

Business Website 1.01

Published on December 12th, 2020.

It is another marvelous sunny day here in New York City – nothing like your regular 24-hour shivery shift in Northern Europe where I grew up in. I go online to check the amount of sunshine people get in NYC probably to remind myself how lucky I am to be here and now. It's 2 535 beautiful hours per year, so, on average I get to indulge myself in a 7-hour sunbath of blissful warmth per day. On contrast, Riga, the capital of Latvia, would allow you to dip your nose and cheeks in a 4.8 hour long (short, too short) keg of glimmer. Even though almost five hours a day on average on paper seems like not so bad of a deal, I honestly remember it being different – less than that, a fraction of what should've been. Most of the time, you would find yourself among the enfolding grayness of the sullen surroundings. Your garbs absorb the soaking moisture out of the humid air that chills you to the bone. The misery spreads into the facial expression and makes you want to hope that everyone else around you feel the same way. I can feel the shivers coming back.

All these thoughts and memories provoke a desire to reflect on my trajectory into the future and my final destination. But, as the three wise men ought to say (I might be making this up): going further requires you to make peace with the past – I decide to use my trusty time machine to travel back in time twelve years ago. I will find my junior self and teach the little bastard a thing or two about websites, entrepreneurship and... well, everything that can make him feel even more miserable. I will enjoy this trip very much indeed.

This ongoing internal dialogue has two major goals. First goal is to reconcile with my past self by admitting to and analyzing my shortcomings of building businesses and websites. The second goal of this endeavor is to help fellow entrepreneurs familiarize themselves with a website creation process: where to start, what to expect, what questions to ask and how much to get involved – a perfect set of actions to rectify the mistakes before they have been made. A rather convoluted concept but totally plausible within the confines of the time-travelling framework.

Essay #1 SEED stage

Web apps, Seeds and Time travel

Back in 2008, I had an idea to open an online store for toys, gifts and gadgets. Since then I had been engaged in multiple e-commerce projects, hired a bunch of web developers, built several web apps and made tons of amateur mistakes. There were frauds, lies, mysteries, failures and some victories along the way. In 2020, I am a professional web developer myself and a slightly more seasoned business person. I couldn't help but notice how aspiring entrepreneurs, founders and CEOs of all ages keep stepping on the same rake that used to whack in my face over and over again. So, I have come to a conclusion that I have accumulated enough experience and it's the right moment to share some valuable insights with you. My personal pilgrimage in a space-time continuum of my consciousness as well as our mutual journey along these lines will be worthwhile.

T minus zero and we have lift off... "Hey! What's up, buddy?"

Businesses differ, says captain Obvious, but they all go through the same development stages during their life cycles. The way I see it, there are roughly four stages a business might go through in terms of growth: SEED, SPROUT, PLANT and GARDEN. SSPG – if you're into that sort of thing. Of course, there is a larger variety of levels, but this amount of vegetation is enough to follow my trail of thought. Shout-out to #gardenersrule. You probably don't have a slightest idea what a hashtag is in 2008. There, there, my cute clueless little caveman. You'll find out soon enough.

SEED is a regular startup that is *probing* the market. It's not a venture capital backed and full-o'-cash kind of startup. This might not even be a registered company yet. The focus here is on building a minimum viable product, minimizing expenses, being ready to pivot, finding the customer, solving the problem, getting paid for the first time.

SPROUT is a small business, a registered legal entity, with growing customer base and revenue. This could be a newly formed company that is entering a known market and is no longer testing the waters. The focus here is on iterating, fixing errors, polishing the product, making existing customers delighted.

PLANT would be an established business with regular customers and steady cash flow. When operating at this level, the focus should be on building a brand: brand strategy, brand

identity, brand personality, brand messaging, voice and tone; scalability, efficiency, processes, sales and communication channels.

GARDEN is an expanding company, a growing brand that is recognized. The focus in web design and development here would be on security, anti-fraud, accountability, social responsibility, trends, rebranding, multi-branding and diversification.

I will split my internal discourse into four phases – one for each SSPG stage. Each phase will have themes or essays that I consider worth looking into. I will be formulating questions and going over them within the confines of the particular phase. My hypothesis is that as business develops, its goals evolve as well, thus the quality, functionality and scale of web applications and their costs should inevitably change to support the transformation. Therefore, there should be a different level of analysis involved in the website creation process at every stage the business entity goes through.

The composition of essays you are looking at right now is meant for the SEED stage bootstrapped startups and small businesses.

I am not trying to make a case that the marketing funnel or social media strategy, for instance, should be attended to exclusively at the PLANT level. These are important activities at every stage of business buildout. However, during the initial growth certain actions are not as significant and should have a lower priority, whereas as the company matures have to become a part of a daily routine.

You might argue that brand strategy and brand identity are crucial at the SEED stage; that if we nail it right at the start and invest heavily into creation of a perfect product, it will promote itself and we're set for life. I would say – sure, it's a really nice dream. If you have lots of cash, tons of time and a steady growing market you can hire professionals to help you with that. Unfortunately, most of us don't have unobstructed access to a hefty gold sack of plenty or have little time to squeeze through the window of opportunity that might only seem to be open.

Truth be told, when you are barely scraping by, strategizing and branding should not take precedence. The most important part at the SEED stage is finding the customer who will pay you for your product and feel good about having done so. When you prioritize strategy over sales at the initial stage, you are consciously choosing to stop the development of your business for whatever made-up reasons you might have. I know what I'm talking about, because I had done exactly that and it was a mistake. Most of your time and effort should be spent on getting sales, but at the same time, that activity would not necessarily exclude strategy and some basic branding if you're sure you can afford it, both money and time-wise.

Whenever there is a task at hand there are actions that are reasonable at one stage and make no sense whatsoever at another. Let me give you an example. If you had a SPROUT type construction company and you had one project that required you to have a forklift, it wouldn't be rational to purchase it. You would rent one because beyond that single contract you wouldn't need the machine. It might've been practical to obtain this tool if you were a PLANT or a GARDEN level business when you have multiple projects and paying year's rent overwhelmingly exceeds the costs of having the apparatus in-house. But for a small operation large, resource-consuming purchases are pointless.

The same applies to websites. It's great to have an e-commerce website like the one adidas has, but do you need to invest hundreds of thousands of dollars when you are a bootstrapped startup testing waters? First of all, I don't think many of us have that much. Secondly, you wouldn't have a clue what to spend it on anyway because you don't know the market and how to find and approach your target audience at the SEED stage yet. You'd burn through cash faster than a dog-pursued scalded cat chasing scattering mice.

You know the primary law of time-travel, don't you? If you ever meet your future self that tries to bore you to death with psalms of unwelcome moralizing advices and then you kill it – there is no future for you. So, I guess, there is no feasible way to make me shut up. And since that is the case... What are we waiting for? Let us look at what you're getting yourself into and try to build us a simple website for you next SEED stage startup.

Essay #2 SEED stage

Simple websites, Car wash and Tsar

This essay will be useful to those entrepreneurs who don't know what a website is made of. Why would you want to know this? Firstly, having at least some basic understanding of the technology behind websites wouldn't hurt. Secondly, you will find yourself harder to be taken advantage of. By the end of this piece you will either learn something new about websites or will learn to look at the website creation process from a different perspective.

“How much for a simple website?” I have encountered this type of inquiry on different platforms numerous times. It seems to me that we are increasingly casually expecting simple solutions to complex problems. It might have something to do with being spoiled

by getting results super fast (in under a few seconds), or being overwhelmed with the amount of choices; maybe it's both. Whenever we are faced with something complicated, our immediate reaction is – *“I want nothing of that, get me a simple answer to my question.”*

Solving complex problems is a tough mental endeavor that consumes a lot of energy and it is no wonder people choose not to act or are drawn towards the promise of a simple solution. We google for questions like: “How to make a simple website” and get “About 3,940,000,000 results” in 0.74 seconds^{1*}. Seeing this ridiculously high number of possible answers inevitably lead us to an overwhelming urge to ‘get this over with’ and clicking on one of the top three links without going further than the first page of the results. Complex problems are hard, irritating and frustrating, and we would rather prefer avoiding them. They contain numerous variables – the backbone of most websites. In order to get to an easy answer, you'd have to break the complexity into smaller pieces and address each of the parts individually. The process is neither straightforward nor fast.

Firstly, you think you need something *simple* but here's a problem – you don't know the technology behind websites and therefore can't fathom what constitutes simple. Does it have to look *simple* like the first page of Google or Craigslist or have some *basic* functionality? And what exactly is – *simple*? If you are making judgments over simplicity based on the looks, then Google's website appears to be really simple, which in fact, it is – in terms of the layout design. However, the technology **behind** this website is so complicated and valuable that the company's revenue is as big as Latvia's GDP² quadrupled. How much would you pay for a website that generates billions every quarter?

Secondly, you have no idea what features your business actually needs and (most importantly) can afford for your excuse of a budget because this is all new to you: you don't know the vocabulary, the ropes, the prices, the difference between complex and basic stuff. You haven't figured out who your customers are, how much you're going to charge for your product and what your revenue would look like – all of it is a big mystery to you at the SEED stage.

Imagine you're at the car wash. You have a few options: an automated soap wash, a hand wash, wax-‘n-seal and a full-service, inside-out, all-inclusive car wash. When you're asking for the basic, simple car wash, you mean the **cheapest** option possible and you know exactly what you're getting and how much it will cost. When it comes to building a simple website – you are not fully aware of what that entails. The point is: when trying to impetuously contemplate over complexity of a website as a whole, you shouldn't view the looks and the inner works separately; you should also take some time to learn about the fundamentals of the technology.

To get a glimpse of what a basic website might be and have a shot at defining what is *simple*, we have to look at the basics. I will assume you know what a computer file is and what folders are. A website is basically a collection of files placed in folders that are organized into a structure. Files have different formats, e.g. *.php, *.html, *.css, *.js and so on. Each of the files contains information (code) that has to be interpreted by the machine (computer) that executes them in order for a website to work. A computer has to ‘speak’ or be able to understand the programming language the website had been written in.

There are different back-end languages that a website can be built with, e.g. PHP, Java, Ruby, Python, etc. PHP is the most widespread one and I will further assume our hypothetical website is written in PHP. You can also build your website with a front-end technology: JavaScript, HTML. Most websites are built with back-end and front-end technologies combined. The most common composition is – PHP plus HTML, plus JavaScript, plus CSS.

In order for our website to be available to visitors, you have to place (deploy) all of the files and folders onto a server that supports the chosen language (PHP in our case). A server is a type of computer made for the purpose of storing data, running code and serving results of computations. Having your website files stored on someone else’s server will cost you money. If your website gets a lot of visitors (high traffic) you will be paying more money. Why? Because you would need to support a bigger infrastructure: more storage, more servers, increased electricity consumption.

