

In this lesson, you'll learn why taking an "all or nothing" attitude towards your work is unhelpful, prevents productivity, and increases your stress levels. You'll *also* learn what to do instead.

Taking an all or nothing approach to your work—whatever it is you do—is basically the same as saying that you either want perfection or you want nothing at all.

So, in the end, all or nothing is equal to being a perfectionist.

Now, even the most ambitious among us know that perfectionism is harmful and holds us back. But, it still creeps into our lives because we don't want to do sub-par work. We want whatever we're doing to be the best it can be.

This is a concept I've struggled with my whole life, so here's an example of how it's affected me.

As a writer, over the years, I've found myself working hard to always find a unique angle on stories. I want my work to be one-of-a-kind and valuable.

This desire drove me to develop systems and routines that ensured that I created the best work I was capable of every time I published something. It felt great!

But, eventually, I allowed that part of me that always optimizes and strives for the best to take over. I allowed it to override everything else.

And I started to convince myself that I needed "more time" to create better work. Now, that's not a problem all by itself—good things really do take more time to create. But, in this case, I'd gone too far.

I wanted my work to be perfect from the very beginning. I wanted to start with perfection and then maintain that perfection all the way through. In order to facilitate that, I decided not to stick to a publishing schedule anymore.

Can probably see the problem building? It's the same one you've probably faced before if you hold yourself to high standards.

Rather than creating the best work I'm capable of, I started to create nothing at all. Nothing was good enough anymore. Even the ideas I would start with became not good enough to pursue. And my productivity came to grinding halt.

I'd taken an all or nothing attitude towards my work. And I got the same result that anyone who does that tends to get. Nothing!

And that pretty well demonstrates the problem with seeking perfection. Not only do you never achieve it but, in the meantime, you achieve nothing at all and, as a bonus, you feel pretty miserable doing it.

All or nothing tends to result in nothing. And when you're trying to systemize your life to get more done, nothing is certainly not optimal.

So what should you do instead? You want to be productive, but you also want great results that you can be proud of. That perfectionist mindset exists for a reason, right? You want to create things that you're proud of.

To beat all or nothing thinking, what you have to do is find a way to trick yourself into accepting imperfect results.

That's hard to do—you naturally want to reject it—but one concept that has helped me and many other productive people is a concept called kaizen.

Now, kaizen is a word the Japanese use to describe the process of continual improvement. It originated in the business sector and Toyota played a big role in popularizing it in the early 2000s by incorporating it into their management and production practices. They call it "The Toyota Way."

And the idea is that your processes and approaches to your work are always imperfect, and gradual improvements are always possible.

The primary mindset shift that takes place when you adopt the philosophy of kaizen is that your work is never really done.

Now, at first, that might sound kind of terrifying. How can you get *more* done when everything you do needs to be approached or dealt with again and again to make it better?

But as you start to take on this mindset, what *actually* happens is that it makes it easier to get started and build momentum.

If you haven't noticed, momentum is the primary theme that runs through all our rules for productivity.

Adopting the kaizen philosophy gives you permission to create something imperfect right now by accepting responsibility to improve it down the road. It's a sort of tradeoff that you make.

You no longer have to envision the optimal end result from the beginning. That's a big benefit, because trying to engineer the perfect outcome from the very beginning isn't just impossible, it eats up all your mental resources and leaves you with none to actually get started.

So, you get all those resources back in the beginning and you can use them to actually build something.

Let's look at a few examples so you can see how this could work in real life.

First, let's just imagine that you're an office manager at a doctor's office, and your office is having a problem. You're getting plenty of patients, but they keep showing up needing a different service than what they described on the phone when they made their appointment.

That ends up causing some chaos, the doctors' time gets wasted, and patients are frustrated.

You sit down and think about where this problem could be coming from so that you can design a process to fix it. And you have a couple ideas:

Maybe the receptionists aren't doing a good job getting the right information from the patients in the first place, and that's where the breakdown is.

Or maybe there's a communication breakdown between the staff and the doctors that's causing the doctors to misunderstand what the patient needs.

You don't actually know where the breakdown is, and testing both of them at the same time won't work. Improvement needs to come right now.

So, you adopt the kaizen philosophy. You do a quick analysis to decide which piece of that patient funnel will yield the best results, and you get to work on it.

Let's say you decide to start at the top of the funnel, where the patient is having their first phone call with a receptionist.

When you listen in on a few calls, it becomes immediately apparent that the receptionists are scheduling patients for services that the doctors don't offer.

That's a big problem, so you focus there and train the receptionists until only the patients that you can actually treat are getting scheduled.

You didn't fix the whole problem—there may still be internal communication issues that cause things to break down, but a big chunk of the confusion went away by fixing just one piece it.

