

# HANDBOOK OF RESEARCH IN MASS CUSTOMIZATION AND PERSONALIZATION

edited by

Frank T Piller • Mitchell M Tseng

Vol. 2

Applications  
and Cases



 World Scientific

## 1.7 A Mass of Customizers: The WordPress Software Ecosystem

Andrew Watson

*Northeastern University, United States*

While mass customization is usually associated with tangible goods, it is also relevant to less tangible goods, such as software and services. This chapter focuses on WordPress, which is both software and service. To be specific, the chapter presents WordPress as: blogging software; widely distributed and deeply customizable; a family of products built on a common platform; the focus of a vibrant community; the keystone of a thriving ecosystem; a for-profit business for Automattic, the company founded by its lead developer. It concludes with implications for WordPress itself, and for mass-customized software more generally.

### Introduction

Software is vital to mass customization. It is widely used in the configuration of mass customized goods, and in the delivery of mass customized services. Conversely, software itself may be mass-customized. In fact, Meyer and Webb (2005) describe the mass customization of software as "extreme" and "pervasive." This article describes the blogging software WordPress in terms of mass customization. The case of WordPress is particularly instructive, due to the number of ways in which WordPress can be and has been customized, and to the importance of customization in the WordPress architecture and in the WordPress ecosystem. As we will see, the diverse members of this ecosystem customize the software in many different ways, and the WordPress platform is architected in order to enable the ecosystem to thrive as it customizes. Mass customization involves "low-cost, volume production of great variety, and even... individually customized goods and services" (Pine: 7). WordPress is software that is coded rather than a tangible object that is produced. Much of the code has been and is being written by people currently employed by a privately-held firm: Automattic. The base software—the mass in mass customization—is architected and developed in such a way as to encourage its customization by others. A fuller discussion of Automattic's strategy will be possible after we have examined the mass, the customization, and some of the others who do much of the customizing.

It is useful to regard Automattic and these other organizations and individuals as an ecosystem. Iansiti and Levien (2004) conceptualize an ecosystem as comprising those organizations that "affect, and are affected by, the creation and delivery of a company's own offerings." The organizations fall in to a number of domains, which may be shared with other ecosystems. One of the domains in Automattic's ecosystem consists of firms providing web hosting. For example, the main WordPress web site provides a link to BlueHost. BlueHost offers, among other things, automatic installation of WordPress software. It also offers tools for eCommerce sites. Hence BlueHost, and the firms with which it shares a domain, are part of eCommerce-related ecosystems, as well as of the WordPress ecosystem. We will consider the WordPress ecosystem toward the end of this chapter. First, we must establish some relevant key terms. Then we will consider the mass aspects of WordPress. After that come the customization aspects, from which discussion of the ecosystem flows.

### **WordPress as Open Source Blogging Software**

WordPress is a blogging software. It is free/open source software, covered by the GNU General Public License (GPL). Developers of such software often refer to themselves as hackers. WordPress is offered, by Automattic and others, as software as a service (SaaS). The purpose of this section is to define the italicized terms. Hence readers familiar with the terms may wish to skip to the next section ("Code Bases").

First, the purpose of WordPress is to enable people to write weblogs, or blogs. Robert Scoble and Shel Israel define blogs as follows:

A blog is really quite simple. It's nothing more than a personal web site with content displayed in reverse-chronological order. New posts are placed at the top of the page... making it easy to see what has changed. In most cases, site visitors can identify the author and leave comments for others to see. Blogs are loosely joined to each other through hyperlinks (Scoble and Israel 2006, p.26).

Although they stress the "personal" aspect of blogging, these authors go on to describe the power of blogging for firms. Similarly, a Business Week cover story insisted that "Blogs will change your business" (Baker and Green 2005).

WordPress is free/open source software. This means that anyone is free to take the WordPress code, run it, examine it, modify it, and to distribute the derivative code resulting from the modification. To be more specific, WordPress is licensed under

the GPL, which means that if the derivative code is distributed, it must itself be distributed under the GPL.