“So, how much would it cost me to make a simple website then?” From my experience both as an entrepreneur who had hired developers and a full-stack web developer who had worked with clients, I have discovered that there is no clear, indisputable answer to this question. It’s also the wrong question to ask. The problem of simplicity of a website is a subject of interpretation. On the one hand, every client has their own vision of what a basic website should look like based on their knowledge and, on the other hand, every developer has **their** own convictions based on experience. From what I’ve seen, these two rarely match.

From the clients’ perspective, a simple website is the most common site template they have encountered a thousand times on the web. Since so many companies use the same widespread solution, clients assume that it must be inexpensive and efficient. Couple that with the ubiquitous ads promising *free* websites that “you can build yourself” and you get a justified misconception of what a basic website should look like with an imagined cost expectancy based on false assumptions. Usually, from clients’ point of view, a three-to-five-page website with a contact form and a content management system is a perfect example of a simple website.

From the developers' perspective, a simple website is anything that could be deployed and launched within a couple of hours. This means that there is no actual development or customization – just a manual, dull copy-pasting pattern. However, it's rarely the case that clients are happy with the generic fit-for-all template, hence they request custom features and additional functionality. Whenever there's a request for something out of the ordinary streamlined process, things get complicated and webdevs no longer consider a website to be *simple*.

During the dialogue (that should but doesn't always take place) between the client and the web designer/developer the code words *simple* and *basic* manifest themselves quite frequently and it takes determination and patience on both ends to untangle what these words actually stand for. Once revealed, it isn't such a rare case that for the client they represent the cheapest possible option while for the developer – the least time and effort they can commit. I think of this approach as being neither efficient nor beneficial to either party as it is akin to a race to the bottom. The less you want to pay me, the less work I'll put in. The less work you're going to put in, the less I'm willing to pay you. This spirals down and produces nothing but remorse, corruption and wasted resources.

“How much would a basic website cost?” is an amateurish, shallow and pointless question at the time of inception of your entrepreneurial adventure. That is why I consider this to be the wrong question to begin with. Nowadays when I hear someone say they need a basic, simple website I know exactly what they mean – they want to pay as little as possible for something they know nothing about. It reminds me of a famous Russian folk tale where tsar orders his guilty servant to “Go there – I don't know where, fetch me that – I know not what.” And Ivan the Fool actually goes about this business venture.

1* Results from Google search made by Sergei Nikolajev on 10 June 2020 from NYC, NY, USA.

2* Latvian GDP approx. 34 billion USD, source:

<https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?locations=LV> ;

Google revenue in 2019 was approx. 160 billion USD, source:

<https://www.statista.com/statistics/266206/googles-annual-global-revenue/>

Essay #3 SEED stage

Elephants, Targets and Beacons

The very first question you would want to answer is – why. Why do you think your SEED stage business needs a website? If the answer is something like: *“Everyone has one. I got to have it too”*, *“My competitors have great websites, so I need one as well”* or *“You just have to have one these days”* – then, while being true, these responses do not provide a comprehensive answer to the actual question.

Is it you who thinks you need a website or is it really your business that requires one? Is this an inevitable investment or a nice-to-have-one expense? If this is a business investment, then you should figure out who you are building this website for – yourself, your employees or your customers, and what is it that your business needs to achieve. If you can’t provide answers to these questions then you’ll have issues with setting the goals properly and usually not having specific targets is a bad idea.

So, let’s talk about the goal of your website first. The goal is the thing you will be aiming at – it will help you choose the direction, set the vector and orient yourself and your business along that axis. Your website’s goal could be something like this: getting followers, selling products online, accumulating subscribers, increasing the number of downloads, growing your readers’ or listeners’ base – whatever that is, the composition of your website will have to accommodate for it.

A website is a tool. It can help you get what you want and get you to where you want to be but it is your job to figure out what and where that is. If you want a digital business card, your website can be that. If you want a robust customer-centric adaptive online store, your can have that too.

Understanding your website’s primary goal helps selecting the right medium to use – a blog, an e-commerce store, an online magazine, a podcast, a landing page, a portfolio or a combination of those. You shouldn’t be dismissive about the existing technology either. Look into utilizing such platforms as YouTube, Instagram, Amazon or Twitter – they can be better in many ways and you won’t have to spend nearly as much time, effort and money on setting them up in comparison to the development of your own website.

I had hired developers to design and build my first online store back in 2008. We had everything done – beautiful images, layouts, type, copy, colors, admin panel with content

management— everything except one tiny little thing called payment processing. We were unable to accept payments online. So, basically it was everything but an online store. It sounds ridiculous, right? But this is exactly what happens when you fail to set the goal properly and, unfortunately, this is how many of us approach problems in life – avoid dealing with the most obvious elephant in the room.

It should've been executed in reverse – payment processing first, everything else – secondary. Since my first e-commerce project I had opened several other online stores. Most of the projects were a total failure as I haven't learned my lesson the first time – I haven't answered the fundamental questions: why you're making a website, who are you building it for and what's the bloody goal.

Let's say your plan is to sell products on your website. That means that the primary focus should always be on making sure this can actually happen. Is it important to have a thought-through, well-designed, easy-to-use admin panel to be able to add and alter products? Yes, of course. But will your customers care about your robust, state-of-the-art internal CMS so much that this feature will help you make the sale? You don't have to answer this one – it's a rhetorical question. In my experience, products sell even when images are not perfect, when the copy lacks the right words, when the layout looks amateurish. And I know for sure that the sale will not happen when you can't accept the payment.

Setting a primary goal for a website is crucial. It has to be realistic and achievable as well. I know it's tempting to say: "There is no limit to how much we can make selling this stuff, dude. So why bother spending time on research and analysis – all the boring stuff?" However, the reality is different. There are forces at play that you can't control and foresee: objective limitations, timing, market size, competition, luck and acts of God, if you will. Set your goal – it will help you navigate the sea of constant storms. Like a dim beacon shining, shimmering in the distance among the ravaging waves and savage wind gusts, helping your shabby boat to stay on course. Sounds dramatic, I know.

All of this leads us to the next inevitable question – *"What functionality do I need to build into my website for my SEED stage start-up and why?"* It's a hard and complicated question to provide an answer to right off the bat, because no one really knows what's best for your startup. The answer will be manifesting itself partially during the lifecycles of your business. You would have to fine-tune your website's functionality to aim at specific targets. And you might have a plethora of those, so you have to limit and constrain yourself. You don't need to spend time and money on payment gateways, security and online carts if you're not going to sell online. You don't need to integrate social media if you're not posting relevant content on a regular basis.

You will be making stupid mistakes and then, hopefully, you'll learn what's important for your business and what is not. You should be prepared for that and voluntarily accept it as one of the main prerequisites of being an entrepreneur. If you aren't terrified of the idea of building a website yet and you have come to a conclusion that your business actually requires one, then you should return to the obvious question – “How much would it cost me to build a ~~simple~~ website?” The major difference is that now you know better than to anticipate a shallow answer. Instead, you'll be looking for something else. Let's look into the composition of a minimum viable website first.

Essay #4 SEED stage Shoe epic, Ironman and J.K.Rowling

“How much would it cost me to make a ~~simple~~ website?”

When you are starting out with just one product or service, you should really consider having a one page website. No “About us” page. No contact page. No blog – nothing secondary. These pages are great for SEO purposes, I grant you that, but the downside is that you will have to fill these pages with some valuable content. Otherwise, the web crawlers wouldn't bother indexing that page. You don't have a clue what the content should be or look like and creating a meaningful content is hard work. So, why don't you save yourself from misery and get rid of the pages your business doesn't need right now.

You may think: *“I am hiring a developer anyway and I want to be able to have this functionality, these pages, so I can use them later.”* Great planning! Just make sure these features are within your budget and the links to vacuous extra pages are invisible to the user. There is absolutely no reason to be rude and make your visitor click on the link, just to find itself on the “Under development” page – a total waste of time if you ask me. Imagine, you'd just met a stranger and this person tells you to go check something out, which you do, and it turns out there is nothing out there. What would you think of that person – a pretentious knucklehead, perhaps? Do you want your company to bear such an association?

This type of approach and behavior undermines trust. And trust is crucial for businesses in general, e-commerce in particular and vital for every SEED stage startups without exceptions. People are more likely to send money to someone who appears to be genuine

and trustworthy. Every link on your website hints at some kind of destination, typically – a page. Every page is a promise of a valuable content. Don't make hollow promises.

So, my point is that having decided to build a one page website is totally fine. Being concise is by a large margin better than being all over the place with multiple pages, shallow content, inconsistent design, cursory copy, random business logic and flimsy strategy. The right approach is to build a minimum viable product (MVP). Of course, the scope of the MVP will be different for every SEED business and thus the amount of features, pages, database tables will vary – there is no silver bullet.

How do you know when the amount of content and functionality is 'juuust right'? Setting a time frame and budget constrains works best. They've proven to be a tremendous help at figuring out if you can afford all of the website's features financially and keeping track of whether the window of opportunity is still ajar. Another practical approach is to question yourself on the decisions you're making.

Let's look at the following example. Imagine, you are starting a website for your new shoe company and you assume that an expert blog would be a major contributor to your online business' success. It will help you drive traffic. Therefore, you are planning to include this functionality into your website: add a blog page with images, edit content, share and like it, etc. You have seen professional bloggers and you want to adapt their ways and replicate their websites. Apart from having someone to build this functionality for you, apart from having someone to design the layout, choose colors and typography, you would have to deal with another major problem – the content.

Writing a meaningful piece is not an easy task. It has to provide some kind of value to the reader but first, you have to define who that reader is. Then you have to write a draft. Then someone professional will have to go through your work and point out all the crappy stuff you'd written. Then you would have to rewrite. Then you will have to adjust your initial assumptions about your target audience and rewrite again. This kind of back and forth could go on for a while. So, basically, if you're doing that, then you are starting a blogging career instead of building your shoe company.

Your writing skills are probably not on par with J.K.Rowling's, so what made you think anyone would bother reading your made-up stories? Google will not do all the heavy-lifting for you by magically getting your scribbles in front of everyone just because it's 'finally out there'. Or do you seriously want the search engine's AI to 'believe' in greatness and immeasurable value of your shoe epic? You need to disprove your own reasoning:

"I need a blog."

"No you don't. Here is why: X, Y, Z."

“I still think I do, because...”

“You are wrong. That’s why: XYZ”

When and if you come to a point where your counter arguments don’t hold, then you probably need to build your blog. But both your reasoning and counter arguments have to be strong. Beware of summoning the Ironman to a face-off match with the Strawman – this kind of self-deceitful practice is the worst favor you can do yourself. You have to conjure two equally vicious, opposing personalities to battle against one another and let the strongest argument win.

