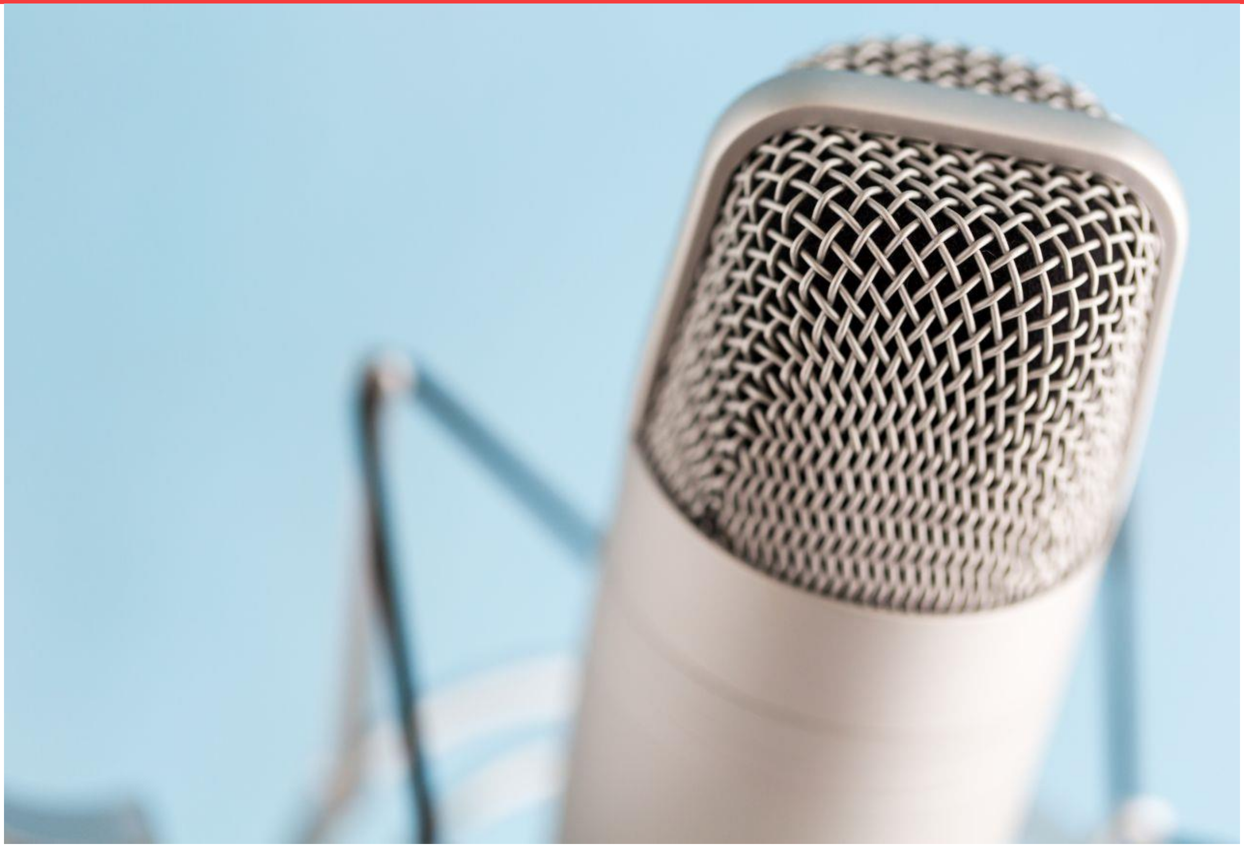
PREDICTABLE DESIGNS

Where Electronics and Entrepreneurship Intersect

Predictable Designs Podcast - Episode #18

Development Costs for Hardware Startups



Podcast Transcript

John Teel: Today, I wanted to talk about development cost, then specifically looking at tips and strategies for those of you that don't have the technical skills to do a lot of the development yourself and you have limited finances to be able to outsource it.

