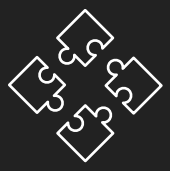
2nd Edition



DESIGN'S IRON FIST

and other essays

by Jarrod Drysdale

Part 1: Design Technique & Process

Part 2: The Designer's Mindset

Part 3: Making Your Own Products

Part 4: Freelancing

Productized

How to Offer Retainers, the Holy Grail of Consulting

You Are Not The Boss

Foreword

Hi there! I'm Jarrod.

Thank you so much for reading this ebook.

Design's Iron Fist is a collection of my most popular articles. I write about design and adventures I've had in my 10+ years of experience as a designer.

I hope that these articles are not just useful to you, but that they also challenge you to try new things and consider new perspectives.

Since I published this ebook, it's been covered by top design sites like the Creative Bloq and AIGA.org as one of the best free design ebooks. I'm very proud of that, so thank you sincerely for being part of this.

You should have received this ebook for free after signing up for my newsletter. If you got it another way, no worries! But you might want to sign up for the newsletter, as I've written an email series to accompany this ebook.

Best wishes for you and your projects,
Jarrod Drysdale
@studiofellow

Part 1

Design Technique & Process

Answers to common questions from junior designers, and a refresher for the rest of us.

Why Write

How writing will make you a better designer.

The great, revered designers in graphic design history created insightful, meaningful work that communicated.

Sure, it was beautiful. You can look at Lubalin's typography work or Vignelli's NYC subway map and marvel at the aesthetic qualities shown.

But these works aren't famous merely because they are pretty. Lubalin and Vignelli are so respected because of their ability to communicate: the former's logo work often revealed a hidden meaning within the letterforms or played some subtle metaphoric trick, while the latter's controversial map untangled a complicated subway system.

The design greats were/are writers, thinkers, and communicators. But somehow, either with the invention of the web or some other influence, that tradition has faded.

So many portfolios I see these days have little substance. The writing is poor and what little text exists in the design is only there to fill the space between retouched photos and illustrations. There's little explanation about why the design ended up that way.

We even have entire communities built around the idea of showing off pretty work that's completely disconnected from context and message. (I'm looking at you, Dribbble.)

Compared to the designers of the past, most of our work, myself included, pales in comparison. Not only do we often fail to demonstrate the purpose of our designs, but sometimes we create designs with no purpose at all—design has become and end in itself.

Great design is supposed to be timeless. What web designer could point to a single project and claim it's as timeless as an Eames chair? I realize this comparison is a bit unfair, but I do think it reveals something about design culture.

Design has become more like a status symbol or luxury good; if you have a nice design, you're legit. Quality design merely filters out the wannabe businesses.

As designers, we describe our job as positioning clients amongst all the noise—a shiny design will make a brand stand out.

But the hypothesis we fear is that a pretty design will make no difference at all.

What if we designers are actually adding to the noise by creating shallow work?

What if there were a better way to serve clients than with aesthetics?

I think I've found a way: through writing.

How writing made me a better designer

I've been running my own products business for the last couple of years. During that time, I've had to learn to do everything myself.

The biggest challenge came from writing.

I have to do all kinds of writing just to keep my business running: a weekly newsletter, blog posts, and landing pages. Even keeping up with social media posts—just writing tweets—can be daunting. That might sound silly, but it's just another thing I have to write. I even went so far as to write an ebook.

Despite the fact that writing takes up more of my time than I ever could have expected, I've stuck with it. Looking back upon a couple years of regular writing, I was startled to realize that I approach a design completely differently than I used to.

A few years back, I tended to get excited about aesthetic concepts or features. Now when I start a design, the exciting part is connecting an audience with a great idea.

Of course we've heard these tired maxims over and over: content is king, design should serve the content, focus on the user, etc.

But after all this practice writing, something is different. It's almost like I'm not even a designer anymore.

After this change, design is just a tool to help me get the

message across. It's an artifact created along the way—not the goal.

When I flip through my Lubalin book, that's the common thread I see in his designs. In each case, he was obviously thinking about how to communicate. He had a clear picture about the intended reader, and used that knowledge to craft clever visual cues to enhance their understanding.

Now, I'm nowhere nearly so skilled as Lubalin. But I'm encouraged that the recent change in my work could mirror his reasoning. A while back, Sacha Grief did a write up of one of my landing pages. I think the only reason he wrote about my design was because of its harmony with the writing. More recently, Joanna Wiebe, author of the incredible CopyHackers books and probably one of the best copywriters anywhere, tweeted about my copywriting. Without me even being conscious of it, my work must have improved to earn those kinds of comments.

Somehow, **my designs are better because I can write. My writing is better because I can design.**

Now, I'm not trying to brag. I just think this is exciting. What could the design community accomplish for our clients and culture if we could lift our discipline beyond mere decoration?

The next time you're looking to level up your design skill, try something different and make a commitment to write.

Maybe you'll become the next Lubalin.

How to Start a Design

New designers are terrified of that blank screen. How are you supposed to fill it with design? Where should you start?

Somehow, that empty white space instills fear. Why? Because it holds power over us. Because we have no reliable method for starting a design.

The cure to this fear is process.

Now, designers talk about process a lot, and you've probably seen those pretty diagrams that explain their processes. Every single one is different. How can that be? Even worse, those diagrams are incredibly complex. How can a person internalize so many steps and be able to follow them instinctively?

I've got good news for you: you're being deceived. No serious designer works by following dozens of linear steps.

A correct design process is simple and it doesn't try to replace critical thinking. Design process isn't meant to think for you; it's meant to help you think clearly and decide what deserves your attention.

Every creative process can be broken down into a handful

of simple stages. But, it's important to understand that these stages are not linear. You'll need to skip back and forth between these stages while you create a design. Every design project will be different, too. You'll use each stage differently for each project. But the important part is that you understand how each stage helps you solve a problem and come up with new ideas.

1. Gather

Find all the elements you need to begin the design, including content, technical requirements, goals, and examples.

2. Edit & Sketch

Revise your content and begin rough sketches that help visualize how you can organize the content and meet goals. Your sketches should be fast and ugly. Get out as many ideas as quickly as possible without committing to anything yet.

3. Mock Up

In Photoshop, Sketch, or HTML/CSS, bring your favorite sketch

to life. The mock up should reveal how the sketch functions and whether each feature fulfills its purpose.

4. Build

Build the design into a fully functioning application. Be faithful to the solution you planned.

5. Test & Correct

Use data and metrics to evaluate the design. When you uncover problems (even the most experienced designers do) return to previous steps to explore smaller solutions to those problems.

...

Now when looking at the blank screen, you know you can't even proceed yet because there's so much to do before you even need a computer.

The only reason the blank screen is scary is because you don't know what to do.

Process removes these fears and gives you the ability to break a problem into smaller parts rather than trying to understand and solve the whole of it at once.

Are You Stuck?

The previous article explained how process makes the blank screen you face at the start of every project less intimidating. But sometimes, even when you follow process, you still get stuck.

When you get stuck, that usually means you are on the wrong step. Move to a different step and evaluate your choices.

Here are some examples:

You are struggling to sketch an interface.

This usually means you don't have enough information about how it should function, what it needs to accomplish, or how it fits into the rest of the application. Go back and Gather.

When mocking up a feature, you keep thinking of extra features to add to the page and get distracted thinking about them.

You didn't do enough sketching. Go back and sketch out all those little features, and make sure they really work well enough to deserve a mock up.

When building a page, you realize the design cannot be supported by the server-side code without substantial, deadline-breaking changes.

When this happens, it means the gathering, sketches, and/or mock ups were insufficient. Before you continue building, research that aspect of the software and then sketch and mock up a few different solutions and see if you can find one that doesn't cause huge implications for the server-side code.

You're done building and are ready to test, but you don't know what you should test.

In the Gather phase, you didn't think about the goals for the project. Now, you're not sure how to measure whether the project is meeting those goals. This is dangerous! Make sure to consider objectives every time you start a new project.

...

So, to recap, process is how you get out of ruts. If you're not sure what do to, think about process. Take a step back and try to identify that missing piece that's holding you up. Sometimes your instincts will just make the problem worse—that's when process has your back.

How To Stop Design Guesswork

When you try to design, do you ever feel like you are making things up?

For example, you are trying UI elements you've never used before. You're not even sure if they will make a difference—it's just a guess. You thought it sounded like a good idea. But how do you really know if it's a good idea or a waste of time?

Design is just like any other kind of work: every time you venture out into new territory, you take a risk of getting stuck. When facing an issue, the natural reaction is to keep working until you fix it. Somehow a week passes, and you realize there was a simpler solution all along. Then you find yourself wishing you could get that week back.

While you're still new to design, getting lost like this is easy. Here are some tips to avoid that trap.

Become a dirty thief

The design problem that's vexing you isn't new. Someone has solved it before, and they are probably more experienced than

you. So before you try to reinvent the wheel, do some looking around to see how other designers have solved your problem.

Look at software you use regularly for similar situations. Sign up for a few free trials to see what the interfaces look like.

Once you figure out how someone else solved that sticky issue, copy them. Steal their solution.

This is what designers often refer to as "finding inspiration", "design convention", and "best practice". These are all fancy euphemisms for copying.

So while I call it stealing, what I'm really suggesting is that you learn from others' experience. That's the best way to solve the most challenging design problems.

(To be clear, I'm not suggesting you infringe on someone else's rights, break the law, or take intellectual property. Don't copy someone's entire design, just the one aspect that solves your challenge.)

Sketch first and you won't waste so much time

Sketching is magic. There's no better way to find and test a multitude of ideas than to sketch.

Sure, you could draw a wireframe in Balsamiq, Illustrator, or whatever. You could build a prototype in code or even create a full-blown mockup in Photoshop. But none of these is

nearly so fast as sketching.

Sketches get exponentially more efficient the more concepts you explore. Your first concept will rarely be correct. To find the best solution, you need to explore lots of concepts. If you try each of these by making a prototype or mockup, you'll end up burning a lot of time.

