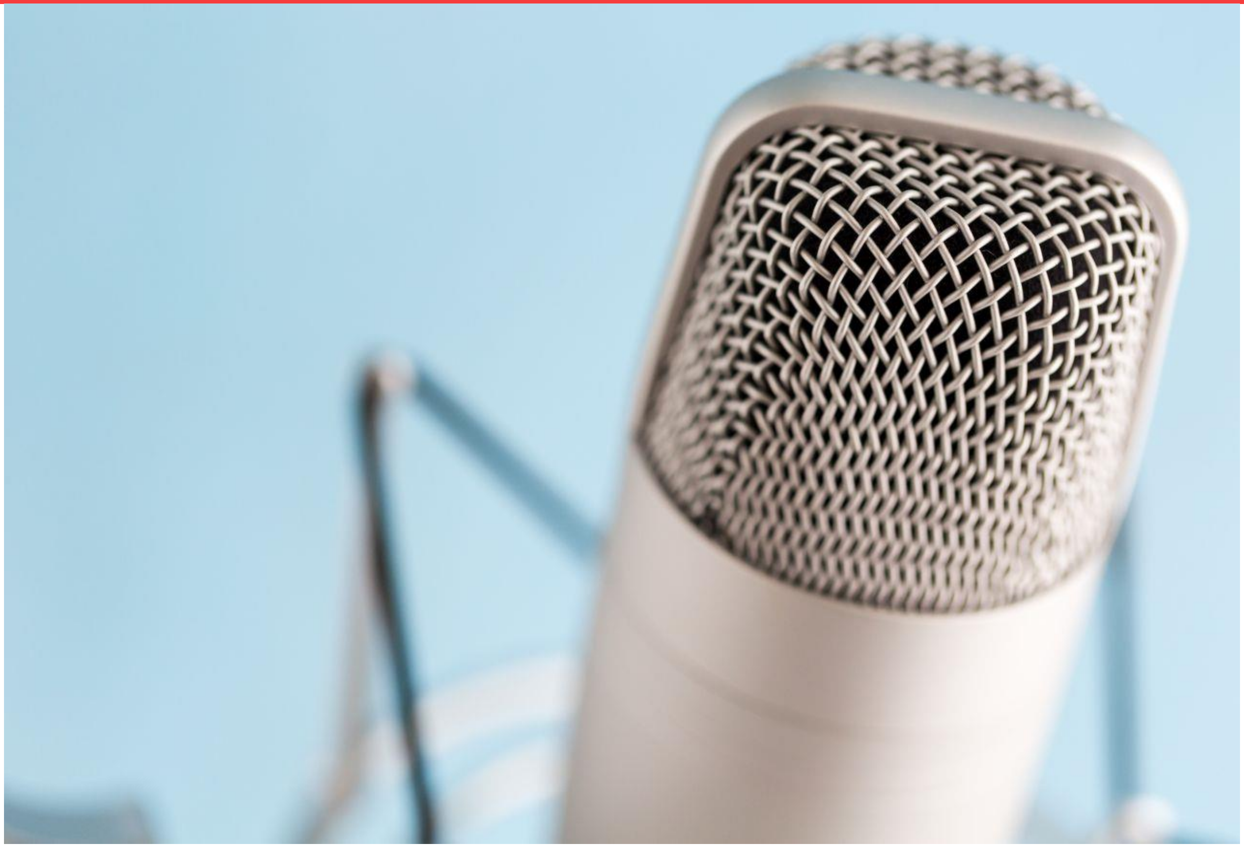
PREDICTABLE DESIGNS
Where Electronics and Entrepreneurship Intersect

Predictable Designs Podcast - Episode #17

6 Strategies to Get Your Product Developed



Podcast Transcript

John Teel: In this episode I discuss 6 different product development strategies.

Which strategy is best for you will depend on several variables including your skills, how much money you have to spend, the complexity of your product, and how fast you need to get the product to market.

This episode is actually just a small part of a video course inside the Hardware Academy that is a roadmap course that walks you through every step from idea to manufacturing.

This is what I'd like to share with you today, and I hope you enjoy it.

The first option is the simplest to start with, at least to explain. That's to design the entire product yourself.

For most engineers and most products, this is not really feasible to do entirely on your own with you being the only person that does everything. There's just a huge amount of variety of technical skills required to fully develop a product.

If this was a larger tech company, they would never assign just one engineer to develop a product like this. They would assign a team of various specializations to develop the product.

Developing a hardware product is going to require that you have a really solid understanding of electronics design. That's not just using Arduino or development kits, but actually designing the custom-printed circuit board.

