

How to Avoid Getting Disappointed By Your Developer and Make Sure You Get What You Paid For (Even If You're Not a Technical Person)

Jean Paul

WPDevelopers.org

[Get in touch now!](#)

“Have you ever been disappointed by a developer?”

The road to hell is paved with failed coding projects. Many business owners risk and lose money, time, and energy on projects that simply don't work out. Thousands of dollars go to waste. And, you're still stuck with the problem that caused you to hire a developer in the first place.

This may have lead you to mistrust all developers, or even worse, to lose confidence in yourself. That's understandable, and for all intents and purposes, it's not your fault.

The thing is, there is an approach that helps you avoid costly mistakes. With this approach

- You'll know if you've got the right developer before the first line of code is written.
- You'll know where your project is going and what could potentially kill it.
- You'll also have a strategy that will show you how to go about your project in the most cost-effective, yet fast way.

Does that sound good? Let's get started!

Summary

This paper provides a complete solution to the problem of failed web development projects. It shows you how to set up working with your developer the right way. By the time you're done, you'll have complete confidence in what you're doing. Most importantly, you'll get your project off the ground so you can get back to other things you'd rather be doing.

The Problem

Here the most common issues you'll run through if you hire the wrong developer:

- in the middle of things, the developer tells you “I can't do it.”
- you run out of budget and the developer says “I need more money”
- the project keeps going on forever. You keep spending money and not seeing anything else for it
- you hire another developer to finish the 95% complete project. The new developer says “The only way to finish this is to start from scratch.”
- you get an OK project but any time you try to fix something, another thing breaks
- you either give up or start to wonder about the point of your existence
- your project fails in a humiliating spectacle (and you've already told people about it!)

The Cause of The Problem

The cause of the problem lies in one thing: starting to write too soon. After a short discussion, the developer jumps right into the code. They don't take a step back and say "You know what, rather than winging it, I think we need a script for this movie."

So, instead of having a Strategic Blueprint, they jump straight in front of the camera. At best, they work from milestones until they get stuck. There may also be a set of instructions, or a feature list. But, without enough digging by you and by your developer you will have a weak foundation.

It gets worse.

“What happens if your project fails?”

The most obvious answer seems to be you'll lose time and money. But the implications are much worse. For instance, if you've promised something to your employees or your customers, you'll break that promise.

Whatever problem the project is meant to solve will probably be worse. You may feel like your reputation is somewhat affected as well. The frustration and disappointment may start creeping into other relationships in your life. It's a downward spiral.

On the other hand...

“What happens if your project succeeds?”

You will have something you can be proud of that you can show to other people. You will solve the problem the code is meant to take care of. You will be able to focus on your business. **You will be the hero in everybody's eyes and rightly so.**

What's the secret to a successful project?

A successful project is like a good movie - it must have a good script. If the script is wack, no amount of special effects or high-profile acting can make up for it. That would explain so many box office flops.

“But, Jean Paul, what has that got to do with coding projects?”

Ah, I'm glad you asked.

The single most important factor that determines the success of your project is what happens right at the beginning.

For your project to succeed, you need to have the right “script”. I call this a Strategic Blueprint. It goes way beyond simply listing the features, mockups, and choosing your stack - although those things are part of it.

“Hmm... it does sound good but... what the heck is the Strategic Blueprint?”

The Strategic Blueprint and is an easy to grasp, simple to follow, clear guide for your project. It contains all that anyone needs to know about your project. This includes

- a clear description of the problem the project is solving for your business
- a clear description of what problem will be solved for your customers
- any “land mines” that could kill the project
- the most essential features the project needs
- nice-to-have features
- explanations for any concepts needed to bring the project to completion
 - and more (see below)

Yes, it does sound like work but...

...this can save you from pain and save you money, time, and energy

The best part is, most of the work will be done by the developer. You provide the information and they shape into something that makes sense to you. Here is how to do it.

“How do you create a Strategic Blueprint?”

The steps to creating your own Strategic Blueprint are simple. You will work on it in partnership with your developer. The goal here is to work together as a team towards a successful product.

Step 1 - Establish what problems are being solved

This is the most important step in the whole process. In fact, if you do this step alone you’ll be way ahead of the pack. But, if you ignore it, you will know deep inside that your final product could be better - even if it seems to work alright.