New designers often want to skip sketching. They want to jump into the exciting part—the real design. They want to get something built. But sketches are important because they prevent you from committing to a solution too soon. If you jump into creating a polished, detailed version of the concept before you are certain it's correct, that's a risk. Better to spend 15 minutes sketching to ensure that time won't be wasted by having to start over.

You might think sketching is a bad idea unless you have some drawing skill. Many designers' sketches look like works of art, but yours shouldn't. Your sketches should be quick and ugly. The only purpose of these sketches is to explore ideas quickly. The more beautiful your sketches, the more time you wasted. Let them be ugly. You don't have to show them to anyone else.

Oh, and just use a pen and paper. Don't try to get fancy with iPad apps or other software. Remember, it's all about cutting distractions to explore ideas as quickly as possible.

Don't start from scratch

Frameworks and libraries are awesome. Never before has our time spent coding been so efficient, and it gets better all the time. However you might not expect that using a front end coding framework or UI kit can save you a lot of design time too.

Frameworks like Bootstrap and Foundation solve common interface problems for you. You don't have to research the best way to design a dropdown menu, or ways to lay out a form. Just choose from implementations built into the framework. Each time you do this, your design burden gets lighter.

That said, using traditional design tools isn't quite so difficult as you'd expect (or might have been led to believe).

Basic Photoshop and Illustrator can be a good use of time when you are ready to build the real concept. Not necessarily for mocking up the whole design, but just for adding some nice graphics. Learning just a few simple techniques can make you look like a bona fide digital artist.

Yes, a lot of designers advise steering clear of graphics software altogether and designing in the browser instead. That's good advice sometimes, but often sites still need graphics. Learn a bit of 'shop and you won't have to use stock photos so frequently.

If you want to learn some Photoshop, set a limit. A few

simple techniques will be sufficient. If you tried to learn every feature in Photoshop, you'd end up with a lot of useless knowledge.

(Also remember you don't need an expensive license to use Adobe software anymore. You can pay to use it by the month these days.)

When you make an assumption, test it

Inevitably there will times you can't find a perfect solution, usually because you don't have enough information to inform your decision. And because of that, you will have to make something up.

Too many designers stop there. They launch that assumption and never think about it again.

As designers, it's easy to come to think that our assumptions are correct. We get comfortable making assumptions because design, as scientific as it can be, isn't always precise.

However, when your assumption can have significant implications, make sure to test it. Real data is always better than a guess. Implement an A/B test or simple analytics to see how people use it out in the wild.

Then, launch it and observe what happens. But keep your sketches handy. You might find that you need to try one of

your other concepts.

Personally, I find CrazyEgg to be the best analytics for evaluating design. The scrollmap is especially useful for seeing how people interact with your content and interface. It's also less time-consuming to implement than event-based analytics.

Remember: if you are guessing, you need information

Do everything you can do avoid making a pure guess. Steal, sketch, and use tools that provide common answers. If you do have to make a guess, test it, and be ready to switch to a backup solution.

How to Navigate the Vast, Sometimes Dubious, and Always Overwhelming World of Web Fonts

Web fonts have become a staple in the web and digital designer's toolbox. But, as the popularity of web fonts has grown, the marketplace for fonts has become more difficult to navigate. Resources like Typewolf, Fonts in Use, and others are helpful, but our industry likes to treat typography like a fashion choice. Here are some more practical concerns for picking web fonts.

Pick fonts based on intended use

You should always evaluate fonts for two kinds of uses:

Display font settings use large font sizes, for content like headlines and logos. You can use more decorative or complex-looking fonts for display settings because the font size is larger. Headlines are supposed to grab attention, so using a more decorative font is a good idea.

Text font settings are what you'd expect: paragraphs and longer sections of content. Here, you want to avoid decorative fonts because, at smaller sizes, they are more difficult to read.

Searching through a huge library is much easier if you know whether you need a **Display** or **Text** font. But, even separating uses like this doesn't keep you from having to look at each font, one at a time.

Ignore fonts that don't have enough styles

To filter out even more picks, look at each font family's available weights and styles. If you are looking for a font for a text setting, make sure it includes at least a normal weight, bold, italic, and bold italic.

If a text font doesn't include these at a minimum, you can run into problems when using web fonts. Sometimes,

when a web browser doesn't have a font to use for a passage of bold italic text, it will use a default font, which can make your design look broken. Even worse, I've even seen browsers replace those passages with garbled text—as if that section of text was mapped to the wrong characters.

However, with display settings, you can be more lax about which fonts to choose. It's okay if you set a headline in a font that doesn't include a normal weight or italic, for example. It's a headline, so all you need is that one bold weight.

Check rendering across browsers

Before you decide on a web font, make sure you test it in all the browsers you plan to support. Many problems can arise when the web fonts are created or converted by the type designer. Some fonts might have bad hinting or kerning that only appears in certain browsers. These issues can even make large display text unreadable.

Don't assume that every font out there is equal. Even expensive ones. Fonts are software and can suffer from the same issues as any other kind of software.

Look up the type designer or foundry

Many free fonts, like on the amazing Google Fonts service, are designed by students or inexperienced designers. There's a lot of junk, but if you're willing to look, you can find some hidden gems.

Sometimes I look up the designer or foundry and see what other fonts they've made. If they've made several fonts or released under one of the more well-known type foundries, chances are the font files will be well made and will render correctly.

(Note that a lot of the well-known foundries use Google Fonts for marketing purposes, so you'll find a lot of single-style font families from these.)

Notice how others use fonts, and imitate them

Typography genres can be confusing. Sometimes, genres are based upon history, but others refer to appearance. Plus, many new font designs are genre benders, which makes classifications nearly useless.

Picking fonts without knowing a lot about typography can be tricky.

To make this easier, take a look at site designs created

by experienced designers. Note how they use fonts in various places, then find similar fonts to use in a similar way in your own design.

There is nothing wrong at all with using fonts that are popular. They are popular for a reason.

And, using popular fonts is a lot more foolproof than trying to invent it all yourself.

Should Designers Learn to Code?

Should designers code? Should you design in Photoshop or HTML? It doesn't matter. Those are the wrong questions.

Gurus tend to take a firm stance to establish their points, but this can be a distraction to those of us who operate outside of ivory towers.

We're building stuff. Right now. Doctrinal debates don't help us get stuff done.

It's great to know that the Basecamp/37signals designers are more efficient when they design in code. Learning from these talented designers' experience is a good thing, but there also a problem with it: just because it works for them, doesn't mean it will work for you.

The correct question is: **which tools help *you* to be the most efficient?** People who suggest that you shouldn't use Photoshop or that every design should be responsive are just trying to help you be effective. They're sharing their experience to help you out.

But when reading this advice, remember that each of us has different knowledge and experience. We each have different strengths and will find varying tools to match those

strengths.

That might seem obvious, but letting advice from gurus dominate your thinking is easy. You have to be the filter because no one else understands your situation. So when you hear this kind of advice, instead of agreeing immediately because you respect the person, think: *"Ok, they're suggesting that a different way of working might be more productive. Is that true for me?"*

Think for yourself. People become gurus by finding success one way, but there are many paths to success. There are many ways to design a project. You have to find your way. If you blindly follow celebrity advice, you will end up with a patchwork of incompatible and even conflicting advice.

So, if Photoshop helps you to be more productive when designing a project, use it. If you can mock up your design concepts more efficiently in HTML & CSS, do that. Pick the tools that work best for you, and not someone out there on a soapbox.

Seek advice anywhere you can get it and know the intentions are good but that people sometimes get a bit too zealous. *(And I'm doling out plenty of advice in this ebook, so make sure to hold me to that same standard!)*

Knowledge vs. Skill

How does one become a good designer? Is it by gaining knowledge or by improving a skill?

You're thinking it's both. And, I'm sorry to say, you're wrong. Here's why.

The knowledge required to become a better-than-average designer is miniscule. In fact, you could easily become a great designer and produce insightful, effective work without any formal training or knowledge of design theory. Many revered designers know nothing about various design topics that the larger community deems essential, but they still produce great work.

This is because designers are communicators.

Designers do not exist merely to make pretty pictures or to imbue everything they touch with fashion. Designers work to make communication more efficient. This communication can take a variety of forms: communication between a human and a computer, a company and a consumer, or software and its user, for example.

Of course, this is not to say design knowledge is worthless. Knowledge will help a person with skill make informed decisions and thus increases the quality of the work.

However, you've probably observed that reading up about design doesn't make much difference in the quality

of your design attempts. That's because when it comes to design, skill is more important than knowledge. You haven't found a way to build your skill yet.

Why does all that matter for you?

It matters because if design is a skill more than a knowledge, it's also more difficult to become a better designer.

This truth is why design can be so challenging, especially earlier in your career. Design isn't something you can just grab off the shelf and plug in. It's also not something you can just read about then implement the next day. There is no quantifiable threshold that makes great design: instead there are a million degrees between awful and incredible, with skill being the determining factor.

So if you're still learning, what are you supposed to do with this? Don't be discouraged. Every time you make another design, you will improve by another degree. Each new design will be less awful and more incredible.

There is no shortcut to making better design. Just keep practicing.

Break the Grid

You know what grids are, and why you should use them. But grids are a common cause of mistakes, too.

Sometimes, even when you use a grid faithfully, the layout still doesn't look right. Columns of text get crammed together so closely that they are difficult to read. Other elements can get grouped too far apart, ruining the effect of proximity.

This is happening because you don't know when to break the grid.

Now, some designers will disagree witht he very premise of breaking the grid. To them, *The Grid* is a force of nature not to be trifled with.

I say that the grid is there to serve you—not the other way around. So, if it gets in the way, break it. (Or switch to a different grid.)

Sometimes, **you have an element with dimensions you can't change**, but it doesn't fit the grid. What do you do? *Break the grid.*

Or, you have **two column text and the grid's built-in padding almost makes the lines run together.** What do you do? *Break the grid.*

Other times, **positioning elements correctly within a grid just looks bad.** What do you do? You should know by

now.

When you break the grid, no one will notice. And even if they do, here's what they'll think: *"This designer broke the grid, but the design still looks nice, so they know what they're doing."*

Great designers sometimes break their grids. Other times, they forgo a grid altogether. Being a designer doesn't mean adhering to one prescribed way of working with religious zeal. Design is about critical thinking. If a tool doesn't fit the job, put it back in the toolbox.

