

The All-In-One WooCommerce Guide for Store Managers



The All-In-One WooCommerce Guide For Store Managers

Intro

WooCommerce powers up hundreds of thousands of online retailers around the globe because it's an extremely powerful and modular software.

This guide will provide store managers with actionable tips and knowledge on how to start and set your WooCommerce store at its best, all backed up by experienced WooCommerce developers we've directly engaged with through the years.

Thank you for downloading it.

Enjoy your reading!

Woo Woo!



Matteo Duò

Content Strategist at Codeable
matteo@codeable.io

Chapter 1: Getting Started With Your WooCommerce Store

How And Where To Find A Proper Theme

Picking the right theme for your WooCommerce store is something that is often overlooked by many first-time business owners and entrepreneurs. The reason being they think a theme is "just" a matter of aesthetics and of how good their developer/designer is.

Even if those are important elements in the equation, they're just part of the story: you need a theme for your eCommerce that's not just a good-looking theme, but one that's enabling your business reach your goals. One theme that has to be fully compatible with WooCommerce so it won't raise technical issues you could have easily prevented from occurring in the first place.

Choosing a theme is ultimately a business decision, and that means factoring in your long-term goals in your decisions.

So where can you find themes for your eCommerce store that are and will always be compatible with WooCommerce?

Go to WooCommerce.com

The number one authority on all things WooCommerce is, of course, WooCommerce.com, the official website from the wonderful people who built and maintain the plugin. Says WooCommerce specialist and Codeable expert [Shadi Manna](#):

What I would suggest at first is to go on WooCommerce.com because you'll be sure to find 100%, future-proof, compatible themes. Many might not know but they started as WooThemes back in the days. Especially, you might want to give a look to Storefront, their flagship theme. If you use this you can always rest assured it will be compatible with WooCommerce.

At first sight, [Storefront](#) might look as a "not-that-advanced" WooCommerce theme if you compare it with many others you can find on [Themeforest](#) (or other theme marketplaces) that come with lots of bells and whistles. Nevertheless, this theme perfectly merges into WooCommerce's vision for their products. As Shadi explains:

With Storefront it's a different approach to an Avada theme, for example, which is one of the most popular on Themeforest. With themes like Avada, they sort of add a lot within the theme files, and they give you a lot of functionality and files and make it a heavy theme. All is different with Storefront because it gives you a core theme, and then you'd need to add plugins and add-ons to extend that functionality. Think of it this way: Avada is like a Swiss Army knife whereas Storefront is more like a lego set you can choose from.

[WooCommerce.com](#) isn't the only choice, of course, to find WooCommerce-compatible themes. Another type of well-known resource for finding quality themes are marketplaces.

Theme marketplaces and third-party websites

There are many places other than [WooCommerce.com](#) where you can find different WooCommerce themes. The key is you need to be sure that a theme you're about to buy has to be compatible with your WooCommerce store.

On [Themeforest](#) and also on other marketplaces, theme authors usually make it clear whether their theme is compatible with WooCommerce. Specifically, they make it clear which version of WooCommerce it's currently compatible with. □□

And that's good news because it'll help you filter out incompatible themes faster. Still, compatibility with WooCommerce doesn't always mean compatibility with all plugins or add-ons you're currently using.

How to quickly check theme compatibility with your plugins and add-ons

Depending on the plugins and add-ons you're using, it could be a bit of a challenge knowing if a WooCommerce theme is fully compatible. You might, therefore, need to look at more things. As Shadi highlights:

I would suggest to look at how long that add-on has been working with WooCommerce, how many comments it's got, what version it's at. You might want to look at the comments on the add-on page itself, and feedback and reviews on compatibility with WooCommerce. All those things typically give you a good idea of how compatible an add-on is, and how responsive the author is.

What to do if your theme is incompatible with WooCommerce

But what if you already have a theme (or have picked on) that you know isn't compatible with WooCommerce? What if, for example, you have a standard website that you want to turn into an eCommerce?

Well, if that's your scenario, things get a bit trickier and you have 3 routes in front of you to take:

- Option a): reach out to the theme provider and gently request compatibility.
- Option b): [hire a WooCommerce expert](#) to make it compatible.
- Option c): toss away your incompatible theme and keep searching for one that might feature most of what you need AND that is also 100% compatible with both WooCommerce and your plugins.

If option a) goes through successfully, bingo! You have the WooCommerce-compliant theme you need, right from the theme provider. But that's not always the case because, as you can easily imagine, a provider will be more inclined to address requests that come from a somewhat big portion of their userbase. Not just from one user. Even though, you never know!

But yeah, having your business dependent on such thing isn't a smart business move.

So, let's keep evaluating options...

[Hiring a developer](#) to make a theme compatible with WooCommerce (option b) comes with additional responsibilities and future costs you might have no clue about. In short, every time a new version of WooCommerce comes out, it's on you (or the developer you hire) to continuously ensure that WooCommerce is compatible with that theme. And that doesn't stop just with the latest versions of WooCommerce, it's with all updates your install will get that could break your store.

This option puts the onus on you to perpetually maintain the plugin code to ensure compatibility with newer versions of WooCommerce. So it's up to you to decide if you like a theme enough to invest in its long-term maintenance to ensure WooCommerce compatibility. Are you willing to keep this burden on your shoulders looking ahead? When considering this option, Shadi suggests to question yourself:

Do I like this theme that much that I'm willing to invest in continuously maintaining it to ensure it's compatible with WooCommerce?

So you'll need to have either your in-house developers take care of that each time an update gets released or you'd need to hire someone to do it for you.

I know what you are thinking: mmmhh, there are more efficient and cost-effective options out there...

With the last option left (option c), you'll need to get rid of your theme and look for one that's as close as possible to the one you had but also compatible with WooCommerce. I know, this will require you a bit more time to invest in your research phase but you'll be saving yourself money and future headaches. It might also require a one-time investment in development work of switching over or changing themes, but if it sets you up better looking ahead, again, it could save you time and money in the long run.

If you ask me, this one is the most cost-effective solution of all the three, which is surely a great one and comes only second maybe to having a custom theme built for your specific needs (if your budget allows it).

Wrapping up

When it comes to choosing a theme for your new store, compatibility with WooCommerce (and additionally WooCommerce add-ons) should be your foremost concern. You might want to first have a look at the themes on WooCommerce.com, as they're made by the same guys who maintain the plugin. ☐☐ But you could also search for themes on marketplaces, just by keeping in mind to check all the info, comments, and changelogs to be 100% sure on what you're about to buy.

One thing is for sure, though: you'll be spoiled with choices as there's a great deal of beautiful premium WooCommerce themes all around the web. Just make sure you're choosing a theme that's well-maintained and is backed by (mostly) positive reviews. Or simply invest in [hiring a WooCommerce specialist](#) to have them help you select or craft a fully-compatible, future-proof custom WooCommerce theme.

Setting up gateways

If you are [running a WooCommerce store](#), having a payment gateway comes as a default necessity. Luckily, WooCommerce offers a platform where you can set up multiple payment gateways of your choice so that your customers will be able to find a payment method that suits them best.

But what are your options? How difficult is it to properly configure a new payment gateway? What do you need if you want to install a new one on your WooCommerce store?

How to enhance WooCommerce payment options

By default, WooCommerce offers you four payment options: check payments, bank transfer, Paypal standard and cash on delivery. These gateways are direct channels that will connect your online store with external merchant accounts. Of course, there are more you could add to enhance your payment options on your store.

One player that has gathered a lot of attention and interest among store owners and developers in recent years is [Stripe](#). This payment gateway accepts all major credit and debit cards, as well as Bitcoin and Apple Pay, and comes as a free official plugin for WooCommerce websites. Robin suggests giving Stripe a try because of its many benefits, even though it might still not be available your country yet or might not integrate with what you're selling:

I highly recommend giving Stripe a try before you try any other gateway. It's got a free official WooCommerce plugin. If there is going to be an issue with Stripe, it may be what country you're in. It doesn't cover all the different countries, and equally, it might be the type of service or product that you're selling. There were some things that Stripe won't work so well with, but you'll see that in their terms and conditions.

If Stripe isn't a viable option for you, you might want to check out Braintree, which was [taken over by PayPal about four years ago](#) and is yet another payment option that has a free WooCommerce plugin as well. This option comes second in efficiency after Stripe because of its some limitations within the admin panel, which is not as nice as Stripe, but also there are compelling reasons that might make this gateway not the right fit for your business. As Robin explains:

Braintree is a close second to Stripe for me. In some cases, it's superior to Stripe. It's slightly harder to work with, slightly harder to develop with and slightly harder to set up. The admin panel is not as nice as Stripe and the reports are frankly awful. But Braintree pays you within 48 hours, whereas Stripe holds the funds for seven days

One aspect on which Braintree tops Stripe, though, is their better anti-fraud tools and being verified by Visa and the MasterCard SecureCode system.

Payment gateways always need to be PCI compliant

As we've seen, the payment options offered by WooCommerce are great but might not be satisfactory to some store owners. That is why your eCommerce store can make use of more other external payment gateways as Stripe or Braintree.

The important thing here to know is that in order to integrate any new gateway, you need to ensure that 3 criteria are met:

1. You have an active merchant account that is properly associated with your WooCommerce.
2. [Your site should also be up-to-date.](#)
3. Your hosting provides you with a supported encryption type for your store.

Furthermore, in order for you to comfortably enjoy and leverage the advanced features from these payment gateway extensions, you'll need to be sure your checkout page is PCI compliant at least. That is, [have an SSL certificate installed](#). As Robin points out:

You need to have a security certificate in place. At least on your checkout, but I would just say it's better if you implement it on the whole store. If you don't have an SSL certificate, your checkout will show a "not secure" warning to your customers, which is something they don't like to see in a checkout page and ultimately it won't complete successfully. It's 100% necessary.

[There are several types of SSL certificates](#) you could choose from, with different features and costs. Yet in most cases, the one offered completely for free by [Let's Encrypt](#) will work great. If your store handles lots of transactions, though, Robin recommends the EV certificate also known as the "green bar certificate":

The only certificate I will consider paying for now is an EV certificate. If you're handling a lot of money transactions, I would recommend getting one so that you can count on a higher level of certification. But if you're at an early stage in your business, just use a free Let's Encrypt certificate. That's what we use on our own site: we used to pay for EV certificate, but then we switched to Let's Encrypt and we noticed zero change apart from we weren't paying for it. So I don't believe it makes a great deal of difference if you're an eCommerce store; it does if you're a bank.

Where to find official payment gateways for WooCommerce

If you're looking to add a new payment gateway, your first stop should be the [WooCommerce official extensions](#).

WooCommerce official extensions grouped here are usually updated constantly and better-maintained when compared to third-party extensions available through other websites. On top of that, official WooCommerce extensions come with a great benefit that's unlikely to occur through other channels:

Here's the general rule: WooCommerce official extensions page should be your starting point. Of course, there are some third-party extensions but, generally, the official ones are better and maintained. Should there be an issue with one of them, you can be sure it's going to get fixed.