The remainder of this chapter describes WordPress as open source software rather than as free software. It would be cumbersome to keep combining the two terms or to retell the history of the arguments between their respective advocates; Moody (2001) and others have already told the story well. It is, however, worth pointing out that advocates of both terms favor the word hacker to describe someone who contributes to software such as WordPress. The current chapter uses hacker in this sense, rather than to describe one who makes unauthorized intrusions into systems. The final point of this section is that the WordPress software is available as a service. You can easily create and post to a blog at WordPress.com without having to install the software yourself. WordPress.com is run by Automattic. It is part of a wider trend toward software as a service (SaaS), the most prominent example of which is Salesforce.com.

### **Code Bases**

The "mass" aspect of WordPress is best described in terms of three code bases. We will refer to the original WordPress code base as "WordPress Classic." It was based on, but split off from, earlier blogging software called b2/cafelog. The possibility of such "forks" of software code is a feature, not a bug, of the GPL. WordPress Classic inherited a simple architecture, wherein each instance of the software supports one and only one blog.

Hence a multiple-blog system required that WordPress be installed multiple times. A multiple-blog WordPress Classic installation still requires this. This helps to keep WordPress Classic a relatively small, simple software product. However, the one-to-one relationship between blog and blogging software installation hinders the scalability of WordPress Classic. It is expensive in terms of system resources: each instance of the software consumes disk space and, when running, memory. It is also expensive in terms of system management resources: each instance must be not only installed prior to blog creation, but also upgraded in order to take advantage of features and fixes introduced in new versions of WordPress.

Hence several projects sprang up to customize WordPress to turn it in to a more scalable multi-blog system. Such customization is of course permitted and indeed encouraged by the GPL. One of those who customized the code was Donncha O Caoimh. His fork of WordPress became WordPress Multi-User (WPMU), and he currently works for Automattic as lead developer of WPMU.

WPMU is the second of the three code bases. A WPMU site administrator can in effect offer WordPress as a service. For example, Edublogs.org offers WordPress as a service to academics, providing the hardware, upgrading the software when necessary, and so on.

The third code base, WordPress.com, is more of a particular WPMU site than a very different code base. It is of particular relevance for at least two reasons. First, it is by far the largest WPMU site, with over two million blogs as of the start of 2008. Second, it is one of the means by which Automattic earns a return on its investment in the WordPress software (Watson 2008). For example, many of the blogs hosted at WordPress.com carry advertisements, with the revenue going to Automattic, rather than to the bloggers.

Table 1 summarizes the differences between the three code bases. Figure 1 shows a WordPress.com blog that has been lightly edited following creation. A WordPress Classic or WPMU blog would look very similar since, as Table 1 shows, much of the difference between the three code bases comes in terms of system management and the back end. Much of the remaining difference comes in terms of customization, and it is to customization that we now turn.

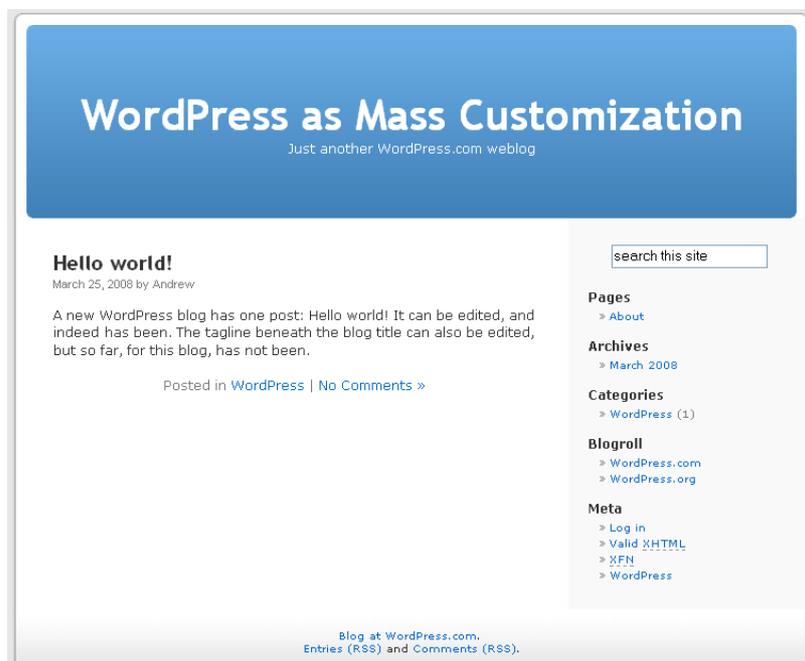


Figure 1: Sample blog with default theme ("Kubrik").

