joke :)*

Thanks! Happy to answer any questions you may have.
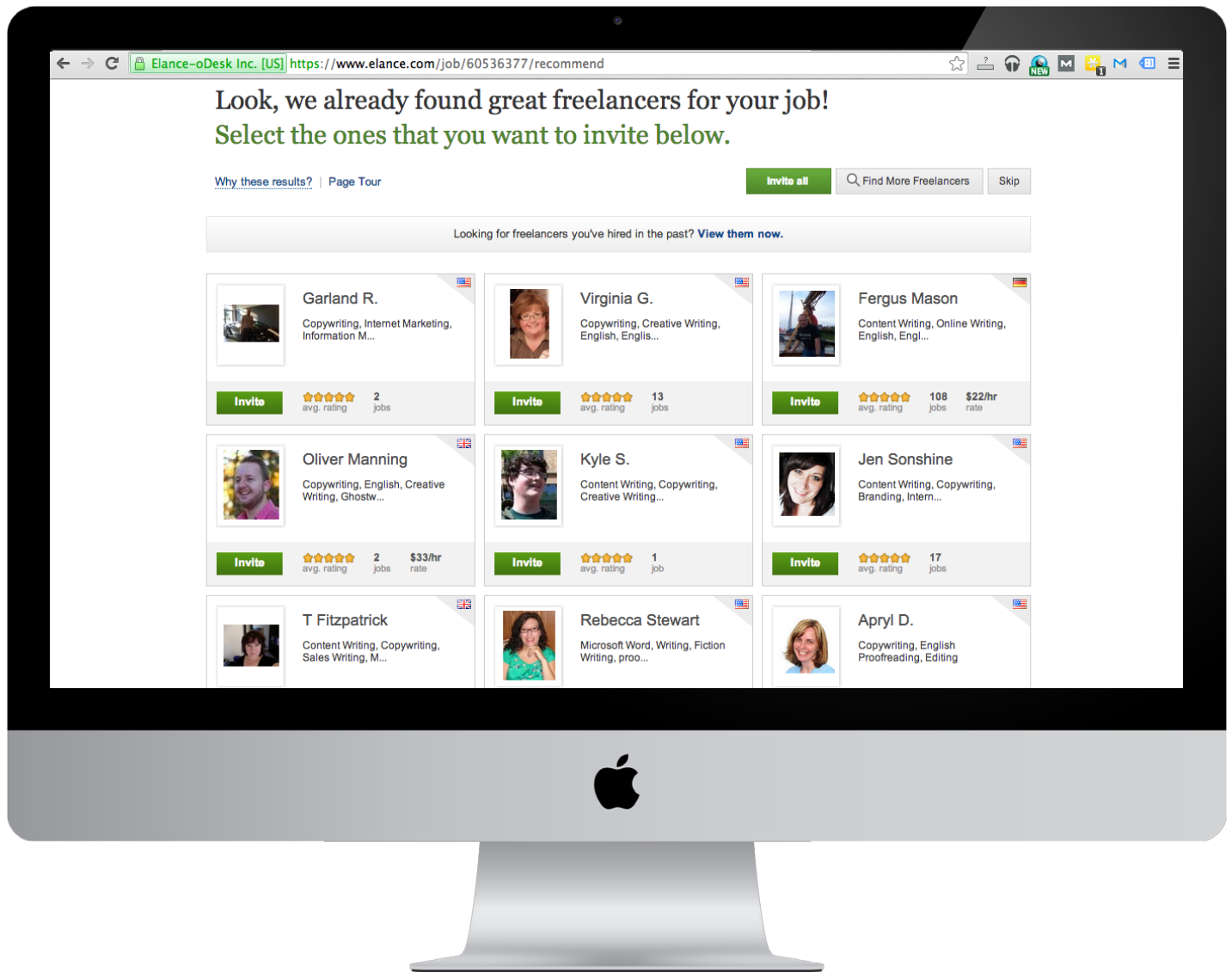
You'll notice that this is for a specific app job (improve existing app) but you can easily update the text to reflect your project.