Setting up the new payment gateway

Once you've picked and installed your new payment gateway, it's time to set it up correctly. Let's see how:

How to set up Stripe payment gateway in WooCommerce



Here you'll find [the related documentation](#).

How to set up Paypal by Braintree payment gateway in WooCommerce



Here you'll find [the related documentation](#)

I know, this setup procedure looks easy and, to an extent, it is easy. Yet some gateways require more advanced configurations and the [testing phase](#), which is crucial to see whether the gateway is working correctly, might not be a piece of cake as you thought in the first place.

Even if you have in-house developers working on your store, you might want to consider [a WooCommerce specialist](#) to set everything up correctly and test it before going live with it before seeing there's something not working. Something, in other words, you could have been anticipating through a specialist's expertise.

So how does the process of testing your new payment gateway look like? Here's the bare-bones version of it to give you the overall idea:

Basically, once you've added your gateway and settings, you'll need to try and test a transaction using their test option. If that works successfully, try a live site transaction using \$0.50 or \$1.00. If that successfully goes through and the money hits your account, then it means it's working.

Multiple payment gateways: which one should you have on your WooCommerce store?

PayPal, Stripe, World Pay, and many others. There's really no shortage of options to choose from when it comes to picking your payment gateway. But that's a decision that strictly relates to single business scenarios and specific customers' needs and it's darn difficult to provide an exhaustive answer here (a [WooCommerce specialist can answer that](#) for you). So I'd like to focus more on which payment gateways are recommended both from a business perspective and a maintenance/development perspective.

Let's start with one important aspect here: owning several payment gateways in your WooCommerce store is not a bad idea at all. This way you'll be able to serve your disparate plethora of customer's payment preference options. Then, it's about implementing one solution that while it allows you to provide more payment options, it also won't require steep setup costs nor lots of maintenance in the future. As Robin highlights:

I would suggest offering PayPal, which comes pre-loaded with WooCommerce, and credit card payment through Stripe. If you think your customers will use it, offer Apple Pay through the Stripe plugin as well, but that basically doesn't require any additional setup.

Wrapping Up

Each eCommerce store has its own specific features and configurations lined up. Yet PayPal paired with Stripe looks as one of the suggested solution suitable to the vast majority of WooCommerce businesses.

When choosing on a new payment gateway you should always keep in mind and evaluate the opportunity cost it might pile up on your shoulders in the future, as it could potentially lead to more complicated configurations and maintenance work for (an undesired) low [return on investment](#).

Shipping configuration

Running a business is hard. And when your business is an online store, there are [many additional aspects you need to take into account](#). Shipping is probably one of the most overlooked yet it is critical for an online WooCommerce store selling goods to customers. Therefore, it requires you to invest a bit of time to have it configured properly so that everything will work smoothly and, most importantly, the way you need.

When it comes to configuring your WooCommerce shipping options, it's mainly a matter of 3 elements that you need to act on and set up.

Let's dive in with the help of a WooCommerce developer!

1. Shipping zones

First and foremost, it is vital to define the areas and countries in which you intend to ship to. In WooCommerce, these are called "[Shipping zones](#)" and, if it's your first time trying to set them up, the WooCommerce dashboard will prompt you to define them. WordPress developer and Codeable expert [Shadi Manna](#) explains:

WooCommerce introduced shipping zones in version 2.6, which allowed customers to add zones to their shipping sort of logic. And what that allows you to do is basically define different shipping methods per zone. The zone can be defined as a country, as a city, and even as a postal code. So, it can be as detailed as you need it to be in order for you to be able to ship potentially at different rates or not. Within the WooCommerce wizard, it prompts you right away to go set up your shipping zones to ensure that shipping is set up correctly.

As you might have already anticipated, if you're selling a digital product such as an eBook, a plugin, or anything digital that will eventually be in a downloadable format, you'd just need to mark it as "Downloadable" in your WooCommerce store as there won't be shipping options for that product.

If that's your case, here's the [official documentation](#) from WooCommerce.

2. Shipping methods

Once you've configured the shipping zones, you're requested to add shipping methods to each of your Shipping zones, which cover how your product will be delivered to your customers. WooCommerce has three shipping methods by default:

- [Local Pickup](#)
- [Flat Rate Shipping](#)
- [Free Shipping](#)

Shipping zones match customer address from top to bottom based on how you set things up in [your WooCommerce backend](#). That means the first one matching your customer address will be the one that gets used. WooCommerce easily allows you to reorder these option by simply drag and dropping each shipping zones up or down to reflect your needs.

If no custom zones are a good fit, then the default "Locations not covered by your other zones" will be used.

3. Weight and size

Unlike the previous two, this aspect isn't something that all WooCommerce store owners and managers should configure. Still, if you need to take into account the weight and size of your products, and want to show them to your customers, things get a bit trickier because your specific settings should be dictated by the shipping provider you're using. Highlights Shadi:

In the United States, we have for example FedEx and UPS which may require the weight, and the dimensions of the products. There is some logic to add products and ship them in different packages. And it is through the carrier's API that you can pass that information over so that you'll get rates dynamically returned to you, assuming you have an account with that carrier.

As with [setting up payment gateways](#), many of the technical aspects are handled by official WooCommerce extensions provided by the shipping providers themselves. Shadi, who's the developer for the [official DHL for WooCommerce plugin](#), elaborates it a bit more:

Establishing and managing the API connection is done through the plugin. You just need to be sure to put in all information within the settings tab under "Shipping". That's all you need to do. Obviously, you must have the weights and dimensions entered into each product in order to ensure that the accurate information and dimensions are being passed to the provider that then returns the corrected rates.

Major shipping companies like FedEx, UPS, USPS, etc. all provide plugins for your WooCommerce store that can be configured to make shipping much easier for you. It saves you the hassle of managing the process separately and you can simply integrate it into your store.

One thing, though, you want to know in advance is that the majority of these plugins comes as premium (paid) plugins starting at \$79 USD for a single-store license, with the exception of DHL that's released as a free product. Shadi further notes:

The major shipping carriers have paid plugins you can use developed either by WooCommerce or trusted third-party partners. The plugin from DHL, which I'm specifically developing, will be given away for free because DHL wants merchants to use their services. So it really depends on the carrier, and this is the sort of the conversation you should have with anyone you want to use as a provider for shipping, and then they'll usually point to the best plugin to use.

Wrapping up

The configuration of the shipping process for your WooCommerce store might seem as an easy task for anyone who's a bit experienced with WooCommerce. And, in a theoretical way, setting up shipping preferences isn't that hard if your shipping doesn't account for discounts, bundled offers, specific rules triggered on your customers' selection, and [shipping classes](#), and so on. On the opposite side, the more options, granularity, and consistency you need your shipping to have, the more advanced its configuration will be.

Chapter 2: Building Up A Solid Environment

WooCommerce Maintenance 101

WooCommerce is one of the top eCommerce solution adopted by online store owners with a [12% market share](#) among the top 10K website in the world, which grows up to an incredible 42% share when considering the entire internet.

This free WordPress plugin, along with many available add-ons and integrations to enhance functionalities, has empowered many business owners to sell their products and services online with a more efficient platform. But even with that, any WooCommerce user still needs to invest time and resources in maintenance activities for maximum flexibility and control over their store.

The question you should be asking yourself by now is: what kind of maintenance do I need to perform for my WooCommerce Store? WordPress developer and Codeable expert [Robin Scott](#) has in-depth knowledge and wide experience working with WooCommerce websites. That's why we asked him to provide us with the most important things to know when it comes to keeping your WooCommerce store in perfect shape.

Let's see them!

Tip #1: Have a process in place to run updates on your WooCommerce store

The first step to a healthy and long-lasting WooCommerce store is having it running on the [latest releases available](#). The reason for this is that every update will result in an improved version of the software. Specifically, be sure that you're running the latest version of your theme, your WooCommerce plugins, and WordPress itself.

Since WordPress 3.7, there's now an [automatic background feature updates](#) running on your website you should pay attention to because it could break your WooCommerce store. The auto updates will only be triggered by minor releases and translation file

updates. Nevertheless, any [software update has to be tested prior implementing it](#) on your live store, unless you're taking chances to having a website not running properly.

As Robin highlights:

All it takes to knock your website offline is for a plugin, or a theme, to be incompatible. And this happens regularly, it's not uncommon. One plugin developer, for example, might have called a function the same thing as another. So it's accidental and coincidental. But when it breaks, it breaks your whole website and that's when the transactions stop happening.

The first thing to do is test any new update in a copy of your website, whether locally or on a staging website, and test everything else out such as website features, transactions, transactional emails, etc. and have it all check before performing any update on the live/production store. Once you've gone through your test phase, you're not done yet as you now have to make sure your live WooCommerce store is running smoothly.

Robin stresses the importance of testing and checking on all different elements working correctly, especially on the live website:

Once the site has gone live we thoroughly test the live website, the live checkout page with the product that costs the minimum amount that the gateway allows. We actually complete an order, pay for it and go through the checkout. The final part on that is to make sure all of the sales emails are being sent and received, and to make sure that basically what they call the transactional email is actually working. So that your users will be getting the receipts, order notifications, and all of that stuff.

And you thought "it was just an update", uh? When running an eCommerce there can't be any room for guessing and chances. That's why having a solid process you (or [your outsourced developer](#)) should follow, will benefit your business in many ways.

Tip #2: Run backups and check whether they're working or not

Backing up WooCommerce data is crucial. [An attack on your website](#) or any loss of data can cause you major disruptions. Therefore, having solid backups for your important information might save you the least of troubles. As Robin says:

Keeping daily backups it's not strictly part of maintenance really, it goes beyond that: you should just have them!

Anything that [a normal WordPress website would require](#) stands for WooCommerce store as well. That means you should automate your backups with a plugin, for example, and have different copies other than the one your hosting provider has stored on third-party websites or locally.

In addition to that, WooCommerce websites require a bit more precautions when it comes to having backups because they are transactional websites, websites where money is transferred from one end to the other. And for WooCommerce users, Robin suggests 3 things for their own piece of mind:

1. Increase the frequency of backups
2. Take snapshots of your database and store before pushing anything live, even if you've tested it on staging
3. Test your backups and see if they're good to rebuild your website

Why so much attention to backups and the testing phase? Well, Robin has a hard truth for you to hear:

With a WooCommerce website, restoring from backup it's a 'bad' thing. You don't want to have to do it, really ever. That's why we're testing in staging so much. Why is that bad? Because, as you can imagine, you will lose orders. If you restore your site to how it was at 12:00 AM (midnight) and it's 4:00 pm and you've done sales all day, those sales will all disappear from your system. So you really don't want to restore from backup. If you can avoid it.

Tip #3: Test thoroughly and test relentlessly

Updates and backups, glued together by a proven process to tackle them, are really important parts of any maintenance activity. But they're not enough. What's missing here is the most critical aspect of them all, one that would make your WooCommerce maintenance either effective or poor: the testing aspect.

