

Why We Haven't Created Our Own

(Insert name of trendy software product here.)

By Michael Stoner

Michael Stoner is the president and a partner in mStoner. He is a pioneer in using the web and Internet as a marketing communications tool for colleges, universities, and schools. You can follow Michael on Twitter @mstonerblog. He and other mStoner staff members post their thoughts on our blog, mStonerblog.com. He can also be reached by email: michael.stoner@mstoner.com.

mStoner has a broad practice. Our clients include independent schools, private liberal arts colleges, medium and large state universities, small to large private universities, community colleges with multiple campuses, and professional schools—in short, a cross-section of American (and, indeed, international) higher education. **The focus of all our work is helping our clients identify how they can reach their strategic goals. We start by asking questions about what their challenges are, what problems they're trying to solve, what audience they need to reach or influence, and what results they require.**

Often, our work involves specifying various technologies for clients to use in accomplishing their goals. Here's an example: Just about every website with more than a few dozen pages needs a tool to facilitate content sharing across the site and to simplify updating and maintaining content. These are some of the things a CMS is designed to do, and the larger and more complex the site—and the more people managing it—the more important the CMS is. For more complex sites, including most of the sites we work on, a content management system is a core business tool.

We encourage anyone selecting any core business tool to do it thoughtfully.

If we're helping an institution select a CMS, we consider many factors in seeking the right tool. Some are obvious, such as initial cost; available features; and the ongoing cost of maintenance, upgrades, and support. Other factors are also important, though not always as apparent. For example, we want to explore how flexible the CMS is; how scalable it is—whether it can grow or change to meet our client's needs; and how it fits with their technical environment. We also want to be sure that the company that created the CMS will provide our clients high-quality, dedicated support and that it will be around to support its product for some time after we license it.

Think about effective search, CRM, a portal, and social networking solutions. These are no longer frills, but essential components of any contemporary business and marketing tool kit. Our clients need these tools to be successful at meeting their challenges and achieving the results that will help their institutions to thrive.

But, while many of our clients have similar challenges and needs, the solutions to meeting those needs are not necessarily the same. So, each time we specify a technology, we're presented with an array of realities and many short- and longer-term considerations, some of them unique to a particular institution.

And, when a prospective or existing client asks us if we have our own software to recommend as part of an mStoner solution, the answer is, unfailingly: No.

No, we do not have a custom CMS that we can implement for your institution.

No, we have not developed our own social networking tool that might serve as your very own mini-Facebook.

No, we haven't created a "portal" that's the "perfect" solution for alumni networking.

Some companies like ours would consider these missed business opportunities. Developing our own system would enable us to derive considerable income from clients: we would be guaranteed an up-front payment when they licensed our product; we'd be able to charge for any customization or new features that a client needed; and we'd derive recurring income from institutions that would pay us fees every year for updates, maintenance, and support. Sounds great, doesn't it?

Maybe so. But we've thought about this a lot, we revisit it every now and again, and we keep coming out at the same place. We've got good reasons for not creating our own "what's-it." And we're happy to share them with you.

1. The “when all you have is a hammer” theory.

As a marketing/communications firm, our first and most important commitment is to discovering what our clients need to help them be smarter and more effective than they already are. Once we’ve learned about their challenges, we work to help them figure out how to meet those challenges. We also recommend the right tools to meet their needs. Making a significant investment in developing a product of our own would mean that we’d need to sell it to our clients. Not just once, but frequently.

Suppose we created our own content management system, an undertaking which could easily cost north of \$250,000 to develop because we’d want to make it really cool.

Don’t you think that we’d be looking for every opportunity to recoup our development cost by reselling the mStoner CMS to every client we could? Of course we’d be enthusiastic about it! We’d be blogging and twittering about our CMS. And as a result, every new client project would begin to look like an opportunity to implement our product, even when it wasn’t exactly the right fit. We’d be thinking of plenty of “reasons” why the mStoner CMS was a “perfect” fit for each and every client. You know the saying, “When all you have is a hammer, everything looks like a nail.”

Instead, not having our own CMS means we have to listen carefully to each of our clients and recommend solutions that fit within their specific culture, budget, and technical environment, instead of advocating something just because we created it.

Maybe our CMS really would be the best choice. But the approach we take—in which we are paid to provide our best professional advice, period—removes even the appearance of a conflict of interest. (In fact, when a client licenses a product that we recommend and on which we could be entitled to a discount from the developer, we pass that discount on to our client.) We believe that’s tremendously reassuring to our clients.

2. Who’s footing the bill?

As a consulting firm, we’re structured to have exactly the number of employees necessary to meet client needs. Developing a proprietary system would require us to do one of three things: a) take time away from our client work, b) hire additional team members who could focus on development, or c) include the development time for this tool within our client projects. Any of those three options would result in a single outcome: We’d have to charge more.

