



# 5 MISTAKES

COMPANIES MAKE

WHEN HIRING A  
**SOFTWARE**

DEVELOPER

[www.squaretree.com](http://www.squaretree.com)

You're getting ready to hire a software developer to create that new program for your business. It's a crucial step that can make or break your business -- so you have to get it right.

Unfortunately, too many people get it wrong.

They hire one developer, spend a lot of money on the project, and emerge with a product that's not ready for the market or still needs lots of tweaking -- and that means shelling out even more money to work.

Fortunately for you, there are a few key things to look out for to ensure that doesn't happen to you.

## **When you're out to hire a software developer, avoid these common mistakes to save you and your business time and money.**

### **1. Hiring a developer after only a brief consultation**

Ever heard of "scope creep?" You will if you hire a developer who quotes you a price after asking only a few short questions.

A development project is a complex undertaking, and developers can't possibly get a quote right after asking you just a few short questions. If your prospective developer asks just a few short questions and then quotes you a price and names a delivery date, you can bet they don't really understand what you need. Most likely, they are more interested in getting you on the hook so they can keep billing you for the project in perpetuity.

An experienced developer, on the other hand, will help you write a goal for the project. From there, they'll help you create a detailed, comprehensive list of functions the program will accomplish to reach that goal. A quality developer might give you a ballpark estimate of what it will cost based on that list -- but here's something really important to understand: A true understanding of the depth of the program details will require significant effort on the part of the developer, and that's typically a billable job.

Taking the time to understand the project is not the same thing as scope creep. Instead, think of it as a discovery process that will get you the most functional product possible. Software development projects are filled with discoveries. As you move through the process and see new screens, reports, and real data, you'll find that assumptions you first made don't actually work like you'd hoped. There may even be better ideas that come out that you then want to incorporate. The Agile process, done correctly, embraces discovery and change and manages it throughout the process. In an Agile process time and cost are fixed and scope is flexible. You build the most important and valuable features first and bring in the bells and whistles as time and budget allows. This allows you to use the basic and most valuable parts of the program sooner.

## **2. Choosing a developer based on the lowest price**

The price tag may seem low at the start, but it's frequently not the lowest-cost in the end. Many developers will quote a price, spend all the cash, and then tell you the project is "almost done" once they've run out of funds. You give more money, they go over budget again, and the cycle repeats itself. It's not an accident; many lowest-price estimate providers depend on that cycle to stay in business.

Instead of thinking of things in terms of price, think in terms of value. If you can find enough value in a project, then price becomes nearly irrelevant, except as a cash flow issue. If your project could attract or retain customers by adding value to the relationship through web based portals, the return could be huge.

Bottom line: Make sure the development company you choose has a proven track record of success, and worry more about success and value than cost.

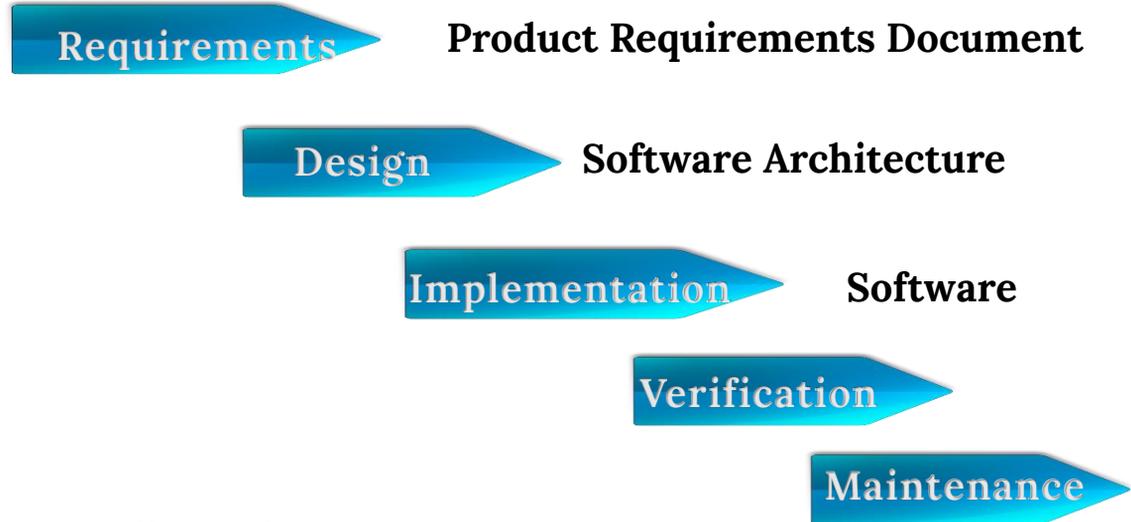
## **3. Choosing a developer based on low hourly rates**