Once you post this, you have the option of inviting freelancer to bid on your job. You can go through and invite certain people to apply.

Then you'll start getting applications coming in!

What to do when you get applications

You'll start to see them trickling in over the next few days. You're going to get a lot of different types of people applying to your job posting - all over the world, different prices. If you invited anyone, you'll see if they denied your proposal or if they are open to submitting their own proposal. Don't take it personally if they denied you - they probably have too much other work!

When the applications do start coming in, it's important to start organizing them and making moves.

Here are the immediate actions you take:

- **If they didn't put your trigger word (above), deny their proposal immediately.** This is non-negotiable. Not only does it mean they can't follow directions, it means that they are not detail oriented. It's a huge red flag. End it there. The best way I do this is to click the *"Hide"* button on their proposal (in Elance).

- **If it looks like a template, be very skeptical.** This is not necessarily grounds for dismissal, but it's not the best thing you want to see. Often, they will start the post with your trigger word, then copy/paste the rest. You want to be sure they answer your points specifically - otherwise you'll only deal with them once they hear back from you.

As applications begin to trickle in start to weed out the field.

- **If it's too short, deny proposal.** Too short equals 3-4 sentences. Typically it's along the lines of

  "Hello Mr Bluest Cloud of the Sky! Thank you so much I will do great job for you! Of course I work very great for others and can do best job for you too! Thanks and accept now so we can begin work."

  Umm....yeah right.

  Sometimes you will get someone who keeps it short but because they want to get more information. That's OK - you should message them and find out what's up.

- **If they promise that they can complete the job before even hearing the full scope, deny proposal.** You're going to get people who tell you that they can complete anything.....but you haven't even told them what the job is yet.

  They will give you a price and tell you they can do *"anything with apps"* which is a red flag. I usually end it here.

- **If they come back asking more questions, that's a good sign.** This means they read your proposal and want to know more before they give you any feedback of price. This is good news and bodes well.

  You can start messaging with them and answer the questions, then go from there.

- **If they come back with questions + no price, that's great.** That's even better - that means they're really trying to gauge what's going on before giving you a price.

  These conversations usually work very well and the prices are fair. By the same token, this is the kind of relationship that can get you to spend more money than you want to spend, because they give you the *"white glove"* service.
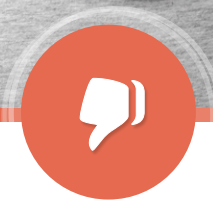
  Always be disciplined. Compare their proposal with others.

Of course this is not an exact science, but will take care of a lot when it comes to cleaning up your proposal list.

GREEN FLAGS

•••

- Responed appropriately to posting

- Asks questions

- Want more details in order to provide accurate cost

Refining the Pool

Ok great. You've gotten your list of 20 down to 3 programmers.

AMAZING!

Now what?

Do you just pick one and hope for the best?

Not so fast. You still want to find the best possible candidate for your project. Even though you've identified the best potential developers, you should really go through one more phase to narrow it down.

Here's what you should be doing to turn the final candidates into a clear winner.

- **Begin private messaging with them.** You may have already done this, but keep this going for a few rounds of back and forth to get a sense of their responsiveness, their English skills, and their ability to grasp ideas. You'll learn a lot about how you jive with them.

- **Have them perform a basic task for ~$30.** Sometimes you can get them to do it for free, sometimes it's $50, but this is like doing a *"try before you buy"* shopping experience.

  When a developer/programmer goes for a job interview they are always asked to do an ability test because programming is a practical skill with a distinct outcome. They know about this and are well aware of the fact that part of the hiring process involves being asked to perform a simple, repeatable task, which is **NOT** a value add for the new employer, simply a quick way for them to show that they can deliver on their promises.