You're also going to have to know how to program with both the firmware, the code that runs on your device itself, then also any, if you have a mobile application that you want programs, you're going to have to know how to do that well.

Those are, even though they're both programming, they're drastically different types of programming. You may be good at one, but not necessarily the other. You're also going to need to know how to do mechanical or industrial design.

At the very least, your product is going to have a plastic enclosure that you're going to have to design, create a three-D model for it and get that ready for mass production, injection molding tech, using injection molding technology.

Then you have all the manufacturing, technical challenges themselves. Like I just mentioned, getting injection molding set up and getting manufacturing setup in general is a sort of a specialized field of engineering.

A lot of engineers will have experience with development, but not necessarily manufacturing.

To have experience with electronics design, programming firmware, programming mobile apps, doing the industrial design and doing the designing the product for manufacturing, that's a lot to take on for one engineer. This becomes even truer, the more complex the product.

This could be a viable option if you're an engineer and you have a wide variety of skills, or if you're really adaptable and can learn these skills and you have a product that isn't overly complex. This is pretty much the route that I had taken for my own product was to design it myself.

The main area where I was lacking knowledge was in the industrial mechanical design. I ended up teaching myself how to do the mechanical design, create the three-D model, design it for manufacturing.

That was a real good strategy for my product, which is really a very simple product compared to a lot of products that I see members developing. Typically, you have to combine this development strategy with other ones.

You may be able to do one part of this, but not necessarily all of it, or you may be able to do a little bit of each, but then you want to bring on an expert or specialist in those areas to help you or to check your work.

The next development strategy that we're going to look at is for you to design only the early prototype yourself. The goal of this strategy is for you to develop the product just far enough that you can have a prototype that's of sufficient quality to get outside investors interested in your product.

What this allows you to do is it allows you to bring your product to market without you having to do all of the development yourself or to fund all of the development yourself.

Instead, you can just focus on the initial parts of the development, and then once you get it far enough along, then you can use that to seek outside funding.

Then with that funding, you could hire professional engineers to go through and clean up the design and get it production-ready or you may not even have to have funding, you could also take the prototype development as far as you can and then bring in professionals to clean it up and get it ready to be a final prototype that you can then use to get funding.

The point is that the key of this strategy is that you're not doing the entire development process yourself, you're only doing as much as you can or as much as necessary to be able to get the funding to take it to the next steps.

This is a good strategy if you have technical skills or you're comfortable learning them, but yet, you're not what you would consider a professional or expert product developer. This would allow you to use your current skills and save the money, what you would normally have to outsource even the early prototype.

You do that yourself and then you just use that to then get the funding that's necessary to take it to the next steps. This, I find, tends to be a good strategy if you have moderate technical skills.

Regardless whether or not you develop just an early prototype or a late-stage prototype, generally, most founders are going to have to eventually seek outside funding. The further you can get the product on your own before you have to seek that funding, the more likely you're going to be able to get that funding that you need.

The next development strategy that we're going to look at is to find a technical co-founder. If you're completely nontechnical, then I highly encourage you to take this as, at least, your main strategy. Most of these strategies can be combined with the other strategies.

If you have no technical skills and you're trying to get a hardware product developed, that's going to be really, really challenging.

You're going to probably end up wasting some money because you're not going to have the skills to judge the quality of the work. You don't necessarily know if you're being scammed by the person doing the work or if they're producing a really poor quality design. If you don't have those technical skills, that's really difficult to judge the quality.

It's very challenging to try to manage other people doing a job or a task that you don't have enough skills to even judge the quality of their work. This is when I highly recommend that you look into bringing on a technical co-founder.

A co-founder would be someone that is equal partners with you. In most cases you need to do an equal split of the equity. Otherwise, it can just cause various complications in your relationship with the co-founder.

It's best to find someone that's just as excited and just as vested in the product as you are, and then make them an equal co-founder along with yourself. Not only can this help you bring in someone with the skills that you're missing, it can also help you bring in money to your company, because now instead of you paying for everything upfront now you're splitting that with your co-founder.

The big challenge with this strategy is finding those co-founders. It's difficult to find a co-founder. It's a really critical relationship that you're trying to build with them.

You need to be extremely, extremely selective in who you choose to bring on as a co-founder, because you are going to be tied to them for years. You guys are going to be working so closely together.

There's going to be a lot of stress you're going to have to deal with. It's really critical that you take the time to find a co-founder that's a good fit for your product.

By that means, first of all, they need to have complimentary skills to yours. They also need to have some income that they can help pay for things in addition to you paying for everything.

They also need to be really, really excited about the product. You don't want to bring on a co-founder that's just not excited. It's excited as you are.