Breaking Design Rules

Running my own business has caused me to relearn certain aspects of design. I've made design decisions I'd have considered a major error only a short time ago. Like putting the logo below the fold. Or using a color that doesn't match the rest of the palette on purpose. On purpose!

When your own livelihood depends upon the design successfully driving product sales, you start to look at design a bit differenly. I personally found myself making changes to my designs to support my business goals and kept them even when I didn't like how the designs looked afterwards.

Put the logo below the fold

Designers revere conventions. Putting the logo in the top left corner of the site is supposed to be an important convention.

But for small businesses like mine, it can be the wrong decision.

Putting the logo in the top left corner is wrong because it makes the statement that the person or company operating

the website is the most important information on the page. It implies: *"Hey visitor, you should know who I am before you read anything. I'm kind of a big deal."*

Brand awareness doesn't exist if your customers number in the hundreds. Designing for a small business is totally different than designing for a large brand (or a brand that wants to become large).

Visitors don't care about your logo and they don't even care who you are (yet). They are looking for the answers to two questions: What problem does it solve? Do I have that problem?

So, I tried placing the logo below the fold and low on the visual hierarchy on a landing page for one of my products. The answers to those two important questions were right at the top where the logo would normally be. The only people who even saw the logo were those who identifed with the problem and intended audience. People who didn't leave the page immediately. That's great because throughout the rest of the page, I knew exactly who I was writing to.

So stop assuming that placing the logo at the top of the visual hierarchy is a requirement for every design. Instead, write about the problem your business is proposing to solve. Write about what type of person usually has that problem. Then, lower down on the page, explain how your business is the answer and ask them to buy. And place the logo there too.

Your job as a designer and writer is to show the visitor

that they have a problem and to convince them to continue reading about your solution. If you succeed in this, people will pay you.

Try an Ugly Button

(I bet you never thought you'd see a designer write that!)

That same product's landing page sported a shiny green button. The button wasn't ugly in and of itself. But set amongst the intense orange that dominated the composition, it clashed—badly. And, it clashed on purpose.

I chose the bright, clashing green color for the button because I wanted visitors to see it. Originally, the color scheme for the site included a complimentary blue. When evaluating the design, I realized that this was a problem. The buttons had similar visual prominence to the other elements on the page, and that didn't fit my goals for the visual hierarchy.

My plan was to place the buttons lower on the page. After all, no one is going to sign up before I've explained why they'd want to. But, when the visitor did scroll to a point where the button is visible, I wanted it to be the most obvious element on screen.

A pretty blue button that matched the rest of the design would not accomplish this. So, I stripped the blue from the design and chose a clashing green color for the buttons. Then, when someone scrolled far enough to see a button, it was

impossible to miss. The ugly aesthetic of the button supported my goals perfectly.

With this change, I made a conscious decision: my business goals were more important than making the design attractive. I didn't care about winning design awards. It didn't matter if other designers liked the page. It wasn't for them, anyway. All that mattered was that the page connected with my specific audience.

Furthermore, rather than merely assuming I met my goals, I used analytics. Every metric showed that my design was successful.

It's okay to break the rules

I'm not trying to convince you to always place the logo below the fold or to always use an ugly button. But I do hope these examples give you the confidence to break best practice when you discover it's necessary.

Be proud that you're prioritizing efficiency over vanity. Making these kinds of decisions is the right way to run a business, and your customers will be grateful. And they'll show that gratitude by paying you.

Further, if your decisions help you deliver better results for design clients, they'll rehire you and refer others to you.

Using design to earn results instead of only aesthetics pays off.

Part 2

The Designer's Mindset

Establish a healthy mindset for creative work. So often, the quality of your design work depends upon how you think and feel about design.

You are a designer
(Yes, you!)

Are you learning design? Are you a design student? Are you a coder, thinking you'll never feel confident in your own designs?

Start calling yourself a designer right now.

"Designer" isn't a title you unlock at some threshold of skill, or a title you earn. People of all skill levels are designers.

If you begin calling yourself a designer, all the sudden you can take that work seriously. You might not have a lot of experience, but at least the time you spend working on a design is serious work. You're not messing around, dipping your toes in the water. You are committed.

That commitment makes all the difference: now you are allowed to make mistakes, because you are going to stick with it. Making mistakes is how you learn and improve your skill.

Too often, fear of making mistakes gets in the way of learning. Make a commitment and allow yourself to make mistakes without beating yourself up about how bad it looks. With each attempt, your skill will improve.

Every senior-level designer started poorly skilled. My work wasn't great when I began my career, but I still called myself a designer.

It's easy to earn the right to call yourself a designer: just

go design things.

"We assign mystical reverence to the work of professional designers. Their elegant color schemes, provocative typography, and eye-scorching aesthetics leave us dumbfounded. Only "creative types" can achieve this; only near-savants who were born with a special talent.

…you'd think design is difficult. You'd think it's complicated, and that gaining basic skill requires hours of studying a multitude of advanced topics. And you'd be wrong.

Anyone can be a great designer with practice. It's both at once liberating and frightening: your future as a designer depends only on how hard you're willing to work. Design is a skill and a trade; you get better at it by practicing. First, learn the basics and go design something. Then, call yourself a designer. The more things you design, the better you will get and the more lovely and insightful your creations will become. No magic knowledge hidden away in design books, blogs, or classes will teach you to be a great designer. All you have to do is practice. Learning design is that simple.

—Excerpt from Chapter 2 of my ebook, *Bootstrapping Design*

Design's Iron Fist

Sometimes while working on a design, you discover a dilemma. You have to choose between your own vision and what's best for the project.

Finding that perfect type setting for a logo, or just the right amount of contrast to draw the eye to the signup button is satisfying. The right combination will have a huge impact upon how people read and explore the composition.

However, sometimes while working on a design, you discover a dilemma. You realize you might need to sacrifice the integrity of the message to make the design look pretty.

Maybe the headline doesn't fit in the font size you want to use. Or there are too many form fields to fit in the area you've reserved for them. In these situations, it's easy to cut the content to save the design. I couldn't count the times I've trimmed a few words or form fields so I could keep the design the way I wanted it.

But when you and I do this, **we're doing it wrong.**

Why is changing the content to fit the design wrong? It seems harmless enough.

You probably expected me to say this: design should serve the content. All design decisions should support the goals of the project. But the implications are big.

"Design serves content" or "Content first" is a cute

aphorism. It sounds nice and even intelligent enough to be correct. But I'm here to tell you that this claim is not a harmless one. It will cause you much strife while working as a designer.

Are you willing to murder your darlings in order to achieve a business goal? Are you willing to change your favorite aspect of a design in order to deliver the message with more clarity?

We designers rule with an iron fist. We mold and shape words, images, and ideas to deliver a message with greater impact. We orchestrate complex factors to form elegant, simple, and usable interfaces. We are ruthless in cutting, shaping, editing, and changing ideas to find the best form.

Design's iron fist rules all. You know that.

But design's iron fist also rules you.

To make great design, you need to be willing to sacrifice the aspects of your vision that impede project goals. That means you shouldn't cut form fields that are essential to the business just so it all fits in a certain spot. Don't remove words that form a better connection with readers just so you can use that particular type setting.

Instead, rework the design. Change the layout. Make more space by shrinking less important elements. Go back and sketch until you find a solution that fits the content and goals, but that will also look pretty. You can have it all if you're willing to put in the effort.

Don't get me wrong. It's not easy. But the affect on your conversion rate (or other key metric) will make the effort worthwhile.

The choice between achieving results through design and realizing your vision is a false dilemma. You can have both.

But you need to be realize that sometimes, your vision isn't the best way forward, and that it needs to change.

All designers want to feel ownership over their work. But the truth is that design isn't just for us, and that we designers are not the only ones who get to decide whether a design is great.

A great design isn't just artistically pure and aesthetically flawless but also functionally and practically sound.

When your vision conflicts with functional and practical matters, your vision is wrong.

Design's iron fist rules you, too.

And that's okay, because when you can allow yourself to work towards making something truly great instead of only catering to yourself, you will find that your work improves.

Sometimes, greatness requires letting go.

Timeless Design

Have you seen those year-end blog posts about design trends? They're fun to read, but they contain a hidden implication you might miss.

Now, any designer will proudly state that great design is timeless. We can point to any number of famous works in architecture, industrial design, and graphic design and show rightly that they are equally effective today as they were upon creation.

But what about digital work, like web design? We see year-end posts about digital design trends every year. Technology turns over so quickly that digital design seems to be forgotten or deemed ineffective much more quickly than physical counterparts. No one points to a great website from 5 years ago and says "Yeah, that would still do the job today."

Is digital design disposable?

The idea that digital design is disposable can be disheartening. You put the same hours, care, and passion into a digital design as you would the design of a physical object. But if you don't update that site design every couple of years, you start to feel like it isn't doing its job anymore.

We feel this way because we forget about the purpose of the digital design in the first place: to deliver a message, experience, or connection. That purpose is fundamentally different than the purpose for designing a physical object. Further, realizing this difference brings the core of the matter into focus. It's not the design that matters. It's the message.

So, when you're working on a digital design, don't worry about whether it's timeless. Don't concern yourself with earning web design awards or whether you're committing a grievous design mistake by following a trend. Instead, focus on the goal of your project. Create a great experience for customers. Deliver your message in a way that people can really connect. And, exploit those trends if they serve the goals of your project.

We all want our work to be timeless. We want to make great things that have an enduring usefulness.

But what we forget about all those famous, timeless designs is that they were only great because of what they helped people accomplish.

Even if a design is eventually forgotten, isn't that design still valuable if it helped people for a time?

The design profession has changed, and often what we produce is not an artifact but a snapshot.

But design can be great and important without being timeless by simply being useful.

Don't Try to Be Creative

While you're learning design, focus on what works, not on having groundbreaking, innovative ideas.

Getting good at design means cultivating your taste. Amd while you're learning, you don't have taste that you can trust. Eventually you will, but for now you cannot trust your creativity. It will only lead you down the wrong paths. While you are still learning basic principles, don't try to be creative. Instead, focus on simplicity, clarity, and the cold, hard science of what works.