Don't think testing is that critical to WooCommerce maintenance? Let me quickly list out when you should invest time in testing things out:

- Before any update comes out, testing is required on your staging website (or your local copy of the website).
- After the updates are performed on the live website, testing is strongly suggested to check and see if everything runs as properly as before.
- For any backup you create, you should test that file to be sure it's able to revert back your store to a full-functioning and stable instance.
- Any payment gateway has to be tested to ensure you're not missing any money transaction on the live website.

Especially transactional emails and gateways require even more of your attention because they're directly connected to your revenue stream. As Robin emphasizes:

With most gateways, the sandbox environment is slightly different to the live environment. Therefore, the best way to know if live transactions are working is to see one happen. And we have that as a checkbox for our maintenance clients. We do it so that we can sign off on it and say 'yes, it's working' and reassure them.

The lesson here to learn is that you should always [keep testing thoroughly](#) that your WooCommerce store website and all its specific "pieces" are working properly. You have to make sure that updates to plugins, add-ons, and your own theme too won't affect your store negatively.

To avoid making mistakes, which might cost you some real money, you should create a testing checklist and strive to follow it. The checklist input should include parties from all corners of the website, and even make it more detail-rich and narrow down to cover specific tasks such as testing your shipping, payment gateways, and so on. With a testing checklist to follow, there's no longer room for guesswork and the process will be more efficient.

Given how much testing is involved when investing resources in maintenance, Robin suggests a nice tool that will test many elements and areas of your WooCommerce store for you:

There's a really great tool out called [Robot Ninja](#). I really recommend looking at that as well for just a test, because it tests whether the things can be added to cart. It tests transactions. It tests the payment gateway and it does all of that automatically and uses different browser simulations.

Wrapping Up

Whether you maintain your WooCommerce store yourself or [have hired a developer for maintenance duties](#), upkeep remains crucial for your website to run properly and, to an extent, last longer. On top of those already-mentioned tips, other measures include choosing a great hosting provider for your store and [taking security really serious](#). As a WooCommerce store owner (or manager), you should always strive to keep your store running smoothly for the benefit of your customers. Soon enough it will turn into that of your business as well.

Your Guide To Prevent Data Disaster

The most valuable asset for your WooCommerce business is your customers who buy your product or service. Their choice of giving your business their email address, credit card numbers, shipping address, brings to the table a key topic for store managers: [trust and data protection](#).

In fact, your store's customers demand your business to protect their data and information they've shared with you. If they're unsure whether your store is a safe and secure place for their data, they'll reconsider how trustworthy you are. And trust is so fragile that could be lost for good.

Data protection isn't strictly related to being under attacks from hackers. It means having a bunch of tools and strategies in place to keep safe and sound your customer's sensitive data. Both in good and bad times. And, we all know, things could break eventually.

As a WooCommerce store manager, you surely want to lower your chances of a store crash, downtimes, broken database issues you've heard of. That's why I've grouped here the critical aspects you ought to know now, when things are running smoothly, so that you can use them if/when disaster strikes.

Let's get started, shall we?

Prevent data loss with database backups

Step one to any data safety strategy is to [maintain a backup](#). If due to any unforeseen circumstance you lose valuable information, you have a place to get your data back and you won't end up at square one with your business. WordPress developer and Codeable expert [Robin Scott](#) stresses out how having a solid system for backing up your store's data is important:

Rule number one to safeguard your store from disaster is having multiple backups, those automatically provided by your hosting company probably at least on a daily basis but also you should use your own backup software. It's recommended. It's simply good practice to have also an offsite copy of them. For that, you can use a

backup service like UpdraftPlus, which stores the backup externally like in Dropbox or Google Drive.

Test your backups and updates

Having regular backups is just one piece of the entire puzzle. Since there are many WordPress backup plugins to choose from, [you have to test run your backups](#) to make sure they're proper and aren't creating corrupted or incomplete files at a crucial moment. The choice of a backup plugin should be based on your store configuration, needs, and in-house resources (who's going to be in charge?). As Robin highlights:

With our clients, we basically use a server-to-server link to generate backups. But there are quite a few options on the WordPress plugins repository and the question really is: 'Which one works in your environment?' So, your goal is to find a good one and then test it: try it and restore from those backups you're getting. It's boring, I know but it's crucial. Some of these tools don't work. And, sometimes, it could be the case that your host might have a restriction on the length that your backup can run. So, actually, you think you've got backups, but what you've got is half a backup. That's why you need to check them.

And it's not just backups, [even your updates need to be tested](#) before they're taken to the live website. It's important to run tests to pinpoint any errors that might cause major headaches in the future. Robin further elaborates:

Testing is going to prevent disasters. You should always test updates after you put them on the live site, at least the transactional side of things. Test everything before you do it but also test everything after you've done it. If you can take up an issue in staging, before it's happened on the live site, that's a good thing. If it happens on the live site, that's when you start to get data issues.

Put hack prevention in place

Beside correct backup files and a thorough testing approach, the next thing you should take care of to prevent data issues with your WooCommerce store is [hardening its security](#). It all starts with [your hosting provider](#), who should already offer some tools to keep the "bad guys" off of your data. But, as you know, when security and data leak is at stake, more is always better.

That's why don't have to forget your store is running on WordPress and all [security best practices for WordPress websites](#) are applicable (and strongly suggested) in this case as well. As Robin comments:

When we're talking about disaster prevention, it's also a matter of safeguarding or hardening the security in WordPress itself. Most of the good hosts have some sort of hack guarantees and they have firewalls in place to prevent bad behavior. At the same time, though, you should have something in place. It's the basics.

One aspect that directly affects the security of your store's data has to do with your users - not customers- and who's been granted access to your backend. Specifically, it's in the way you have configured and manage user roles in your WooCommerce store. In Robin's words:

Don't allow anyone access to your store who doesn't need it. Don't give anybody admin access, unless they absolutely require it. For most scenarios, that's really only developers and one person in the business. Anyone who's just doing data entry and all the management doesn't need to have admin-level access to the website. WooCommerce has got a Shop Manager role and you should use it. It's the best thing to do because it prevents that person from making mistakes. But it also it keeps the admin panel pretty clear of all of the options that they don't need to be seeing as well.

Here's a good resource to learn more about [roles and what they can do \(or can't do\)](#) in your WooCommerce store.

What you need to know before your WooCommerce store crashes or gets hacked

Bad things happen in our life, hence in your life of a store manager. That means your store could break and even become unreachable for no apparent reason. When that happens, you'll need to keep in mind what's the most important data you need to secure and always have handy:

- Order numbers
- Customer IDs

These two types of data are the most critical to a store and it'd be pretty darn hard to retrieve them if things go south. As Robin exemplifies:

The situation we're trying to avoid is the loss of transactions, which is really a disaster in itself. It's really hard to recover from as well. The primary thing you'd want to be backing up is the order numbers and customer IDs. In an ideal world,

they'll be backed up every instant. We use AWS a lot and we create a snapshot of the whole database server and a snapshot of the whole file system the website is living on. That's done very regularly and on a rolling cycle and it gives us a complete image of everything involved.

That's the best disaster-prevention stack for lowering the chances of not incurring in major data losses and nightmare-ish times to get everything back correctly. Yet, given the advanced configuration and tools required, many small business budgets will hardly be able to accommodate that advance setup. In Robin's words:

That configuration stack is an ideal scenario, but not every store is in that situation. It then goes down to daily and automated backups performed by someone at your company and by good hosts like WP Engine, Kinsta, or any provider like that. At the end of the day, the whole thing is really about as much as you can afford and as often as you can.

This aspect is a critical choice for any WooCommerce store manager but might require some technical knowledge to clearly understand how all the parts work together.

[Consulting with an expert](#) could help you get a better picture of it all.

You know, though, that if you had all the measures in place to prevent disaster from happening, it could still be happening, right?

Let's see what you should do, then!

What you need to do when your WooCommerce store crashes or gets hacked

Breathe, it's game on!

What would you do when your WooCommerce store breaks? Would you just easy-peasy click-restore it from a backup? Would you ask your host to do that for you and revert your store to an old functioning instance? Wait, please! There's a key thing you must do before touching your backup files or do anything else.

Recall to your mind what you just learned: you don't want to lose order IDs and order numbers. So if the store drops out, [it's hacked](#) or you have a disaster happening, you have to hold your horses a bit longer and refrain your desire to reverting to a previous state.

Why that? Because doing so would make you lose all the data that had happened between crash and restoration of your WooCommerce store. And that's something no store manager wants to do to their business. When your website crashes to the point that you need to restore it, there might be data that was being entered or that is entered during restoration. If you're not careful you might end up losing that data. Robin explains:

If orders have happened or maybe new customers have signed up accounts since the time that the disaster occurred, you're going to need to secure that information from the current live database before you restore from backup. To do that, copy that database straight away. It might sound counterintuitive because it's gone wrong, but that's the data you need. So, copy your live database, get a copy of it. It might not be accessible. It might be down. But you need to get a copy because an expert can go in there and can pull out anything that's salvageable. And particularly we're talking about customer IDs and order numbers. If somebody's paid money into your store, you want to find that order.

Your goal, at this point, is to do everything to lose the least data available in your database as possible. And if you had reverted to an "older" version of your store, you would have lost that data.

Wrapping up

Your first means of safeguarding your WooCommerce store from a disaster are as strong as your disaster plan and backup strategy have been set up. The weakest the latter, the worst the scenario you'll be experiencing.

WooCommerce stores are vulnerable to attacks and major issues that could result in downtimes, corrupted database, or simply being hacked. Things could fizzle out pretty fast sometimes, and it's at the times that a good disaster plan would show its real value. You know the old saying "Cross that bridge when you come to it", right? Well, when things go bad - and they do eventually - there could be no more sales for your store by the time you'd crossed that bridge... think accordingly!

Hosting and Backup Strategy

Your WooCommerce store has loads of customer data in its database and there are new ones coming every now and then. In case of a failure, all this valuable data might be lost and it could potentially bring your business back to square one. Odds are against you and your store as it just a matter of one bad apple to spoil the whole bunch.

You just need one failed element to start losing money

[WooCommerce relies extensively on themes](#) and plugins that can be added for a variety of purposes. Sometimes, ill-maintained or poorly coded products can cause your website to crash. What's worse is that using these incompatible plugins or themes on your website might result in losses for your company.

Things don't get easier as there could be [several reasons for your website to go down](#). But if you're using an incompatible plugin or version of a theme on your WooCommerce store, you're paving your business path with a lot of future headaches. As WordPress developer and Codeable expert [Shadi Manna](#) explains:

I had a client, who had 16,000 orders, and one plugin was simply crashing their site all the time. They weren't aware of that. All we needed to do was disable it. And this was costing them a lot of money because it will crash their site every few days. So it's very important to only use also reputable plugins by companies that work with WooCommerce, understand what WooCommerce does, and provide well-maintained and tested products.