MVP website is the assembly of **key** elements that have minimum functionality – just enough for your project to reach the defined goal. Keep your fantasies and desires along with “I don’t need this now” mentality in mind at all times. You will find the right balance of features for your project through constant negotiations with yourself.

Now your question can be formulated like this: “How much would it cost me to make a **MVP** website?”

Essay #5 SEED stage Piggyback rides, Baskets and a Lovely bonus

“Which technology works best for my web project?”

At the SEED stage you need to get your MVP website out there as quickly as possible, so the technology shouldn’t be your primary concern for now. You will have to address this problem at later stages as your online business scales and your company grows. So, unless you’re creating some kind of software, there is no need to bother investigating which programming language or web stack is better. I would go with a most widespread technology (a PHP-based website at the moment in 2020), because there’s just more skilled people who can help when things go sideways.

“What are the costs and are there any unavoidable expenses?”

I’m glad you asked. There is a domain registration fee and there are hosting expenses. Select a hosting plan you can afford. Shared hosting option should be the least expensive one but is available only to open-source software (e.g., Magento, OpenCart, WordPress), frameworks and when you’re developing from scratch. Choose a service provider that has the proper support for your chosen web application. WordPress (if that’s the choice) is usually available on most hosting servers but it’s best to double check with the provider.

It’s a good idea to have the servers located closer to your target audience – this will improve the loading speed and search engines’ rankings of your website. How closer? The closest you can find. You should be able to get a decent hosting from 5 USD to 10 USD per month. Many hosting companies offer a bundle: a domain name, email support and hosting. So make sure to check it out.

Online website builders (e.g., Shopify, Webflow, Wix) host their respective applications on their own servers. This means that you will not have control over the hosting, which might not be such a bad idea for a non-tech-savvy entrepreneur at the beginning of the web journey, but you will neither have all the flexibility nor access to all of the server settings which in most cases is OK unless you’re building something ‘techie’. The monthly costs are typically higher in comparison to the shared hosting option but you are not paying for the hosting alone, so it isn’t that straightforward to assess which one is more cost-effective.

Can you get a hosting service for free? Sure. You can use your own personal computer but you would have to know how to set it up and then run it 24/7. There is also an AWS (Amazon web services) option that offers some free tier services for a year. Just a word of caution – without a certain level of technical skill it could prove to be quite challenging. Bear in mind that you would also have to set up your own email server. Not just an email address but the server that is capable of not only sending but also **delivering** and receiving emails. That is – in case you want to have a professional email like peas@ina.pod, of course.

Another way to host your website for free is to find a friend with an active website and ask for a piggyback ride. Basically, if your friend has a dedicated server or uses cloud hosting, it shouldn’t be much of a problem to deploy your website there. This won’t cost you a penny or you could agree to split the costs. Asketh for help and thou shalt receive.

Next stop is to purchase a domain name that is cheap. Up to 12 USD a year should do it. When (**and if**) your business starts generating income, you will be able to afford a catchier name and redirect your traffic without losing SEO rankings. Instagram, for example, as we

know it today, started with a totally different business idea and name – Burbn. It seems to me that they’re doing just fine, so don’t worry about the ‘perfect’ domain name too much.

In case of an e-commerce store, you would need to obtain an SSL certificate for security reasons. The cost varies from 10 USD to 200 USD a year or sometimes even more than that. Some platforms (e.g., Shopify, Webflow, Wix) do not let you choose where to host and don’t give you access to their servers, but they come with a preinstalled SSL certificate that is included in monthly fees. There is an organization that issues these certificates free of charge called “Let’s encrypt”. If you’re not into spending money on the certificate, consider hosting your website with providers that come with a free “Let’s encrypt” SSL certificate. If you are running your own server with AWS, you should be able to install this free certificate yourself. When your business starts making money, please [consider donating to “Let’s encrypt”](#) so that they can keep this great opportunity available for small businesses.

If your choice falls on WordPress, then, providing you’ve chosen a free theme, you can expect the cost of setting up the website to be around 100 USD. An experienced WordPress developer should be able to deploy it within 2 hours. This price tag does not include any design changes or content creation. Your website will look exactly like the theme you’ve chosen, with no difference whatsoever. If you ever find yourself receiving an offer to have a custom made website for 100 USD – you are either **extremely** lucky (which I doubt is the case) or someone is trying to take advantage of you by selling something you could get for free. Usually you would also get a couple of ‘lovely bonuses’ to sweeten the deal: a custom made logo that had been generated on a free-logo-generator website, your email included in digital marketers’ (aka spammers) database and a piece of malicious code. Everything cheap comes at a price.

“What drives up the costs?” Features, products and pages are the key factors. The more elements, plugins, add-ons, apps, extensions, pages and products you want to have, the more complicated your website gets. If you are using a free plugin that enables a certain feature, it usually enables more than just that one thing and that’s an overkill for your website’s performance. Usually these plugins or apps come from different third-party developers and sometimes these add-ons conflict with each other. In addition, they all fight for the same resources, are not sequenced or fine-tuned to work in unison. So, if you want your website to work well, every feature you’re adding to your website has to be thought through and developed to co-exist with everything else in the application.

When it comes to pages, you would come to a point where you would have to make a decision how they should look and whether or not they should all appear the same. Website owners usually want their pages to look differently, or to be more precise – have

different layouts. That means that someone has to design those layouts and then code them. You could use a free theme that has multiple pages, but you'd have to accept the way it looks and also the fact that this design is adopted by thousands, if not hundreds of thousands of other websites.

The quantity of products is another major contributor that has an impact on the costs. Every product has to be entered into the database. It has to have images, variations, description, title, price, tax, availability, quantity and so on. If you are selling just one product or service, that shouldn't be much of an issue, but if you have more, it will take time to input all the details. You can hire someone to do it for you, sure. And I'm certain that you can outsource it to someone who is offering a really low price. When doing so, please consider the risk of having to redo everything, in case the job is not done, let's say, conscientiously enough. Because cheap doesn't necessarily equal to diligent and attentive.

So far, so good. Have I ever personally experienced any complications with this? In fact, I had. My first e-commerce gift and novelty store had been developed by a web development company in 2008. They had produced layout designs, admin panel for content management and provided hosting for additional 20 USD a month. They were supposed to provide maintenance and bug fixing services for a period of two years providing this monthly fee arrangement had been met. Bottom line – I was responsible for a development fee and monthly payments and they were responsible for making sure everything worked.

Once the honeymoon was over – the initial web development service had been paid for, the winds have changed. Constant battles over bugs, snags and deficiencies became a weekly routine that went on for two years. When push came to shove, I had absolutely no leverage other than those 20 USD and so I became a full board hostage to those dungeon dwarfs. In the end, I refused to pay my monthly fee because they failed to fix some ridiculous bugs and they refused to put my store back online because I refused to pay. This was a typical vicious lose-lose situation where I had lost considerably more: the code of the website I had paid for, customer database and a peaceful slumber.

Was there anything I could have done to minimize the damage? Now I know that I could have. Try to avoid keeping all eggs in the same basket. When you commission someone to build a website for you, make sure you have all the code that you have paid for. You can get domain name, hosting and email services from one provider (or even three separate providers). There is absolutely no need to use a one-stop-shop. It seems very convenient, I know, but you might become a permanent guest to your dungeon keeper.

Now, your question can be slightly adjusted: “How much would it cost me to make a MVP website with **XYZ features?**”

Essay #6 SEED stage Pivots, VCs and Failures

“How much time, resource and effort should I put into this website?”

There is no golden rule of thumb or anything. You should invest as much as needed to reach specific business goals. My personal advice is to be realistic and expect and perhaps even plan for failure. It's not a given that your initial business idea will be successful. The statistics are there: “Roughly a third of new businesses exit within their first two years, and half exit within their first five years^{1*}.” And this is not a happy sell-to-VC-firm kind of exit; this means shutting the business down.

“A third of small businesses get started with less than 5.000 USD and 58 percent got started with less than 25.000 USD^{2*}.” I'm sure you see a correlation here. Every third business fails in two years and every third business' initial investment was less than 5k. Half of businesses fail within five years and half of businesses invested less than 25k. I'm sure the data isn't sufficient to prove anything, but there is still something to it. Based on my personal experience and that of my fellow entrepreneurs, we tend to underestimate the required financial input and overestimate the income. We make ourselves believe in the fairy tale and hope that it will come true while turning the blind eye to the unpleasant reality.

What made you think that your business will succeed and not go bankrupt as almost half of all businesses do? Venture capital backed start-ups, for instance, suffer the same fate as regular companies. When VCs assemble their portfolios they actually plan that 6 start-ups will fail completely, 3 will return the original investment and only one of them will have a substantial return. That is what they're aiming at, but the reality might be different. Now, it might look great to you – 10% of start-ups can guarantee a substantial return, but let's not forget that these are vetted, audited and supported companies. They get connections, mentorship and financial support from their investors. It's not the same for a regular bootstrapped SEED business.

Improper prioritization of activities, excessive time and resource investments into a handful of assets isn't the wisest set of actions for a SEED stage company. Get ready to pivot mentally and financially. Your website will have to be reimagined and redesigned sooner than you might think.

“But am I not setting myself for failure – this way of thinking? Shouldn't a true entrepreneurial spirit strive for making it happen no matter what?” Yes, that is true, of

course. The only problem is that most of the time you (and I mean everyone) is clueless about what's the next big thing in the market and whether your value proposition is viable for the current market. There are too many variables at play to be able to predict the success of a venture:

Is the product right?

Is the packaging done right?

Is this the right time for this particular product?

Is this the right market?

Are we targeting the right customer?

Is this the right team for this business?

What will be the reaction of my competitors?

Is there anything better on the market that I'm not aware of?

And then there is an inadequate you and your biased business partner. Try putting all that into an equation and see what the outcome will be.

Time is arguably the most important of all variables. You need to set specific time constraints to build the website and another set of constraints for the website to achieve your business goals. Think about building your website within a specific period of time, keeping your eyes wide open for the narrowing window of business opportunity. If you figure out that time is pressing, you might have to hire additional people who will have to work extra hours that will inevitably result in additional expenses.

How long can you afford to lose money financing your project from personal savings? The longer you are able to endure, the better your chances get, unless the market is shrinking. But in order to keep the company afloat you will have to have cash. And you can get cash by selling products. So, if you are not selling or your sales are going down, it doesn't make sense to wait for better times – there won't be any at the current disposition. Pay attention to the variables and figure out what can be rearranged. Set a time frame for each business goal you have and a deadline for completing your website. You will be wrong, the deadlines will shift, but at least you would have something to aim at.