That's what success through kaizen looks like.

Now, in working with the receptionists, you might have fixed one problem only to discover during that process that there are other issues with the front office staff that are holding the whole practice back.

At that point, you look at all the problems you know about, and simply pick whichever one is easiest or most important to work on next. It doesn't really matter because, at the end of the day, you're making the whole practice better one little bit at a time.

There will always be more problems to fix and, each time you fix one, you might uncover another. But your whole practice is functioning along the way and it's getting closer to reaching its ultimate goal of providing great patient care with each improvement.

Let's look at another example—building a piece of software.

There's a genesis question that has to be answered whenever a software developer decides to build something new.

Do you sit down and architect the whole system from the beginning so that every step of the way is planned out? Or, do you just build something small, test it, and go from there?

What if you wanted to build a web tool for, let's say, real estate agents? Just an example.

Should you try to figure out every single feature that needs to go in the product before you build it? Or, should you just try to find the biggest pain point that real estate agents have and try to build something simple that makes that pain hurt a little less?

Well, the kaizen philosophy would tell that the second approach is the best solution.

If you tried to fix every problem all at once, it would take forever and you might not even get it right. That's the risk of trying to create something enormous without regular feedback.

Instead, what most successful developers do is build a tiny version 1.0 of their product. In our case, they would talk to 10 real estate agents to figure out what the primary pain point is, and then they would create the simplest thing they possibly could that would make that pain hurt a little less.

Then, they would go and get feedback. It almost certainly wouldn't be good enough, but having something to actually look at would help them have a better conversation about what to fix next.

Little by little, they would quickly identify and fix the problems that would eventually leave them with an incredibly useful piece of software. And they benefited from getting to launch that software from Day One. They didn't take the massive risk of spending months or years building something privately that might not even work.

And this is, in fact, how most software companies work today. They call the process "agile development" and it follows all the same kaizen principles. Quick, iterative improvements towards a larger goal.

They make progress by accepting imperfection.

So, if you want to banish all or nothing thinking and start implementing kaizen in your own life, you only need to accept and implement three key principles:

1. Find the heart of the problem. When you know what your ultimate goal is, it's much easier to take small but decisive actions to get you there.
2. Embrace the suck. You have to just get over the fear of being imperfect. Everything new sucks. And if you want to create something great, the only way to do it is to create something that sucks first. It's how every great thing today came to exist, so you'll be in good company.
3. Something is always better than nothing. Every chance you get, choose something over nothing even if it isn't ideal. It's easier to improve something that already exists than it is to create something in the first place. So just get started to make your life easier.

Now, here's a quick exercise you can do that will help you actually do this.

First, pick a problem you'd like to solve that you've been procrastinating on. Something that you want to get right, but you don't know *how* to get it right.

Now, ask yourself: What is at the very heart of this problem. What is the underlying pain that I'm trying to remove? Go past all the surface level stuff. Each time you answer this question, ask yourself if the answer is *actually* the heart of the problem. If it isn't, ask again until you find it. Act like a five-year-old that keeps asking "Why?"

Here's an example. Imagine that your diet is a mess, and you want to fix it.

What is at the heart of this problem? Why do you *really* want to have a better diet? Is it just because you know it's the right thing to do? Probably not. Is it because you feel sick when you make poor food choices? Well, now we're getting closer. But *how* do you feel about yourself when you eat something awful that makes you feel sick? Is there some shame involved? Or some fear?

Now, I'm not saying that's it for *you*. But that's the kind of self-questioning process you should follow to figure out what the root of the issue really is.

Now, when you find that, go ahead and skip straight to a solution. Right now, what is your *ideal* solution to this problem? Describe it in detail. How different would your life be if you had it or if you did it all at once. What's happening when you're firing on all cylinders?

Okay, do you have that in your head?

Now, I want you to tell yourself that you can *never* have that solution. It isn't true, but that's what I want you to tell yourself right now, because it's critical to what comes next.

And what comes next is to ask yourself, "Okay, if I can't have that ideal solution, what's something a few steps down that from that? What's something that's *close* to the ideal solution since I can't have perfection?"

Now, imagine that you can't have that, either!

Keep going through this process until you have something really small and really actionable that you *can* have right now.

When you complete this process, what you'll have is a small "version 1.0" solution to the very deepest part of the problem that you discovered. It won't be perfect, but it *will* be a step in the right direction.

And because you know what that ideal and perfect solution looks like, you can connect what you're doing right now to that end goal. You can see how what you create today is a step on that path to the perfect solution.

And that is what keeps you motivated. It's seeing how the small steps you're taking really lead towards the ideal end goal.

And that is how you destroy all or nothing thinking and become more productive.

See you in the next lesson!