As you become more comfortable with design fundamentals, allow yourself to branch out and experiment. You will make mistakes. However, making mistakes is part of creativity, so don't beat yourself up. Instead, try a different decision next time.

We all think our creative ideas are great, but design is not just about having ideas—it's about choosing the correct ones. Doing that takes experience. Through practice and hard work you will gain that experience, and you will then be able to indulge in the luxury of your creative ideas.

Until then, don't try to be creative. Instead, observe what works and imitate it. Practice. When you look back at older

work and see its flaws, don't be ashamed. Seeing the flaws in your past work only means your skill has increased. Be proud. Use that knowledge to do better on your next project.

Not designers. Not coders. Just builders.

We've been talking about how designers should code for years, and the attention this issue rightfully earned has produced an interesting and unintended consequence: the line between designer and coder is blurring.

I propose we remove the line altogether.

Why? Because design concerns and code concerns are the same concerns. Internalize that for a moment. Designers and coders contribute to the project at hand equally—we both want the project to succeed. Because of that, our goals and concerns are the same.

Designers don't want to see an app they designed running slowly because of agonizingly complex SQL. Coders don't want to build an app that is hard to use because of a poor interface design. An incredible app requires both incredible code and incredible design. Without either one, it's not incredible.

Designers are tired of being the gatekeepers for visual detail and user experience. They're tired of having to explain to the team that their concerns are not petty but are instead a

major factor in a project's success.

Coders are tired of dealing with designers who don't understand how things get built. They're tired of turning down silly features that have dire coding implications or are otherwise untenable.

Too often, designers and coders are at odds. Their interests compete. They disagree. If our goals are the same, the classic face-off between designers and coders needs to end.

The influx of coding designers proves we can do it. We can align our goals and start to alleviate all that mutual frustration by sharing our duties and skill sets.

Of course, specialization is important. If each of us were to just dabble in every possible skill rather than pursuing mastery, nothing would get done. However, we need to stop defining ourselves by too-narrow sets of concerns, and instead become well-rounded professionals who each excel in a specific area.

But there's a problem: coders need to catch up. I could make a broad generalization that most coders don't know about design, and it would be a little unfair but perhaps a tiny bit true too. Instead, I'll say that I don't anecdotally know of many coders who understand even design basics. Do you?

Here is my plea: let's all start learning about the opposite discipline and continue blurring the line between them. Designers: learn to code and understand why technical

proficiency brings about great software. Coders: learn about design and understand why the little details affect the entire project. Let's stop assuming our colleagues are out to undermine our interests and get on the same side.

Let's stop worrying about whose concerns are more important. All designers and all coders are builders.

We share the same concerns.

We all want to make smart, useful, and delightful things.

We are all builders.

It's Good to Doubt

Doubt. With creative pursuits, it's inevitable.

Not just with design, but anything creative—starting a business, writing a blog, or even choosing a color to paint your bedroom walls.

If you are learning design, you might think that once you get some more experience, you'll be more comfortable and will doubt your work less and less.

But doubt happens regardless of your experience level.

I just wanted to share a quick story about doubt. I remember back when I was writing, designing, and recording for the second edition of my ebook, *Bootstrapping Design*. (This isn't a sales pitch, but this story is a great example of doubt.)

I was designing a landing page and recording the entire process for a new video series, and I was a little worried that the design I made during that live recording wouldn't be good enough.

Why? I've probably designed a hundred websites over the past 10 years. Designing one little example landing page, rationally, should be no big deal, right?

But I've never had to show every step of a design on camera—in front of a bunch of people. With creative work, mistakes are an essential part of the process because that's

how you find the right solutions. Opening up and showing those mistakes to the entire world (or in my case thousands of people) was, to put in mildly, outside my comfort zone.

I started having thoughts like:

"I'm going to look like a complete idiot if this design doesn't look good."

And doubt started creeping into the back of my mind.

Even though I've worked on all kinds of high pressure design projects and have years of experience in this kind of work, I still doubt myself.

But what we forget is that **doubt makes your creative work better.** If you are so confident that you never question your ideas, how could you improve?

It's good to doubt.

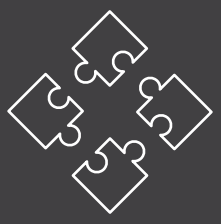
When I started to doubt that example landing page design, you know what I did next? I worked harder. I paid more attention to details. I made sure the design wouldn't turn out horrible by addressing every one of my doubts. The result was a better design example and a better way to teach people about design.

So when you work on your own design, embrace the doubt. Don't beat up on yourself for being unsure—doubt your ideas and know it's okay to make mistakes. Work through the questions, and you will know when you arrive at the right idea.

The trick to managing your doubt is to **avoid making decisions based on fear**. Instead, find the exact reason you

are doubtful about the idea, and consider how to fix it. Make a rational decision about how to make the design better.

Also, be honest with yourself: is your doubt really about the design, or are you just fearful in general? Part of creative work is knowing when to push ahead, even when you aren't confident. Confidence comes after you take a risk and put that design out there, despite your fear, and find that it was a good design all along. It's exhilarating. You deserve that kind of experience.



Part 3

Making Your Own Products

Many designers only create on behalf of others. But why not make your own products? Here are some of my adventures in product making. Learn from my successes and failures.

Reboot. Relaunch. Redesign.
Pivot. Sunset. Shutter.

The Knack, a Web App, Story

Editor's note: This article started it all. It was my first public post about design and business, and laid the groundwork for what eventually became my first ebook and the very newsletter you are subscribed to. This article hit the front page of Hacker News, and brought me 3 full time job offers, and connections with many smart people who have taught me a lot. What follows is a heavily edited and partially rewritten version of the original. If you've ever considered making your own products, this story will teach you a lot about how not to start a business.

This article is about the demise of my labor of love for over a year, Knack, an analytics-focused gradebook SaaS for schoolteachers.

The first version of Knack took me ~2 months to build, spread over 5 months of freelancing. Launch was August 2010. I worked on it another 4 months of the following year, and on Sept 21, 2011, I shut it down.

Entrepreneurship, and the Act of Creation vs. the Pit of Despair!

My first year of entrepreneurship was spectacular and melodramatic. It was unlike anything else I have done in my life or my career. I made something awesome.

Throughout the year after launch, I struggled to find users and make sales. I tried every marketing tactic I could afford. I wrote personal emails to bloggers asking for coverage, or at least feedback. I took a stance. (That teachers are unfairly blamed for the problems in education). I lucked out and had a chance to write about my stance for GOOD online. My thoughts and writing were a bit undeveloped, but it was exciting. I got a few hundred visits. This was the high point of the year, and came a month or so after launch. It didn't last long; the traffic stalled. I struggled to get users. I added features, relaunched, redesigned, repositioned. I rewrote the marketing website over and over. I ran Facebook, LinkedIn, StumpleUpon, and AdWords ads. I wrote link bait blog posts. I tweeted. I facebooked. Sent more emails. I offered free subscriptions to every teacher I met.

A few people loved what I had to say, but none of them used my web app. (Almost none.) Over the year I had about 120 free trial sign ups (100 of which I bought with AdWords), and a total of exactly 10 users who paid for at least one month. I lost about $2000 out of pocket, which I easily financed

through freelance work. Monthly costs were about $150 total for my merchant account, hosting, and other services. The price in time was more substantial. Knack occupied about 6 months of work stretched out over more than a year.

My Great Idea

Knack was supposed be an alternative in an enterprise software dominated market. It would underdo the competition. It would save users measurable time every day. Knack would empower downtrodden educators who are blamed for the problems in the public education system.

My goal was to carve off a niche of young, tech-savvy educators who are passionate about education reform. I'd build an app for them. I wasn't going to revolutionize an industry. I didn't need every teacher everywhere to use it or even like it. Just a few.

I focused on building a great app. I reworked core UI features until they were right—sometimes 4 or 5 times. I ignored the laundry list of other features most gradebooks have, and just built what I thought I knew teachers needed. The only blogger who graciously covered Knack called it a "beta version." That stung, but it was fair. I had launched early on purpose, following the advice of tech industry maxims. I knew I could always fix it up later (and eventually I did).

I charged money for my software. I could not risk

freemium and getting stuck with server bills I couldn't afford for people who didn't want to pay me. I set the price low—only $4.99/mo. That's cheap for a web app.

The same month I launched, a funded silicon valley darling launched too. I felt blindsided. Worried. Then I reread Getting Real and Rework. That new competitor had just validated my market, or so I thought. I put it out of my mind and kept working.

I thought Knack was a viable business. It was at the time the best work of my career, built for people who could reap tangible benefits. The hard truth that it was doomed from the start took that full year to set in.

Sometimes People Don't Want Their Problems Solved

Before I built the app, I talked to friends and family who are educators. Trying to be supportive, they innocently told me what I wanted to hear. They didn't know much about business or software. I didn't know how to get real answers out of them.

I should have done more research about teachers before I started coding. I thought what I'd learned was enough. There were quite a few online gradebooks that charged money and seemed to be healthy businesses. I had found research that said most teachers spend their own money on their

classrooms. I asked educators I knew if they'd pay for a web app. None of this was enough.

I didn't learn about the simple psychology that drives teachers' desires and purchasing decisions. I didn't realize there is a stigma amongst teachers—that they often deeply resent having to spend their own money on their classrooms and careers. Many businesses, both online and off, have special discounts and freebies for teachers. This has warped teachers' sense of value and, frankly, fostered a sense of entitlement.

Beyond this, I've learned that teachers do not want technology solutions. Using my web app or any web app would require a small minimum of effort and change, and any amount of these is too much for many teachers.

Teachers are really busy. A hell of a lot busier than you'd expect. Really. Visited a local school lately? Most teachers are busy putting out fires or trying not to get fired. (Or lose their collective bargaining rights.) I feel for them. I really do.

But because of so many external factors, individual teachers make terrible customers.

One Designer's Arrogance

Years later, looking back on the experience I had with Knack and rereading the original version of this article, I've realized that as designers we face a temptation to assume we know

better than everyone else.

When I planned Knack, I saw the daily frustrations my wife had with managing grades, data, and expectations as a teacher.