When you think that just a single element could tear down your entire WooCommerce business, you might get scared of going out of business, if that ever happens to you. That's why backups are so (darn) indispensable for any website.

Why is a backup strategy so important?

To prevent your heartbeat from skyrocketing each time you load your store pages, there's one thing you can start working on now, and of which we're all firm supporters here at Codeable: have a [solid backup strategy in place](#).

Having backups in place from the start allows you to be insured against any loss in the future which might be immense if your store has grown in size over time. If your WooCommerce store crashes, it is like your retail outlet suddenly shuts down. The consequence? You lose customers shopping at that point in time!

With a WooCommerce store, you might lose much more than that as it could result in potential loss of customer history too, causing greater damages to your business and harder times getting back on track. As Shadi highlights the need for a backup:

Backup is a very important topic in general for websites, but it's particularly important for WooCommerce sites because it's a transactional website as opposed to a more "static" WordPress website. When you have transactions occurring on the website continuously, such as new accounts, new orders, anything which is stored on your site, it is vital for you to have backups because you're losing customer data, if anything were to happen to the server. This could be thousands of dollars lost because of these [failed transactions](#).

Having a backup strategy means [setting up an automated system](#) through which you can generate exact copies of your WooCommerce store, mainly your database where all of the important data is stored.

But here's the twist many often forget to mention...

A solid backup strategy is strictly linked to your hosting provider. You'll see why in a second.

Why choosing a reliable hosting service is so crucial to your backup strategy?

Hosts are typically responsible for placing and keeping your site over the internet. That's the basic offer, I'd say. But if things go south with your WooCommerce store, you'd want a quick way to have a running backup of your store and a smooth process to restore it as fast as you can, wouldn't you?

Unfortunately, these types of features don't usually come with the cheapest solutions you find on the market. Yet, for a small business owner who maybe can't rely on in-house technical staff, they're worth every single dime in the long run. As Shadi highlights:

Picking the right hosting provider is crucial. It's extremely important for WooCommerce sites to have reputable hosts that are familiar with WooCommerce, they understand the implications of losing data, they have great backup systems, they have great servers, great uptime. I typically recommend the [WP Engine](#) and

SiteGround type hosts that manage only WordPress-type websites and are very familiar with WooCommerce, and can even give advice on these types of things.

Choosing a reliable hosting provider puts your business in the hands of someone who understands your business, has the infrastructure it might require, and might anticipate some of the issues you've experienced in the past and suggests solutions or partners you can reach if needed.

But let's dig deeper into an important yet often neglected aspect of a good hosting provider. Specifically, their "getting your store back online" feature, or what it's usually referred to as the restore process of your WooCommerce store.

How does your hosting provider relate to the restore process of your store?

Having a reputable and reliable hosting service is key to being in control of your store's availability. Specifically, the host's server size, configurations, and infrastructure stack make a huge difference to the number of downtimes you might incur.

But it's in the event of a crash, a [WSOD](#), or a [failed update](#) that your choice of a good hosting provider will show its true value.

For many small businesses, the easiest way to restore their WooCommerce store would be to just click a button and revert everything up to a previously recorded and fully-functioning instance.

Think about it for a second: what if you had to call in and hire a developer or a sysadmin each time your WooCommerce goes offline? Would your budget be able to make room for that? Well, you know how paying for quality tools and services it's always worthy.

And when it comes to the easiness of backup and restore process, that means looking for higher level providers, namely WordPress managed hosting providers.

As Shadi further comments:

For small businesses, most of the time it's in the hands of the host providers in terms of the server uptime, and in terms of the backups. I would suggest using a reputable host who's familiar with WooCommerce and WordPress and how the

database works. They have backups so you can easily restore them, and you should have one-click restores where you can restore the site right away.

Easy-to-perform and efficient restore is necessary because all the time the website is down, you're losing customers.

Why backup and restoration for large businesses is a different game

The dynamics of backup and restore processes are completely different for larger WooCommerce stores for they have greater traffic and customer volumes. With more elaborate setups, they need more effective systems for backing their data and being able to get it back in case of a crash.

These include, for example, having their own servers and data mirroring, i.e. a duplicate backup for the database created every minutes or so to lower the chances of losing data. Shadi notes:

With bigger WooCommerce stores, I'd suggest to really hone in on the backup strategy and set up hourly backups, or, in some cases, might even do some mirroring of the whole database every five minutes, if the shop is that popular and large. This setup is complex and should be performed only by experienced developers who know how WooCommerce works. This can't be stressed enough: if your database was to crash, you would be losing all of the transactional data within that time, to when the latest backup was made and there would be no way to retrieve it.

Beyond hosting: When it comes to backups, it's never "too much"

Backups are never enough and there are plenty of plugins who can take care of that for you, either free or paid. You just have to pick one, configure it, and fire it up in your WooCommerce install to clear the worst case scenario off your head.

That's just the bare minimum, though, as you should enhance your backup strategy with several ways to make and store your backups and [a more thorough process to follow](#) when your store goes offline.

Snapshot of your database, mirroring, offline backups are just some of the many ways you could have a better backup and restoration plan for your store. If you care about your WooCommerce business, [getting a WooCommerce expert](#) setting a solid backup strategy plan for you might be more than helpful.

Wrapping up

Backups are sort of insurance policies: you have to have them, hoping you'll never need them. When it comes to your WooCommerce store, backups are your fail-safe mechanisms in case things go awry and help you avoid major losses. But backups alone are just one side of the coin, as without the ability to seamlessly restore them would end up being a major business drag. That's where your hosting plays a prevailing role over any backup plugin solutions you might pick.

Given that you can easily and cost-effectively change your backup plugins, picking the right hosting provider for your business is a more difficult choice as it'd be tougher to move your store to a new provider than just installing a new backup plugin. So invest some time on it and, when in doubt, ask a developer what's your best call based on your configuration.

Chapter 3: Speeding Up Your Store

How To Boost Your Slow Admin Panel

We're big fans of speed because, as it's been widely proved, [speed makes all the difference](#): better UX for your users, higher SEO rankings, improved click-through rate. But speed isn't important only for the visible part of your WooCommerce store (aka the frontend). Speed can also affect how you manage it, how smoothly and quickly you'll be able to make changes "behind the scene", how efficiently you'll be able to work in your admin panel (aka the backend).

So, this time, I'd like to focus on this part of your WooCommerce store. Thanks to WordPress developer and Codeable expert [Nabeel Molham](#), I'll go through some of the most common issues that might be causing your WooCommerce admin panel to lag and provide tips on how to fix them.

But before that, there's something you should do...

Update everything before you start

Before any fix, or improvement work with software there's always a step 0 that you need to take, otherwise, your work will be of less impact, if not useless to an extent. And when it comes to WooCommerce, this preliminary tasks you're strongly suggested to take is [updating your store](#) to its latest available releases. The benefits of keeping your WooCommerce and website up-to-date have already been discussed.

It's only after you're sure all of WordPress core files, the WooCommerce plugin along with all other plugins, and your theme as well are at their latest stable release that you can start working on fixing and improving your slow WooCommerce backed.

It's now time to see which are the top 6 reasons for a slow admin panel in WooCommerce.

Let's start!

Reason #1 for a slow WooCommerce admin panel: Object caching

WordPress and WooCommerce users rely heavily on plugins because they make their life so much easier. Problem is that, not only all plugins are created equal (some are

great, others good, others again are simply poorly coded), but also a plugin might affect several areas or aspects that you don't expect to be directly touched by it. For example, one type of plugins that has gathered lots of attention among WordPress users is [caching plugins](#). The main reason behind this choice is that, when properly configured, [having a good caching system in place](#) can dramatically increase the performance of any WordPress website.

I said properly configured because wrong setups within [W3 Total Cache](#), one of the most famous WordPress caching plugin, could lead you to [very slow pages](#) both on your frontend and backend. As Nabeel explains:

If you're using W3 Total Cache, based on your specific WooCommerce builds and features, it sometimes might get you a slow backend because of the 'Object caching' feature. If you're seeing that happening, I'd suggest to simply disable the object caching option in that plugin so that your backend will almost immediately have some performance boost out of it.

As a good alternative, Nabeel suggests giving [WP Rocket](#) a try:

If you don't want to use W3 Total Cache, there is another one that started to get a lot of attention: it's called WP Rocket. Especially, the latest versions of WP Rocket are providing really good results and can speed up a website dramatically. And, most importantly, WP Rocket doesn't affect your backend in any way.

Another little tip on top of disabling object caching you might want to try is also disabling database cache as well to gain a little more speed boost in your admin panel.

Never forget, though, that playing around with caching preferences isn't as easy as the possibility to check/uncheck options provided by the plugins UI makes you think. Speed and performance are serious matters you should know how to properly handle. If you don't want to mess things up and get your store slow, [ask help to a performance specialist](#).

Reason #2 for a slow WooCommerce admin panel:

Translation plugins

If your clients are spread around the world, there's a high chance your WooCommerce store is available in more than just a language. For that reason, it could become quite heavy and result in slower speeds. For example, [WPML](#) is known for making WordPress admin run slow and there are several things [you'd need to do to fix its slow performance](#).

As Nabeel highlights:

One of the most used plugins for multiple-language stores (and websites) is a plugin called WPML. This one might be really heavy on the CPU, especially on the backend. If you need to translate your pages into several languages, I'd suggest switching to different plugins such as [Polylang](#) or [qTranslate X](#), which don't have as many features as WPML, but they're lightweight on your admin panel. For basic translation needs, both of these plugins will do the trick.

Reason #3 for a slow WooCommerce admin panel: Old transients

You might be not aware of it but your WooCommerce store saves temporarily cached information in your database, like customer sessions data, in the form of [transients](#). If you haven't configured any automated way to remove those you no longer need to store, they can quickly start cluttering up your database.

Result? Slow admin panel in WooCommerce! Explains Nabeel:

Sometimes old and expired transients aren't successfully cleared in WooCommerce. Unfortunately, when they are stacking on 'top of one another' over time, they cause the slow down your backend. What I'd strongly suggest here is for you to install a plugin that will help you manage transients. There is a great one called [Transient Manager](#) thanks to which you'll be able to delete all transients, or just delete the expired ones.

Regularly clearing useless transients will help you keeping your WooCommerce backend run smoothly.

Reason #4 for a slow WooCommerce admin panel: Unused plugins/themes

As with any WordPress and WooCommerce install having many unused and outdated theme files and plugins might affect your performances. The main reason is that all these files, even if deactivated, they keep storing their settings in your database. But there's more: WordPress will keep checking for new updates as well, which will cause unnecessary HTTP requests in your backend. [With WooCommerce there are also add-ons](#) you should be taking into account. That's why deleting all these files you're not using will benefit your store's speed as a whole.