Table 1: WordPress code bases.

Code Base	Source	Software as a service?	Blogs per code base	Potential for customization
WordPress Classic	Wordpress.org	No	One	Extensive
WordPress Multi-User	Mu.wordpress.org	Software used to provide service	Many	Depends on system administrator
WordPress.com	Not available	Yes	Hundreds of thousands	Within limits

### Customization

Table 2 shows the ways in which a blog can be customized by the blogger and site administrator. For each of these ways, the table identifies the customization and the means by which it is implemented and the way in which the customization currently applies to each of the three code bases. The current (March 2008) versions of WordPress Classic and WPMU are 2.3 and 1.3 respectively. WordPress.com does not have a public version number: software upgrades are not performed by users, since timely upgrading is part of the service aspect of SaaS.

Table 2: WordPress customization.

Customization	Mechanism	WordPress Classic v2.2	WPMU v1.2	WordPress.com
Free/open source	GPL v2	Yes	Yes, for site administrator	No
Adding function	Plugins	Yes	Admin provides menu, blogger selects from it	No
Changing page structure	Themes	Yes	Admin provides menu, blogger selects from it	Admin provides menu, blogger selects from it
Changing page style	CSS	Yes	Admin may enable	Yes, as premium service
Changing sidebar content	Widgets	Yes (requires plugin)	Admin provides menu, blogger selects from it	Admin provides menu, blogger selects from it
Localization	Translation files	Yes	Yes, although not well-documented	Yes, language selected by blogger

WordPress Classic is the most customizable of the three code bases. This is to some extent because the source code is open, and so any hacker is at liberty to modify WordPress. This is not to say that one must be an accomplished hacker in order to customize one's WordPress Classic blog. For example, a blogger wishing to extend the function of WordPress Classic may well be able to find and "plug in" code already written and made available by another person. Hence the blogger provides administration, as well as content, for her WordPress Classic blog. When it comes to WPMU, customization is possible at two levels. At the "higher" level, the administrator of the WPMU system in effect sets up a series of menus for the whole site. Each blogger then customizes her blog by selecting from the menu. WordPress.com is to a large extent a WPMU site, with site administration provided by Automattic.

The first row of Table 2 refers to the fact that WordPress is open source software, under the GPL. Given the above-discussed implications of the GPL, it may seem curious that anything other than "Yes" appears in any cell of this row. However, WordPress.com offers the software as a service. It does not, strictly speaking, distribute the software. Hence Automattic is at liberty to modify the WPMU code for use at WordPress.com without having to release the modifications under the GPL. This is a point about the GPL, rather than about WordPress in particular. It is sometimes described as a loophole in the GPL, and one that will become increasingly important if the trend toward SaaS continues.

The next means of customization is well introduced in the WordPress documentation. Plugins are tools to extend the functionality of WordPress. The core of WordPress is designed to be lean, to maximize flexibility and minimize code bloat. Plugins offer custom functions and features so that each user can tailor their site to their specific needs. (WordPress 2008)

As an example, one of the most widely-used plugins is "Subscribe to Comments." When visitors to a blog leave comments, they may wish to be informed of further comments, rather than having to revisit the posts on which they commented in order to check for subsequent comments. If the blogger has used "Subscribe to Comments," visitors leaving comments will be invited to check a box requesting that subsequent comments be emailed to them.

That is just one example. There are hundreds of plugins, varying along dimensions such as complexity, popularity, and visibility to visitors. Plugins have many advantages in addition to those explicitly identified in the above quote from the documentation. First, they allow function to be added in a modular fashion. Second, plugins allow anyone to extend the function of their WordPress blog, and to make that extension available to the WordPress community by contributing it to

the WordPress plugins directory. This may mean that the same function is offered to the community by means of multiple different plugins. This allows a blogger to select the implementation of the function best suited to her needs.