I thought I could ride in with a new design and fix it all.

But, now I realize that there was so much I didn't understand.

Often, people use certain tools for a very strong reason and will be opposed to changing or even unable to. For example, many teachers are pressured into using specific software from much higher up the food chain—state and federal laws often affect which standardized tests a school district uses, and often the software they pick relates to that. These kinds of deals are often negotiated between large companies and legislators. In many cases, the audience I picked had little control over which tools they use.

It didn't matter how well-designed my new software tool was. There were some forces that were too large for a solo founder to overcome.

Further, I didn't really understand my audience well enough to try to sell them something.

I looked at an average teacher's workday, identified problems, and then set out to fix them.

But, now I realize that distilling teachers' responsibilities into software workflows is remarkably complex, and probably not worthwhile. The resulting software would suffer an

incredible amount of feature bloat, simply because teachers themselves have too many responsibilities. Carving off a small set of workflows would make my product irrelevant to much of my audience, which is what ended up happening.

Perhaps if I'd made a simpler app with smaller implications, I'd have fared better with this audience. It's hard to know.

But, that said, there are many realities in the education system that make teachers a bad choice for an audience. There are better audiences to serve, and problems that are more ripe for software and design solutions.

With Knack, I certainly learned a painful lesson:

Sometimes, my vision as a designer is naive. Creating a design for someone isn't just about solving their problems my way. It's about understanding them.

When I made Knack, I didn't understand the people who were supposed to use it, and because of that, I designed the wrong thing.

As a designer starting a business for the first time, I approached it like I would any design; find a problem and then fix it.

But business is more complex than problems and solutions. You can't just pick a problem and fix it. You have to find an audience, come to understand them, and then determine whether that have a problem they will pay to fix.

$30,000 eBook Sales In 2 Months

Editor's note: This is the most popular article I've ever written. I wrote it years ago, but I'm including it here because the lessons I learned are still relevant.

Just over 2 months after I launched my design ebook for startup founders, I broke $30,000 in sales volume. Here's what I learned.

Research Customers

6 months ago, the idea of writing a book was inconceivable. I've never wanted to write a book. I didn't think I had anything to say.

When I started—with research—what would become my next project, I was surprised. Not surprised at myself, that I had discovered some new ambition, but surprised at what people needed and how well I could meet that need. Me, a designer, not a writer.

See, by beginning my project with research rather than an idea, I found an opportunity. It wasn't an opportunity I could have imagined nor one for which I would have intentionally searched. It was an opportunity that already existed out

there—on the web, in tweets, on blogs, and appearing in the frustrations of certain people. (The audience I researched was programmers who are bootstrapping software businesses.)

The primary reason for the success of this eBook is that the idea came from my customers, not from me.

My business is succeeding because it began with an understanding of customers. This understanding includes not just what they need, but also more important insights: what they buy, what they value, how they communicate, and where they hang out.

You think you already know these things about your own customers, but you don't. Your assumptions are wrong.

My first business was based on such assumptions, and it crashed and burned in silence. Don't make the same mistake. Research first.

Price by Value

I set a price for my ebook that some consider too high. Their opinion demonstrates that these people are not really in the audience for my book.

My audience is composed of professionals—they're good at what they do and they are paid well for it. While many in my audience can't afford to hire a designer outright to work on their bootstrapped side projects, they are comfortable paying for products and services as part of doing business.

They donate time and money to open source projects and they enjoy supporting products they like. For them, it doesn't matter that much if the eBook costs $12 or $39. All that matters is that it helps them to build a more viable business.

The numbers prove my strategy worked well enough. I earned $30,286 in total sales volume (revenue) as of 66 days after the launch.

When Mistakes Happen, Just Apologize

I made an honest mistake after launch. The coupon I had promised everyone on my mailing list expired before I said it would.

I reactivated the coupon, extended the expiration by an extra week, and sent an email to the list apologizing. This apology email drove about $1,200 in sales. Other sales newsletters have not converted quite so well.

Would I have earned more sales if I hadn't made that mistake? Did the apology completely close the gap? I have no way of knowing. Regardless of speculation, the apology email made an impact.

Later on, my payment system had an issue where credit cards were being declined for no reason. A couple of customers were very kind to notify me, and I got in touch with the support teams for those third-party systems and they fixed the issue quickly.

Rather than leave it there, I sorted through the logs to find about 10 people who had been declined when trying to purchase. I wrote a personal email to each of them, and included a $5 discount coupon as an apology. A few of them had already purchased after the error was fixed, so I refunded them $5 instead.

These apologies have led to customer relationships and great feedback, and I know that several of these customers recommended my book to others just because of this experience. Apologizing and fixing the problem was not only the right thing to do, but it drove more sales by word of mouth.

So, what now?

The last couple of months have been strange because I have no idea how to run a profitable business. I'm making it up as I go.

I've completely skipped marketing practices like SEO and A/B testing. Many people claim these techniques are essential to running a business, but the truth is they are long-term strategies for optimizing something that already works. SEO and multivariate testing can't do much if people don't want your product. This time out, I learned to focus on getting the product right before I worried about following every best practice. I can always work on that stuff later.

Next, the revised edition of the eBook is in the works. I have dozens of lengthy feedback emails and even full copies of the ebook annotated by customers to inform the revisions.

After that, I'll start researching the next project. I have an awesome customer base to learn from, and they're already asking for more.

Don't Let Trolls Hobble Your Marketing

Some months ago, I decided to change how I write. I'd realized that much of my writing came across as preachy. "People just want helpful information," I theorized. "People don't want to be preached at."

So, I wrote posts geared towards being helpful and did my best to avoid sounding judgemental.

And something strange happened. These posts didn't get so much traffic. People started unfollowing me on Twitter. My writing frequency decreased.

I had a difficult time deciding what to write. I'd often open up Twitter, then immediately close it because, well, I didn't have anything to say.

My decision to avoid sounding preachy had removed the best aspect of my writing: the passion. And people lost interest.

When I changed back to my old, ranty style on Twitter, people immediately began engaging me. Even though I was standing on a soapbox. Even though my bold claims had some slight inaccuracies, or didn't apply to everyone. Even though

some of my statements were a bit idealistic and simplistic.

Looking back over my blog and tweets, the difference is amazing. Bold, controversial content always gets more interest. Some people get mad, but even more thanked me for speaking up. And then they subscribed to my newsletter, followed me on Twitter, or bought my ebook.

I'd changed my writing because of my fear of dissenters. I was tired of having people nitpick my arguments. I was afraid the only reason people were reading was because of the controversy—that they didn't really care what I had to say. I didn't want people to think I was just rocking the boat so they'd buy my stuff.

I was wrong. Developing a voice and taking a stance are what earn you an audience. It's how you get people to care about what you're doing.

So when you're writing landing pages, blog posts, and emails for your business, take a stance. Say something bold.

> *"If no one's upset by what you're saying, you're probably not pushing hard enough. (And you're probably boring, too.)*
>
> *—Jason Fried and David Heinemeier Hansson in* Rework

Make some people mad. You'll find that people who share your worldview will gravitate toward you.

And do yourself a favor: spend your time talking to those people, the people you like, rather than arguing with grumpy strangers on the internet. You and your audience will be a lot happier.

Good Enough Design

When you're building a business, you read about all kinds of best practices you aren't supposed to neglect: design, user testing, marketing, security, etc. So you start bookmarking blog posts and make a to do list. That list gets long really quickly, and you start to wonder. How can I do all this? How could anyone do all this?

Many, many disciplines are required to build a business, and you're learning about how to trim your idea down into a minimum viable product. But even that tiny remainder is intimidating if you aren't already an expert in each discipline. How will you develop a marketing strategy when you've never done that before? How will you build it if you can't code?

The problem is, everyone has an opinion about how you should build your business, but no one is willing to stop short and say "That's good enough for launch." The marketing experts say you should have a flawless marketing plan. The design blogs say nothing short of the best will do.

Then, what started as the minimum gets bigger. And bigger.

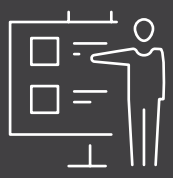
If you're bootstrapping a business, there is such a thing as good enough design. Furthermore, *there's such a thing as*

too much design, too.

On launch day, your design doesn't need to be perfect because your goal is to sell a product, not to win a design award. Similarly, you can still earn sales if your marketing is imperfect.

If you keep refining the design, the marketing, the elegance of the code, and whatever else, you won't get to launch. **So, you have to learn to be comfortable with good enough.** You have to learn to filter all that advice, and decide which pieces are essential for your business.

Work towards good enough and launch your product. You can fix and tinker later.



Part 4

Freelancing

Strategies for making more money, managing clients, and tackling the unpredictable aspects of freelancing.

Fire Me

How Saying No Gets You Hired

"I think I just told my client to fire me."

Over dinner, I was telling Rachel, my wife, about my day. I'd sent a really difficult-to-write email to my best client, telling them how the project they wanted to hire me to design was wrong for them.

When I start working with a new client, I have a "Getting Started" PDF I send them that includes this paragraph:

"I don't just sell a coat of paint. If you can't measure whether a project affects your bottom line, it's not worth the investment. So I'll never try to sell you just a pretty website. I don't accept a project if I don't believe my work will pay for itself."

Scoping out this project and comparing it to my client's goals, I knew it was going to be a waste of their budget. And therein was the dilemma:

Do I tell my client not to do this project and end up not getting hired?

It's obviously in my interest for them to hire me. I want work and I want to pay my mortgage. But do I really mean what I wrote in that PDF? Am I really the type of consultant who will deny the project if it's wrong for the client, even when I want the money?

At the time, I was concerned I might be about to head into a dry spell in my business. Turning down the work when facing that uncertainty was even more difficult.

But I started consulting because I want to be proud of my work. If I wanted to go do mindless design work and overcharge clients, I could go back to working for The Man.

So I turned down the project. I felt is was the right thing to do.

(Sorry for the humblebrag. There really is a point. Stay with me.)

You know how the client replied to that email? It went something like this:

"You're right. Instead, we want you to handle our product design and marketing from now on."

They hired me anyway. For a bigger project.