So, you want to list:

- *What problems does the final product solve for your business?*

This is where you discuss what the project will actually do for your business. You could have various goals, ranging from saving time to increasing profits, and everything in between.

For example, you're not happy with your current scheduling system. Having a new one will make it easier for you and your employees to track time and money.

This should increase productivity, reduce conflicts with clients, and eventually make more money.

- What problems does the final product solve for your customers?

Using our earlier example, you could say customers have been complaining about double bookings. They are getting frustrated and some are venting on online forums.

Your new system will give them a more pleasant experience.

You may also be struggling with bookings from mobile devices. Your new system will address those issues.

Remember! Write as much as you can. This is the core of your Strategic Blueprint and will ensure that everyone is on the same page. Don't worry about whether it sounds good or not. It's not a literary competition. You just want to bang out as much useful info as possible.

- What are the three most important things your final product must have?

This will let you know what the priorities are for the project - things you can't live without. In other words, these are the features/items/concepts that will make or break the projects. Of course, there could be more or less than three. But, three tends to be the sweet spot. **When budgeting, you must focus on getting these core features done.**

Example:

- It has to have an intuitive, easy to grasp interface
- You should be able to see at a glance what's going on in your admin dashboard
- The code must have checks to ensure no double bookings occur

Step 2 - Be clear on what the data is going to look like

This part will be mostly done by the developer, based on the info you provide. You will share a list of all the information your product must store. In a similar vein, you will also decide what the outputs should be.

This will help create the database schema, API endpoints, etc.

For example, you need the first and last name of each employee. You also need their employee number. You also need to know which bookings they are assigned to. (In reality, you will probably have way more data than that.)

Step 3 - Be clear on the workflow

You want to know what the workflow is going to be. You also need to decide what data goes on what screen. This is especially crucial when working for mobile devices.

For example, the user picks which service they want. The user then gets to pick a date and time for the booking. Finally, they add a note with any special requests they may have.

Step 4 - Start creating wireframes and mockups

Wireframes are a way to picture where each element will go on a page. This is not the design stage - you just want to work out how the system will work. Your designer can later take this stuff and make it look pretty.

More importantly, though, it will start showing you potential issues with your project. For example, you may realize that the booking process doesn't really flow on the small screen. You will also see if your workflow needs revising. This tends to be one of the hardest parts of any project. But, once you get past it, you will be way on your way towards success.

Once you're happy with your wireframes and workflow, you can start talking about pretty design.

Step 5 - Look at the technical aspects

Now that you have a clear picture of what needs to be done, your developer takes over for a bit. They will start looking at what solutions you'll need to make your project a reality. They will revise the data structures and review the database design.

Also, this is the time to figure out potential issues on the technical side. For instance, it may take some time to figure out how to smoothly integrate with other systems. Again, if you use two different systems, you have to make sure bookings don't overlap.

Depending on the complexity of your project, this is also the time to look at the architecture of your project. If you're building an API (you probably will) this is the time to design it as well. Possibly things like hosting will be looked at at this stage as well.

Step 6 - Finalize the development stack

At this stage, your developer will have a clear idea of what stack will be best for your project. You will know the frameworks and any other software/scripts your project depends on (dependencies).

If you have separate front and backend teams, they may collaborate on this. They can share what they need and what they expect from each other as well.

REMEMBER: your stack should be one of the last decisions not first!

Step 7 - Review your Strategic Blueprint and create your milestones

Now, you can look at all the information you've gathered. You have everything you need to create your milestones. Notice you're working from a complete set of data rather than hoping and guessing. Everybody knows what needs to be done.

Of course, make any changes you feel are necessary at this stage. And you're good to go!

“Wow, this looks good but... isn't that a lot of work?”

Yes, it is. Work and being smart is how you win. Without it, you won't get the best project you and your customers deserve.

“Ok, so what if I have questions or I need help with this?”

Well, just get in touch. You can reach me by clicking this link(<http://bit.ly/wpdevelopers-form>). I'm more than happy to help you get started. Feel free to ask any questions you have - and there's no obligation on your part. I'm excited to see how I can help you move your project forward. So, get in touch now!

In your service,

Jean Paul

WPDevelopers.org

[Get in touch now!](#)