I'm going to give you some cost ranges for the different strategies of developing a product, then we'll hone in on focusing on these strategies if you are severely cash limited and you have limited technical skills.

First of all, the cost to develop a product. Obviously, the cost can vary all over the place for few thousand dollars for a really simple product all the way up into millions if you get really, crazy complicated, which is never the type of product that I would recommend that an entrepreneur should try to take on. Leave the million-dollar projects to the larger companies.

Instead, you need to generally focus on something that's a little more feasible for a small startup to develop. The cost, like I said, it's going to depend a lot on the complexity of the product, but it's also going to depend a lot on the strategy that you use to develop your product.

The cost estimates I'm going to give you, I'm going to assume it's a "Typical complexity product." This is something generally has a microcontroller for the processor instead of a high-speed microprocessor with an operating system.

It's like an STM32 or some other microcontroller-based processor. It also, perhaps, has some wireless technologies, Wi-Fi, Bluetooth, various sensors, rechargeable battery.

Those are what I would consider typical complexity features versus if you have really high-speed processing requirements like facial recognition or a custom radio application. Anything like that, I would consider more of a complicated product.

I'm going to focus mostly on the more middle of the road typical complexity product that I see. Let's just run through some of the different ways to get the product developed and look at the cost.

We'll start with the most expensive and work our way toward the cheapest. The most expensive is going to be to hire a really large, well-known design firm that's most likely located in the US. This is really a strategy that I almost never recommend. The reason being is just the cost.

Most of these larger companies, a minimum that they're even going to consider is $100,000. For a lot of you listening, that's just going to be more money than you have to spend or more money than you should risk spending without significant proof that you're going to be able to get that money back.

A big firm is most likely not going to be the route for most of the people listening to this podcast just because of the cost. It's just a preventative cost for most of you.

The next option is a smaller firm. In the past, some of my older articles and things, I haven't differentiated as much as I maybe should have between larger firms and small firms.

There are smaller firms that typically maybe just have a few engineers or perhaps they have more engineers, but they're outsourcing some aspects, or maybe they have some of their managements in the US, but some of their engineers are in Asia.

They implemented different things like that to get the cost much lower than what you would be charged for a huge firm that's in the US, that has a full in-house team of highly paid engineers.

These smaller firms can be a middle ground between the big firms and the freelancer route. For a small firm, I would say, you're looking at probably in the price range of $10,000 to $30,000 to get a product developed.

Keep in mind, these are to get it fairly far along. This is for a production version of the product. I'll talk more about this in a little bit, should a production version of your product be your goal?

Because the costs that we're talking are pretty significant. We'll talk about that here in a moment, but for a small firm, your typical product, you're looking at least around $10,000. It's still significant.

For a lot of you, it's outside the price range a lot of entrepreneurs can probably afford, but it's also in the range that a lot of entrepreneurs can afford. It's much more realistic than $100,000. $10,000 gets it down into where a lot of people could potentially afford it, but it's still a significant amount of cash. I realize a lot of you listening that $10,000 is just an unimaginable amount of money to invest in your startup initially.

That's generally, especially if you're a nontechnical founder, I really have evolved over the years of thinking that if you're nontechnical, you really need to be working ideally with a firm of some sort instead of the freelancer route because it's just so challenging to manage a bunch of freelance engineers if you don't have the technical skills to manage those.

I tend to think going the small firm route is probably ideal if you can afford it. If you have the $10,000 to $20,000 to spend on it, then a small firm that is likely either overseas or has some portion of their business overseas is going to be one of the best strategies for you.

I also know there are a lot of you that can't afford $10,000. I want to give you options as well, even though the less money you spend, then there's going to be compromises. There's always compromises in everything business-related and especially, with doing a hardware startup.

If you have no technical skills and you have no money, then that presents two really big challenges. Obviously, if you have technical skills, then all these costs are going to be a lot lower because you're going to be doing at least a big portion of that development.

