

In this lesson, you're going to learn a little more about why it's important to break up big systems into smaller ones, and how to go about doing it so that they stay organized and keep you as productive as possible.

And to illustrate the importance of this, I'll tell you a little story from my childhood.

When I was a little kid, I liked to collect things. Action figures, coins, and things like that. But my favorite thing to collect, like most kids growing up in the 80s and 90s, was baseball cards.

I absolutely loved them and, over time, I amassed a pretty big collection for a little kid—about 5,000 cards.

And I never organized them. They were just sitting in piles on the floor of my room, and it was starting to drive me crazy looking at the mess. Probably drove my parents crazy, too.

I had a birthday coming up, and my grandparents asked me what I wanted. So, I told them, I wanted to finally organize my baseball cards. They took me to the local card and collector store (I don't think those really exist anymore, but it was my favorite store as a kid).

The collector store had all kinds of boxes in different sizes for me to choose from. And it just so happened that the biggest box they offered could hold 5,000 cards.

What a coincidence!

Being about 12 years old and not realizing that breaking big things up into little things is the key to progress and productivity, I immediately latched onto that big box.

My grandparents, being much older and wiser, knew exactly the mistake I was making and encouraged me to get a few smaller boxes instead. But, I was certain, and it was my birthday, so they bought it for me.

Well, I'm in my 30s now and the baseball cards are still not organized. Oops! I guess the real birthday present was a life lesson that 12-year-old me couldn't fully appreciate at the time.

But I'm glad I had that experience because it helped me understand how important it is to take big things and break them up into smaller ones to avoid overwhelm.

And this absolutely applies to systems and routines.

If there's some work or some process you're trying to organize with a system, and it's really big and overwhelming, then you should take note here because, unless you're careful, you may end up with an equally big and overwhelming system to manage it. Or, more likely, you never systemize it because it's too much work to put together.

You just trade one kind of overwhelm for another.

If you remember from an earlier lesson, we talked about a rule of productivity that states "all or nothing equals nothing." It's the idea that if you try to tackle everything at once, you won't get anywhere. If everything has to be perfect from the start, you probably won't even get started.

So, just by embracing that principle, you can see how much easier it is to build systems one or two pieces at a time and then build more or reorganize them later.

If you do that, though, you may still end up with a big honkin' system that's hard to navigate.

That's why, in this lesson, I'm encouraging you to break those big systems up into smaller sub-systems that can be looked at and managed independently.

At the end of the day, it doesn't really change the amount of work you do, but it totally changes the amount of work you have to do at any one time. It lessens it a lot.

This can lead to big gains because, once things are separated out a bit, they can be better organized and it's a lot easier to focus and work on the parts independently. I think it's more calming for your mind that way. Less stress.

A website or an application is another example that comes to mind. Some websites are really big and take a lot of code to work. In some cases, hundreds of developers are writing more code for it every day. It's ever-expanding. Now, technically, you need all the code. But, the website also has a lot of pages, and some of the code is used on some of the pages, but not others.

If you put all the code that the entire website needs in one file, it would be enormous and probably wouldn't even load if you tried to visit it.

Also, the people working on it would have to wait a long time for the file to open just so they could edit it. And once they had the file open, they'd have to spend ages searching through it to find the part they want to adjust.

They'd probably have to add a lot of comments in the code just so they could find what they need, and that would make the file *even bigger!*

Worse, if they're not really careful, making an adjustment in one part of the file might mess things up in a different part.

Basically, the whole system would be a giant, tangled up mess that no one could use or work on because there are too many variables all in one place.

#####

So, how do the people managing these big websites solve this problem? They need all the code, but they don't necessarily need it all in one place.

What they do is break up all that code into pieces and store those pieces in different places where they're easily accessible if needed. Then, when they want to make a page appear on their website, they create a new file and write some code that references just the bits that they need.

And now they can find what they need to work on much easier, too.

So, maybe across the whole website there are 100 (or even 100,000) different blocks of code needed to do everything that the website can do.

But on the home page, all that's really needed is the logo and a few buttons that take you to different parts of the site.

The file for the home page could be as simple as writing a couple lines of code that say, "Get the logo and get the buttons and put 'em here."

There's a bank of code blocks, but you only need a few of them at any given time, so things move a lot faster because you only ever have to draw what you need from the bank.

Speaking of banks, it's kind of similar to keeping a bank account. You keep all your money in the bank, where they organize it and keep it safe. Then, you use your debit card to get little pieces of your money any time you need it to buy a sandwich, or fill your gas tank, or buy groceries. You only access what you need when you need it.

You can do the same thing with your own systems.

The work you do might be really complex and interconnected, but you don't have to try to keep everything about it in one place. And you don't have to try to build and perfect all of it at one time.

Instead, try breaking a big, confusing system up into sub-systems for each step. You can keep a list of all those steps in one place, but make them really simple. Just have each step reference something or somewhere else where more information can be found.

This ties in with the previous lesson about documenting your systems.

Let's say you run a retail store, and every night, you need your staff to follow a process to make sure the store is closed properly. They need to clean up, clear out the cash register, organize some things, et cetera et cetera.

And maybe the steps involved are different on different days. Maybe you get a delivery on Wednesday afternoons that need to be handled in a specific way. Or maybe some other process needs to be followed on Fridays.

To make things even more complicated, you have different staff with different levels of knowledge working on different days. And the way the steps in the closing process are completed is really important, so you can't rely on everyone knowing exactly what to do at any given time.

Rather than trying to capture all of that information in one place, you could break it down into different pieces and reference what you need to streamline it and make it easier for you to work on and for everyone else to follow.

You could have one little flowchart that says, "What day is it?" and then direct your team to a checklist that is specifically for that day.

So you have a little bank of checklists, and that flowchart sends people to the right one at the right time.

Then, that checklist just says, "here are the ten things to do to close the store." But each of those 10 steps references a set of instructions in a binder behind the register.

In that binder, you keep a set of instructions for how to do each task on all of your checklists. That binder is your bank of instructions. There might be thirty or forty sets of instructions, but only ten or so will be used on any given day.

Now you have a three layer system that can get anyone to all the information they need to succeed as quickly as possible.

All someone has to do is look at the flowchart to figure out what checklist to use. Then they reference the right set of instructions for each item on the checklist so that they do everything correctly.

At the base of the system is a binder full of tons of information. But no one has to learn all of it all at once. They only have to learn what's relevant for them at the time that it's relevant.

This makes things a lot simpler for everyone.

And maybe you're thinking, "Yeah, simpler for everyone but me!"

If you're thinking that, I'd remind you of two things. First, think of how much simpler your life will be once there's a system that gets everyone the information they need exactly when they need it, and you don't have to train people anymore or answer the same question over and over, or deal with the problems that come when the process isn't followed properly because someone doesn't even realize they're doing something wrong.

Second, remember that all or nothing = nothing! You don't have to build something like this all at once. You can do it over time and as needed. Since the different pieces of the system all live in their own little place, it's easy to find and work on the parts you need to whenever you want. And since the system is now modular, you can take your time because you can work on one piece without disrupting the other pieces.

Your system is now faster, better, and stronger.

So go ahead and think about what big, overwhelming systems in your life could benefit by breaking them up into smaller ones that work together.

Think about how the system *wants* to be organized. How could you divide things up so that you only have to look at or use what you need without wasting time wading through everything else to find what you want.

When you're done with that, I'll see you in the next lesson.