Custom coding or plugin overrides?

Having your website suited to be the most comfortable in usage is one of the goals of custom coding. This way you can ensure that the user experience is 100% as it is designed by your strategy. It also allows you to be **better** than your competition in delivering exactly what your client needs. Sometimes you are using a Joomla or WordPress plugin, which is **almost perfect** – core override would allow you to ensure that it is fully perfect. Should you do it?

Jude was a young programmer – she loved creating websites on WordPress. She was inexperienced, but promising, and had more and more clients. They wanted her not only to code their website but also to take care of it. That Tuesday she received a call: *Hey, would you like to take care of my website? It requires some updates and a few small adjustments.* Sure thing, she agreed and shortly after received login information. The website was not really taken care of in the past – many plugins and core were outdated, leaving it vulnerable to hackers. Jude agreed on the rate and started updating core and plugins.

At the end of the day, all was done – but she received a **phone call**. *What is going on with the website? The contact form is looking completely different! Also, business postings are not displaying half of the information!*

What the hell – Jude thought. Since she was inexperienced, she never encountered such bad practice, as hacking core files. Unfortunately, that's exactly a trap, which the previous developer left for her. The **website could not be updated** without manually searching for files, checking differences and applying them to the newest plugin versions! Furthermore, neither plugins were written properly, nor hacks were done according to the framework model. This situation led to **a lot** of confusion.

Thankfully, she had her homework done before. Tired and angry after her work hours, she was able to restore the backup and go home with a promise to herself – if she ever finds out who did it, he'll regret it.

As you can see in this case, Jude was put in a quite hard situation. On the one hand, she had a website, that needed updates, on the other hand, a lot was hacked in core by the previous developer. The previous developer was a mindless guy, who wanted his job done fast and cheap. This leads to the conclusion that doing core hacks will get him right to his aim – because a proper approach would take twice as much time and money. He achieved his goal **but left the website**

unmaintainable and vulnerable.

Can you make some code adjustments, when you're not plugin owner?

Are there other ways besides hacking core to make simple code adjustments, if you are not a plugin owner? Of course there are:

- You can collaborate with the original developer. Send him a message, with your changes to core prepared. If a developer is caring, he will review your code and append it to the core plugin. With the next plugin version, you will have the functionality you need! Minus is, it will take some time before your changes appear in the public repository
- Extend with a separate plugin – you can leave the original plugin untouched and write your own that runs alongside the original. This approach is not always possible though if changes are needed inside the core
- Use custom hooks – this method is only for WordPress. Hooks allow you to customize the behavior of the plugin. But – be aware – this method can only be used if the original developer prepared hooks! And unfortunately, it's not that common for WordPress plugins to have proper hooks
- Overriding callbacks – a method for WordPress. While at first sight, it might seem like a good idea, it also has a drawback. Often functions aren't modular enough to call only pieces of the code, you care about
- Adding custom hooks – a method for WordPress. It is a solution somehow between overriding callbacks and collaborating with the developer. In this case you override core files with your hooks, and then call them from separate files. On update, you still need to do custom work, although it's only adding your own hooks back – unless of course, the original developer adds your amendment to the master page
- Fork! – for any CMS. If a plugin license allows it, you can copy code and make your own version of the plugin. But you lose the ability to update the plugin if any updates are published by the original author!
- MVC overrides – a method for Joomla. It's a very powerful method because it allows you to override either views or whole files, without touching the core at all. It's also very convenient in use and does not require any type of hooks or callbacks.

Summary

Overall, the approach you should choose depends on many factors. Is the hack only temporary? Are you switching over to another CMS in a short time (you can read about different CMS advantages and disadvantages here)? Do you want to stay compatible with the plugin or core updates in the long term? Should plugins be maintained for security or maybe used on other websites? Are you willing to trade new functions and security for a fast hack? Answering those questions will allow you to make the right choice.