That's always going to be the cheapest way to get a product developed, is if you can develop it yourself, but for a lot of you, that's just not feasible and hiring a large firm isn't really feasible. Typically, the cheapest option is to go the freelancer route, but there are so many issues with hiring freelancers.

I can't tell you over the years, I just see more and more stories of entrepreneurs that have been burned by going the freelancer route. I've recently even had a few people contact me that they had hired engineers through the website, Fiverr, I guess it's how you say it.

It's not something I use very often, but I know it's a website where you can get extremely cheap freelancers to do various projects like build a logo for your company for $5, that type of thing.

That's what I've always associated Fiverr with, is a really, really, really cheap freelancers that I think can be feasible to use if you have a simple project like build a logo, or design a flyer, or something like that, but it's not going to be the way most likely to try to get a full product developed, especially if you're nontechnical.

I know one person, a member of my Hardware Academy had reached out to me. He had hired someone on Fiverr for only $250 and he got some schematics and designs back and he submitted his design in the academy community to get feedback.

The general feedback was this design's incomplete, it's not at all going to work. It's not ready for production. Perhaps, he wasted $250. I'm going to talk about that more in a moment, a way that this can still be a reasonable strategy.

Then, another person, he had spent $2,500, 10 times more, but still quite cheap for getting a product developed on Fiverr, but they kept going past the deadline and they essentially never delivered. He ended up having to go through Fiverr and submit a dispute and get his money back.

Both of those were either wasted money or definitely wasted time, but I understand the appeal of-- Especially, if you only have hundreds of dollars, I understand the appeal of trying to hire someone really, really cheap, but it's almost always going to be money that's going to be wasted money if you don't have the technical skills to really monitor and manage them.

What I recommend, and this applies regardless of the strategy that you're going to use, whether you're going to hire a firm or a really low-cost freelancer who's in India that you found on Fiverr.

If you're nontechnical, you need to have a technical advisor. I think that should be priority one is to find a technical advisor. Ideally, a technical co-founder that you bring into your company, but I also know it's extremely challenging in the early days.

Especially, if you don't have a lot of progress on the product, it's really hard to find founders that want to commit to your product to the same level that you do and come in.

Finding a technical co-founder is ideal probably, but it's also very, very, very challenging for a lot of people. What I recommend is that you definitely do is bring on a technical advisor.

This is something that is part of the service that we provide in The Hardware Academy is you get access to a team of experts. You have a lot of technical advisors, including myself, we can communicate privately.

I'm always there to be an advisor to you. That's the cheapest route to getting your product developed is to hire someone really cheap on, say Fiverr, but then perhaps join the Academy so that we can help provide some oversight to what they're developing.

What I would really probably recommend is that you do that, so you hire someone cheap if that's all you can afford, you also join the Academy to just get our consulting, and help, and advice, and basically, be your technical advisor.

Also, most likely is to hire a more experienced and usually probably more expensive, engineer who can be your technical advisor, and do design reviews, and really keep track of the progress that the developers, the low-cost developers that you've hired.

That engineer, the technical advisor can be the one to just, obviously, advise you on technical decisions and challenges but also could provide design reviews of anything that they design so that you're just making sure that you're getting quality that you're paying for.

Most of this you can do in the Academy. You're welcome to submit your design in the open community. For no additional cost other than just the membership cost, you can get reviews and feedback.

That's what happened with the one member that had paid $250 to get a schematic design. He just submitted in the Academy and he got a bunch of feedback from various engineers.

In conclusion, was, is you need to hire someone that can either take over that design or someone, like I'm suggesting, is bring on a technical advisor who can help you manage those people that are actually the ones doing the schematic, and the PCB layout, and all the detailed design work.

This is something I really encourage that regardless of your strategy, the firm, how great they are, how much you trust them, just always have second opinions there.

Just always have as many advisors as you can find. Score.org if you're in the US, is one option I've used in the past for finding some advisors. You can find general business advisors. You're probably not going to find any advisors through Score which is basically a non-profit.

