BIZCOMMUNITY

6 signs it's time to re-evaluate your software stack

Issued by LeBrau Press

26 May 2020

Your software stack is your company's secret sauce. It keeps you competitive, productive, and capable.

But software doesn’t stay good forever. Needs change, and not all software updates improve programs for the better. So how can you know when it’s time to look for new solutions?

You don’t want to waste resources auditing your software only to find everything is fine. If you’re going to up-end your software stack to save time, money, and mental bandwidth on things like payroll, accounting, employee scheduling, customer emails (or whatever else you use technology for), you need to know the change will be effective.

The secret is to look for the signs. Here are six reasons you might want to re-evaluate your software stack:

1. It’s not scaling with your business

As you build your business, keep in mind you can easily outgrow your software’s capabilities. Some programs are designed specifically for small businesses, while others are targeted at enterprises.

Look into your software tools to see what their limitations are. How much data can they store? How many employees or customers can use the software at one time?

If your employee count grows beyond that number, that platform is going to struggle to support your staff. Either a limited number of workers can use the software at one time, or the platform will try to support all of them and crash.

2. It’s not user-friendly

How easy is it to learn the tools and systems your company uses? If they’re not intuitive, they may be more trouble than they’re worth.

At the end of the day, software doesn’t accomplish goals; people do. Do new recruits need tons of training before they can accomplish a task? Do they get frustrated and avoid using the software anyway?

If not, look elsewhere. Software should promote productivity, not hinder it.

In every software sector, there are options with better and weaker user interfaces. Beware, though, that you may sacrifice features or customisation opportunities for something that's easier to use.

3. It's unreliable

Even a solid user experience can't make up for a bad service. Don't waste your time with tools that don't work when you need them to.

If you're using a cloud software, does it have 99% uptime? If it's a custom application hosted on your servers, is maintaining the codebase a nightmare?

If you're spending too much time restarting programs and calling IT, it's time to ask whether you should just replace the program. Developers' time doesn't come cheap.

4. It doesn't contribute to company goals

If your current software isn't helping you reach your goals, don't change your goals; change your software.

For example, say you want to expand into a new country. Your current e-commerce platform may not be compliant with that country's laws. If you're trying to do something like cut down on HR costs, could self-service payroll software that also handles employee onboarding save time and keep you more organised?

If software holds the key to your goals, don't settle for "good enough." A cheaper program that requires workarounds is costlier than a more capable one at a higher price point.

5. It's outdated

Technology evolves quickly. If you're not sure where your software is in its lifecycle, consult your tech staff or providers.

One thing to look for? Whether the software is still supported. After a couple of years, companies stop putting out patches and updates for their programs.

Ask, too, whether your current software has customisation and AI capabilities. Can it automate recurring tasks? Does it have the workflow options you need?

If you think you don't need all the newest features, remember: Your competitors are taking advantage of them, even if you aren't.

6. It costs a fortune

Software can help build your company, but it can also drain it. Like any business decision, you should ask whether your software is delivering a good return on investment.

Maybe your software makes your life easier, and its per-month subscription price is low. If it takes a ton of IT support or requires extensive training, though, it may not be worth the trouble.

Do a cost-benefit analysis. Create a T-chart, laying out cons on one side and pros on the other. Do your best to associate dollar values with each. Add them up, and decide: Are the costs outweighed by the benefits, or vice versa?

Don’t fall for the sunk cost fallacy: the belief that you should keep investing money in something because you’ve lost money on it and don’t want it to go to waste.

While evaluating your software stack, take particular care with programs you don’t use yourself. You may not see their true value, but you might also miss key warning signs. When in doubt, check with the people who use each program in their daily work.

If they report a red flag, trust them. It can’t hurt to look into other options, and you just might discover a new tool that makes all the difference for your business.