

New Dot Spot

How much is this going
to cost and how long
will it take?

by **Michael A. Borthwick** Founder, New Dot Spot on January, 2017



This is a free excerpt from the book, “What’s Next? Everything I wish startup founders knew about the software development lifecycle” by Michael Borthwick

New Dot Spot
4732 1st Street SW
Calgary, Alberta
newdotspot.com

This eBook is for you if...

1. You're a new software startup or considering starting up.
2. You want a prototype, a minimum viable product or don't know what those are
3. You have a budget of at least \$25,000 (If not, you'll really want to [sign up for the mailing list](#) as we will cover a lot of DIY and fundraising solutions).

This book is organized into four parts

1. Alignment

Why it's good to get on track with your dev shop and why it could cost you more than money.

2. Types of projects

Different types of software have different costs.

3. Missing Pieces are Costly

Getting a better estimate requires detail and clarity. Why you need a roadmap, how much they cost and 3 costs you've failed to account for.

4. Roadmap Calculator

How you can get clarity on your own. Use this exercise to find out how many hours it might take to build your prototype or MVP.

Introduction

As a former developer, project manager, and solutions architect I can tell you that the two most common questions I hear revolve around money and time. In this eBook I intend to help you answer those two questions.

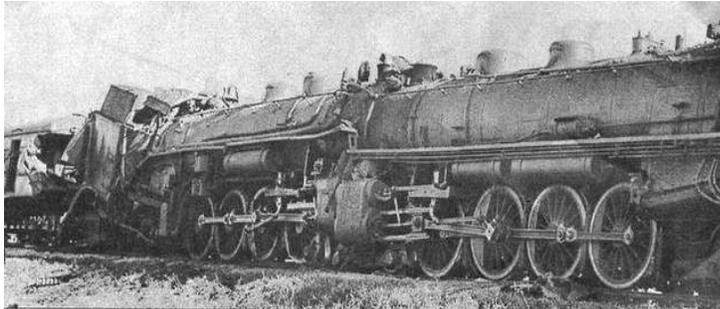
"HOW MUCH IS THIS GOING TO COST?"

"HOW LONG WILL IT TAKE?"

I'll bet you're already wondering how accurate I am at answering those questions. I'll come back around to that in a bit, but before I get too far into this I want to back up the train a little and talk about what is usually done to answer those questions and why I think those things are just. plain. stupid.

**The first thing we need to do is agree to keep both
of our trains travelling in the same direction.**

Alignment



Imagine if we were on the same track, but headed in different directions?

This won't work any other way. Yes, we both have goals. Yes, they are aligned already or we wouldn't be doing business (or about to do business). You have an app that you want to build and take to market. We want to build that app and help you take it to market.

We are **Aligned**.

Now imagine if we were on the same track, but headed in different directions. We would quickly collide or quickly grow apart - neither scenario is good for business. Let me show you what I mean.

IDEolium¹ is a security startup. They are trying to solve the problem of authorizing employees using various "no-touch" methods. They want to be able to authorize employees to get through a secure door hands free, card free, lag free, pause free - quickly, easily, and securely. They're currently working on a prototype that will measure the walking gait of employees. They are in need of software for their prototype.

¹ Does IDEolium sound too good to be true? Your suspicions are correct. IDEolium is a fake company. I pulled that name out of a random business name generator and improvised what they do based on the name alone.

IDEolium has 3 choices.

1. Hire a developer
2. Get a quote
3. **New Dot Spot**

1. Hire

There are three problems with hiring developers that startups fail to account for.

First, attracting people to come and work for a new startup is difficult. Every developer I know wants a reliable paycheck from a big stable company - low risk. How is that any different than everyone else you know? Startup life is risky and we've all been on the losing end of a bankrupt startup. It ain't pretty, Joe.

On that same point, not everyone wants to be an entrepreneur. The developers I've met who do have that entrepreneurial spirit will all tell you the same thing, 50% of your idea is worth less than 100% of my own idea. Those kinds of developers have their own ideas for a startup and they're making it happen on their own, like an entrepreneur would.

This isn't to say that there aren't developers out there who don't thrive in startups. On the contrary, there are some of us who actively seek out positions where we can be employee number 1 or 2. We are a rare breed with a rare skill set and we either burn out on the idea real fast or we hit gold and ride the elevator to the top.