Reason #5 for a slow WooCommerce admin panel: Wrong settings within your Content Delivery Network

Content Delivery Network (CDN) services like [Cloudflare](#) or [KeyCDN](#) are great tools to help you serve pages faster from your WooCommerce store. Specifically, they "duplicate" your pages across multiple servers in different locations so that the time to retrieve those pages is reduced because they're served from the closest location. If you're relying on a CDN, you might experience a slow admin panel because you haven't set up a specific rule within their settings. As Nabeel warns:

Depending on the Content Delivery Network that you're using, it might be the case that your backend gets delivered through the CDN as well, resulting in a slow admin panel. With some providers, you don't need to take care of it because they automatically exclude your backend but others don't do that right from the start. If that's your case, I'd recommended excluding your admin dashboard from the delivery network to improve its performance.

In other words, you should invest a little more time to create a custom rule that will allow your admin panel not to be delivered through the CDN (here's [how to do it on CloudFlare](#) and [how to set KeyCDN up](#) properly).

Reason #6 for a slow WooCommerce admin panel: Slow servers

I kept probably the most important as the last one: even if you put in place all these tactics, if [your servers suck big time](#), your WooCommerce store's backend will be no different. That's why you'd need to realize that if you're relying on a cheap hosting provider, it won't get you very far. If you have no idea how your servers (and hosting provider) are doing, you could talk to a [speed and performance specialist](#) to get some advice.

Additional things you should try to make your WooCommerce admin panel faster:

With these six elements above here to look into you should already be able to fix, or at least improve, your WooCommerce's backend speed. I'd like, though, to give you a couple more inputs that will help you get a better understanding of what's happening

"underneath" your store so that you can monitor what's clogging everything up and act accordingly:

Install the [Query Monitor plugin](#) or [Debug Bar](#) to get a more effective way to uncover slow scripts/database queries. As Nabeel describe it:

This is not a fix for your slow admin panel. It's more of a way to investigate or to get more information on what's happening in your backend. There are two plugins I'd like to suggest you, one of which is called WP Query Monitor, while the other is Admin Debug Bar. The combination of these two plugins will give you clear insights and alerts on any process that's taking too long, which might lead to a slow admin.

Install [Heartbeat Control plugin](#) to have more control

WordPress introduced [WordPress Heartbeat API](#) with WordPress 3.6 in 2013, which introduced some cool features such as the autosave feature (Thank God!), the post locking feature warning while two users are working on the same post, and so on.

What's important here to know is that the Heartbeat API will keep sending AJAX calls to the server as soon as you - or any other of your users - is logged into the admin area. Continuous calls and server request?! This can't be good for performance! That's why thanks to a plugin such as Heartbeat Control you could stop it from being triggered in some or all areas of your store.

Make sure your WooCommerce store has enough memory

This isn't something that's strictly related to your admin panel yet it could help you prevent [PHP memory limits](#). Here's the [documented process](#) from WooCommerce.

Wrapping up

Speed is important for your customers but should also be powering your admin panel to make both your work and your developers' life more efficient. A WooCommerce store is a complex business asset that connects lots of "moving parts", some of which [you can surely optimize and improve](#). Never forget, though, the two most important elements that directly affect your store's performance: its foundations (aka how [it's been built in the first place](#)) and the quality of your hosting.

How To Speed Up Your Slow Product Pages

Are your users dropping their purchase because your product pages are painfully slow? Been there, done that! I mean, in a world where we're accustomed to buying almost anything online, your users can't stand to wait for your WooCommerce pages to load. Besides that, Google will also retouch your rankings pretty hard and position your pages down under. Or even worse, to the second page.

According to latest usability research, users begin to drop off a site when its response time is [longer than 2.5 seconds](#)! Just to give you a real-world example: a 100ms load time increase for Amazon would result in 1% loss of their revenues.

That's all to say speed isn't just the icing on the cake, something you could ignore when you run a WooCommerce store. Speed and performance optimization has to be one of the pillars [your store's maintenance](#) is built upon.

So the question is: how long do your WooCommerce product pages take to load? Are they fast enough for your users to buy?

If you're not sure what to answer here, or just [have no idea where to look at](#) to fix your slow product pages, here are five things for you to check out and see if they're causing your pages to lag!

Let's dive in...

The "size matters" side

The overall size of the product page contributes a lot to the speed at which it loads. As a rule of thumb, if your product page is loaded with images and is more than 2MB, it'll surely affect the loading speed and will stall your website. WordPress Developer and Codeable expert [Robin Scott](#) explains how to determine if your pages are too heavy:

If you need to check whether your page is heavy (or not), what I would do is run a page test on tools.pingdom.com. That will tell you the size of the whole page (HTML and images). Just as a rule of thumb, if your page is 10 MB or even 20 MB, which can happen in a store that has a lot of gallery pictures, it's just too big and almost always because of too many, very large images.

When you want not only to fix but also improve your WooCommerce store's performance, you always need to start by collecting the data related to your pages. This way you'll earn a more detailed and precise picture of whether something is clogging up your product pages on the frontend and plan how to fix it accordingly.

The "good hosting vs bad hosting" side

I can't stress this even enough: a [good hosting service plays a major role](#) in the amount of time your product pages take to load. It is vital to choose a reputable host so that your site is not marred by lag issues. On this matter, Robin names some of these hosting providers:

It's best to use one provider that's demonstrated itself to be effective for hosting WordPress and WooCommerce websites. There are probably four off the top of my head that are very good. These are all the managed hosts. We got Cloudways, we got WP Engine, we got [Kinsta](#) and then there is [Pantheon](#) as well. Those four, I will basically be happy to fully endorse them as being unlikely to be the problem if you've got slowness in the frontend.

This slowing down might not necessarily be blamed on the host all the times, rather on how your host handles requests or how it's configured, as Robin further elaborates:

It's not necessarily that you've got a bad host, but potentially you're using too many resources. Let's say there are too many concurrent connections. Which is people all accessing the website at the same time, or more to the point, transactions all happening at the same time.

You can optimize a lot of things and aspects in your WooCommerce store to make your product pages faster, but if "the engine" isn't able to perform at maximum speed, you'll see little tangible improvements. Investing in a more robust provider, with higher-performance servers stack and optimized configuration, is an effective way to prevent future headaches in the long run while getting almost immediately speed boots.

The "caching setup" side

Roughly speaking, caching is a process that speeds up website loading times by storing webpage data temporarily and bringing it up when requested, instead of loading the whole site again. As you can easily imagine, how your caching solution is set up directly affects your store and pages' speed, either for good or for bad. As Robin further stresses developers often ask questions about your caching configuration:

One aspect that I'd suggest to look into - or have an expert do that for you - is 'Have you got caching set up properly?' because that's something you have to have correctly configured. Whether it is WP Rocket, W3 Total Cache or, if you're using a good managed WordPress hostings, they come with their own pre-rolled caching systems which are, to save a long conversation, pretty good.

These outlined here are pretty common elements to always check when your product pages seem slow on the frontend. Of course, they're far from being the only ones you could perform on your WooCommerce store to optimize its performance.

In fact, if your store handles lots of transactions, visitors, and requests, you might want to [focus on the backend as well](#) and check if it's underperforming. Specifically, on advanced techniques that have to do with your database.

The "larger store" side

With large WooCommerce stores, the fixes are different because the dynamics of these stores are different and there are much more technicalities involved. One of the options for bigger sites is database optimization, with a key role played by a change in the database engine that's been used since. Robin points out:

For large stores with lots of orders and lots of products, there are some advanced database techniques that can speed up the way that WordPress works once it gets to a very large size. They have zero impact on a small store. Changing the database storage engine is one of them. In fact, MySQL has changed it's default storage engine quite a while ago (MyISAM), and stores now could benefit greatly by using the storage engine InnoDB. This is a simplified description, as there's more to do in the database to make a larger store faster, but it's really a developer-level conversation, and can get very advanced. Larger stores should pay a developer who knows WordPress and WooCommerce to optimize their database.

These optimization procedures, as well as doing some clean up on the WordPress options table, will bring in a very noticeable speed impact.

The "Now what?!" side

If you've checked off all the aforementioned things, and [everything is up-to-date](#) chances are your theme or a plugin or several plugins are either misconfigured, or there's something wrong happening in the system. As Robin highlights:

If despite these steps, your website performance doesn't improve, it might be time [to call in a specialist](#) to look over the site and help you figure out what's wrong.

Is WooCommerce a slow software, in the end?

Speaking about WooCommerce and speed brings up a whole different story here, as there's a school of thought that thinks WooCommerce isn't the right piece of software to handle big eCommerce stores. The main reason? Because - they say - it doesn't scale well. But that's just a short answer to a deeper question related to the software ability to adapt and grow smoothly when visits, transactions, queries, database records, and so on, exponentially increase.

As Robin explains, such short answer is biased:

It was mentioned at a conference two years ago that WooCommerce doesn't scale particularly well; it kind of is slow is the short answer. To that, I'd say that yes, out of the box and without performing database optimizations, as you get bigger, it will get slow. But it's worth saying the database is being improved: [WooCommerce 3.0 brought a load of problems](#). It broke stores. But the reason it did that was to create an environment that can be much faster in future. Overall, there are improvements being made and there are some massive improvements coming. But even now, we are comparing WooCommerce favourably with the very best hosted eCommerce platforms, because it's a great software, and a real market leader: it can stand up to this level of competition.

Wrapping up

With a strong and growing market share of 28% of all online stores, WooCommerce is the most adopted eCommerce platform worldwide. As any other living piece of software, it is under constant development to bring in better performances and a more stable environment. Given that, though, if you're experiencing slow product pages with your store, it won't be that easy to fix them as many factors have either directly or indirectly an impact on them.

But that doesn't have to stop you from acting on them, whether yourself or with the [help of an experienced developer](#), because it's your business and your revenue the elements that are ultimately at stake here!

Chapter 4: Fixing Your Store's Common Issues

How To Fix Image Issues On WooCommerce

Can you imagine buying a t-shirt from an online store without seeing it?! What if Amazon, for example, had no images at all for its product. Wouldn't that affect their sales numbers? I strongly believe it would! And I do that because we, as human beings, are visual-thirsty creatures: it's been proved that a well-curated image of a product directly [increases the conversion rate](#) of people buying that product.

That's why WooCommerce store managers need to invest in the [imagery related to the products they want to sell](#). Having nice images is just one part of the equation here, as it could be the case that your product images aren't just showing properly on your pages, scaring away prospects and customers.

Product images are one of the most effective tools in your sales pipeline. It's not simply a matter of making your website more attractive: it's about selling more. That's why I dug into how to fix the most common image issues in WooCommerce and provided you with quick solutions to that!

Are you ready? Let's start!

Types of images in WooCommerce

Before jumping onto how to fix issues with images, let's clear the air on which different images types are available. WooCommerce comes with three types of images you can set up and take advantage of with your store. Specifically, you have:

- Catalog Images: medium-sized image used in a product loops (shop page, product category pages, related products, up-sells, cross-sells, etc.).
- Single Product Image: the largest image on the individual product details page.
- Product Thumbnails: this is the thumbnail, commonly used underneath your Product Image, the cart, and widgets.