Even better, they followed my advice, and we got to work on a project that had a much better chance for success. I taught my client something: I really am invested in their success. I might have turned down work, but I earned trust.

Consultants fear the lull. I've written about the feast or famine cycle before, and have explained how earning repeat work with the same clients can be so much better than having to find new clients over and over.

But to earn repeat work, you have to build trust and act as a partner. You have to deliver results or you're not getting hired a second or third time. You have to make hard calls and

be the expert, even if it means you're putting yourself out of a job.

So much freelancing advice is structured around pitching and landing gigs. Learn how to write emails to clients or position your services. Learn how to raise your rates.

But there's little advice about how to partner with clients. I don't mean to criticize these folks teaching you how to run your business—many of them are my friends, actually—but it's so important to consider the ethics of freelancing.

Freelancers have a bad reputation. Many clients learn to see us as unreliable and only interested in racking up those billable hours. How many clients have told you that they've had a bad experience with another freelancer? I hear it all the time.

Here's why that happens. Because of fear, we book any project we possibly can, and in the process, take on work that's not good for the client. The project inevitably doesn't get results, which adds to the bad reputation.

Master your fear of losing clients, and you'll earn respect and get hired more often.

But if you can master your fear and truly be an expert—even when it's better for the client than for you—you'll be able to distance yourself from that bad reputation. You'll get the respect you deserve as an expert.

All because you acted like one.

Productized

How I launched a productized consulting service in 2 days that brought $15,450 worth of work in the first month.

"Productized consulting" is the all the rage in the product and freelancing communities. You might be tempted to pass it off as merely a buzzword.

I felt the same way, so, as an experiment, I launched one of my own. What I learned is going to surprise you.

Why fixed-length, "productized" consulting is a powerful marketing tool for freelancers

That somewhat confusing term refers to a freelancing service with a fixed length, scope, and price. For example, I offer a "landing page in a day" (LPIAD) productized consulting service, which includes copywriting, design, and front end code for a 1-page website at $1000.

Regardless of what you call it, the benefit is that the client knows exactly what they will get and for what price—up front—which is attractive when freelancing projects have notoriously unpredictable structures.

This strategy isn't new. As with so many of the best

strategies, 37signals, before they built and became Basecamp, pioneered it. They offered a fixed-length and fixed-price design service. It was called 37express: pay $2500 and in 1 week 37signals would redesign a single page of a client's website.

According to Matt Linderman on SvN, 37express was one of the ways that 37signals built up an audience before they launched Basecamp.

More recently, other talented consultants are offering similar fixed services. Nick Disabato offers long-term site optimization and A/B testing strategy for a fixed monthly fee. Jane Portman offers monthly creative direction for software businesses. Adam Clark offers wordpress theme customization and deployment in 1 day. Brian Casel is even teaching a course on how to start a productized consulting service (Disclosure: he interviewed me for the course but I have no financial incentive to mention it).

With all this buzz about productized consulting, the natural question is: why? Why are all these talented folks offering tiny, fixed consulting services?

If you're cynical like me, you assume the answer is simply money. It's an easy way for consultants to make a lot of fast income. But the truth is, this is good for the client too.

Let's start with why this is good for you, the freelancer (hint: it's more than just money). Then I'll address the criticisms.

A massive volume of leads that seeks you out (no more cold calls)

The most exciting part of productized consulting, to me, is that I don't have to search for leads on job boards. No cold calls.

Instead, I got to announce LPIAD like a product, and because of that, get the word about my services out in a completely different way and to completely different people.

A fixed consulting service is easy to position as a product. "Landing page in a day" is immediately understandable, and I don't have to pour as much effort into selling it as I would a 2-week consulting project. I can cast a wider net.

Instead of talking to 1 lead at a time, I can talk to thousands.

This has some interesting side benefits.

While a broadcast method like this might not drive thousands of sales (I hope not: help me I'm drowning in landing pages, oh god, help), it does get my name out in front of thousands of people. I can post it to Hacker News, and thousands of people I've never met, people who I'm so far removed from that maybe no one in my immediate network has met either, will learn who I am and that I offer consulting.

Clients post projects to job boards because they don't know anyone who can do the work. But if they've already heard of someone, what will they do? They'll reach out.

Even after I shut down my productized consulting sales because I had too much work, people kept emailing me to ask if I was available.

Let that sink in for a moment. I didn't have to go looking for work. It was finding me. That's the single best reason to start productized consulting right this very second.

Offering a low, fixed price brought leads for larger projects

Maybe this is unfair, but before I launched my productized consulting service, I thought the price would attract penny pinchers. The types who mean well enough, but don't understand that I can't design a brand new Pinterest clone for them in 2 hours.

I was completely wrong. The people who emailed me interested in the service were respectful of the price and my time. Those who hired me are even better—not a single one has crossed the line by asking for free, extra work.

However, a few of those emails turned into leads for larger projects. Three nice folks emailed me saying they'd like to purchase several 1-day projects, inquiring how they should do so. I responded by explaining my process for standard consulting work. So far, two of these have booked paid projects (accounting for slightly more than half of the total

above).

Thus, offering a lower-priced fixed consulting service didn't cannibalize standard consulting work. In fact, the opposite is true—it served to get my name out in front of clients with regular-sized budgets too.

My existing clients didn't mind

The other fear about launching productized consulting was that existing clients would see it and immediately fire me, after seeing that I offer a service for so much cheaper.

One client saw my productized consulting service launch, and mentioned it. The only reason he brought it up was to congratulate me—he wasn't phased by it at all.

I think that launching this type of service carries little risk, so long as you're smart about explaining it. Existing clients aren't going to fire you, especially when they see that a fixed consulting service provides something different than what they need.

Could fixed consulting groom future clients?

I have no data to support this yet, but if clients from my LPIAD service do well, there's a good chance they'll return and hire

me for a bigger project. I've already established a relationship and trust, so why would they hire someone else?

Even if they don't return, it's still a new connection. If they're building businesses, they certainly know others who are too. That means more leads through referrals.

Either way, as a consultant, there's a strong incentive to do a good job.

The criticism: are we over-billing for a small amount of work?

After I launched my "landing page in a day" (LPIAD) service, through some fluke it reached the #2 spot on Hacker News for the better part of a day.

However, the questions and criticisms began pouring in. Numerous HN commenters posed that the price is too high and the result too unpredictable. I received lots of email from potential clients, asking "Can you really write and design a website in 1 day?"

I absolutely can, but not without a strict structure. Let me teach you how to do the same.

A strict process ensures quality results

Calls, meetings, contracts, proposals, and onboarding take

a lot of time and drive up consulting fees. To make a 1-day project realistic, I had to strip out as much process as possible but still ensure I get the information I need in order to do the work.

I automated my onboarding process to limit the amount of time invested before the project starts. After the client pays up front, they automatically receive an email questionnaire. They reply with answers and a date to schedule the project. I confirm over email. On the day of the project we talk on Skype for 15 minutes. I run this call very strictly to ensure I get all the information I need to do the work—unfortunately, there's no time to chew the fat.

(Note: I'm changing my onboarding process for the next round of projects, but that's a different article in itself.)

The other way I save on time is that I use my design framework. It's a set of tools, such as color schemes, font pairings, and other base styles, that I've built on top of the Bootstrap framework. It allows me to produce custom results very quickly, and without it, there's no way I'd be able to finish these projects in 1 day.

Every single person I know who offers productized consulting has a similar limit to make it realistic. I my case, it's using a framework. Others use a specific tool or technology, or set a hard limit on monthly hours.

As a client, there's less risk because of these constraints; the entire process is engineered to be predictable. We

consultants couldn't sell a productized service if it wasn't. However as a consultant, it's a major test of your project management skill. If you don't plan well, you'll end up spending way more time on the project than you get paid for. That also means there's no downside for the client: if the project goes as planned, they get what they paid for, but if the project plan goes wrong, they get more than they paid for!

To offer a productized consulting service, it's important to add constraints and manage time well. But if you're confident in your project management skills, the rewards can be substantial.

The requirement to pay up front filters out window shoppers and leaves more time for clients

Completely automating the lead-up to a consulting project has a side effect: what do I do about window shoppers?

If you're a freelancer, you know what I mean: those people who email you, talk to you on the phone, and ask you to send a proposal, but who probably never intended to hire you.

Reducing window shoppers is obviously good for the consultant, but it's good for the clients who do hire us too. As a consultant, if I can reduce the amount of effort I put into finding work, I'll have more energy for my clients. Nothing is quite so soul-sucking as sending proposals to job board posts.

If I can do that less, the quality of my creative work will only be better.

With the 1-day projects, I was especially concerned about people taking up time but never hiring me because there was a potential to reach a much larger volume of people. The easiest way to filter out these (certainly well-meaning but still time-draining) folks was the price.

If a client pays me $1000 up front, they are serious. I can't think of a better way to chase off window shoppers.

The truth is, $1000 is a not a large budget in the broad sense. My consulting fees average higher, and I'm sure yours do too. That's cheap to a serious client who might be accustomed to projects with 5-digit price tags. To those clients it's also a fairly low-risk way to begin a relationship with a new consultant.

However, most people purchasing my 1-day projects are launching new businesses. For most of them, it's their first business and/or it's a side project.

You'd think the price would scare them off. But for these clients, $1000 is a steal for what I deliver. I don't just design the page: I write it, give marketing advice, and code it. Finding freelancers to complete all that work would certainly price well beyond $1000.

(Frankly, at the end of a landing page in a day project, I'm exhausted. I put a lot into these and could never manage to do two in a row.)

But from talking to my clients, I've learned the value isn't even those deliverables. It's the opportunity to get advice—and frankly, to have an experienced person do the hard part of marketing or launching a business for them.

"Productized consulting" is hip slang for old-school sales tactics that work (really, really well)

Let's be honest with ourselves: consultants of all types and industries have probably experimented with varied fee structures for as long as consulting has existed. This is nothing new.

An iron-clad price and product are a simple way to explain the benefit of hiring a consultant. There are a hundred comparisons, but here are a few: financial advisor order fees/commissions, realtor fees, and Roto-Rooter (extremely popular, productized plumbing!).

