Wordpress cringe

The reasons for a so-called ‘WordPress cringe’ that exists for both its users and the web developers working with it are understandable: the plug-ins break with each update, and the back-end interface is, well, pretty *ugly*. Web developers cringing at WordPress software made me wonder why a different software with the same listing system couldn’t be a solution (such as, for example, Kirby).

WordPress is a web builder software that is, at the time of writing this, nineteen years old. When it was released for free in 2003, it soon became the market leader of the blogging tools industry, allowing both users and developers to extend its default functionalities by writing their own plug-ins and sharing these with the rest of the online community.

In 2005, the world discovers WordPress version 1.5, an incredibly flexible theme system that adapts to the user rather than expecting the users to adapt to *it*. This already hints at the fact that there were some software limitations, but what this specifically meant was that headers, footers and sidebars could be placed on specific pages and not necessarily on all of them, such as before. I guess some users didn’t like feeling constrained by component repetition across pages.

By 2010, WordPress version 3.0 enabled the user to customize the styling of the default *Twenty Ten* theme. The theme became an annual mirror to the user’s taste in grids, and continues updating itself every year.

The year is Twenty Twenty-Two: The Most Flexible Default WordPress Theme of all times. Built to reflect the modern user needs, it is inspired by birds, offers six color palettes, and encourages as little use of CSS as possible—because *Global styles* are just a few clicks away. Apart from style, WordPress stays focused on improving a full website editing experience, for its future direction solely depends on the needs of millions of web publishers around the world, a.k.a. its users.

Years of WordPress’ changing according to the user’s and web developer’s needs made WordPress survive as the leading publishing software on the internet, with more than 43 percent of all websites built under its wings. So, why oh why, do users *cringe* so hard at the software that updates itself recursively along with their digital wishes? To answer this question, I’d like to explore the technical, stylistic, and personal perspectives of Wordpress’ cringe.

1. What is Technical Cringe?

Openness is an unruly concept. While free tends toward ambiguity (free as in speech, or free as in beer?), open tends toward obfuscation. Everyone claims to be open; everyone has something to share, everyone agrees that being open is the obvious thing to do—after all, openness is the other half of “open source”—but for all of its obviousness, being “open” is perhaps the most complex component of Free Software. It is never quite clear whether being open is a means or an end. Worse, the opposite of open in this case (specifically, “open systems”) is not closed, but “proprietary”—signaling the complicated imbrication of the technical, the legal, and the commercial.

— Christopher M. Kelty, *Two Bits:* *The Cultural Significance of Free Software*

When a software cracks open and enables web developers to access and expand its set of functionalities through modifying the default code and contributing to it with plugins, it makes itself technically vulnerable. It leads to standardization, civilization, exponential growth:

Standardization was at the heart of the contest, but by whom and by what means was never resolved… We don’t live in a world of “The Computer” but in a world of computers: myriad, incompatible, specific machines.

— Christopher M. Kelty, *Two Bits: The Cultural Significance of Free Software*

Under the current conditions of the tech market, all of this/the software becomes a playground for competition; a potential space for exploitation. Not just a few, but *many* of the plugins that perform the same task, all coded by amazing web developers who don’t really communicate with each other, end up competing for what the study material of these developers taught them: user attention. Each plugin, instead of finding itself one too many to choose from, finds itself wanting to be one in a million.

Openness is an unruly concept. The price of software’s freedom is the user’s unfriendliness: the technical slowdown that comes with every new software update.

Here, we end up confronted again with the simple fact that there is no *universal* user, although most of our software expects there to be. Some users are more attuned to the tools and technologies they’re using, while others prefer to stay busy with non-technical things. Some users are curious enough to learn more about the back-end and maintenance of their digital presentation, while others prefer their websites to be independent.

The ecosystem of WordPress puts most web developers in a position of not questioning or even understanding the extent to which their clients and/or users want to express themselves on their respective digital interfaces. How do they want to use the back-end? How often? What is the function of the website?

Only when true user needs are understood can technology become a cultural meeting ground upon which we can look away from the image of a universal user and learn about their socio-political situation, class, and culture through their relationship with technology. This takes time, for this often constitutes the unpaid analysis of the process of web development. And because this analysis is not per se encouraged by the web developer during the whole development process, he opts for a standard setup of the back-end, one that leaves the users in fear of clicking that crucial update button.

2. What is Stylistic Cringe?

Because the WordPress’s community shares all issue-related matters, the users can easily find out what exactly is wrong with the issues they’re having. Sometimes the most curious of users even go further in researching possible solutions than the web developers do. Because of this transparency, developers refrain from identifying with the software and would rather opt to settle for a less user accessible techne-pack, or their own proprietary version of WordPress (such as BlockSmith).

This rarely discussed reason behind the web developer’s pride has a cleaned-up version: instead of confessing to their opinions, a web developer finds himself persuading their clients and/or users that the interface design of WordPress is *just horrible*. Because of the personal sensitivity that style installs in every user, the user is easily persuaded and charges extra for the time it took their (presumably male) web developer to build their own CMS.

Irony alert: everyone cringes at the underlying fact that WordPress is accessible to all in the blink of an eye and can make anyone feel like they are able to code and build websites. I, personally, don’t see why anyone *shouldn’t* just build their own websites.

3. What is Personal Cringe?

Besides being the very best version of a Post-WordPress web developer I can be today, just because it felt cute, my reasons for *cringing* will most likely bring me to delete this identification later. Throughout my cringe studies, I was hoping that someone would complain about the listing and/or archival logic of the WordPress and WordPress inspired softwares, and how its modularity promises a false freedom that one eventually ends up feeling a victim of, or at the very least dubious of.

My cringe derives from confusion: I’m still puzzled by the technical differences between proprietary CMSs and WordPress; I still wonder why (and this is probably a question of governance) the users cringe when openness is technically allowed according to their wishes. Let’s talk about this.