Issues with images in WooCommerce

There are mainly 2 issues when it comes to WooCommerce and images not showing correctly:

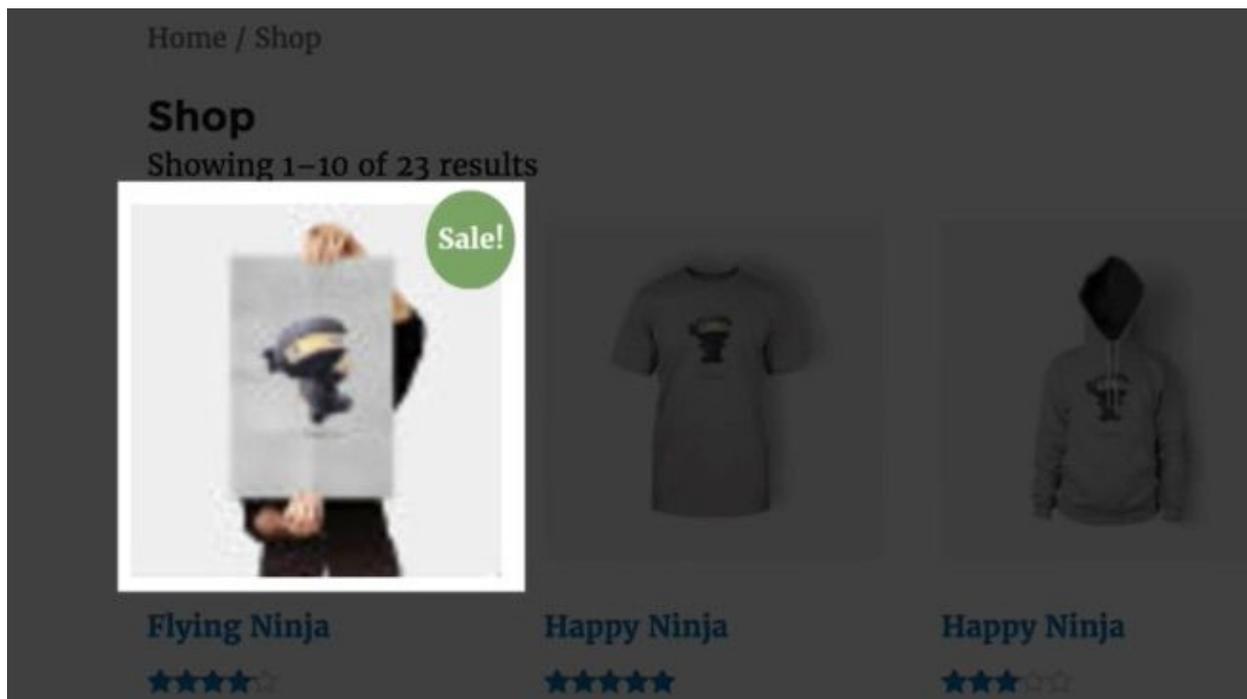
a) having blurry product images

and

b) having one of the new gallery features not working properly.

Let's start with blurry images.

Image issues in WooCommerce: Blurry product images because of wrong image size settings



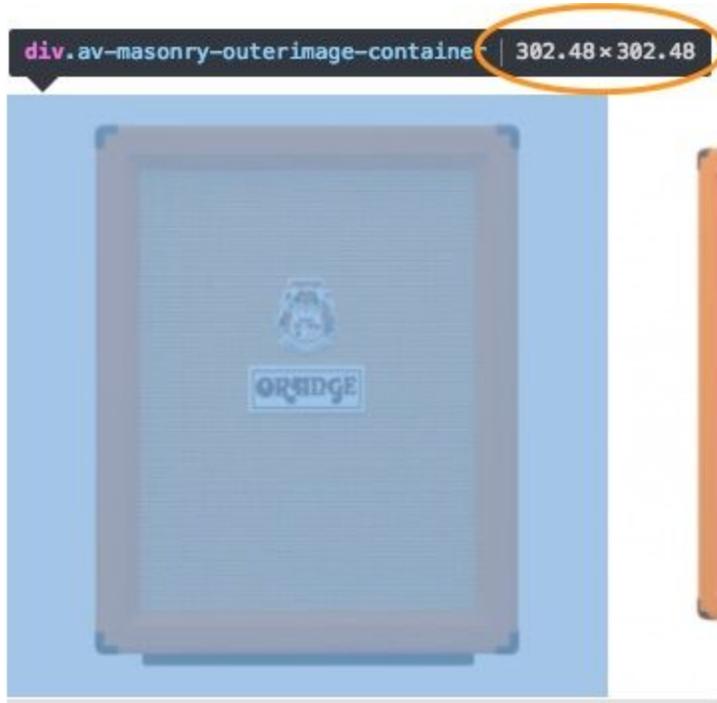
When your images aren't uploaded properly, as a result, you get a blurry image in WooCommerce. Why that? Well, the reason could be that your Product Image settings aren't properly configured based on your current theme's image size requirements. For example, your theme requires Catalog Images to be at least 200x200 pixels, while your WooCommerce image settings have been configured to 150x150 pixels.

How can you see which are your theme's image requirements?

As WooCommerce has three different types of product images, you'll need to repeat this procedure for all the three types. Here's how I did it for a Catalog Image. I'm on Chrome, [here's how to enable the inspect element](#). Here's how to do the same if you're on [Safari](#) or [FireFox](#).

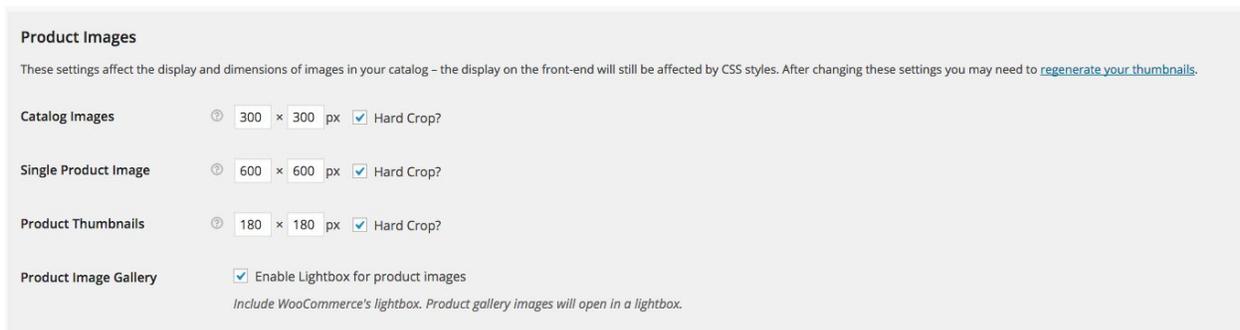
So, back to business...

I used [Orange Guitar Amps' website](#) as an example:



Here's what you need to do:

- Open up your store
- Navigate to your Catalog Image
- Right-click on it and choose "Inspect"
- Note down those image sizes



Now, do that for the two remaining types of images (if you use them on your store) and head back to your WooCommerce display settings. In there, update the Product images box (shown above) with the image sizes your theme requires you just noted down. It's

advisable to add slightly higher sizes here to lower your chances of incurring in this issue again in the future.

Once the new image settings are saved, you might need to regenerate your images and you could do that with the [Regenerate Thumbnails plugin](#). As WordPress developer and Codeable expert [Nabeel Molham](#) comments:

It's really important to have your image sizes correctly configured in your settings, otherwise your product images could load improperly. You need to specify the width and height of each type of picture based on your theme specifications and, ultimately, on what looks better on your store. And when you do that, make sure to regenerate all your images to reflect your updates.

Image issues in WooCommerce: Blurry product images because of low-quality images

If you're still getting blurry product images it could be a problem with your images not being high enough resolution. As Nabeel explains:

If you're still experiencing issues with your images, you might need to re-upload them with the highest quality possible available. This step is critical to generating the different image types sizes on your store because they'll be created based on this image resolution. This way, the blurry or pixilated issues won't be there anymore.

When going through this new upload task for your images, WooCommerce suggests that anything at 800 x 800px or higher would work for most themes. Now you have another great reason to invest in some higher-quality product images!

The second most recurring image issue in WooCommerce is related to the new gallery features [introduced by WooCommerce 3.0](#). Let's see why!

Image issues in WooCommerce: zoom, lightbox, slider feature not working properly because of incompatibility or plugin/theme conflict

With WooCommerce 3.0 a new set of product gallery features has been introduced. Specifically, I'm talking about:

- Image zoom (also referred to as "magnification")

- Lightbox
- Slider

Because of these new features, many WooCommerce store managers had incurred into image issues with their store when updated to WooCommerce 3.0. One of the main reason is that the theme they were using didn't support the new WooCommerce 3.0 gallery features. That's why it's strongly suggested to only [use WooCommerce compatible themes](#) right from the start.

But that isn't the only reason to prevent you from not being able to enjoy these new gallery features. Another reason can be found in a [plugin/theme conflict](#). If that's your case, you should follow the standard troubleshooting procedure for such types of conflicts and try disabling your plugins (and add-ons) one by one to check which plugin is causing the issue. If you aren't lucky with plugins and add-ons, do that with your theme.

Given the technical implications, having [a WooCommerce expert](#) checking that up for you is probably your best call. As Nabeel reminds:

Finding out what's causing you issues with images is a complex task, as we'd need to identify several things before actually fixing it. In most cases I've seen, it has to do with a plugin conflicting within your WooCommerce store or the theme itself, which might have some CSS that's not properly configured.

Wrapping up

Images are important for any website for a number of well-known reasons. And when it comes to eCommerce stores, great, high-quality [product pictures become a business necessity](#) because they could mean higher or lower sales numbers.

If you think "it's just a picture", you're completely failing to see all the implications at a business level. If you don't make sure your store has everything it needs to showcase your product, the foundation of which relies on whether your image settings are properly set up, you're paving your store's future with low-quality business results.

And that would be the hardest issue to fix!

How To Fix Checkout Issues On WooCommerce

Have you ever had any problems while you were just about to finish an online purchase? I mean all was going smoothly, you had already put in your cart all the items you wanted to buy so badly, but then on that last page (the checkout page), nothing happened. Maybe you got an error message, you were unable to complete the payment, or worse, you felt confused because of a redirect to the homepage with no warning.

I'm sorry I had to recall that painful trace in your memory (consumerism, what have you done to us!), but that's what your customers experience when your checkout page in [your WooCommerce store](#) isn't working as it's supposed to do.

Given how critical are checkout pages for all store owners and managers, we've talked with WordPress developer and Codeable expert [Robin Scott](#) to explore some of the probable causes of a failing checkout and collected insights on how to go about them.

So, what do you need to do when your WooCommerce checkout isn't working properly? Let's see!