Fixed consulting earns all types of experts more reliable income and better-educated customers. It has a strong history, and that's just one more reason why you should try it.

How to Offer Retainers, the Holy Grail of Consulting

Most freelancers burn through clients faster than King Arthur loses knights (especially if you're enjoying the Monty Python version).

A thriving consulting business needs a constant stream of new clients, after all. You write proposals regularly to try and keep new projects coming in. If you're doing really well, maybe you can line up several projects in advance.

Freelancing sites, books, and courses (many of them very, very good and written by my friends) advise you on how to optimize lead finding. Build an onboarding process, like Paul Jarvis teaches. If you refine your pitching practices, like Brennan Dunn does, you can raise your rates and write fewer proposals. Or, simply consider how you email with clients, like Robert Williams advises.

This is great advice and you should follow it.

However, no matter how efficient you are, earning that constant stream of new work gets exhausting sometimes: writing cold call emails, responding to job board posts, and answering referral emails. You end up booking only a fraction

of proposals, and you never quite have enough leads for next month.

There's a different way to consistently earn freelance work, without needing a constant stream of new clients.

Instead of trying to book single, one-off projects, pick your favorite clients and guide them towards ongoing work and a long-term relationship.

Start offering retainers to the people you enjoy working with most. Your quest is not so dangerous as you might expect. Here are some tips to avoid the killer rabbits, black knights, and bridges of death that await you in finding the holy grail of consulting.

Why try retainers? Retainers in practice: a short story.

I restarted my design consulting business 10 months ago (although I've been a professional designer for over a decade).

Editor's note: As of this publication, I've actually been working with clients exclusively under retainer for over a year and a half!

As I started looking for clients, I quickly tired of the lead-finding slog. The first couple of months, I'd managed to work with some phenomenal clients who rehired me, and slowly (and densely) started to realize that they had ongoing

challenges.

I assumed that the ongoing work wouldn't be desirable. Usually, it entails making little textual website updates once a month or monitoring Google Analytics. That sounded like the exact opposite of the kind of business I wanted to build.

But as I listened to what my clients were saying, I realized their challenges were deeper than tedious maintenance work. In some cases, maybe a client didn't really understand how to use the assets I'd just made for them. Or, they had aspects of the business they really just didn't want to have to think about.

I had a crazy idea: what if I pitched them on retainers? I could be the solution to those long-term frustrations my clients were describing. These needs didn't fit neatly into a one-off project, and were happening on a regular basis. A retainer just made sense for them.

A retainer would be great for me too. If I could count on small projects each month from a handful of clients, I wouldn't need so many new leads.

So, I did a ton of research on how retainers are structured and how to sell them, and broached the topic with my clients informally. What I learned would inform the biggest success of my consulting business to date.

Here's what landing several retainers did for my business. My total monthly income for retainers is equivalent to a low salary, and is more than enough to pay my monthly bills. My

retainers free up the time I would have spent looking for new leads. The impact has been so significant that I can work on my own products and even accept one additional large project every month. Often times, my retainer clients hire me for larger one-off projects (when that happens, I simply credit the retainer fee towards the larger project).

My consulting income is consistent. It's also higher than before I had retainer clients.

Even better, I get to work with my favorite clients; the people who value and respect my work and the ones who pay me on time. I even have one client who often pays the retainer fee before I've even invoiced them!

Most people think retainers are a con job, and they usually are

Back when I mentioned retainers to each client, I was wary of the word, "retainer", and didn't speak it until much later. What I did say was something like: "It seems like you have some ongoing needs for design support. How can I help you with that? Is there a way I could support your goals on a monthly basis?"

Of course, once each client began to see the value of a long-term partnership, the term reared its ugly head, and some of the reactions were severe. "What if I don't use all the

hours I paid you for in a month?" "How many hours do I get?" "Can you give me a discount on your rate?"

Few people in our industry offer retainers because they have a bad reputation. Clients expect a "use it or lose it" allotment of hours that they purchase in advance at a discount. Some consultants might try to sell those blocks of hours and hope they go unused, which means free money.

I don't blame clients for being suspicious, because a "use it or lose it" block of hours really is a complete con job. It's unethical. I would never feel right about billing for hours I didn't work, and I'm sure you wouldn't either. Plus, scamming your clients isn't a great way to get referrals when you do need them.

So, with this negative reputation in mind, I set out to make a totally different type of retainer. Here's how I did it.

(Note: maybe this is bad reputation why many consultants are using the term "productized consulting", some of which are just highly structured retainer agreements.)

Offer retainers only to clients you've already worked with

In my business, a retainer starts out as a regular old project. I design or write something for a client and deliver it.

That first one-off project is critical; I get to evaluate the

client, and they get to evaluate me. We figure out how well we really work together. Further, I get additional chances to learn about the client's business and what struggles they're having. After the project ends, if everything went well, I use those insights to pitch a retainer.

Never pitch a retainer to a new client for three reasons:

First, you wouldn't know what kind of work to offer on retainer. Only the experience of actually working with them, side-by-side, provides that knowledge. I haven't been able to get enough out of interviews, questionnaires, or any other qualifying method. The client isn't always aware of ongoing challenges, or doesn't expect you could help solve them. So, you need to see these things happening.

Second, you don't want to get trapped working with a client who doesn't follow your advice or who you don't like. No matter how hard you try to ask clever questions to weed out difficult clients, when you start a real project, there are always surprises. Use fixed-length projects to qualify clients for retainers.

Third, a new client won't trust you and will say no every time. You need to build some trust and prove you do good work during a real project before a client will commit to recurring work.

Pricing: no discounts or counting hours

During the pitch process, the question of rates will naturally arise. It's also the most difficult part of selling a retainer.

What do you do when the client calculates an hourly rate and asks for a discount?

First, talk about profits instead of rates

Never negotiate your rates. You deserve to be paid fairly for your time and experience. I know from first-hand experience that this kind of refusal is difficult; it's confrontational and a little scary. You feel pressured to slash your rates or lose the client.

Stay strong. You can survive this part of the negotiation. It all starts with a single sentence:

"I sincerely appreciate the opportunity to work with you, but I can't negotiate the rate."

I use that same sentence every time a client asks for a discount. It explains that I value the relationship while also politely refusing to offer a discount.

Immediately following that, I'll write about the results the monthly work can deliver. If it's uncertain, I suggest we crunch some numbers before signing a retainer, so that the potential for profit is clear.

Note that I'm completely redirecting the conversation.

Instead of falling into the trap of talking about hours, rates, and discounts, I change the topic to talk about what the client wants to get out of hiring me in the first place: profit.

The client isn't considering a retainer because they have money to burn. It's about achieving a business goal. So, remind them of that, and position your monthly services as an investment they can put towards achieving that goal.

(Of course, you have to sign the retainer too, which means you have to help them make real progress, or they'll fire you!)

Second, remind your client of the value

Negotiating a discount and counting hours steers the conversation away from the value you provide every month.

I realize that this value thing might sound like complete doublespeak. The easiest way of figuring out the value is simple math: how much did I pay for this? How much did I earn? If I could pay less, I'd earn more. That's more value.

Some people will find the value argument to be disingenuous—that it's just a different form of the same old con; instead of "use it or lose it" hours, these critics might think I'm just trying to jack up my rates under the guise of "value".

Profit is value. However, profit, while powerful, is only one part of the value you can provide to a client. The other

side is simply removing frustration. The ongoing, long-term structure that a retainer provides is arguably more suited to being a problem solver than a one-off project could be. You can observe problems growing and changing over time and adapt your solutions. You get more than one chance.

Removing frustration is a key part in pricing your retainers, because it allows you to steer away from the idea of hours. Frustration is difficult to quantify into hours, but easy to compare to a price. By focusing on frustration, can also demonstrate that you're more concerned with being helpful than you are about rates.

Set a flat monthly fee

I set a flat monthly fee on my retainers. My clients trust me to tell them when they ask for too much retainer work in a given month and when we need to talk about additional fees or make cuts to the scope. I do this all without counting hours.

Does that sound insane and impossible? It's not. Since you've already worked with the client, you have an expectation about how projects will flow.

It's also about setting expectations with the client and wording contracts clearly. Each of my retainers includes a "small monthly project" which can take "up to X days of work", in addition to extra deliverables that are less about hours and more about value (which is advice I followed from Brennan

Dunn's book).

My retainer contract states:

"I'll advise you on how to get the best value each month and I will act in good faith to use the time efficiently. Further, I will notify you if I expect the monthly project to be especially short and recommend how we can accomplish the most together each month."

I clarify how overages work like this:

"I will notify you before we agree upon the monthly project if it will require extra work beyond the scope of this agreement. I will also provide a written estimate for the overages. I will not complete extra work without your permission."

These statements help my clients to be comfortable with the scope of the retainer without requiring an allotment of hours.

Of course, you have to earn that trust by delivering great work and results. Retainers aren't for you if you just want to sit on the beach.

The first month of a new retainer, I work very very hard to prove my value. Retainers are about the long term, and if you invest some work into them up front, it pays off.

Provide ongoing value, not just a deliverable

So often with consulting, we fall into the trap of thinking we just make a product or a thing. With retainers especially (but also with your regular projects), it's so important to focus upon value.

I keep mentioning value. Everyone writes about value these days (especially value pricing). What does value mean? It means you're earning your keep. It means you are making your client money and making her life easier.

With a retainer, your client will invest a significant budget over the long term and will expect a lot more from you than just a Photoshop file or a few lines of code. They want results and piece of mind in addition to output.

So, when you pitch a retainer, hone in on the stickiest, most obnoxious aspects of your client's business. What is the day-to-day? What tasks are they not keeping up with? Where are you best positioned to make them a boatload of cash?

For example, one of my retainer clients has an aspect of marketing they just don't want to manage, so I took it over for them completely. Another needs me to keep her accountable to follow the plan we set in our first project. I've also handled other recurring work in retainers, like: running a monthly newsletter, managing ad campaigns, and designing new marketing and product assets each month.

Listen to your client. Then, write a nice proposal about how they can pay you every month to make their pain disappear, only to find it replaced by stacks of cash. That, my friend, is value. They'll hire you on the spot.