  Examples of basic tasks can be to create something super basic *"create a basic iPhone app that uses my logo on a stopwatch"* or it can be something like *"Design an icon that looks like this app's style."* There are an infinite number of possibilities.

- **Give the same task to all the teams you have left.** You'll be able to do a side-by-side comparison and make your final decision.

☀ **PRO TIP**

•••

"try before you buy"

If you're a beginner, this is also a great way to get a sense of how the development process works. Like I said before - it's a **PROCESS**. It may take a week or two weeks, but it's much better to spend a bit more time now finding someone you like than waking up later on and wishing you had someone else.

What's interesting is that you're usually learning just as much about yourself and your ability to work with programmers as you are trying to find a great developer. That's important to remember - it's not all about finding the person...it's about finding the right person for **YOU** and your business.

I've seen people work with world class developers and have the whole thing fail because personalities weren't aligned. I've also seen people create million dollar businesses with very inexpensive talent because they knew how to cultivate the skills.

Selecting Your Programmer

You've gotten the projects back, you've been able to work the developers. You've learned a bit about the process. Now.....you choose one! This is the first big step in app development and an exciting one.

Before you accept the proposal, here are some things to communicate and keep in mind.

- **Be clear about the deliverable.** Going back to what was mentioned before, make sure everyone is on the same page about what you're getting and what it costs. It's a good idea to run the list by a friend or someone in the app space to make sure you've got it all.

  A good list involves:

    - coding

    - design

    - icon

    - screenshots

    - testing

    - uploading to app store.

  Make sure this is all written down and there is a record of it. That way there is no confusion down the road as the project comes to a close.

- **Each day have them upload their work to a shared space (like Dropbox).** Shared storage spaces like Dropbox and Box are in the cloud of the internet and allow multiple people to upload files to them. It's very easy to setup and allows both parties to easily keep one central location where all the files are.

  Have them do this. This will give you a lot of security in case they disappear for some reason and you need the code - you'll have a place where you can always pull from.

  Similarly, have them update the dropbox folders before you release every milestone payment.

- **Agree on a timeline and set milestones.** As mentioned earlier, agree on a timeline and then set milestones.

  I typically do 4 phases: 25% up front, 25% after completion of the first phase, 25% after second phase, 25% on delivery.

? ASK YOURSELF

●●●

"Are their any other workflow requirements that you need to communicate to the developer before you begin?"

- **Agree on communication method.** Typically Skype works the best (Instant messaging and chat) if you want to get out of the outsourcing sites' messaging service. Google chat is also good.

  BUT, do not move your payments outside of the outsourcing sites until you fully trust the teams down the road. You lose the escrow and security those site provide if you do so.

Then, rock & roll!

At this point you've taken as many positive moves as you can. Be confident in the fact that you did your homework and know that you're already ahead of most of the population simply by reading this ebook!

As the weeks go on, you'll learn a lot. It's rare that the first team is the one you stick with forever, but it happens.

Remember: it's **OK** to fire the team if they're not performing. A lot of people convince themselves that they don't want to go through the process again so they *"stick it out."* The sooner you dump the programmers you don't like, the sooner you find the ones you do like.

☀ **PRO TIP**

•••

Return to this guide whenever you're about to hire or fire a team… knowing where things went right/wrong is powerful for future success.

6

NEXT STEPS

Critical Momentum

Well, you made it!

You've learned the exact processes I (and many other successful app business people) use to hire top tier developers. But it's important that you don't stop now.

This is a very important moment in your journey because you're starting to **MULTIPLY** your progress with new people working for you. While that is super exciting, it can also lead to downtime while you wait for progress to be completed.

You do **NOT** want to wait days or even weeks while your developer starts the project, you want to take action and keep your momentum going strong.

Next up you will learn how to turn your project into a live app in the Publishing ebook. We have **TONS** of information and walkthroughs in here. See you then!

And everything in between.

"You do NOT want to wait days or even weeks while your developer starts the project."