Think backward

The checkout page stopped working all of a sudden, or at least you've just been informed about that. Well, as with many other issues, the very first thing to do is to try to think or trace back what new actions, edits, and changes have been done recently. As Robin puts it:

The first thing to have a look at is how recently your checkout page stopped working and also what things you might have changed in between the time when it was working and when it stopped.

If you don't have records of past changes, or you actually didn't change anything, then the next thing to do is to start looking into your hosting setup.

Check your hosting configuration

Sometimes your checkout page could be misbehaving because your current hosting is somehow incompatible with WooCommerce. This could occur if you enabled a *force https* on your checkout where then the problem could be a result of a PHP setting gone wrong. As Robin suggests:

If you haven't changed anything your side, the first place I would look at is your hosting configuration. One of the best way to check that is to look in the WooCommerce status and see if you've got all ticks in there. Check if there's anything like a template out of date, or if it says something about your server configuration which is out of date. That's definitely something that we want to investigate.

To confirm whether it's your hosting the one causing the problem, you should be able to put a copy of your website onto the same hosting and then disable all plugins but WooCommerce and your current theme. Leaving just WooCommerce and using the standard WordPress 2017 as your theme, you can now put through a transaction and see what happens:

With the standard WordPress theme and just WooCommerce active, if you'll find your checkout page still not working we know there's something wrong with the hosting setup.

Next thing to investigate is your payment gateway.

Check your payment gateway

Which payment gateway are you using? If the payment service you have installed is not working, chances are that your checkout will also not function. At times the payment gateway could just not be showing up on the checkout page, or an authentication error occurs every time you try to check out. In the first scenario, your payment option probably just requires [an SSL certificate](#). In the second, there could be a problem with the connection between your store and your payment gateway service provider.

Specifically, you (or your developer) might [need to check](#) whether your server uses an unsupported encryption type because - to put it simply - not all secure is equal.

Explains Robin:

You server might be secure, but using "weak" encryption. This is like comparing Fort Knox with razor wire to your own double lock and burglar alarm. Ecommerce transactions require higher standards. And, if your hosting provider doesn't meet these criteria, you either need to move host or change your security setup.

Delve into Paypal Instant Payment Notifications

If you're one of the many using PayPal to collect money on your store, there's something that could negatively affect the completion of your checkout and that has to

do with [PayPal Instant Payment Notifications \(IPN\)](#). It's a bit more technical than usual, I know, but I'll try to make it as easy as possible.

There are two main things important for you to understand here:

1. If your server uses an unsupported encryption type (to PayPal), the Instant Payment Notifications might not work, hence your checkout won't complete.
2. If your server configuration is missing a specific "thing", which should be there instead, your checkout won't complete.

In Robin's words:

PayPal has a thing called IPN, which is Instant Payment Notifications. If your server uses an [unsupported encryption type](#), the Instant Payment Notifications might not work. Another thing that it could be the reason here is if either the [SoapClient or CURL](#) isn't installed on your server. You'll find that one in your WooCommerce status page: you'll see an X next to it saying that this is required for PayPal Instant Payment Notification.

Don't despair, though, if you've felt lost for a moment. Just be aware that there's [this plugin](#) to test whether your system is compatible with the current types of encryption accepted by PayPal.

Run a troubleshooting routine

If you haven't found anything yet that might be causing the issues with your checkout, then it's time to unroll the [typical WordPress troubleshooting routine](#). As you did it to check whether it was your hosting causing the issues, you should do the same to see if it's something related to one of your plugins or your theme.

Highlights Robin:

Once you know your server is configured correctly because you've put in a successful test transaction, the next step is to try isolating the issue to something in your plugins or your theme that's preventing checkouts from completing. Firstly, try by disabling your theme. If the transaction doesn't complete, it sure has to do with your current theme. If it doesn't, then it might be a plugin. The quick way to find out which one is turning on each plugin and then test your checkout each time.

By disabling and then enabling back your theme and each of the plugin you're using, you'll have better chances to pinpoint the faulty "element" raising the problems.

Wrapping Up

A failing checkout can cause your customers to leave abruptly from your WooCommerce store, taking their wallet elsewhere. An incomplete checkout leads to a disappointed potential customer. And if your store is experiencing issues, there are several areas you - and your developers - should look into like the one outlined above. As always, [hiring a WooCommerce specialist](#) is always the best solution when you have none available in-house or just want things to be fixed as quickly as possible. But there's a process could start doing right this second that will have a tremendous impact on how likely your store will have issues in the future. One action, which is more of an approach to better managing and running a website: [test everything before making it go live](#).

Chapter 5: Getting Help For Your Store

WooCommerce Specialist vs WordPress Developer

Running a WooCommerce store is already a difficult thing itself because on top of the standard business aspect you're required to take care of, there are lots of "moving parts" under the hood. And when something goes wrong, or you just need to [add a new custom feature](#), for example, you're also faced with another big challenge: should you hire a WordPress developer or you'd better off looking for a WooCommerce specialist? Which one is the most appropriate for your current needs?

This isn't an easy answer to provide, as the steps that contribute towards a successful project or a fixed issue can range widely from simple ones to those extremely complex. Therefore, the right person to address them can be different when scenarios change. So when things with your WooCommerce store aren't as they're supposed to be, is it a hard and fast rule to go for a WooCommerce specialist? Or, will any experienced WordPress developer do? Well, fret not! WordPress developer and Codeable expert [Shadi Manna](#) gives valuable insights to master the trade.

To make it more clear, I'll be developing and grouping the topic based on if's. Let's start!

If it's aesthetics and frontend work

WooCommerce is extensive and the work that goes into it varies as such. There could be simple changes that you wish to make like [selecting a different theme](#) or changing the positioning of some items. If that's your case, these tasks can be handled by a normal WordPress developer. The difference is further elaborated by Shadi who says:

Anything that has to do mostly with frontend such as templating, structure, CSS, HTML, JavaScript, or any other cosmetic changes, either a WordPress developer or a WooCommerce expert will be able to do help you. Anything which starts changing functionality, or adding functionality, or changing the way the flow of the ordering works, I believe you need a specialist.

Simply put, a WordPress developer can handle frontend customizations and successfully work on the structure/template of your WooCommerce store but when it

comes to other areas and elements, it is advisable to hire a WooCommerce specialist. Let's see why that is.

If it's functionality and backend work

A [WooCommerce specialist](#) is required for a number of reasons and these are usually related to backend technicalities of the platform. Shadi explains this by bringing it down to two main points:

There's a lot happening in the backend of a store and you need to understand the mechanics of what's happening within WooCommerce to be able to change it. So, if you're working with a developer who's not familiar with that, it's going to be very difficult for them to work on your store. On top of this, you should be aware, for example, that WooCommerce is moving away from the standard database of posts and post meta to their own database structures for optimizing the speed of WooCommerce. So you can't - or just shouldn't - use standard post functions from WordPress that are given today because tomorrow they won't work.

A standard WordPress developer might not be aware of these WooCommerce-specific functions or any other aspects that differentiate a WooCommerce store from a WordPress website right from the beginning. That's why for anything which is highly technical and functional, you should probably rely on a specialist. Otherwise, if things aren't properly coded and handled, it'll be like pulling the rug from under the feet (of your website) causing it to crash.

There are plenty of examples that show how WooCommerce is an ever-evolving piece of software that requires thorough understanding, one of which has to do with the [wp_update_postfunction](#). As Shadi explains:

There are many examples of bad practices, for example, with wp_update_post. As of today, you could use that on an order, post, or product because they are posts in the database. You could also update all their meta info with no problem. But I'm pretty sure in future versions this function won't work because it's going to be removed from the database of posts and put in its own database. When that happens, your code will then simply fail. There are at least tens - if not hundreds - of functions that should not be used.

Too technical? Don't worry. The moral here for you to understand is that you need to be aware of that a WooCommerce store comes with many specific functions that are constantly changing to improve. And if you need help with your store, you'd be better

hiring someone who's updated on these changes happening in WooCommerce in order to deliver you proper development work.

If your budget can't afford a WooCommerce specialist

There's no reason to sugar coat this because we're talking about business: hiring a WooCommerce specialist usually will require you a higher investment. Be honest, though: this can't come completely out of the blue to you, as it happens in any private and professional domain. If you need someone who's experienced within a specific environment, you'll be charged more because they'll provide greater benefit eventually. A WooCommerce specialist will cost you more but there's an effective way to being able to hire specialists. I'll explain.

When it comes to development work, specifically for bigger projects, or projects that require more resources from you, there's a proven approach that enabled thousands of business owners to afford WooCommerce specialists they had no idea they could do in the first place. Have you ever heard of "agile development"? [Maybe Minimum Viable Product](#) or MVP? Well, these notions strictly relate to the idea of iterative development, which could be the right solution to work with WooCommerce when your budget seems too tight. Highlights Shadi:

If hiring a WooCommerce specialist it's a question of money, I always advise my clients to use the iterative approach. If you can't build everything today, that's fine. Let's do a Minimal Viable Product first to make sure you're making money and your business runs smoothly. Then, once you need additional functionality, you need additional changes, we can create another iteration, another version of your store and add those changes.

In other words, by following this approach, you'll be to run a newer version (iteration) of your store while being already capable of making money so that you can reinvest it in additional changes and build up on top of each other (future iterations).

If you already know a really good WordPress developer

It's not that uncommon for many business owners to have already engaged and [successfully worked with great WordPress developers](#). So it's no surprise if you think of reaching out to your preferred developer and ask for some advanced WooCommerce-related work. And to an extent, it might even work for you but it's ultimately related to what you need to get done on your store how beneficial this solution can be. As Shadi stresses:

From a business perspective, I'd suggest to simply work with your preferred developer as much as possible but I'd also ask them to be honest about how far they can still provide value to you. At some stage, they'll reach a turning point about which they should inform you and explain that you'd need either to [bring in a WooCommerce specialist](#) into the project or completely move this part to a professional.

If you don't think proactively about your store's future

[Hiring a WordPress developer](#) today to save a few bucks might be more harmful than you imagine. If even it's the most prominent and grabs our attention at first, there are other resources besides money that are at stake:

I think a problematic issue that has a great impact on the business is the cleanup process. Trying to fix something after the store is launched, and backtrack, and try to fix it again, could be much more costly in terms of lost sales. Doing it after your store has been launched, and more custom-code has been added on top of that code, will be exponentially more expensive to you than simply investing in a specialist from the beginning.

Cleanup is a business expense and [taking care of things properly right from the start](#) it's one of the best ways to save on costs. The main reason being you won't have to pay again to get things fixed, even though you had already paid to have them in the first place:

I'd suggest sticking with someone who is building the right code from the beginning, because I've had to clean up other developers' issues and, to be honest, it costs clients way more. In terms of lost sales, and in terms of having to double pay for things that were done incorrectly, and now having to pay a specialist to fix it anyway.

It is a sensible choice to invest in a woocommerce specialist from the beginning instead of looking to save some money "here and there" and end up spending more time and efforts cleaning the mess afterward.