The second point startup's often miss is that experienced developers aren't readily available like other staff. One way or another experienced startup developers leave the hiring pool. It's almost a certainty that the more experience they have the less likely they are to pursue something new. This means the hiring pool is shallow and concentrated with junior developers who haven't burned out or haven't struck gold (yet). Think about it. You want a developer with 5 years of experience right? Well, what kind of experience? Where did that experience take them? You think you can attract them away from what they've spent the last 5 years building? That would be very rare.

When you are balancing the cost of paying for development versus the cost of hiring developers, one of the costs startup founders fail to account for is the overhead costs. There's more to hiring a developer than office space, computers, breaks, snacks, coffee... more coffee... pizza and beer. New startups look at the rate an agency is asking for and think, "I could hire someone for half that!". Well, could you? Really? I have literally worked out of a garage as a dev in a new startup with 3 people and I can tell you from experience, I don't recommend it. This leads us into the third issue most startup

founders fail to account for - the amount of time and the rare skills required to organize, communicate with and manage developers well.

If you haven't already come to a deep understanding of Agile vs Waterfall, continuous integration (CI), rapid application development, application development lifecycle, object oriented programming, functional programming, reactive, state mutations or even what an if statement is - then you're going to need to hire a project manager to translate 1337 into English. For example, as you talk to different developers ask them what "suboptimal" or PEBCAK really mean.

Developers truly are a different breed of employee don't just speak in their own language they come with a unique culture as well. I believe that the core of good project management is communicating with developers, but if you're not familiar with the culture and language it can be a very time consuming challenge. Consider that the cost of failure here means the loss of a developer.

2. Get a Quote

The interesting thing about quotes is that they take a very long time to prepare. They go into radical levels of detail. Then they multiply that number by something I like to call "the professional factor". The bigger the company you take your roadmap to for a quote, the higher the multiplier. That's not unexpected, but I wanted to get you to think about why and how that happens. You're not hiring just a developer, you're hiring the people around the developer too. The kind of people that developers need to support their efforts.

Think of a sports team. They have more than just the players on the team. There's a coach, a trainer, a bus driver, the ticket takers, concession stand workers, etc... etc... When you hire developers you need to take on all of the supporting roles. Maybe that's the thing you bring to the table? More likely it's not.

If you're a new startup working with a big agency and you just got a quote for \$500k you're likely recovering from sticker shock. I don't take issue with their price. Building software at that level is a very costly endeavour indeed. However...

As soon as you sign the contract your interests head in different directions.

They will now do exactly what they agreed to do, but no more. They are constrained by the quote and their motivation is to get your app built at or below what they quoted. When things change, and they always change, you will find yourself at odds. This is not an appropriate business model for new startups.

This is, however, an appropriate business model for an established business with domain experience who are looking to take a product out of startup stage and build a fully scalable, professionally crafted, well polished product. If you aren't a new startup and are interested in pursuing this, New Dot Spot is part of a larger group of companies. Our sister company is fully capable of building mature software on budget and on time. Contact us for more information and we will happily set up a referral.

.....

3. New Dot Spot

Here at New Dot Spot we specialize in building prototypes and minimum viable products for new startups. When we made that choice,

**We knew we couldn't attract
new startups by charging
agency rates.**

We knew we wanted to keep our goals aligned with our clients, but knowing our clients as well as we do, we know that change in a new startup is the only certainty. If we did quotes, we would quickly find ourselves in the unenviable situation of having to refuse change requests or break the budget and that's not an appropriate business model for a new startup when you know that the only certainty is change.

When the only certainty is change...

What we came up with is a strategy that utilises the best traits of both tracts. It allows us to keep overhead low, not unlike hiring your own team. It gives the flexibility that new startups need to handle change when change happens. Most importantly, our approach gives us the ability to identify potential technical or user experience issues early.

Validated learning is at the heart of our process and the reason we do things the way we do. We focus on finding out if your idea has what it takes to succeed and if not, what needs to change and how? A professionally crafted, highly scalable and robust software

solution is great, but not if the **idea** has yet to quantify its ability to attract users. Scale is useless for a startup.

You wanted to know how much it's going to cost to build your app, but did you budget for studying the user experience and updating your app over the next 3 - 12 months? Change.

Too Long!

Didn't read!

I want numbers!

We can't tell you what your idea is going to cost without knowing your idea in detail and frankly, neither can you. We can help you figure out what it might cost, but we can only get you in the right ballpark.

If you find yourself about to go in and pitch to a venture capitalist, a finance group, Dragon's Den or Shark Tank... please come in and talk to us. Don't use these figures for that. (If you're doing that you probably don't need this eBook anyway.)