If your website is not meant to make money, the above pivoting mentality still applies. You will be investing time and resources to build something. You will be aiming at meeting certain objectives: followers, likes, readers, viewers, subscribers, etc. You will have to make changes if things did not work out the way you had initially planned.

Let us reformulate the question again: “How much would it cost me to make a MVP website with XYZ features in **N days** and maintain it for **T time**?”

1- <https://www.jpmorganchase.com/corporate/institute/small-business-longevity.htm>

2- <https://smallbiztrends.com/2019/03/startup-statistics-small-business.html>

Essay #7 SEED stage Toothpaste, JAMES and 3 seconds

At some point or another, depending on your business model, you will have to monetize your enterprise. For this to happen, you will have to identify and then differentiate your target audience from your product’s users and your customers. Someday you will also find out that not all of the customers are made equal. But for now you have to prepare the ground for a strong foundation and figure out who your core target customers (CTC) are.

Why would you want to do that? If you don’t define the pool of your potential customers then **everyone** is your customer which is a terrible idea for any SEED stage business. If you don’t extract the CTC out of this pool, you will be wasting resources and efforts chasing after the wrong fish.

“My product is so good – everyone needs it. Great products always draw customers like flies to ...” Honey, sure. But no. That’s where every overly optimistic inexperienced entrepreneur is wrong. It might be the case that in the future your product will be so good that everyone would want to have it (still, highly improbable). Right now though, at the SEED stage no one cares about it because no one knows it exists. And so your job is to find people who are ready to believe in you and your team and purchase your crappy stuff, and be so excited about doing so, that they would become your product’s ambassadors. Click here if you want to know why your new product is a disaster and why everyone is terrified of it.

“How do you come up with a CTC profile?”

You should spend some time conceptualizing who your ideal customer might be. That’s someone who is eager to make a prepayment for your product even if you don’t have it in stock yet, even if they have to wait, even if they have to commute for hours to get to you. You should come up with a clear profile of such a person and then find them. When it

comes to building your [MVP web application](#), the very first public version of the website should be made for that particular person. Don't build it for everyone (especially, not for yourself). Develop it for your core target customer.

Let's imagine you are introducing brand new toothpaste to the market. Everyone has to brush their teeth, correct? Wrong! Almost all newborns, the majority of infants and almost every fourth senior over 65 don't have teeth. They might need a toothbrush for gums, but they don't need the toothpaste. That's roughly 25% of the population (in the US), who don't need and, frankly, probably don't want your product. And how about those who don't brush their teeth out of sheer spite, like pesky puberty teens?

Next, there is an abundance of toothpastes in the market – a huge variety of scents, colors, shapes, volumes, with different properties, badges, endorsements and promises. Some of them have a huge advantage over the others – they have an established name, they are available on every corner and customers who had already taken their chances know that at least their teeth didn't fall off on the first go. Your new toothpaste doesn't have that edge yet and surely it will be challenging to graft early confidence.

How is your product different? You have to have an answer to this one. Let's assume that apart from making your teeth cleaner, your miraculous substance helps in preventing or treating gum disease. That's great! Who has a higher chance of getting exposed to that infection? Is it smokers? Is smoking allowed for minors under 18(21) in your state? That's another 10% of population that you should deduct from the total target market. Which state are you targeting? Is it New York? What is a women-to-men ratio for this state? Do you think that there are more male than female smokers? Do you have the data to support your opinion? Are there any racial or cultural differences? Is it true that Hispanic and Asian people smoke less than White? What should the income of the person look like to be able to afford your toothpaste? Do these people have a more affordable alternative solution to this problem? Do they have a dental plan? Do they **believe** that having healthy gums is a good idea? The more questions you can come up with, the more answers you can provide, the better your understanding of the core target customer will become.

Your CTC's profile could be something like this: "JAMES the Driver" – a 35 year old single White male smoker with 35k - 60k USD annual income, without dental insurance, no kids, living in greater NY area, thinking that he's probably "gotta do somn' 'bout 'em gums". Now that you have defined who your CTCs are, you would have to figure out how to get your offer in front of them.

"What does this have to do with my website?"

Everything. Your website has to provide a clear, concise value proposition for JAMES. There are 3 seconds^{1*} while the page loads and then you have another 5 seconds to deliver the right message. Think in terms of an elevator pitch. The longer the first 3 loading seconds last, the less time you would have left to pitch the proposal. So you have to be sure that JAMES gets to see it fast while simultaneously making sure it's comprehensible. If by the time of 10 seconds (as we all know, $3+5=10$) JAMES hasn't left yet, and mind that you are doing something right if less than 40% of JAMESes bounce^{2*}, let him buy your product. Extend your arms and embrace the warm feeling of money. Hug and caress JAMES – he deserves it.

Now, you can expand your question even further: “How much would it cost me to make a MVP website with XYZ features in N days that will help me attract **JAMES'** attention within T time?”

1- DoubleClick by Google found 53% of mobile site visits were abandoned if a page took longer than 3 seconds to load. Source: <https://developers.google.com/web/fundamentals/performance/why-performance-matters>

2- Bounce rate benchmarks. Source: <https://cxl.com/guides/bounce-rate/benchmarks/>

Essay #8 SEED stage

Neurohelmets, Layouts and Kettles

If you have established who your core target customer is, it will become much easier to come up with the design for your website. Since you are at the SEED stage, on a budgetary diet and can't afford to hire a decent designer, your chances of building a web masterpiece are slim. But still you rightfully wonder how your website should look like.

Seriously, how should anything look like if it doesn't exist yet? Search for similar things. If you're designing an electric Wi-Fi enabled kettle and it looks like a shoe box without any hope for a handle, your product will not be appealing because consumers will be looking for a thing that can hold water, boil it and have a stick attached to it so they can carry it around without having to burn the skin off. Your competition is a great source of inspiration for both your product and website's design.

Does your product or service have competitors? If you think you don't have any direct competition because you are that unique, then there should be indirect substitutes. If you fathom along the lines of: “Our product doesn't have competitors”, then, sorry, but your

product doesn't have a market. Move on. "We are developing a brand new technology – a neurohelmet that you put on before going to sleep and you can experience any dream you want." That's great! The market is huge. But there is another solution. I'm using it on a daily basis – I go to sleep every night and I see dreams free of charge. And I actually kinda enjoy the element of occasional surprise, you know.

There is always competition. Even if there are no direct substitutes, people can go and do something else instead. Or they can choose to do nothing and continue their suffering without your miraculous snake oil. You will have to contend with that as well.

Pick the strongest competitors and research their websites. Try to spot things that you like, things that you dislike, things that could be better in your opinion. Look at the colors and type they use. Try to figure out who they're targeting and what they might be missing out on. Keeping that in mind, you don't necessarily have to copy everything (and you actually shouldn't), especially if you did a better job at figuring out your CTC, but you could surely implement some ideas. The fact that these companies are in the business is that they must be doing something right. Figure out what that is and what you could do to make it even better.

I assume you remember the MVP, minimum expenses and ready to pivot concepts. I also conclude, you have agreed that the perfect looks aren't important at the moment. And therefore we have decided to go with the cheapest possible option – to use WordPress for our website (you didn't have a say in this, but, oh, well...). It doesn't have any upfront payments and you can set it up either yourself or at an affordable price. We've established that there are thousands of themes available. All that's left to do is to spend some time selecting the one that is right for you. Tsk-tsk. Not for you, but for JAMES, remember?

As of April 2020, there were 3926^{1*} WordPress themes. It will take you months to select one unless you restrain yourself. Use filters and search, juxtapose results to your competitors' websites and distill the most important parts into a one page layout. Take three letter (or A4) sheets of paper and attach one to another with a scotch tape so that you would have 3x length and 1x width – that will be your screen. It will look like a gigantic smartphone that doesn't fit in anyone's hand. It doesn't matter if you can't draw – scratch, scribble, scrawl or doodle how you think your web page should look like. Aim at throwing out your first draft and repeat. Your free DIY master class is over.

There are three major blocks in a **common** layout: header, content area and footer. Header is where your logo and menu will be. Footer is where your contact details, privacy, terms and conditions are. The content is a place where you will describe your offer. The first

thing JAMES will see is the so-called ‘hero’ section. That’s where your value proposition has to be. That’s where the clock starts ticking.

Set yourself some constraints. For example:

- analyze competitors – 3 days;
- sketch five page design options – 2 days;
- investigate available themes – 3 days;
- pick the most lucrative layout – 1 day.

If you allow yourself to spend more time, you would find yourself overwhelmed with options, get anxious and indecisive. Remember, you will definitely have to change everything anyway – no matter how well-planned you had it initially. Don’t overcomplicate things. Less is more especially at the SEED stage.

The time has come to update your question again: “How much would it cost me to make a MVP website that looks like **ABC** with XYZ features in N days that will help me attract JAMES’ attention within T time?”

1 - source <https://WordPress.org/themes/> (29 April, 2020)

Essay #9 SEED stage

Apple orchards, ROI and Suffering

You are starting a new SEED business and creating a website for it. You already know who you’re targeting, what kind of web application you want to build, what features should it have and how your web pages should look like. You are also aware of your time constraints and expect something in return for your labor – some kind of reward: money, fame, acknowledgement or influence.

Let’s say you have this idea – a seed – that could grow into an apple tree and later someday perhaps transform into a beautiful apple orchard. But right now what would you have to do (apart from dreaming about it) to start making this a reality? You know that it can’t become a tree and bear fruit overnight, that it would take time for the seed to grow and develop and transform. What would you do with the seed today to help it become a sprout? Perhaps, you need to plant it in the soil instead of tossing it on the ground. You

might need to ensure certain amounts of temperature, sunshine and water. You might have to protect it against excessive watering, cold and heat. You will constantly have to pay attention to the environment, conditions, obstacles and opportunities. Are you ready to kneel and get your hands dirty “removing the weed”, aka your personal biases and inadequacies? The point is – you will have to pull your socks up and go that extra mile of blood, sweat and tears to merely have a **chance** of sprouting your business idea.

The market is under no obligation to make you successful. The odds are stacked against you. There is no guarantee that your actions will bear any fruit. One thing you can claim a guarantee on is this – you will suffer. Make no mistake about it. The only way to make it through and not to succumb to an overwhelmingly enticing desire of self pity and universal anger is to accept the suffering **voluntarily** and to aim at something worthwhile. You should know it will be hard, you should know that you will probably fail and lose some if not all of your money, but even when the probability of your success tends to zero, it's still best to take a shot at it.

Realizing that your start-up idea is probably another screw-up of yours, it totally makes sense to act inside a structure of restricting parameters to test your hypothesis. If you don't hit certain targets within the confines of a specified framework, then it's best to move on, isn't it? How does one set such parameters to determine this threshold – this fine line between failure and non-failure? I find it useful to pinpoint business goals, allocate resources and measure the return on investment (ROI).