On the other hand, plugins have their drawbacks as well as their advantages. While the existence of multiple plugins with similar function allows the blogger choice and flexibility, it may also create confusion for the blogger—and for a reader who finds the same function provided in different ways at different WordPress blogs. Plugins can also make upgrading to a new version of WordPress more complex, in that new versions may "break" plugins, and plugin writers do not always provide timely upgrades.

Plugins fit differently into each of the three code bases. A blogger running WordPress Classic finds the plugin she needs, uploads it to the appropriate directory of her blog site, and activates it. A blogger using WPMU may activate any plugin uploaded by the site administrator. She may not upload plugins herself, although she may, depending on the WPMU site and its administrator, be able to lobby for a particular plugin to be available. A blogger using WordPress.com never sees the plugins administration menu. We can say, at the risk of oversimplifying, that Automatic, as WordPress.com site administrator, selects, uploads, and activates plugins on a site-wide basis.

The third row of Table 2 describes themes, which determine the layout of the web pages comprising a blog. The contrast between Figure 1 and Figure 2 illustrates the different that themes make. Figure 1 shows the demo blog with the default theme: Kubrik. Figure 2 shows the same blog with a different theme: Benevolence. Although the content has not changed, almost everything about the blog looks different: the header; the sidebar, which has moved from right to left; etc..

The demo blog is at WordPress.com, where the number of themes available is currently in the dozens. The length of the theme menu for WPMU differs from site to site, under the control of the site administrator. A blogger using WordPress Classic can choose from over a thousand themes that have been developed and made available by members of the WordPress community. As with plugins, the blogger uploads the code she needs to her blog site.

What if the blogger likes some aspects of a theme, but not others? One answer is that, if she is using WordPress Classic and has some proficiency in PHP, she can modify the theme, or even write her own theme. There are other answers that do not require programming, and that work at WordPress.com. Some themes allow replacement of the header image. For example, our blogger could retain the Benevolence theme, but replace the grassy header image with an image of her own choosing, provided it had the same dimensions.



Figure 2: Sample blog with different theme ("Benevolence").

The fourth and fifth rows of the table describe two other ways in which the blogger can customize her preferred theme. One is by using CSS: Cascading Style Sheets. The emphasis belongs on the word style because CSS is a means of describing the style of a web page separately from its structure. Both the structure and the style of the demo blog in Figure 2 are determined by the Benevolence theme. Structure refers to things such as the placement beneath the header of the post of the date and time on which the post was made. Note from Figure 1 that the Kubrik theme defines a different structure; for example, it does not include the time of the post.

Style refers to things such as the font. Hence our blogger could use CSS in order to change the size of the type for the post headings. She could also use it to change the color of link text from red to a more restful blue. She could not use it to change the structure of the web pages comprising the blog; for example, if she wanted the date and time of posts to appear below the post content, rather than above it, she would have to switch to a theme other than Benevolence.

CSS is important to WordPress in several ways. First, it is an example of modularity and of sound web design. If our blogger wants to change the size of the type, or any other style-related aspect of the theme, she has to change only the CSS file for the theme, rather than having to look through all the PHP files for points at which style is specified. Moreover, when she changes the style by editing the CSS, she can be confident that she is not at the same time making unintended

changes to the structure of the page. Second, CSS is important to WPMU and to WordPress.com in terms of security. Allowing bloggers to edit PHP and HTML presents a security risk to other bloggers and to the site in general. Allowing bloggers to edit CSS presents no such risk.

The third way in which CSS is important to WordPress follows from each of the first two. Due to the modularity and security implications of allowing CSS editing, it is possible to allow bloggers to edit CSS, and to charge them a premium for this. Indeed, Custom CSS is one of the for-pay addons available at WordPress.com. The availability of Custom CSS at a WPMU site is a decision for the site administrator.