It's where retired executives and such that, want to give back and help others. They are there doing offering to give advice and consulting for free. You can find some good business people, but the odds of finding someone that's got a significant hardware experience is going to be really unlikely.

That's when something like the Academy can be really beneficial as being part of your team of advisors.

That's the first thing you need to do, is you need to get some technical advisor. The worst thing you can do is just to go blind and without any technical experience and try to hire someone off of Fiverr, or even Upwork, or any of these freelance websites. You're almost always going to be burned.

I don't know the exact ratio, but I feel like more freelancers end up losing money, getting a product that's not manufacturable, or that won't even work, and just wasting money and time. That's not the route that I suggest. Start with the advisor.

The other tip is, simplify your product. If you don't have a lot of money to spend to develop a product or the skills to do it yourself, then you have to simplify the product.

That's going to be the best way to get your costs reduced, to make it more manageable for maybe a less experienced engineer that's a lower cost to develop.

Hiring an inexperienced engineer to develop a Bluetooth low-energy product that's got a, say, a temperature sensor on it and that's it is going to be much easier than finding someone that has the capabilities to design a really complicated product. Focus on simplifying your product.

There's an article on my blog, I think it's 10 Tweaks That You Can Make to Your Product to Drastically Simplify It. I'll make sure I link to that in the share notes, but definitely take a look at that. It just gives you different ways of simplifying your product to keep the development cost down and speed up the time that you can-- How fast you can get the product to market.

This goes along with the minimum viable product, MVP concept is just you need to simplify it down to the absolute core functionality to solve the intended problem.

Get rid of all the extra bells and whistles that you think people may want. Just get it down to where it just solves that most basic function. Focus on that. Get that in the hands of customers, then you can build from that feedback. You can build from there.

Another tip, along the lines of simplifying your design, is to find ways of using existing solutions for your initial proof of concept prototype, for instance.

Is there a way that you can completely avoid having any custom development on your first version that you just want to get in customers' hands? I've seen other entrepreneurs do this successfully. Where the one Academy member, they were doing a fitness wearable and they were able to find a development kit that was very close to what they needed for their final product.

It wasn't going to be exactly what they needed and the cost was going to be prohibitive for mass manufacturing, but it allowed them to get something in customers' hands and start testing and getting feedback as soon as possible.

Try to think of any ways that you can use existing solutions to get some market feedback or customer feedback before you start trying to go down the path of fully custom development.

Another tip regardless of whether you hire a firm or a freelancer is to get recommendations or referrals.

Especially, once again, if you don't have the technical skills to judge how good they are, it's just going to be a shot in the dark if you just do a Google search and you try to find an engineer or a developer or even a firm and you hire them, you may get someone really good and you may not. It's always one of those things where it's definitely an area where I recommend that you get some referrals or recommendations.

To be able to judge an engineer, and how good they are, it really takes another engineer to be able to judge how good another engineer is. I would start by getting referrals and recommendations from other engineers.

Then, this, once again, is something that we have in the Academy is we have referrals for developers and supplies and manufacturers that either I've worked with and recommended or other members have recommended. That's one way to get some recommendations for freelancers, and firms, and manufacturers, and everything. That's always something good to do is to get recommendations.

When getting a product developed, once again, especially, if you have little finance and really regardless of your finances, you need to focus on minimizing your development cost, how long it takes to develop. Those are the two key things that you want to minimize.

You need to focus on reducing those costs and not worrying about the product cost yet. For instance, let's say, your product is fairly large, you can fit an Arduino in your product enclosure. You could make a proof of concept prototype that you can potentially even sell, but you're just embedding an Arduino.

You're like, "Okay, I'm not going to even show this to customers." You decide to get a fully custom design done so that it can be more affordable and something you can actually sell at a profit. I think that's the wrong strategy to take. You need to, of course, know that your product can eventually be made and sold at a profit, but initially, don't worry about profit. You need to just worry about minimizing the cost of getting it developed.