Business goals are the overall value you are aiming at. It could be revenue, subscriptions, donations, followers, viewers, likes or shares, for example. It should be something quantifiable. What are **your** business goals, by the way? I know you don't have the answer to this question – no one does. But it doesn't mean you can't get closer to finding the answer or getting further away from being wrong. Your daydreams about being on the cover of the magazines as the entrepreneur of the year are pathetic and unsubstantiated. You should stop avoiding this question, figure out what it is that you are after and grow the hell up.

Your resources are: time, money and effort that you are going to invest to get closer to reaching your business goals. There will be obstacles in your way that will hinder your progress and produce negative emotions; there will be boons that will facilitate advancement and make you experience positive emotions; there will also be everything else – things that will neither help, nor obstruct your movement towards your goal. You will have to make effort, spend time and money to identify those boons and obstacles and fathom how to stumble upon the former more often and get through the latter faster.

The simplified version of the ROI is akin to an interest rate you could get from a bank. If your annualized ROI yields more than what you would get from a bank deposit, then it's probably a good idea to continue with your project. Otherwise, the risk of losing money is bigger than just having it stashed somewhere relatively safe. When it tends to zero, maybe, it isn't such a bad idea to pivot or, perhaps, to get it over with.

Let's finalize our primary question: "How much **X** should I **invest** to build a **MVP** website that looks like **ABC** with **XYZ** features in **N** days that will help me attract **JAMES'** attention and acquire **Z(ero) value** in **T** time?"

I know, you wanted to have a basic web app for your SEED start-up and you wanted to have a simple answer to a simple question – how much would it cost you to make a website. You were anticipating a confirmation to your hopes and dreams of this being a fast, easy and painless problem. I'm so sorry to have disappointed you. After all that we've been through, there's this ugly, tough and painful question in front of you. It looks like a cow's dung that you have to delve into with your bare hands knowing perfectly well that there is absolutely no guarantee that a rose might be buried somewhere underneath it. And here is another bonus for you on top of that pile – now you know that if you don't answer this question, you will blame yourself for not doing it when your startup fails. Because you knew **exactly** what you had to do but had decided to take a cheaty shortcut.

As you've probably guessed by now, apart from making you aware of the web technology and affixed challenges, my side objective is to try to dissuade you from taking the entrepreneurial path altogether. If you haven't changed your mind yet, we can continue pursuing our goal – creating a website for our SEED startup. All that is left now is to find the right people to make your website project happen. Let's find out how easy that is.

Essay #10 SEED stage

Rockstars, Unicorns and Ninjas (RUN)

Your startup is at the SEED stage. You have formulated the first scrupulous question about the features, target audience and business goals. You have your time frame and budget in place. What are your next steps? Figuring out who exactly is required for your web project and where to locate these people are your next obvious priorities.

Since you're building a specific type of application – the website, searching for anyone with coding skills like a software engineer or an IT developer might not yield the results you'd expect. The same is true with respect to a graphic designer. You don't need just anyone, but rather a person with a particular skill set – someone who can build you a website. What you are looking for is called a web designer and a web developer. I will further refer to these specialists as **web D&Ds**.

To find a decent web D&D I would recommend starting with your family, friends and colleagues at work. Ask around whether they have or know anyone who has had an experience with creating a website. If you were unsuccessful at finding a web D&D among the people in your close network, you would be looking at classifieds, freelancer websites, fivers, upyourwork and similar hornets' nests.

I'm giving you a heads-up here: the majority of the '*regulars*' on these platforms are there for a quick buck. Don't expect to find a devoted partner who would fall in love with your business idea and put in her/his best effort into making it great. I'm not saying you can't stumble upon a good developer there. All I'm saying, that it's going to be highly unlikely. Like a one in a thousand chance. At first it would seem like there's plenty of great pros at low prices (just what you need) with amazing all-5-star reviews, so you can have a luxury of being picky, but the reality is **so** much different from what it appears to be. Your venture is at the stage where you're probably looking not so much for prowess but for interpersonal skills, trustworthiness and reliability. So virtual badges, digital certificates and unsubstantiated claims of hundreds of successful projects are pretty much irrelevant.

Browsing through ads and looking at the candidates' profiles, you might encounter the following names / occupations freelancers assign themselves: full-stack developer, front-end developer, back-end developer, UI/UX designer, web designer, graphic designer – that's a mouthful of different skill sets and professions. It isn't such a bad idea to familiarize yourself with what these titles stand for. Let's try to understand the differences between web specialists.

Front-end developer deals with how things look and work on client's (your user's) device. When you open a website in your browser on your computer or smartphone, everything that you see is the job of the front-end developer.

Back-end is what you don't see, but the code is there – on the server, and someone had to make sure it works as intended. Adding users to a database table, sending notification emails, handling payments – that's a back-end dev's job.

Full-stack developer is someone who can do both front-end and back-end.

User Interface (UI) designer makes all the buttons and images look and interact the way they do. The job is similar to the front-end dev's but involves a more in-depth analysis of why interface elements should look and perform in a certain way.

User Experience (UX) designer focuses on the user journey and the experiences along the way. When you're adding an item to your shopping basket, notice what happens when you click the "add" button – you either stay on the same page or get redirected to the checkout page – that's the job of the UX person to decide.

Web designers can produce an artwork that they know can be implemented using web technology like HTML, CSS and JavaScript, but they might not do the actual coding. Web designers can do a good job with drag-&-drop website builders.

Graphic designers are not limited to the confines of the code and can produce pretty much anything in terms of visual elements and layouts; however, that doesn't always mean that their work could be a good fit for the web due to the fact that websites have to be fast and responsive. The work produced by a graphic designer might have to be accommodated for the web by a web designer or a front-end developer.

Whom should you hire? The decision on involving any of the above professionals will depend on your business objectives, your budget and timeframe. I would argue that for a SEED stage startup it's best to hire a jack of all trades – a full-stack developer in our case. However, this is not a 'fits all' solution. When you are utilizing a generic free theme template without serious layout changes or any server-side logic, a front-end developer can do the job well. If you're customizing the designs heavily, you might want to tilt towards hiring a web designer. If you're into altering or creating a new plugin, then you'd do better with a back-end developer.

There are several other types of the 'so-called' web developers, but you will not find anyone admitting to that. There is a CMS specialist and a WordPress enthusiast. The former is someone who had experience with deploying or installing (or activating) a certain CMS platform, knows the features and can operate within the confines of that particular

system. The latter can deploy a WordPress website using an existing theme with existing plugins but will be unable and unfit to design or even change the given layout. Don't expect that either of two would have the knowledge, skill or experience to make changes to the code without breaking the functionality. When you know which theme, plugins and apps you want for your website and you need someone to either deploy it on the server or go through a basic setup, these two types are (should be) the cheapest option you can get.

Heaven forbid you encounter one of the following specimens: a code ninja, a rockstar developer or a unicorn desdev. All of these personalities are clearly possessed with illusions of self-grandeur. They all relish a baseless fantasy of their superior skills in every imaginable web-stack that had ever existed, so unless you can't harness your curiosity, I wouldn't recommend looking for these fantastic beasts and try to figure out where to find them.

Seriously, why would anyone call themselves a ninja web developer? Is this a subtle sense of humor that everyone is supposed to understand and appreciate? Think about it – a ninja is a trained war machine with a purpose. Ninjas wear face masks so that other people are unable to identify them. They live and act in the shadows. They are skilled with deadly tools. They have mastered battle tricks and use the element of surprise. I think that anyone who proudly promotes this occupation of a code ninja has exactly the intensions of a shinobi – to deal massive amounts of damage anonymously in the shortest period of time.

“And what about rockstars and unicorns?” Do you want me to analyze those creatures too? Fine. Rockstar developers are easy to define – they are always late, arrogant, drunk, and they don't give a shit. I doubt you crave for this upstart to crash your next web project party. The examination of *know-it-all, been-there-done-that, front-to-back-inside-out* unicorns is a much easier task since there's nothing to talk about, really, because they don't exist.

Essay #11 SEED stage

Students, Gnomes and Shame

I used to imagine and accept it as a fact that a website was a sort of servitude you'd get yourself into voluntarily. First, you would commission web developers to make something for you. Then they would stash it somewhere out of sight like some precious gem or their best masterpiece. And then you'd pay your monthly dues to these brass-bearded hunched gnomes in lab coats so they would graciously agree to consider tinkering with whatever it was they had conjured. But I could've never imagined that I would be ransomed to get to the source – to get the source code that I had paid for. Well, I couldn't imagine many things back then. Here is hoping to help you along your journey, young apprentice. (*Stop patronizing, you maggot!*)

When I was opening my first online store back in 2008, I decided to hire a web developer, because I didn't have the skill set to do the job at the time. I had made the first mistake by not turning to my close network of friends and relatives for help and went straight for the classifieds. I had found a student who was willing to take the job together with his friend for 500 USD. I was ok with spending this sum for an e-commerce website with admin panel and a custom design. I had drafted a legal agreement, met with this 17-year-old guy, made a 100 USD deposit, took a copy of his passport, his contact details and address and moved on to other pressing business matters while being assured that the website was in the works.

Two weeks later I received a JPG image of a main page. It didn't look impressive but I thought it was just a draft, so things would get better. Long story short – after about another two weeks the dude had switched off his phone and took a hike with my deposit jingling in his pocket. That was a major setback for me as not only did the timeframe shift – I had lost at least a month, but my faith in making this within a certain budget had staggered.

What had happened and who was there to blame? The guy who got off the grid? Of course he was, but not solely. This was on me – I was supposed to know better than getting into a business relationship with a student who had zero real world experience. I understand now why those guys had vanished – they've promised something they could not deliver. With the technological landscape being so much different in 2008, the kind of project I wanted done with custom design, CMS and e-commerce functionality was worth **at least** 5000 USD back then. They've underpriced their services at 10X. It's no wonder they had 'apparated' once the self-cast *Fantassium Believiosa* had been dispelled.

My lessons learned were lucid: I was supposed to foresee and workout the probable scenarios; I should've established clear weekly milestones to keep my finger on a pulse of the project; I should've secured more offers and had a backup team to step in. Instead of asking the right questions, I had chosen to stay in Wonderland because it was so nice and calm and peaceful there. I didn't want to hear anything that could permit the idea of a failure or contradict my self-hypnotizing beliefs. I was looking for someone to confirm what I wanted to hear:

"Can you build this?" – Yes.

"Can you make it look like that website?" – Yes.

"Can you build this in a month?" – Yes.

Now I know that I was supposed to ask different questions or, rather, ask the questions differently:

What challenges do you see with this project?

What complications do you think might occur?

What's the backup plan if this doesn't work?

What are our options in terms of technology for this project?