Cleanup is always a business expense and poorly managed time is always a cost you can't profit from.

Wrapping up

WordPress developers and WooCommerce specialists live into two distinct sets of the world where the overlaps are common. For a business owner, this intersection might be

difficult to understand on an operational level: "who should I hire to help me with my issue?", you might ask yourself.

As in many other aspects of our life, a specialist is the professional to reach out to for specific issues. Specifically, if you need help with your store hiring a WooCommerce specialist has multiple benefits that will allow you to save on maintenance costs, time, and will help you anticipate future issues in the long-run. That doesn't mean a WordPress developer can't be of help for your store. Anything that has to do with your theme, template, and frontend is completely up their alley.

How To Get An Accurate Estimate For Your WooCommerce Store

[Estimating the actual costs for a fix](#) or for [adding a new feature](#) to a WooCommerce store can be tricky and depend on a number of factors. As a client, you should be able to obtain clear figures on the amount of money that you'll need to budget for.

The clarity and accuracy of the estimate that you receive will be directly proportional to the attention to detail with which you describe the issue or feature that needs to be added.

So how can you get the perfect estimate for a fix or new feature for your WooCommerce store?

Enrich your brief with a test case

A test case helps the developer identify the ideal circumstances and what has gone wrong with the problem. It allows them to understand how much work will go into fixing the problem and, as a result, how much it will cost. WordPress developer and Codeable expert [Ashley Shaw](#) highlights this point:

If you want a solution to a specific feature or fix on your WooCommerce store, first of all, provide a test case. The test case should be very specific. The more information about the user journey to where the issue occurs, the better. If you're requesting a feature, on the other hand, you'd need to explain in just normal user terms what your desired feature should do and if some of the base functionality already exists on the site.

The more details the better

Details are everything. They are extremely crucial for developers to be able to tackle your request and give you an accurate estimate. And [when you're engaging with a developer](#) to help you fix an issue on your WooCommerce store, the more details you'll be able to provide them, the better your task will be understood.

So, for example, some additional information you should be adding can include the process that the user who encountered the issue followed, where exactly the problem did occur, a screenshot of the issue and what browser the person was using etc. Ashley further explains:

It doesn't have to be all in technical terms, you can simply describe what happened in plain English: 'They go to this page, they add this product to cart, but when they click this option - or whatever the case is - there's an issue with the checkout page when they're using a certain browser.'

Add screenshots with annotations to your brief

As you should already know, [screenshots are incredibly helpful](#) when communicating with developers. Ashley also stresses this:

Including screenshots for the fix or the required features can really help the developer's work, especially if they have annotations. You take a screenshot, and then you circle the problem area or the thing you like, and then you put in little comments on the screenshot. That always does give a visual indication. And I can't stress enough the importance of explaining what browser is being used when you're doing things.

There's never "enough" information when communicating with developers

When in need to [find the perfect developer through outsourcing platforms](#), it's common for the clients - you - to [publish their project requirements and brief](#) and then wait for the developers to start engaging with you.

Codeable only allows up to 5 developers estimating on the same project so you don't get overwhelmed by offers. It's a controlled environment in which you and the developers applying for your project can discuss your requests more in-depth.

As important as within your project brief, even during this pre-work chat and communication you should share the most information you can and answer questions coming your way.

To get a good estimate, keep in mind that the developers you're chatting with are keen to understand everything related to your WooCommerce project at their best. Ashley comments:

When you post the project on Codeable and you start getting questions from the various developers, hopefully, they will re-explain how they understand your user journey and then ask questions. Try to answer those questions proactively. More often than not, being explicit in your communication about what you need can help extending the brief that you've written because your understanding might not be technical. And we - the developers - might then mention technical aspects that we need to consider for achieving an accurate estimate. So, in my opinion, it's always important to be as verbose and explicit as possible.

Provide links to the actual sites or staging environment

There are a couple of things people miss out on when they're looking for estimates for a fix or a new feature for their WooCommerce store.

You probably won't believe it but this is true and one of the crucial elements clients miss probably because they're too busy focusing on the issue or the feature they want to have. As Ashley reminds:

There are many clients who don't provide the exact URLs to their actual site or their staging environment. So, very often I have to ask: 'Which site are you referring to?' I realize sometimes you're working on a development site or staging environment, but it doesn't prevent you from sharing a link: I can't work on something that I don't know about.

I can (partially) understand the client's little distraction because maybe they were caught up in the number of problems the issue had already caused to their WooCommerce store. But not providing the URLs you're referring to either when asking for an estimate for a fix or a new feature makes both lose time with back and forth communications.

Wrapping up

[Running a WooCommerce store](#) and business is already a major and tough task on its own. That's why you should make the process for getting a precise estimate for a fix or a new feature as smooth as possible.

At the end of the day, whatever the issue or feature your WooCommerce store might need, little room for misunderstanding and effective communication with the developer are always what is going to win the battle for you.

Benefits Of A Consultation

When we have little or even no idea at all on something, we google it. "How to..." has become our standard prefix to bring up what we hope could be the solution to what we see as our troublesome existence in that moment.

What are the usual results we get from that? Hundreds of articles and resources listing out things you should do. But what's the answer you were in need of? What's the right solution to your problem (not others')? Well, to get it you need to invest time to assess whether those links are reputable sources or not.

And even if they are, you'd still need to adapt them to your current situation. Wouldn't it be better to have a direct answer to your question right from the start? That's the value of [consulting with a WordPress professional](#).

But there's more to that...

Access to a breadth of experience

When it comes to your business, you don't want to lose nights and days to find a solution. You want an answer, an in-depth one, and as soon as possible, with no time wasted on inconclusive research. As WordPress developer and Codeable expert [Daniel Stanojevic](#) explains:

As with all things in life, the more you work with something the more experience you get. As Codeable experts, our in-depth knowledge and expertise allows us to advise you on the best path forward much better than random Google search results will. There is a huge difference in quality between various themes and plugins that all appear to do similar things at first glance, but then underneath differ in approach and implementation significantly. Just knowing these key differences can help save you a huge amount of stress and time down the road, once the time

comes to update your site. After all, you don't want a broken site from clicking update on the wrong plugin.

Freedom to focus on your needs rather than the technology required

Having a consultation with an expert in their fields provides you with something that very few things enable you in life: freedom. From a business perspective, it's the freedom to being able to focus all your time on what you really need to get done, on what issues you need to have fixed, all without having to waste resources on finding solutions. Besides that, having a 1-to-1 consultation with an experienced developer will allow you to foresee and anticipate further scenarios you haven't realized yet, like what's the [best way to maintain your website](#), which hosting can better serve your business and the like. Elaborates Daniel:

One of the best value out of a consultation is that you - the client - don't need to think about technology at all. I don't think you even need to think about WordPress at all. When talking with an expert, you're completely focused on what your business needs are and what you actually want to build and then whoever you consult with can advise you on the best path forward to get there.

Not having to think about all the technicalities and the next steps takes a heavy burden off your shoulders that was surely interfering with your commitment to [growing your business](#).

Opportunity to figure out if the idea is feasible for you

Before you begin treading down the path of launching, developing, or improving your website, it is important to ascertain if the project is worth the time, money and effort that'll go into it.

Having the possibility to directly engage with a developer through a consultation gives you just that, as Daniel highlights:

If your end goal is to launch a website, an application or get a better understanding of how feasible your idea is, [the discovery phase](#), in that case, will be getting confirmation from an expert that your idea is technically feasible, is doable, and makes sense from a professional's point of view other than your own. Through that process, you'll be able to uncover aspects you haven't thought of yet, gaining an enriched vision of what you'll need to focus on and, ultimately, your goals.

Better picture of deliverables, costs, and timeline for midsize and large projects

Expectations for a project vary from person to person. A professional and experienced WordPress developer knows, at first glance, the depths of things that might look simple to many [non-technical business owners](#).

When it comes to projects that are above \$2,000/\$2,500 - which are usually referred to as midsize projects - just having a list of project requests like "I need X to do Y" isn't going to make the cut for a successful project.

Don't raise your eyebrow yet. Let me explain.

Having a clear, informative, and as exhaustive as possible [project brief](#) is a must for any project. It's in your own interests to craft one up that embraces these standards. Why that? Well, you're free to have a sloppy one, one that required 15 minutes tops to be created, and then fire it away. "It's now all in the developer's hand", you think. No, it's not. If you do that, [you're setting your business on a very risky path](#). Your project brief is your insurance: if something isn't written there, you can't have it fixed, improved, changed.

And when there's more money involved in a project - don't forget: it's your money -, you'd need to have as many details as possible to be [part of the documentation](#) you and your developer agree on. These further investigations, details, and enhancements are generated through what it's called a "discovery phase". In Daniel's words:

Having a discovery phase on larger projects is an absolute must so that everything is clearly defined. I usually type out a project specification after a consultation if we're going that route. And I define in detail what exactly will be done, how it will be done, how long it will take, and then that's something I send to the client and they can review and let me know if they have any changes they want.

Comprehend and align your expectations with the process

[The discovery phase](#) is strongly suggested to empower you with even more information on your project.

A real-life example will help you see the value of it: as soon you grant access to your website, the developer starts noticing that the WooCommerce extension you'd like to

customize needs to be updated. On top of that, it also has some dependencies with other custom plugins, some of which need updates as well.

Now, the developer is stuck. Your project is on-hold.

If you had a discovery phase before the project kickoff, your developer would have highlighted those things and you'd have been able to make a decision in a stress-free manner.

Larger projects come with their own specific requirements and technicalities that, if ignored in the beginning, could cost you a lot of time and money at the end or mid-way through the project. Daniel further explains:

For larger projects, which can take weeks or even months, I insist on having a discovery phase. Usually, in my discovery phase, I like to work at least on some beginning wireframes for the project because a wireframe is an outline of how the page will function and what it'll contain. It's not about design, it's about knowing that this page will consist of the following things, and a user will be able to do A, B, C, and D. Without that, larger projects generally end up being a mess.

Wrapping up

[Working with outsourced developers](#) has proved to be a cost-effective way to run and grow businesses. Technology is breaking the barrier of location, making it easier for your business to attract and work with the best developers worldwide.

I understand, though, it could be a challenge for some because most of the work is done through written communication. Also, not being able to talk with the developer you just hired might seem "weird" or at least uncommon to some. That's why on Codeable you can have a 1-to-1 private call with one of our [pre-vetted developers](#).

Even for those that are more used to written communication, jumping on a call with a developer can be beneficial because, for example, an unintended and critical issue popped up or your project might require further analysis up front. Having a direct and preferred channel of communication with your developer goes well beyond your writing skills and time constraints.

At the end of the day, the possibilities with a consultation are endless but you get always the same beneficial outcome: one hour of quality information is worth many hours of aimlessly wandering and frantically googling.



The Quality Promise

What you can learn from over 60k successfully outsourced projects.

Download PDF >