One example of this is, it's very enticing when you're past development and you're wanting to buy some inventory, that if you only bought 100 units, trust me, you're going to pay a lot more for them than you would if you ordered 10,000 units.

I've seen a lot of entrepreneurs fall into that trap of, "I know I can sell them. I'm going to get the 10,000-piece order so I can get a much lower cost." That's the opposite of the advice I'm giving.

In that case, you're spending more upfront in an effort to try to lower the product cost and make more profit.

Forget about any profit on the first 1,000 units that you're going to sell, most likely, unless you're selling a really high-dollar product, and especially on the first few hundred units.

If I had a product and I was able to get a prototype that I could actually sell to a few customers, even though it was going to be at a loss. Let's say, I had 100 customers that were going to buy the prototype, even if I lost $20 on every unit that I sold. I sell 100 of them and I lost $2,000.

That's generally not what you're striving for, as you want to make money with sales, not lose money. That's not the goal when you're first starting off.

First starting off is to get that data, to get the product in customers' hands. If this cost you $2,000 because you lost $20 on 100 units, but it gave you valuable data from customers, then that is well worth the cost of that $2,000.

That's just an example. I'm just making up the numbers, but the point is, focus on your getting that data, that customer feedback data even if it costs you to get that data.

That's a much better way to approach it than pushing past the data collection right into trying to minimize your cost because you may go to all this trouble minimizing your cost, you have a fully custom design, you get it in customers' hands, and you find out about half of the features you included, they don't even want or need.

It would be much better to have gotten something cheaper or cheaper to develop and get that in their hands and collect that data before you go off and spend all this money in developing the full production version.

Those are my general tips, then some estimates guidelines on how much it costs to get a product developed.

In general, I think that your best strategy and where your focus needs to be if you don't have technical skills and you don't have the money to spend, then like I said, first of all, start with a technical advisor. That should be priority one. There's absolutely no way you should be doing any of this on your own without having a technical advisor.

Then, after that, it's pretty much going to come down to, you need to be constantly networking, making new connections because that's how you're going to get other people that can be interested in your product at some point.

Perhaps, you get it to the point of you get a proof-of-concept prototype, you get 100 of them made, you get some data, people love the product, and now, because of the connections you've made, you can get outside investors. You're always going to be looking for doing networking, whether that be investors or co-founders in the future.

The point is if you don't have technical skills and you don't have any money, you need help, you need other people. You need to be making networking, and getting advisors, and connecting with people, one of your top priorities. Secondly, you need to learn.

There's so much you're going to have to learn. I'm not advising you to try to learn to be an engineer if you're not an engineer and try to develop your own product.

That's feasible if you've got technical skills or you're an engineer, but it's not something I would ever recommend that you try to do if you don't have any technical skills.

You're going to need to still learn, though, enough about it to manage it and know how you're going to overcome the obstacles associated with that area.

Don't know how to design a PCB, design your own but understand the process of it. You're just always going to have to be in the networking, making new connections, and constantly learning, but then, also, be sure you try to capitalize on your strengths.

Even though you may not have any technical skills, everyone has areas that they are good at and things that they're really strong in.

If you're a people person with a lot of marketing experience and you enjoy networking, then make that a focus while you're still doing some of the other stuff in the background because it's always much easier to get people interested, invested in your product once you've, at least, made some serious upfront progress on your own.

Those are my general tips for those of you out there looking to outsource product development and specifically, for those of you that have severely limited budgets. These are just some of the tips on ways to overcome these obstacles.

There's no easy path. Even if you've got lots of money and lots of technical skills, it's still complicated to get a product developed in the market. It's not like, "This is easy for other people and hard for you." It's hard no matter what your path is, so you're just going to have to keep that in mind and just keep pushing forward.

I'm John Teel. I hope you've enjoyed this podcast and I'll see you next week for another episode the *Predictable Designs Podcast*.

PREDICTABLE
DESIGNS
Where Electronics and Entrepreneurship Intersect