No matter how we approach it, taking the time to create a custom product would pull our attention and time away from our day-to-day client work, and the costs would end up getting passed on to our clients. Particularly in today's economy, we don't like the sound of that.

3. Getting just what you want.

Getting just what you want is a complicated proposition when it comes to software. Even if a solution is purpose-built just for your institution, you won't have every feature you want. So, you're going to have to make some compromises. Part of our work with our clients involves helping them manage the process of deciding just what compromises are acceptable at their institution when it comes to selecting a technology like a CMS.

We know that no solution is perfect; every one is a hybrid. Here's what I mean: Suppose we built a CMS. It would do some things really well, and some people would appreciate its versatility or specific features. But others would want a particular feature to work in a slightly different way. And still others would need an additional feature to make the CMS exactly right for them.

And guess what? Pretty soon, those people would start to complain if they didn't get what they needed. Meanwhile, our development team would still be working really hard to make other improvements on the system—big improvements, like making it more stable or coding a new feature that a lot of people really wanted. Those could be big improvements that would benefit everyone who was using it. But, by the time we rolled them out, we could have angry customers whose specific needs weren't being addressed.

So, we always ask the vendor of a CMS or other core business technology some probing questions: How big is the development team that's working on keeping the product going? How fast do they turn out releases, and what do those releases include? And, more importantly, what do people who are using the system have to say? Do they get responses to their queries and needs?

Seriously, we've heard people complain about almost every system out there. And we know that one of the challenges of being the software developer of a product with any reasonably sized installed base is that you can never develop features fast enough. We don't want to put our clients in the position of working with unresponsive developers (or those who *appear* to be unresponsive)—and that includes us.

4. Keeping the wheels turning.

And what about support? Actually, how about bug resolution, maintenance, and upgrades? Creating a product is just the tip of the iceberg when you look at the big picture. Supporting that product, whether that means providing a help line for instant assistance or just fixing bugs, is the rest of the story.

The companies that successfully develop and manage products like this have teams of people devoted to creating them and making them work. We wouldn't presume to be able to give you that same quality of support while still doing all that we do now. Frankly, we'd prefer to have our time and attention focused where they should be: on our clients.

Of course, we could try to get clients to pay us for product support. Now, when a project is complete, we provide various kinds of support to our clients. But what we do is substantially different from the kind of support we'd need to provide for a software product (e.g., bug resolution, technical support, maintenance, upgrades, and feature development). And, in general, we believe that the best way to serve our clients is to specify solutions that they can truly "own" and support themselves, or at least be able to work with a vendor that specializes in supporting its product.

5. The power of the community.

One of the benefits that advocates often cite for open source software is that users contribute new ideas and new features to software that they are effectively co-developing. In general, we think, the more people who use a piece of software, the more ideas available to the developer for improving it. So, one benefit of using a content management system like Sitecore (to name just one example), is that every new user benefits from the insights offered by the folks at 40 education institutions worldwide that use this product—not to mention the ideas provided by hundreds of other Sitecore users. And Sitecore has a product development team of more than 20 people working on its product full time, with 45 people providing quality assurance and support. With this intensity of effort, there are constant improvements, large and small, being made to the product. This kind of community takes time—and numbers—to build; more importantly, it requires time and budget if it is to be managed as an effective resource for users. In our case, this would divert us from our core business and make us less able to serve the needs of our clients.

6. Other choices. Lots of them. Lots and lots and lots of them.

Our approach means that our clients work with best-of-class products that are created by companies with a core competence in developing and supporting great software.

That's what they do, period. They don't also dabble in design or content development or strategy, except as related to their products.

Seriously, we have given this a lot of thought.

We faced the decision to develop a CMS back in the late 1990s. We were working with an institution that recognized that it needed a CMS and were ready to take the plunge. There were two choices: Vignette (think conservatively \$1.5 million for licensing and basic implementation for a solution that was far out of scale for the institution) and another product that, upon evaluation, had serious flaws. We could have developed a CMS for this institution and gotten in on the ground floor in higher education: in fact, we even had code that we'd written for a CMS that ended up as a commercial product. And we saw a real opportunity: At the time, no college had launched a website using a CMS, although we recognized that every

college and university would need one. In the end, though, we decided not to pursue development of the mStoner CMS. Many of the reasons that emerged as we made that decision guide our thinking to this day.

There are many software solutions out there, with different feature sets, price points, and levels of complexity. We believe that serving our clients means helping them to navigate the sometimes overwhelming landscape that's already out there, and working with them to find tools that will work for individual institutions.
