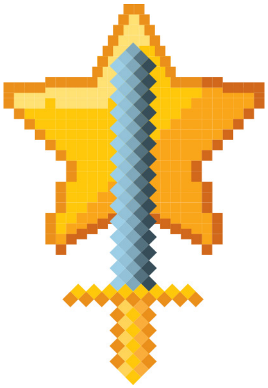
MOTUM B2B

Quest: Choose the ideal web platform for your business

When you're rebuilding or upgrading your website, battling through the unending waves of options, features, and fine print can be a harrowing adventure. To make your journey a little less perilous, we've broken down the pros and cons of website platforms to help you on the path to a glorious, sales amplifying B2B website.

It's dangerous to go alone! Take this.



Knowing the answers to the following questions will greatly assist you on your journey:

- How do you use your existing website? Are you happy with it? Does it do what you need?
- Why do you want to upgrade?
- What is it not doing?
- What other pieces of tech do you use? CRM? Salesforce? Pardot?
- How does it fit into your business goals? Is it a tool for your sales people? A tool for your customers? Both?
- How often do you maintain it? Why or why not?
- How often do you want to update it or add content to it?
- Who, if anyone, manages your existing website? Are you planning on bringing in someone to manage it? What are there capabilities? Do they know HTML?
- What stats do you currently track? Metrics?

LEVEL 1: WYSIWYG site builders

EXAMPLES: SquareSpace, WYSIWYG components of marketing automation platforms, large corporate CMSs with fixed templates.

What is it?

Companies selling unbreakable websites-in-a-box in the form of a service model.

Right fit for:

Small projects that aren't going to evolve, such as an online brochures, landing pages or microsites.

Not suitable for:

Anything involving JavaScript or dynamic or relational content, like a product engine calculator, or resource library.

Pros:

- Comes to life reasonably fast
- Typically-optimized for SEO
- Lowest cost to skin with basic branding elements
- Bug fixes and code glitches are nonexistent
- Support is usually someone else's problem
- Non-developers can maintain words and pictures

Cons:

- Most JavaScript or advanced analytics is a no-go
- In practice, "What you See is What You Get" often isn't what you get. There is a lot of erratic behavior with a graphical editor writing code behind the scenes that you never see.
- Templates provided probably won't match your branding
- Restrictive, expensive or outright impossible to modify
- No flexibility on design or layout

LEVEL 2: Template-based frontend + “all-in-one” WYSIWYG CMS

EXAMPLES: Drupal, WordPress, Joomla

What is it?

A free-to-download universal standard for building websites that typically comes with a mature, comparatively easy to use CMS as part of the package.

Right fit for:

Companies who have simple, one-language/region web requirements and who employ dedicated marketing staff on-hand to actively maintain and enhance their site outside of agency help.

Not suitable for:

Functionally complicated sites, sites spanning multiple audiences or sites involving custom components or processes is tied specifically to business objectives.

Pros:

- Portable and well-known. Easy to find developers who can work with this technology, so you are not stuck with a vendor
- Fairly easy to update content and maintain pages if set up correctly
- Strong ecosystem of plug-ins and add-ons potentially make life easier
- Initial shell comes to life quickly
- Lots of available templates

Cons:

- Notoriously hackable. Highest security risk of all technologies
- Extremely difficult in practice to maintain dev/production environments well without solid hosting middleware and periodic code freezes
- Costly to customize or maintain long-term. Law of diminishing returns in terms of Dev ROI after the initial shell is built
- Doing anything requires writing plug-ins
- Template structure limits outward distinctiveness

LEVEL 3:
Handmade frontend + “off-the-shelf” CMS

EXAMPLE: Craft CMS

What is it?

Attaching a tried-and-tested admin framework to a 100% custom built frontend, which the admin CMS can “talk” to via an API-like process.

Right fit for:

Mid-functionality websites, web apps, medium complexity projects, or anything built in a frontend JavaScript framework like React or Angular

Not suitable for:

High-functionality sites, or sites with a strong MVC or business logic requirement.

Pros:

- Faster than a custom 100% build
- Total flexibility as far as front-end website presentation and display features go
- Cost-effective to build out new management components
- Total customization of output. Want to dump your entire website in JSON format? Go for it.
- Once set and working, is extremely intuitive for clients and non-developers to use
- Once the core structure and relationships of your site are set the rest is quick to develop
- Core functionality such as language translations
- “Hard to break”
- You can still use a lot of plugins developed by the community
- Search engine is standard issue, no third party needed

Cons:

- Admin functionality still needs to be thought through and documented with a discovery and planning process
- Even with GUI tools, build-out of the admin back-end still takes time and effort
- Front end presentation of your site must still be explicitly built out – no templates
- Non-devs can’t WYSIWYG web pages as completely as a Wordpress-type site without some extensive setup and customization
- More expensive than just rolling out a bunch of templates for your front end
- Can solve most but not all functional problems
- Still can’t handle complex business logic and functional requirements as well as custom
- If you choose to use other peoples’ plugins it can be a security risk

LEVEL 4: Custom website built using open-source framework

EXAMPLES: Zend Framework, ASP.net, Pure Play PHP

What is it?

The most custom-crafted solution, but also the longest to roll out from vision to production

Right fit for:

Enterprise level websites, sophisticated business tools, apps, sites with complex data, 3rd party connections or integration with other tech

Not suitable for:

Microsites, landing pages, sites without an admin back-end

Pros:

- Most secure technology
- Total flexibility across all components, front and back
- Integrates most completely with your process and business requirements
- No trying to shoehorn requirements into off-the-shelf templates
- Fastest to customize or maintain once established
- Easiest to customize for speed
- Can still leverage framework libraries for things like database connections, API connections, authentication, etc.

Cons:

- Longest development time to get initial prototype off the ground
- Requires extensive discovery exercise and thorough planning process to execute
- Build-out of the admin back-end takes more time and effort compared to other solutions
- Front end of your site must still be explicitly built out – no templates, no drag-and-drop
- No WYSIWYG unless it is explicitly built
- Requires developers at every step of the process, including post-launch support
- Search solution must be explicitly planned out – no “out of the box” search engine
- Can require special hosting requirements