There are four degrees of software (like sunburns, some hurt more)

1. Prototypes (low cost)
2. Minimum viable products (low-medium cost)
3. Internal production software (medium cost)
4. Scalable commodity software (eg. SaaS) (high cost)

1. Prototypes

Cost Range: \$5,000 - \$99,000

Clarity: Low

Change: Chaotic and constant

Risk: Very High

Prototypes have a specific purpose and it's not to make money. If you're considering a prototype it's because you've gone through the discovery process with New Dot Spot or you've done it on your own and you have identified a technical uncertainty. Basically, if you don't know if the technology can do what you want it to do, you should build a prototype and find out.

I said prototypes aren't about making money, but they do have a very high value. Nothing protects investment quite like a prototype. Prototypes are purpose built to defend investment where high levels of uncertainty exist. This is why investors always ask about a prototype. Have you done your utmost to mitigate risk? Here at New Dot Spot, we love to battle risk. We've done it for so long, we can usually answer your questions in our initial meeting. We can identify the technical uncertainty in our discovery process and work quickly to produce a prototype that proves if it can (or can't) be done. We love the challenge.

2. Minimum Viable Products

Cost Range: \$25,000 to \$500,000

Clarity: Medium

Change: Constant, planned, managed

Risk: High to Medium

Prototypes test hardware and software. MVP's test business models and customers. If you have an idea and you want to find out if people will actually buy it, use it, adopt it, or like it - you should build an MVP. An MVP can be built rather quickly and for low cost, but it doesn't last out there very long. At some point you will need to build your full IPS (Internal Production Software) and scale up. **New Dot Spot specializes in this area** and focuses on usability, testability and employs a highly specialized software development lifecycle to help you develop the healthy problem of not being able to scale fast enough. The kind of problem investors are looking to solve.

3. Internal Production Software (IPS)

Cost Range: \$100,000 - \$1,000,000

Clarity: High

Change: Slow, well planned and executed

Risk: Medium to Low

Internal production software is software that is custom built to handle the needs of staff internal to a company. This may also include client access to some degree, but the focus is on service to "internal clients". IPS is custom built to scale to the size of the organization and their capacity for growth over 5 years. This software is constantly under development and commonly built by a team internal to the company.

4. Scalable Commodity Software (SCS)

Cost Range: \$500,000 - \$500,000,000+

Clarity: Extremely High

Change: Very slow

Risk: Low.

Scalable commodity software is a service offered to the Internet at large. It is intended to handle billions of users. MVP's will be (hopefully) replaced by SCS once the business gains traction and scaling begins. SCS is under constant development by large teams of highly specialized developers. You'll often see companies refer to this as their version 2.0.

3 Expenses You (Probably) Failed to Account For

If you've been following along closely you'll know that I already alluded to the most common expense that startups fail to account for - change. Changes to the software after it's built account for far more than the total cost to implement a first version.

Post Launch Modifications

Price Range: minimum 10% of initial budget per month.

If there is one thing that I would like you to know about what this is going to cost it's that it never stops costing. I have never ceased to be amazed by the mindset of "we'll launch when it's done". You will never be done. There will always be changes, new features, and bugs. Always. Software development that slows down or stops, dies - just like a shark.

Even as I write this book I know I'll be coming back and editing it before I publish it. After I publish it I'll continue to make changes and at some point a "second edition" will be released. The day I stop making changes and updating the content of this very book is the day this book becomes irrelevant.

Change continues in all industries, but for some reason, people seem to hold on to this mythical notion that software is somehow immune to change. Maybe there was a day back in the time of gaming cartridges that meant shipped equaled done, but those days are long gone.

Managed Hosting

Price Range: \$50-\$1,500+/mo

Does your app rely on a server to function? Do you know what runs a server? That's right, software and just like we talked above, software that doesn't change becomes

irrelevant. While I'm sure you don't want to use software that's irrelevant, have you budgeted for changes to the software on your server? From the operating system to the software you built for your app, you'll need to keep that up to date. From uptime guarantees to basic security you'll need to keep that up to date. You can look at hosting costs from any provider out there and go with the cheapest or free provider you can find, but you're still going to have to maintain that server. Sometimes people think they can host for free, until they realize that you get what you pay for. Or they think they can go without managed hosting, until the server goes offline and they find themselves on hold with their hosting provider or worse, talking to a level 1 service technician. The ultimate nightmare also happens when they discover that their hosting provider doesn't do technical support for failed Wordpress updates and no, they don't do backups for free.