Tailor your retainer services to each client

Each of your clients are unique. They have a specific audience, product, and voice. Match your retainer pitch to this.

However, it's not just about adapting your pitch. You need to adapt your *self* and your *services*.

I learned pretty quickly after restarting my consulting biz that clients weren't just buying a pretty design. They want an outcome, regardless of how much work or complexity is required to achieve one. When finding people with the right skills to hire can be so difficult, I handle every aspect of the project personally and make that outcome happen.

So, to book retainers, stop thinking of yourself as a designer, coder, marketer, writer, or whatever. You need to be a partner. Diversify the services you offer. Be the go-to person. Be the one consultant they've ever met who will steer every aspect of the project like it's magic.

You can do it. If you don't write, start practicing on your own portfolio site. If you can't design, pick up some basics so

you can advise your client.

Being a generalist has brought me more retainer work than being a designer. Not because anyone out there is looking to hire a generalist, but because it allows me to adapt my services to each client.

By being adaptable, I can do a better job of solving my client's problems and I can work with a wider variety of clients than if I were a specialist.

That said, if you aren't convinced about diversifying your skillset and the type of work needs to stay the same, at the very least consider how you can tailor your work each month to meet the client's needs. Even if you're essentially still doing the same thing, you can also deliver a monthly report, teach them a new marketing tactic, or provide another artifact that goes a bit deeper than the usual deliverables.

Contractual requirements

That doesn't even begin to cover all the possible ways a retainer can turn into a trainwreck. I've also written up several other requirements that I put in place to avoid issues. I discuss these with clients so they aren't surprised when they see the contract.

2 weeks advance notice required for booking monthly

project

I'm not on call. All retainer projects are scheduled as normal, but with a smaller lead time than I usually require because I already know to reserve a slot for that client. The result is that I end up rearranging my schedule occasionally, but everything still fits. Also, no one is expecting me to make a website update at 3am on a Saturday.

My contract states that I am allowed to waive this requirement if my availability changes, and I often do. I even encourage clients to ask if something urgent comes up.

However, it's important to set the expectation that availability works just the same as with normal projects.

30 day notice required to cancel retainer on either end

At first, a client might worry they will pay for a month and then I could walk away without doing work. (Having worked with the client previously reduces this concern.) The 30 day notice to cancel the retainer contract is there so they know I'm on the hook for the work and that I'm not going to walk away and keep the fee on a contractual loophole. Also, they can rely on me and have a longer-term plan in place.

The 30-day notice also protects me. In the same way, I don't want to do work and not get paid. So the client is on the hook too.

Last, I depend on my retainers for my livelihood. Because I spend less time each month looking for new clients, losing

the retainer would be more of a concern than a one-off project would be. The 30-day notice gives me time to prepare and make up for the loss of income if the relationship ends.

Planned billing: when to send invoices and when payment is due each month

Paying invoices is a major hassle for many clients. It's part of why we consultants so often get paid late.

So, in my retainer contract, I detail the invoicing and payment process, including dates, to ease the headache of paying me each month. I want the retainer to be pain-free and smooth. Nailing down how transactions happen is a big part of this, as no one likes dealing with missed payments or unexpected invoices.

No rollover work or hours

I don't allow projects or time to "roll over" to the next month (like mobile phone minutes used to). I realize that this sounds like a "use it or lose it" setup, but I don't require it in hopes of getting free money. Instead, it's to encourage the client to keep me busy. I don't want the work to fizzle out and the client to cancel the retainer. It's in my interest to keep the work consistent so the retainer doesn't fall into question.

Trial month: cancel anytime, without notice, in the first month. (But no refunds.)

For a client, signing a retainer is a bit scary. They're making a big commitment, and investing a lot into it. To alleviate that fear, I offer a trial month on my retainers. During the first month, either of us can cancel at any time, effectively waiving the 30-day notice that's normally required.

This allows client to feel good about making the commitment. They aren't stuck with me until they've seen how it will go, and if they don't like the results for the first month, they aren't stuck with me.

Similarly, I can cut and run if a client turns out to be difficult despite our previous project and every other indication.

However, it's important to note that my contract requires that even in the event of someone cancelling during the trial month, both that month's fee and monthly project are still due. Both parties still need to make good on the agreement for that month, then it ends.

Get off the lead-finding treadmill

By jumping from one client to the next, so many of us are leaving client needs unmet. Stop abandoning your clients!

Now, I'll acknowledge that for many of us, the appeal to

working for yourself is the constant newness. You get to meet new people and find new challenges regularly.

As a designer, so often I just want to dig into a cool new design style or come up with a new brand from scratch. I don't often get to do that for my retainer clients.

However, having retainers doesn't replace those fun, new projects completely. But you know what it does replace? Crawling job boards. It also provides a more reliable income while only occupying part of my time, so that I can still take on exciting new challenges every month.

Of course, I couldn't end this article without preaching at you again for just a moment:

Who says new is more fun, anyway? Working through challenge after challenge with retainer clients and seeing their success grow is one of the most fulfilling experiences of my career. It's definitely more substantial than the fleeting coolness of designing a new logo.

How many of us consultants can say we helped even a single client's business in a truly substantial way? This doesn't often happen in a single drive-by project. But a long-term client relationship can become a major achievement.

This is how you can do the best work of your career (and you won't need even a single holy hand grenade).

You Are Not The Boss

I write about how designers can establish ourselves as experts, persuade people to follow our recommendations, and build trust so that we can avoid revisions and other frustrating aspects of working with non-designers.

But sometimes you have to do what they say, and there's no avoiding it.

As a designer, you will always be working for someone else. It's either a client or a customer, always.

The person who writes the checks has final say, even if they are wrong.

This introduces frustrating influences into our creative process. We designers sometimes feel we are losing control over our work, just because we do our work for others. If they are paying, we have to do everything they say, or we have to work hard to persuade them to follow our advice.

The day-to-day work of a practicing designer is a twilight battle zone. We fight on two fronts: creative battles against our own ideas and inspiration; and practical battles against the people who pay for and use our design.

But sometimes, it's okay to wave the white flag and retreat.

You are not the boss, and you shouldn't try to be.

That might sound strange coming from me. After all, I wrote a whole ebook that will teach you how to fight those battles, earn trust, and persuade clients and coworkers to follow your advice. The strategies work most of the time, and most designers can see a lot of benefit from changing how they work with non-designers.

But, on rare occasions, the strategies don't work. This happened to me recently (and I'm willing to tell you about it because while I do have a book to sell, I'm more interested in helping you fix this problem).

Sometimes, the person paying you wants want they want. And you think that you have to comply and get paid, or walk away unpaid.

In these situations, I encourage you to realize that this is a false dichotomy. There is another option.

If you are unable to persuade your client or boss to use the design as you made it, you still don't have to agree to change requests you know are bad.

Instead, you can find a new solution.

Usually, when unstoppable force meets immovable object and you see no path forward for the amazing design you have made, what you actually have lurking beneath the struggle is a new design constraint.

It's frustrating to realize that you have a new design constraint after having already made a design.

But, while frustrating, when you hit a dead end in a design project, you have an opportunity and a choice:

You can either remain stubborn about your recommendations, which can make you look like a jerk. (I did this recently, and do not advise it.)

Or, you can back down and find a new solution that fits the new constraint. You can be the designer instead of trying to be the boss.

Yes, it is hard work. Yes, it sucks to step back from an awesome design and rethink many of your decisions. Yes, you want to avoid doing this whenever you can.

But, finding a new solution when you are at an impasse shows the real value of working with a design professional.

A client might see a design, refuse to approve it, and think that if you can't solve it, their project must be hopeless. They fear there is no solution to the impossible problem they have given you. They don't feel great about the impasse, either.

If you take that opportunity and use your awesome creativity to solve the new problem, again, in the face of seemingly impossible-to-everyone-else odds, here's what will happen:

Upon seeing the new design, your client will be pounding her fist upon her desk, screaming *"HELL YES THIS IS AMAZING"* at you over the phone. And then she will proceed to rehire you for another project.

Sometimes, backing down from our vision for a design is

the best way to be valuable. Sometimes, we designers need to set aside our own vision, and just solve the problem the client actually has. Even if they didn't clearly explain it to us at the beginning of the project.

Sometimes, our job as designers is to find the problem before we can solve it. This means rework and scrapping great ideas. This is certainly an exercise in humility. It is a difficult pill to swallow.

But I have never had a client react so strongly and passionately to one of my designs.

For me at least, that is what success as a designer looks like:

Solve the problem no one else can solve.
Solve the problem they didn't know they had.
Solve the problem, even if it's not the way you prefer.

About The Author

“One of those guys who is just good at everything he does (except sports)” (and writing bios), Jarrod Drysdale is a designer, writer, developer, and product maker. He’s written 4 books, founded 2 startups, and writes Critique, a weekly design newsletter, to over 18,000 subscribers.

A practicing designer for over a decade, Jarrod has completed work for companies like State Farm, LensCrafters, McDonald’s, Kraft Foods, New Line Cinema, Domino’s Pizza, Scottrade, Intuit, Bloomberg, and startups and online businesses of all sizes. Many millions of people have directly experienced his design work.

His writing and work have appeared in GOOD, .Net magazine and the Creative Bloq, CommArts.com, AIGA.org, Smashing Magazine, Design Taxi, and Despreneur.

A lover of weird music, and no digital nomad at all but more digital hobbit, Jarrod makes awesome design from the safe confines of his lavish basement design studio/hobbit hole with his trusty Jazzmaster guitar always nearby. Jarrod is a both devoted and devout husband, father, and cat-person-who-also-loves-dogs. 6'2", bespectacled, and bearded, Jarrod loves fancy food and indie rock but don't call him a foodie or hipster or there will be blood. It is however safe to call him tall. (And seriously, he is very bad at sports.)

Get more of my writing at studiofellow.com.

Design's Iron Fist and other essays

By Jarrod Drysdale

Second Edition, October 2016

This ebook is available for free upon subscribing to my newsletter at http://studiofellow.com.

If you enjoyed this ebook, sign up for my design newsletter, Critique, which is read by over 18,000 designers every week. I share exlusive articles there which I don't publish anywhere else.

Email me anytime at: hello@studiofellow.com