What happens with the work you'd already done if you abandon the project?

Don't be afraid of asking difficult questions. You are partnering with a stranger to do something together. It should be a mutually beneficial business relationship, not an indentured servitude on either side.

This wasn't the only episode I have had dealing with web enthusiasts. Here is a good summary of my experience with web developers. It is actually one of the reasons why I decided to write these essays – to help fellow entrepreneurs familiarize themselves with the technology, to lower expectations, learn to ask uncomfortable but necessary questions, acknowledge the danger of wishful thinking and voluntarily become an active participant in the process of website creation. Here it goes:

*I dreamed a dream in times gone by
When hope was high and life worth living.
I dreamed of website so divine,
I dreamed webdevs would be forgiving.*

*Then I had dough and needed aid.
I'd find a nerd who wouldn't waste it.
There'll be no ransom to be paid
For website fast, unique and blessed.*

*Webdevs calling me at night
With their voices soft as thunder,
As they tear my phone apart,
As they turn my dream to shame.*

*And still I dream he'll come to me,
That we will build websites together.
But there are dreams that cannot be,
There is cloud tech I cannot weather.*

*I had a dream my site would be
So different from this hell I'm living,
So different now from what it seemed.
Web nerds have killed the dream
I dreamed.*

Essay #12 SEED stage Bombings, Easter eggs and Hard clients

“How many web development and design offers should I get?”

It's always good to have some options on the table. I should probably amend that – it's a good idea to have the right balance of options on the table. You don't want to have only one offer from one web D&D, because in that case you're stuck with it and you might become a hostage to the situation. Then also, one offer means that there is only one opinion and obviously it is not enough. Another extreme that you don't want to find yourself drowned in is too many offers. The problem here is that you will be overwhelmed, become irritated and as a result, will probably make a bad decision.

Let's say you have decided to publish your request on a freelance platform or in classifieds. Receiving a hundred offers from different freelancers and a bunch of companies wouldn't be such a rare occurrence in this scenario. However, going through all of the offers thoroughly would be ridiculous – you shouldn't have that much time, you've business to launch. But a cursory examination on the other hand will definitely yield poor results. So, you have to decrease the amount of submitted offers and set a procedure to filter out the irrelevant ones.

First, you need to improve on the input. The quality of proposals you get will always correlate with the quality of your posted request. If you have asked: "How much for a basic website?" then don't complain when you get absurd replies. You need to publish a thorough considered brief to have a chance at getting meaningful offers. Secondly, you need to find a way to filter the output. Some responses will be too generic, some will look like a copy-pasted ad letting you know that this is a carpet bombing meant to produce a random hit by one of the thousands shards.

As a goal, during this distillery process, I'd suggest abstaining from having more than three offers to evaluate, maybe five – tops. Aim at that and put a procedure in place to facilitate the selection. Here are a few practical tricks to have under your sleeve. Include an Easter egg question to make sure the candidate had actually read the brief. For example, somewhere in the middle or towards the end of the request ask their opinion on who is stronger – a bear or a shark, and explain that you want this included in their offer as a first sentence. If you don't see the answer to this tricky question as requested you can save yourself from trouble reading further. Another life hack is to dismiss all offers that come within the first hour from publication. It takes time to provide a meaningful offer to a thorough brief, so, fast replies are most probably duplicates of the cloned proposals. Your ultimate goal is to avoid dealing with folk who haven't even bothered reading your request and prefer to cut corners.

There'll be plenty of sudden problems with your website: the server gets retired, emails don't get through, third-party plugins conflict with each other, SSL certificates aren't properly renewed, the pages load forever, etc. You want to find a resilient personality who wouldn't shy away at the first sight of trouble. Ask tough questions to let weak candidates 'fall off'. Some of them will, because these folks want **nothing** to do with answering complicated questions or having to deal with a cognizant client. Your preferred prospect should have and be courageous at expressing their opinion about the subject, because they are supposed to be the experts and their job is to help you avoid obvious mistakes that you are unaware of.

When you have chosen 3-5 most promising offers that you'll be investigating, start by comparing what's different about them. Then prepare a set of questions that you will send to each candidate. Keep your communications private. Ask your candidates about what they think about a specific detail offered to you by other contenders that they haven't mentioned in their offer.

For instance, a candidate A is offering to build a website with CMS while candidate B suggests building a website on WordPress. While both options would provide you with a CMS feature, firstly, you might not be aware of that, secondly, it isn't clear which technology will be used by candidate A to secure the CMS. So your question to A could be: "Why do you think my project requires a CMS and which technology/platform are you suggesting to use?" And your question to B could be (it rhymes): "Why do you think my project requires a CMS and why are you proposing to use WordPress?" If you're paying attention, you'll learn a lot, trust me.

Whenever you ask a question to one candidate, make sure to ask the same question to everyone else, providing they haven't given an answer already. That way you would be able to compare the candidates more efficiently. Ask **open questions** that cannot be answered with a simple Yes/No. Never ask a question along the lines of this example: "Can you build me a website within a month?" Rephrase your questions in such a way that the candidate would have to actually **think** before giving you an answer, for example:

"What might be the obstacles that could prevent you from making this website in a month?"

or

"What makes you say that you'll be able to get this done within a month?"

Keep the amount of questions reasonable. If you start asking a hundred questions before hiring someone, you'd be increasing the probability of a perceptive professional taking a hike because you will appear to be a hard client. No one wants to work with a hard client. There's a balance you would want to find here as well. If you want to employ a trustworthy expert (who is usually in demand due to being professional and reliable), you need to present yourself as a reasonable client who they would be able to communicate and work with.

Think about this as well: when your entrepreneurial pockets are 100 USD deep, does it make any sense to ask so many questions that LT could effortlessly compose the seventh volume of "War and Peace" out of them? If it takes two hours for web D&D to go through your inquiry and another couple of hours to write you an answer, then by the time it has happened your budget will have dried up.

Essay #13 SEED stage

Soup chefs, Nuggets and Rampage

“How would I compare the candidates?”

You might think that it's great to hire a web D&D who had already done exactly the thing you want, but here is the problem. The fact that they claim to have done something similar does not guarantee a success with your gig. Every web project is different. It might look like the exact same thing but there will always be constraints, extra features, unexpected twists, variation in technology stack, outdated dependencies, etc.

Here is a good example. You can have two people side by side making the soup using the same recipe from the same ingredients and they would yield different results. One dish would be edible, while the other will provide a mouthwatering *can't-get-enough-of* and *bring-me-some-more* type of experience. It's the prowess of the soup chef that makes all the difference. The same holds true in regards to web developers.

There are CMS specialists and WordPress enthusiasts, as I had mentioned before, but they are far from being web developers. Not going into much detail, I would argue that the aforementioned couple can install, deploy or implement a plugin written by a web developer, but they cannot create one or modify the existing code without breaking it. Yes, they know more than you do, but far less than a true web developer. These specialists lack the fundamental knowledge and therefore have serious limitations for the work they can produce. A real web developer, on the other hand, can build a custom solution or add some functionality even without prior knowledge of the specific platform. Take note – I'm not making an argument that the skills of former web devotees are inferior to true webdev's, I'm just trying to help with identifying who is who and who's capable of what so that my fellow entrepreneurs would avoid having futile expectations.

Let's look at the following example. You have two candidates to choose from. The first candidate A claims to have built 500 websites with WordPress and installed thousands of plugins. The second candidate B says (s)he does not have a huge experience with WordPress but had built web applications with Zend and Laravel. Assuming this to be true, if your project involves a development of a custom WordPress plugin, then your obvious choice falls upon someone with thousands of installed plugins (A), right?

Well, it's not that simple. You would need to ask a few questions to both candidates first. But even now, with the information provided, there is no way of telling whether the person A would be able to build a custom plugin for you, because (s)he only claims to have

installed them. There is nothing mentioned about having actually built one. The person B, on the other hand, appears to have a deeper knowledge of the underlying technology and therefore is more likely to deliver and here is why – WordPress is built on PHP; Zend is a PHP framework too; Laravel is based on Symphony, which in turn is a PHP framework as well. That means that anyone who is proficient with PHP is capable of altering, creating and fixing **any** PHP-based application including WordPress. Thus, based on the info provided, the candidate B looks more promising at the moment.

You want to find a web D&D who had not necessarily done exactly the same project but has had an experience and knowledge that allows her/him to accomplish the task. A great soup chef isn't someone who had made thousands of borscht soups, it's the chef who knows the culinary basics well enough and had made hundreds of different dishes and now specializes in soups.

Keep in mind that there are usually multiple ways in solving a particular problem in web development. For example, if you want an e-commerce store built, it's doable, for instance, with WooCommerce plugin for WordPress. But you can also do it with Shopify. The former would require you to deal with hosting, SSL certificates, updates, but you would have all the open-source code in your possession and therefore more flexibility. The latter is a ready-to-use solution with monthly fees and limited access to the code. There are a lot of other tools and platforms as well. Compare candidates based on their expertise and their ability to approach the project from different angles clearly articulating the pros and cons of each option.

Reach out to your prospects with a request to showcase their work and challenge them to compare it with a similar website that you find professional and of high quality. Or even better, without disclosing your intentions make them compare their work against the work of another contender. You will end up having both appraisal and critique for the same showcased website from different people. The surface of plastered-to-perfection proposals will crack open and nuggets of valuable information will emerge. As a result you will have a better clarity on what's inside the packaging that says “candy” and who'd wrapped it up.

I would argue that trustworthiness, reliability and decency are the most important selection criteria for a candidate at the SEED stage, not the experience or speed. Your business has neither spare resources nor time that you could afford to lose on a sneaky code ninja. Look out for any signs of something that smells fishy. Remember – if it looks like a fish, smells like a fish and talks like a fish – it's probably a self-proclaimed prancing unicorn on a rampage.

Essay #14 SEED stage

Gibberish, Fellas and Chickens

“Any tips on how our communication should commence?”

Your SEED stage startup should be looking for a web D&D who can get things done in a timely fashion within a set budget while not having to sit in your office with her/his throat at your arm’s length. Recent COVID-19 events had shown that working from a distance is an inevitable reality. The ability to clearly articulate complex ideas without being physically present is an invaluable asset.

Efficient communication is crucial to the success of the web project. You should be looking for a person who is comfortable interacting over a multitude of channels and capable of adapting to the ever-changing environment. This is particularly important for web development as the technological landscape changes very rapidly. You want to find someone who is articulate enough to be able to explain project’s challenges in non-technical terms. They don’t have to know **all** the programming languages, libraries and plugins – that’s superfluous; they are not required to have experience with every new tech that has been labeled ‘*hot*’ for the last few years either – that’s unnecessary. What is crucial is that they should comprehend the basics of the coding well enough to be able to implement that knowledge and deliver result.