It creates just a lot of challenging dynamics between the two of you. If one founder is much more motivated and excited than the other, because the one that's the most motivated and the most excited tends to do the most work and then that just creates challenges in the relationship.

The other big downside with bringing on a technical co-founder is it reduces your equity in your company. However, it's always much better to have a small piece of a big pie than to have all of the pieces of a really, really tiny pie or zero pie.

That may be the case if you don't bring on a co-founder, if you just try to do this on your own and you never get it to market.

Now, you own 100% of nothing, versus if you bring on a co-founder, you may end up finding yourself owning 50% of a multi-million dollar company, eventually. There's always trade-offs with all of these different strategies.

I highly recommend this strategy. If you are completely nontechnical, it's really going to be essential for you to bring on a technical co-founder.

If you happen to be highly technical, then a co-founder can still be a good fit if their expertise is in marketing yourselves or something. I'm not going to get into that in this lesson, but the point is the same, is to bring on a co-founder that has complementary skills to your own.

The next development strategy that we're looking at is hiring freelance engineers through potentially a website like upwork.com or guru.com or just referral-based.

A freelance engineer is typically just a solo engineer that works on his own. You're going to pay them typically an hourly rate, but also in some cases you may be able to strike a deal when you pay them on a per milestone basis.

Like all of these strategies, there are some challenges with hiring freelance engineers.

First of all, as I had mentioned in the first strategy is few engineers have all of the skills required to develop a product to bring to market. This likely means you're also not going to be able to find us a single freelance engineer that can do everything for you. This ends up meaning that you will most likely have to hire multiple engineers with various specialties.

Then, what happens in that case is, now you're managing a team of engineers and you have to make sure all the pieces fit together. The hardware has to work with the firmware, the firmware has to work with the mobile app, and the printed circuit board has to fit in the enclosure.

You may be working with three to four or five engineers. If you don't have the technical expertise, it's really challenging to manage all of those engineers, because you're going to be the one that has to solve any problems that are caused by the interaction between the two engineering specialties.

For instance, if all of a sudden your printed circuit board you find is too big to fit in your enclosure, well, then you may run into a case where the electrical engineer is blaming it on the mechanical engineer.

I made the board the right size, he made the enclosure too small. Then, the mechanical engineer may say just the opposite and blame the electrical engineer, or especially with hardware and software it gets really challenging to manage a whole bunch of different freelancers, especially if they're spread out around the globe.

It becomes very, very challenging to manage all of that.

That's the biggest complication with freelance engineers. The biggest advantage is, if you don't have the technical skills, this tends to be the cheapest and the easiest option, is to hire a freelance engineer to do that work for you.

It's not typically something I would recommend if you're completely nontechnical, because you're going to have some technical skills to manage these. You do have the option if you could hire just a general consultant to help you manage the various engineers, so you hire a bunch of contract engineers, then you have one consultant who helps you manage those engineers.

That can be one option, but it's still opening up for the possibility of a whole lot of issues. Hiring the freelance engineer route is typically only a feasible option.

If you have at least some project management background or you feel comfortable managing a team of engineers. You don't have to necessarily know how to do everything that they're doing, but you need to have the skills to be able to judge what they're doing and to solve any problems when the different pieces have to come together.

You're the person that will be responsible for the final product, so you have to have all these different parts come together and work together to create the final product. You're going to be the one responsible for that. That's the big challenge with freelance engineers.

The big positive with freelance engineers, it's one of the cheapest ways to develop a product if you don't have the skills to do it yourself. It's going to be much cheaper than hiring a development firm.

The fifth option we're going to look at for developing your product is to hire a full design firm. A design firm, unlike freelancers where you're hiring individual people, and most likely multiple individual people, and then you have to manage and make sure everything comes together.

With a firm all of the engineers already work together.

Typically in the same building but not always, they may be remote teams. The point is all the engineers already work together, in the firm itself will manage the entire project and manage all of those engineers.

You don't have to be responsible for making sure that the hardware works, that the firmware works with the mobile app. The firm itself will take care of that as part of the project management.

If you lack the skills to manage a team of engineers, then a design can be probably your safest option for getting your product developed.

The big downside with the development firm is it's the most expensive route to develop a new product as well, especially if you try to hire like a large US or North American based development firm.

Typically, you may be looking at very minimum of $100,000 to develop your product, so that's typically not viable for most startups.

There are smaller firms that's potentially made up, a lot of them can be just various freelancers that ended up working together, and now they work together and have a small firm, so they don't have the overhead of a large firm with dozens and dozens of engineers and big office building.