The last mechanism for customizing WordPress themes is sidebar widgets. Again, this mechanism is best illustrated by a "before and after" contrast. This time the relevant contrast is the one between Figures 2 and 3. Figure 2 shows the blog with the standard Benevolence sidebar. In Figure 3, the standard sidebar has been replaced by three sidebar widgets: a calendar, a set of links, and a more freeform "text widget." Many available widgets are not shown: these include a search box, a selection of photos from the blogger's Flickr.com account, and many more.

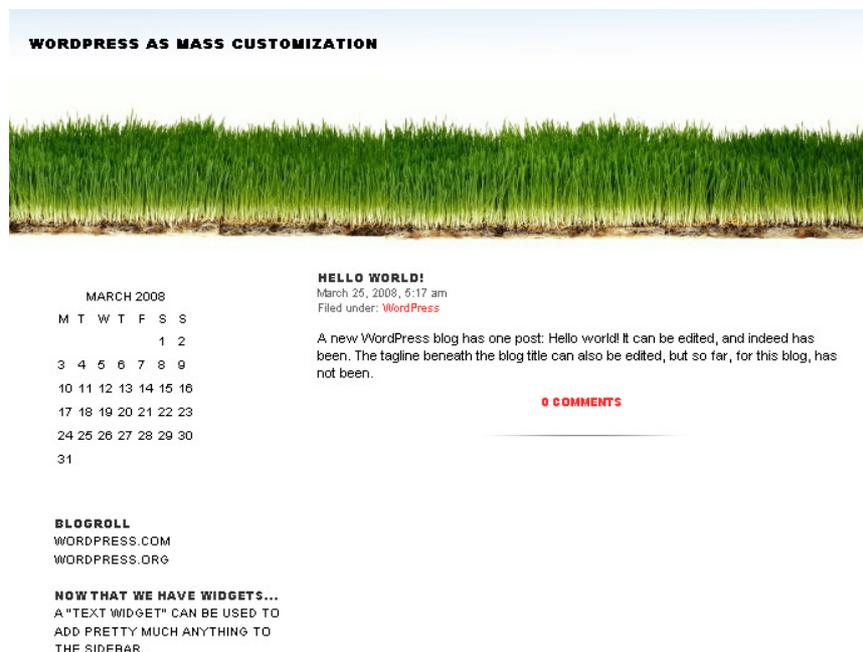


Figure 3: Sample blog with sidebar widgets.

### **The Ecosystem**

Automattic, as the firm that defines WordPress, occupies the central position in the WordPress ecosystem, and is its own single-firm domain. A second domain in the WordPress ecosystem was introduced early in the current chapter; that domain comprises web hosts. Then there are the bloggers themselves, who comprise multiple domains. Before identifying and describing these domains, it bears remarking that almost every member of the ecosystem is a blogger. Matt Mullenweg, for example, was blogging (atPhotoMatt.net) before WordPress, let alone Automattic, existed. Nevertheless, Mullenweg and his colleagues are more accurately and usefully placed in the Automattic domain than in a more general domain of WordPress bloggers.

A similar point applies to another category of WordPress blogger. This category comprises what von Hippel (2005) describes as "lead users." These include the bloggers who customize WordPress, and make the customization available to others. For example, only a small percentage of the WordPress themes come from Automattic. Many are developed by individuals, sometimes in response to a theme design competition. WordPress.com is designed for customization. In fact, Table 2 is not a bad illustration of the very architecture of WordPress. The effect of this is to encourage and expand the lead user domain.

We can now turn from the architecture of WordPress to the strategy of Automattic. It is what Iansiti and Levien (2004) describe as a keystone strategy. This contrasts with the strategy of dominating the ecosystem at the expense of other members. The keystone strategy aims at a central, although not dominant, position within a thriving ecosystem. Iansiti and Levien argue that it is preferable to the strategy of ecosystem dominance in complex and turbulent environments; the web is certainly such an environment.