Over the course of interaction you might bump into a language barrier or even a dialect issue; there is also an industry specific jargon you might have to deal with. So, if you find yourself in a situation where a web specialist throws a bunch of technical terms while muttering gibberish in a thick slang – you have a serious problem. You shouldn’t pretend it is OK and dive deeper into this business relationship with someone you can barely understand. I’d suggest you find a better fit for your project.

Some people complicate communications on purpose. When terms and conditions aren’t clearly defined, it’s easier to change the rules of the game in one’s favor and slither out of the corner when convenient. There is actually a technique used by notorious salesmen – uttering a lot of nonsensical abbreviations fast in order to make the other person feel like a total idiot while elevating oneself to the level of an expert. And when things go south you might hear the following excuses: “I never said that. You got it all wrong!” Hopefully, you don’t want to fall for that.

“Should I expect my candidates to have any follow-up questions?”

If your posting says something like: *“Make me a simple website”*, (God, I hope it’s not) and your prospects aren’t asking any meaningful questions, then you shouldn’t be considering them as an option. Why? Do you want to work with someone who does what they’re told even if doing so will inevitably result in something stupid and then get a response: *“I know this was an idiotic decision, but I did what you told me to”*? If that’s what you’re after, then be my guest – it’s your money, you can flush it down the drain of your particular liking. Personally, I would prefer working with someone who knows a thing or two better than me and can point me in the shit-swamp-free direction.

Generally, if a web enthusiast isn’t asking any follow-up questions it either means that they have absolutely no idea what they are going to do, or they think they know what they’re doing, but are afraid to ask for clarification because of the fear of you thinking of them as being unprofessional or lacking experience. Which is true, by the way – an experienced, seasoned web D&D would have had many clients with similar misconceptions, attitudes and behaviors, and they know how important it is to inquire, to stand your ground and not to be a pushover. They know the value of their work and thus aren’t in need of acknowledgements. Junior candidates aren’t that versed yet.

Another reason why you are not getting any questions might be (and I hope you won’t fall prey to one of these *‘fellas’*) that the candidate doesn’t care about the success of your business or how much money you are going to waste. They want to make a quick buck and wash their hands clean of your bloody disaster of a website. They might just as well have the amount of experience to be aware of what, when and how things will go wrong and go about doing it anyways, knowing perfectly well that they will be commissioned to fix it later. Cheap services are typically like that – *“I’m not getting paid for advice, so I’ll do what you ask and it’s not my fault you’ve asked for the wrong thing.”*

The next subject to be mindful of is whether your prospect is answering all of your questions. If I had asked a bunch of questions in my email and received answers to just a few it either means that my questions are so obviously stupid that it’s embarrassing for the web specialist to lower themselves to my level or that the candidate is just not paying attention. Or maybe it’s the case that the person is intentionally avoiding uncomfortable questions. “How many chickens does it take to trample the elephant?” – unless it somehow ascends us to the next level in our web design progress, I would abstain from answering this one myself. But other than that, I think that it should be OK for me as a non-tech-savvy client to expect answers to a fair amount of dilettante technical questions since that’s actually the reason why I’m hiring a professional. Either way, I sense a firm odor of arrogance when candidates don’t provide answers – very suspicious indeed.

When you are communicating with a web D&D via email or messenger, keep track of the response rates. If it's a regular business day and you're both from the same country (aka it's not a national holiday somewhere) then you should expect to receive replies no later than within 24 hours (weekends excluded). Don't think your developer is such a huge pro that they're just busy, think – rude. Maybe that particular email requires some additional effort on behalf of the web contractor, then (s)he should communicate that by saying that it will take some time to provide you with relevant information. They should actually get back to you within a specified time frame as promised.

You should act in a professional manner as well. Don't send emails and texts with “URGENT!!!” subject over the weekends when you want to change the color of the heading on one of the pages. That's a sign of a terrible client. Professional web D&Ds accumulate requests on bug fixes, changes and updates and then go about solving these problems in bulk. Don't expect your web contractor to be erect at the desk waiting for your signal, have a dedicated computer monitoring your website's performance with all of your project's folders open and connections established so they could get things done the minute you had thought about it. It doesn't work that way. Make a list of all the things you need done and send it over in one big chunk.

Essay #15 SEED stage

Fifty shades of Red flags

“Are there any red flags that I should be aware of?”

This is a very powerful question at your disposal. It's not the question itself that's so potent, but the idea that something might (probably will) go wrong and you don't know what it'll be. Nevertheless, you are willing to open your perceptions to things that might get in your way and burden yourself with the awareness of probable troubles. These kinds of questions will start popping up in your head when you get the right approach – assume you know little and that **you are wrong** in regards to what you already know, then figure out how to **get less wrong**.

For starters, there's a red flag of availability. I have observed candidates sending initial offers that say they could start right away. Is there anything wrong here? If we look at the provided information with our usual *shallow* approach then nothing is wrong and it's

actually great that people are just sitting there waiting for me to allow them to start working on my grand project, aka The Next Big Thing. “They’re there for ME. The only reason why they exist is to fulfill their destiny by designing and developing my website.” It sounds very enticing – this way of thinking, doesn’t it?

The problem here (apart from my magnificence) is that the real professionals are rarely available right off the bat – they might have a day job and this is just a side hustle, or maybe they are seasoned full-time freelancers – either way serious designers and developers wouldn’t start the conversation by saying they’re available right now as if it’s the most important characteristic that distinguishes them from other candidates. *“Hire me now, while supplies (my generosity) last. Available through February 29. Limited offer.”*

When I started out with coding, I didn’t have projects on a daily basis but I had tons of other things to do – searching for clients, writing these essays, improving my awezzom website, polishing [the ultimate brand strategy workshop](#), planning the social media strategy and so on – that’s at least one full-time job in and of itself. I wouldn’t start a conversation by saying that I was available because it might’ve given a false impression of being out of work which, in turn, implies that I could get commissioned for less. Web D&Ds who genuinely want to be helpful and inform the potential client in advance about their happenstance availability are actually doing themselves a bear’s favor.

So why would anyone write an introduction saying that they’re available? I understand their reasoning – differentiation. “We know – clients have problems with their developers (because, as it turns out, we are exactly the kind of developers who create problems to our customers, but so does everyone else). There are tons of developers on the market, but they’re not rockstars as we are, of course. We can come in and fix your problems (and create a bunch of additional ones along the way; again – just like everyone else). And, hey, we are available unlike your current third-world good-for-nothing nerdy amateurs, and therefore we’re better. Hire us!” Personally, I think that this sort of shallow reasoning is a product of the lack of experience, empathy and humility. Such an appalling value proposition is also disrespectful in some vague sense.

I have witnessed offers along the following lines: “We are available now. If you send a prepayment, we can start today!” Wait, what?! Are you for real? Your main competitive advantage is your availability due to the fact that no one sane wants to hire you because you produce more problems than solutions and you want a prepayment to pull the client into your little version of hell? Good luck with that. I’ll pass this “**Next stop: Abyss**” train.

The next red flag is a full prepayment ask. Let’s look into what the prepayment is for and why would a web D&D ask for that. The prepayment is a guarantee that you as a client will

not change your mind in regards to going ahead with the project. Web designers and developers know that if they start working and provide a deliverable there is a chance a client might not pay. There might be many reasons for that but one of them is a change of heart. Prepayment is meant to mitigate that.

As a client you are not protected against fraud or an incompetent service provider. So it's not a very bright idea to make a full prepayment. The only way to solve this problem properly is to minimize the risks. You could try to make an arrangement to pay after the job's done but if you're not flexible with this position some quality web D&Ds would decline working with you because that shifts all the risks on them. A very potent technique which I consider to be something approximating a win-win situation is breaking the project into milestones with clear deliverables.

For example, you want to have a WordPress website for your e-commerce flower shop. You want to hire a full-stack web developer to setup the system on the server, design five different page layouts and then optimize images and website's performance. You could suggest splitting payments like this:

- 1st stage – deploy WP on server; 100 USD prepayment;
- 2nd stage – create 5 layout mockups using graphic design tools; 500 USD prepayment;
- 3rd stage – mockups' revisions; 500 USD final payment for mockups;
- 4th stage – coding; 500 USD prepayment;
- 5th stage – coding revision; 500 USD final payment for coding;
- and so on.

This way you will be able to have tangible deliverables and keep your hand on the pulse of the project. If you ever find yourself deciding that your current web contractor is not a good fit, you can keep the project going with another person. You won't have to start from scratch.

Does this technique work for every scenario? Of course not – there are always exceptions. There are cases when it makes sense to make a full prepayment – when there are no stages, the price tag is small and no revisions are necessary. For example, if you are hiring a web developer to add a subdomain to your website then there are no milestones. When the task is simple enough and your developer is charging a relatively small fee, a full prepayment is totally justifiable.

Whatever the project, you can always ask your web D&Ds to break it into milestones and see what they come up with. The amount of prepayment should always be based on the budget. The budget is formulated based on the scope of work. The scope of work is

determined through a series of Q&A. Whenever I see a candidate asking for a full prepayment without knowing the full scope and budget of the project – that’s a major scarlet flag for me.

Yet another red flag is a no bargaining statement. “Why would that be an issue? People don’t bargain over a haircut at the barber’s, why would I haggle with a web designer for their services?” Web development is not as simple of a task as it might sound. There are many variables at play. It’s similar to building a townhouse in some sense: you can have one built from wood or from bricks; you can have a dozen floor-to-ceiling stationary windows or just a few awing windows. Every variable significantly alters the entire architecture and the final cost. The same applies to web design and development: the amount of pages, plugins, features, database tables and their relationships – everything has an impact on the complexity.

There is always room for negotiations: payment model (per hour, per project or per value), scope of work, deliverables, time frame, revisions, etc. Some services aren’t negotiable like a website’s maintenance fee or an installation of a generic WordPress template on the server, I get that. But when it comes to making a custom website – it’s just not the case. Your prospects might be super busy and are submitting an overpriced estimate without giving it a second thought. Think: if they are so preoccupied with other things, would they be able to devote time, attention and resources to take care of your business problems? Ask why they’re not interested in discussing the matter.

Let’s get back to the flower shop’s example. If we assume the initial budget for the project was 2 000 USD, you as a client could investigate what would that price tag encompass. If we further assume that the project consists of the following deliverables: website’s deployment, 5 page layout designs, coding, testing and 3 revisions, and your goal would be to decrease the costs, you could ask to reduce the number of page layout mockups and limit the number of free revisions. Your web D&D should be able to entertain the possibility of adjusting the price. If they don’t – that’s a crimson flag right there.

The offer of unlimited revisions is another red sign of trouble. As far as I’m concerned, unlimited revisions technically mean that I (as a client) am allowed to ask for any amount of changes at any rate forever for free. It could look something like this:

Monday. “Hey Code Ninja, please change the heading color to brown.”