It's just a few freelance engineers with various specialties that have come together and work together.

If you can find that, that can be a really good option. It's not going to be quite as hands off for you as a full design firm would be, but it's a nice mix between going the freelance route or hiring the full design firm.

In addition to the cost, the other major problem with hiring a full design firm to develop your product is you're removing yourself from the product development phase to some extent.

If you're wanting to build a hardware company, it's going to be really challenging to build that company if you're not the one that's developing the products that you're planning to sell.

You may be able to do that on the first product, but, eventually, you're going to want to have to manage and oversee and do the development yourself.

A big part of a hardware startup is the product development. If you're outsourcing all of that, then you have to begin to question what is it that you do.

Are you just a marketing arm? What it your purpose?

That's something you need to consider when you go to pick out or pick a development strategy, and, specifically, if you choose to hire a full design firm is, are you outsourcing too much, are you giving away the core part of your business?

That's going to raise problems down the road, especially if you try to seek outside funding, a lot of investors can be really skeptical that you're wanting to outsource everything about development.

Keep that in mind. It's typically not a good strategy to think that you create a spec and then you give it to a firm and say, "You got to develop this. Just let me know in six months when it's ready."

If you're going to go the design firm route, then you need to be heavily involved, communicating with them, understanding the design decisions and the design trade-offs that they're making, because they don't understand your market, they're just engineers.

They're going to pick most of their decisions based on engineering criteria, and not marketing criteria or business criteria. Just keep that in mind if you do decide to go the route of hiring a full design firm.

The final development strategy that we're going to look at is to partner with a manufacturer.

One option is if you can find an existing manufacturer that is already producing products that are similar to yours, then instead of you redoing everything that they've already developed plus incorporating your new features, a lot of times it can make a lot more sense if you strike a partnership with them so you can take one of their products and either you modify it or more likely they would have their development team to modify one of their existing designs to meet the criteria that you need for your product.

This can be a really good strategy, especially if you have a more complicated product, but yet something that's only incrementally different than other products that are already out there.

Especially if the product is really complex, you may be able to partner with a manufacturer to only-- they take care of reusing one of their products for part of it. Let's say you have a remote control car and you have, I don't know, let's say it's got WiFi and some special electronics that may differentiate it from other remote control cars.

Well, there's not a whole lot of point for, in most cases, for you to develop the entire car itself, the plastic and the wheels and the axles and all the different plastic pieces because that's really complicated and really expensive. What you may be able to do is strike a deal with the car manufacturers so that you use the body of one of their cars, but then they incorporate your custom electronics into it.

This can be a good strategy that allows you to not have to start from scratch that you can just build off of what someone else has already developed. For this, I would recommend that you go to alibaba.com and just search for your product and products similar to yours and then just reach out to them. The one thing you'll find is that most Chinese manufacturers are very flexible in regards to working with you or customizing one of their products for your particular application.

Of course, you have to approach them as a serious business. This will take some time commitment on them. Don't just reach out to them as Joe Schmo with a Yahoo email address.

You need to present yourself as a business, which is almost always the case when reaching out to anyone to help you or for any service, I recommend that you reach out to them as a business and not just a person. You'll get a much higher response rate and companies will be much more willing to work with you.

The downside to this strategy is finding the right manufacturer can be challenging and you also have to question who will own the intellectual property. If they do the development work, then they may want to own the intellectual property so that can create complications.

They may want a total exclusive manufacturing agreement. It could be that they require that you produce, manufacture the product only with them for a certain number of years and that could lock you in. If they end up performing badly, then you're locked in with this manufacturer and you don't have any way to get out of it.

Those are the trade-offs that you have to consider when partnering with a manufacturer. If your product is just an incremental improvement to an existing product or a big portion of yours is based on an existing product, then this is a strategy that I highly recommend that you pursue.

Those are the six strategies for developing your product. It's not always going to be only one of these six strategies. A lot of times you'll have to mix and match different parts.

Like with the partner, with the manufacturer, you may do that for, like I said, for the remote control car enclosure. You partner with the manufacturer, but then you hire freelancers or a development firm to design the electronics.

A lot of times you're going to do a mix and match or if you have design skills of your own, let's say, you're a really good programmer, then it may be three of these. You may partner with the manufacturer for the enclosure, outsource the electronics to a firm or a freelancer and then do the programming yourself.

There are various ways that you can mix and match these. It really depends on the skill set of you and your team, how much money you have and the complexity of your product.

That’s it for today, be sure to tune in next week for another episode of the Predictable Designs Podcast.

PREDICTABLE
DESIGNS
Where Electronics and Entrepreneurship Intersect