### **Conclusions**

Although Automattic is privately held, and thus does not release financials, recent events suggest that its keystone strategy is working well. In January 2008, CEO Toni Schneider blogged that Automattic closed a \$29.5M round of financing. Over the last two years, Automattic's business has been expanding at a rapid rate. Our most prominent service, WordPress.com has grown to over 2 million bloggers. Their blogs are read by an astounding 114 million unique visitors from all over the world... Revenues have been growing as well, we've been profitable as a business, and we've accomplished all of this with a fantastic team of fewer than 20 people.

Late last year we sat down to figure out how we'd like to expand our business in 2008 and beyond. Since things are working well, we didn't want to make any major changes. However, we did set a couple of new goals. One was to put enough money in the bank to have financial security for years to come. Another was to invest more aggressively into our "other" products and services.

Of particular relevant to the keystone strategy is the intention to grow, not only the WordPress business, but other businesses. These other businesses are related to WordPress. Akismet provides a good example. Most bloggers allow readers of their blogs to add comments. Unfortunately, "spammers" have discovered this, and so many comments are in fact generated by software rather than written by humans, and are attempts to drive traffic to commercial sites. Akismet is a software service to detect and quarantine spam comments. It is implemented as a plugin for WordPress, but can be used to fight spam for software other than WordPress. Akismet is free of charge for individual use, but there is a charge for its commercial use. Two comments linking Akismet back to WordPress will complete this account of Automattic's keystone strategy. First, the decision to branch out from WordPress itself reflects an intention to grow Automattic without trying to dominate the WordPress ecosystem. Second, Akismet is not the only spam-fighting plugin. It faces competition from other members of the WordPress ecosystem. Such competition is one of the things that makes the ecosystem thrive.

## References

- Baker, S. and Green, H. (2005). Blogs will change your business. *Business Week*, May 2. [www.businessweek.com/magazine/content/05\\_18/b3931001\\_mz001.htm](http://www.businessweek.com/magazine/content/05_18/b3931001_mz001.htm).
- Ianstiti, M. and Levien, R. (2004). Strategy as Ecology. *Harvard Business Review*. 82(3).
- Langer, M. and Jordan, M. (2006). *Visual Quickstart Guide: WordPress 2*, Berkeley, CA: Peachpit.
- Meyer, M. and Webb, P. (2005). Modular, Layered Architecture: The Necessary Foundation for Effective Mass Customization in Software. *International Journal of Mass Customization*. 1(1): 14–36.
- Moody, G. (2001). *Rebel code: Linux and the open source revolution*, New York: Basic Books.
- Mullenweg, M. (2003). The blogging software dilemma. *Photo Matt*. Jan 24. [photomatt.net/2003/01/24/the-blogging-software-dilemma/](http://photomatt.net/2003/01/24/the-blogging-software-dilemma/).
- Pine, B. J. (1992). *Mass customization*. Cambridge, MA: Harvard Business School Press.
- Schneider, T. (2008). Automattic fundraising. Jan 22. [toni.schneidersf.com/2008/01/22/automattic-fundraising/](http://toni.schneidersf.com/2008/01/22/automattic-fundraising/).
- Scoble, R. and Israel, S. (2006). *Naked conversations: How blogs are changing the way businesses talk with customers*. Wiley.
- Von Hippel (2005). *Democratizing Innovation*, Cambridge, MA: MIT Press.

Watson, A. (2008). Making Money From WordPress.com. March 12. [changingway.org/2008/03/12/making-money-from-wordpresscom/](http://changingway.org/2008/03/12/making-money-from-wordpresscom/)

WordPress (2008). Plugins. [codex.wordpress.org/Plugins](http://codex.wordpress.org/Plugins)

### **Author Biography**

**Dr Andrew Watson** is currently an independent writer and consultant, focusing on social media and strategy. His particular focus within social media has been on the WordPress blogging/publishing platform. More recently he has turned his attention to web widgets: modules that can be shared and reused across multiple web pages. Dr Watson was previously on the faculty of Northeastern University's College of Business. He was based there while preparing the earlier versions of his account of the WordPress ecosystem. He remains Boston-based.

Contact: [changingway.org/](http://changingway.org/) & [widgetstrategist.com/](http://widgetstrategist.com/) | [andrew@changingway.org](mailto:andrew@changingway.org)