Tuesday. “Change the size of the body copy slightly.”

Wednesday. “Change the heading color back, but increase the size of the font.”

Thursday. “Change the font of the body to Comic Sans.”

Friday. “Make everything the way it used to be. Please hurry!”

Sunday. “URGENT!!! Change the color of the heading to brown, but slightly darker.”
Monday...

This is the kind of work process that would surely make our web shinobi denounce her/his loyalty to the craft. I understand – they are using the amount of revisions as a competitive advantage to lure the not-so-savvy (~~idiot~~) customer and they have a very limited experience with hard clients who would go about doing exactly the thing outlined above. Otherwise, no one sane would ever offer unlimited changes. The reason why I consider this to be a vermillion flag is my suspicion of a malicious intent. I can’t shake it off. It always seems to me that the sneaky code ninja won’t deliver on the promise and will be looking for pathetic excuses, ridiculous clauses and made-up exceptions.

Essay #16 SEED stage Backups, Primates and a Drowning man

“Is there anything I can do to minimize the risks?”

Unless you’ve worked together before and you can make a strong case of the person’s integrity, consider your web D&D to be a stranger. What is it that you would never do that could compromise your own safety and security? I’m sure you wouldn’t hand over your wallet, apartment keys and alarm system’s pin to someone you barely know unless you are forced to. Don’t force yourself: don’t disclose your credit/debit card information to your web specialist; don’t disclose your login credentials from your hosting, payment processing and email accounts.

Whenever possible, input all logins, passwords and digital keys into your web application yourself after the web D&D has finished the job. Revoke her/his access and change your website’s passwords. If you don’t know where exactly your user credentials should be entered, ask your developer about that and state clearly and confidently that you would prefer doing it yourself. A decent web contractor would **never** challenge you on this decision. They know how vulnerable a client can get when a webmaster holds all the keys.

Let me give you just a few illustrations of what could be done. If you have provided your sneaky code ninja with a full access to your hosting, they could register your domain name rights to someone else, they could also change the user credentials in your hosting so that you would be unable to access it and revert to the original settings. All of your websites located on that hosting will be held hostage by the dark web shinobi and it will take you months to get it back, providing you have used your personal credit/debit card to pay for the hosting and domains. They can move all of your website's files from the server and, unless you have a backup copy of those, you would be ransomed to have them returned. If frenzied unicorns have access to your bank, PayPal or Stripe accounts – they have your wallet in their hooves. If inebriated rockstar developers get access to your main email that is used to manage hosting, payment processors and other accounts – you are doomed.

To prevent these problems from happening, set up a separate email account with gmail or similar independent reliable service and use it as an admin email for your website. Never use your main email (the one used for payments, hosting, etc.) as a login anywhere in your website. This is especially important in WordPress and any other open-source software, as anyone who would have access to your website's files will be able to get and decipher your user credentials.

If your hosting provider or web builder has the functionality to invite or grant access to a web specialist, I suggest you utilize this option. Shopify, for instance, has a nice feature called Partner's Access. You'd be able to decide what changes the web contractor can and cannot make, revoke access at any time and guarantee they are unable to change your personal account settings.

In general, your web D&D doesn't need access to your hosting panel. They should be able to do their job having an FTP access to your server. You can check with your hosting provider how to obtain it. Make sure the ftp access credentials differ from your personal account's logins. But there are times when you will have to provide access to your hosting if certain settings have to be tweaked, compression enabled or SSL certificate installed. Ask your web D&D what access they would need and **why**.

Having made plenty of stupid mistakes on my own, what steps would I take to protect my work if I were to launch a new website today? A SEED stage bootstrapped start-up probably doesn't have a vast array of valuable assets but here are some things you could do to sleep tighter:

- Register a website's domain on your name from a company that explicitly allows domain transfers and has a streamlined procedure in place to ensure that. This way you

will be able to transfer the domain name to a different registrar should you decide to do so.

- Select a hosting company and create an account on your name. This will fortify your exclusive rights to make changes to the settings of the server that will hold your website's files. Hence, you will be able to copy, delete and move files to another server or hosting provider should you deem it necessary.
- If you are planning to use emails with your domain name, like `itsme@mycooldomain.com`, then you should register your email addresses with either your hosting provider or use a separate service like Google's GSuite or Migadu. In consequence, you will be able to create and manage as many emails as you like.

These steps will guarantee several things. First, the name of the website will always be in your possession (if you continue paying) and no one would be able to transfer it and ransom you. If you allow your blasted rockstar developer to register the domain name on anyone else other than yourself or your company, eventually you might find yourself in big trouble. Imagine you would somehow end up obtaining the rights for `google.com` domain. What's your primate brain's rudimentary thought – push it for billions, right?

Second, you will be entitled to manage the server, its settings and contents. If a berserk unicorn ever manages to befoul your digital sanctum, you would always be able to revert to the original state by proving that you are the account holder. You will also be able to make as many backups of your website as you like not having to rely on the mercy of the foul beast.

Third, you won't have to beg your rogue code ninja to open yet another email account. You will have the full control over the quantity of emails, its contents, their names, passwords, redirects and quotas.

There's an infinite amount of ways how things can go wrong and how you could get screwed over. That doesn't mean you should be careless and hope that everything will turn out just fine. When your web D&D has finished the job, change the passwords you'd provided her/him with and make a backup of the website. Store it somewhere safe: your PC, Mac, HDD, flash drive, Dropbox or Google Drive. Use your common sense and really think this through. Don't be naïve. The rescue of a drowning man is the drowning man's own job.

Gathering all the SEEDs

If you were to ask me to provide a concise advice on how one should go about creating a website for a SEED stage start-up, I would give the following answer:

1. If you care about the outcome, your business and your customers don't expect this to be an easy, simple task. There will be unexpected challenges and you should be ready for that both mentally and financially.
2. Think about why you are building a website for your business, who you are building it for and what is the main goal of the project. Your website is a tool and you're the one who can make it either sharp or blunt.
3. Concentrate on the minimum viable product. Get rid of all the features you can. Leave only the inevitable ones. Having a one page thought-through modest website is totally fine. It's much better than a large site with vacuous pages and shallow content.
4. Minimize your expenses: get a cheap domain name, choose shared hosting with an email server and a free SSL certificate. Select any technology that is within your budget. Keep the amount of plugins, add-ons, features and pages at minimum. Remember that you will have to redo your website as your startup takes off.
5. Procure hosting and domain name yourself in order to avoid keeping all of the eggs in one basket. A convenient one-stop-shop might backfire on you.
6. There is no need to hire a graphic designer or a web developer when you're on a really low budget – choose a free-for-all theme from well-known website builders.
7. Plan for failure. Every third start-up fails within the first two years; half of them exit within five. Don't consider yourself above the statistics. The odds are stacked against you.
8. Be ready to pivot. Set aside a budget for making that U-turn when you figure out that you were going in the wrong direction.
9. Set and follow through with time constraints. You need to know when it's time to reassess your initial assumptions and make the pivotal decision.
10. Know your target audience and your core target customer. Dig deeper into understanding who you are creating your website for, what problems they have, where these people are and what they think is true. Build your website for that particular customer.
11. Investigate your competition. Figure out who they're targeting and try to do a better job at solving the problem in a different way. Use what works for your competition and innovate on what doesn't.
12. Define your business goals. You will be wrong and it's fine. It is a lot better to set precise targets, aim at them and miss than to have a blurred vision of vague objectives.

The more you get to be wrong, the better you will become at formulating the right objectives, thus increasing your chances of attaining your goals.

13. Keep constantly asking yourself what is it that you can do to become less wrong.
14. Figure out your desired return on investment. If your personal journey is about getting money rather than experience then you should define your ROI and keep it in sight. When you're falling short, it's OK to admit your faults, then pivot or move on.
15. Don't ask amateurish questions like "How much is it going to cost me to make a simple website?" The right question to ask is - "How much X should I invest to build a MVP website that looks like ABC with XYZ features in N days that will help me attract JAMES' attention and acquire Z value in T time?"
16. Whatever the budget you came up with – double it; whatever the revenue you've dreamt up – divide it by ten.
17. Find a web designer or developer through the network of friends, relatives, colleagues or acquaintances. Consider freelance platforms and classifieds to be your last resort, because they swarm with digital predators.
18. Take the full responsibility for both success and failure of the web project yourself. Don't expect anyone else to assume it.
19. Familiarize yourself with different titles web specialists assign themselves. Fathom which one is more suitable for your project's requirements. Your SEED stage start-up might suffice with a full-stack developer.
20. Beware of sneaky code ninjas, frenzied unicorns and intoxicated rockstar developers who are possessed by the image of their godlike magnificence.
21. When in doubt, hire a professional consultant to advise on the best web practices for your business case.
22. Abandon your wishful thinking mentality. Stop living in Wonderland.
23. Have a handful of options, offers and candidates for your project to be able to juxtapose them and select the best fit and not fall prey to carnivorous unicorns or become hostage of a dark web shinobi.
24. Ask your candidates tough, open questions that facilitate thinking and discussion.
25. Identify a web D&D who is easy to understand and communicate with.
26. Find a trustworthy business partner for your web project. You want to have a competent, reliable and nimble problem solver by your side.
27. Look out for red flags of immediate availability, low response rates, full prepayment requests, unlimited revisions, claims of superiority and grandeur.
28. Break the project into milestones with clear deliverables. Avoid making full prepayments for the job that can be divided into chunks.
29. Be very mindful of what logins, passwords, account details and other user credentials you are providing your web contractor with and why. Don't allow yourself to be naive.

30. Rather than making yourself believe in the fairy tale that everyone needs your product, reverse the reasoning and start by convincing yourself that no one wants or needs your stuff.
31. Aim at something better – the best functionality, user experience or web design you can come up with.

It is tempting to believe that you can set aside a hundred bucks, pick a random web specialist and get a great website within a couple of days. That's an impression you get from looking at ads promoting easy-to-build, free websites and from web enthusiasts promoting two-digit gigs with a shiny lure of potentially having a six-digit-income e-commerce store. Sober up. You should know better by now that a fancy French cheese is a courtesy of a chef only in a "Mice Catchér de Royale".

Even if your budget is ridiculously low, your website doesn't have to be crappy. Even if every competitor has terrible websites, you don't have to live up to their standards. You can aim at making things better for yourself, your employees, your customers, your family and broader community. Put in a little more effort, a little more attention to detail, adopt responsibility for every decision you make and things will get better. Aspire to create something beautiful and meaningful and you just might make a difference in the world.

A word from my Sponsor who hadn't paid a dime for this:

awezzom

Websites Worth Making

<https://awezzom.com>