Roadmap

Price Range: \$1,000 - \$7,500

A roadmap is like a business plan, but for the development of software. Like a business plan, it is only an estimate of what might happen if you execute well and are lucky enough to avoid any unforeseen pitfalls or technical hurdles (did you build a prototype?).

A Roadmap contains descriptions of the primary users, the user story (or stories), wireframes of every screen, outlines of key functionality and is used to organize and identify every component of the project. Once those components are identified, it becomes easier to estimate how long each component might take. Even more valuable, is that those estimates can be tracked and compared as we move through the development stage giving you a clear velocity (how well the project is moving along). With that understanding, it should be easy to see that your roadmap is critical to your project management efforts.

Roadmaps, like business plans, are best done by highly specialized individuals and take a lot of time to produce. They are, therefore, somewhat costly. The primary value of a roadmap is that they provide the clarity you need to answer the question, "What is this going to cost to develop?" If you are going to raise funds, this is the document you need and the answer you produce will be of a level potential investors will appreciate.

The roadmap has many other valuable uses as well. Developers follow the roadmap while building the software. Marketing uses the roadmap to create their message. Design uses the roadmap to build the user interface, etc... As powerful and as important as your business plan, the roadmap is a critical piece of documentation that no software startup should be without.

To answer your question, "How much is this going to cost?" The best way to find out is to build a roadmap.

We love to do roadmaps here at **New Dot Spot**. Contact us if you're ready to get serious, but until then be aware that the next section is very rough and filled with generalizations.

Build Your Own Roadmap

Are you building a prototype or MVP? Try this little exercise to find out how many hours might be involved in creating your project. Once you know the number of hours, email your spreadsheet to me and I'll help you finish the estimate.

Step 1: Do your low fidelity wireframes.

We'll be putting together an eBook on that subject soon, until then, you'll have to resort to Google for instructions. In short, draw out every screen of your app using a whiteboard and dry erase markers or paper and pencil crayons. Don't forget the header, the menu, the login screen and the user profile page.

Step 2: Create a sitemap. Make a list of every screen, page, menu item, and function for your project. Refer to your wireframes. Add to that the things everyone always forgets like the footer and any email notifications that may be sent.

Take that list and put them in a spreadsheet. Give that column the title "Pages". Across the top add the column titles "back end", "front end", "design", "database", "documentation", and "project management".

Using the table of items below, fill in the back end and front end columns. Feel free to add or subtract to these numbers for items you think are going to run longer or shorter.

Page Item	Back End	Front End
Paragraph of Text	0	0.25
Form	0.1 x Number of fields	0.25 x number of fields
Button	0.1	0.25
Link	0	0.25
Image	0.25	0.25
Menu	0.5	1.5
Table	5	5

Pagination	5	5
Dynamic Element	5	20
API	20	5
Email	7	15 (5 for subsequent emails)
Custom Functionality	10	10

Keep in mind that these are just average numbers. You have to adjust them.

Using your spreadsheet skills, you can calculate the design cost by taking the front end number here and multiplying it by 2.3. You can calculate the database costs by multiplying the back end costs by 0.5. You can calculate the documentation costs by adding front and back end together and then multiplying by 1.5. You can calculate the costs of project management by adding everything up in the other columns and multiplying by 0.25. You can then add a column that sums each row. Then sum that column. Then take that number, multiply it by 1.2 for average team velocity and that is your estimated number of hours if you really know what you're doing.

The 'n'certainty Factor

The last calculation you're going to do is your uncertainty factor. Take your last number and multiply it by a number 'n' somewhere between 1.0 and 2.0. Use 1 if you've done this a hundred times before and use 2 if you have no idea what you're doing with these numbers and everything seems really confusing and kind of crazy. We need to know what you can can and can't handle. Show me a quote you've ever received from any other agency that does that and I'll be amazed.

Yes, the entire estimate could end up being doubled, but that's okay. That's why we're here. We'll get your roadmap in shape and help you get that 'n'certainty factor down.

When you've progressed far enough that you feel comfortable sharing your work, free to email it to us. We're happy to have a look and give you some feedback.

Thank you for reading!

Be sure to [sign up to our mailing list](#) for more information on workshops, presentations and future eBooks (like when we do a roadmapping workshop in your area our when the Wireframing eBook is ready, for example).

Send your spreadsheets to:

Michael Borthwick
Founder
mike@newdotspot.com

New Dot Spot
newdotspot.com