We've all seen the articles and advertisements for off-shore programming at very low hourly rates. I have personally tried them and here is what I found. If the rate is 25% of the cost of a programmer in the U.S., expect the estimate to be take four times as long as it would for the on-shore programmer. I don't know how it happens, but in the end you are not getting the deal you think you are.

The other problem: Communication. You will typically be talking to a project manager who is not a programmer who then communicates your needs to a programmer who then interprets them and writes what he/she thinks they heard. Think of the children's game "Telephone," where you line up the kids and whisper in their ear a phrase. They each repeat it "exactly" into the ear of their neighbor, and by the time it passes through just a few kids the message is very garbled. It's the same for business communication.

## **4. Choosing a developer who doesn't follow standard processes**

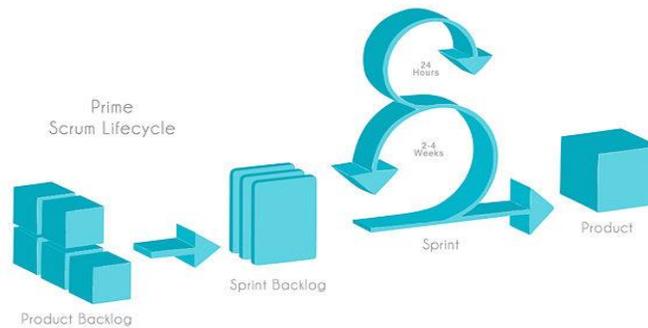
Every developer has his or her own process. Some follow the standardized process, and some follow the "cowboy" method. With the latter, the cowboy will listen a little, create some code, and then skip over the peer-review process. The only testing involved is the testing the customer does. Cowboys may be great coders, but without a peer review, there are often big holes in their approach. Professional developers, meanwhile, use one of two crucial processes to ensure the quality of your product: The Waterfall Development Process, or the Agile Scrum process.



## Waterfall Development Process

**With the Waterfall Development Process, there are five main steps:**

- a. Requirements definition. One or more documents are created to define what the system needs to do. This document is the bible for development, and changes due to discoveries in the process are discouraged.
- b. The design process begins and the architecture of the project is created.
- c. Implementation, in which the code is actually written.
- d. Verification starts with testing by the development team followed by testing by the customer. In a strict waterfall, this is the first time the customer sees the actual program they say they wanted. With this approach, customers are often disappointed with the outcome, because foresight during the requirement stage is never 20/20.
- e. Maintenance, including ongoing changes, enhancements and bug or anomaly repair



## Agile Scrum (for use with programming teams)

### With the Agile Scrum process, there are six steps:

- a. **Product Backlog.** Create a product backlog which lists all the program functions. Depending on the size and complexity of the project, it can be as extensive as a waterfall requirements document but it will be organized in potentially-deliverable pieces and prioritized with the most valuable pieces first. Perhaps most importantly, it's subject to change and re-prioritization throughout the project.
- b. **Sprint Planning:** Take the top backlog items that can be accomplished in a timebox period of time and recast them as a sprint backlog. The sprint is the amount of time to work on these functions and the result should be a fully tested and functional part of the program. The timebox can be between two and four weeks.
- c. **Sprint.** A sprint begins and the team works on the items they've committed to delivering to the customer. Each day there's a check-in to share what the team has accomplished and to set new goals for the following day.
- d. **Sprint Review.** At the end of the two-week period the team shows the client the results of the sprint and the client gives feedback.
- e. **Retrospective.** The team discusses what went right, what worked not-so-well and creates action items for the next sprint.
- f. If a release is warranted the code is handed off to a release team and a new sprint planning meeting is started. If a release is not warranted then the team starts a new sprint planning meeting.

Both of these methodologies have their benefits, and modifications can happen based on the size of the team and the complexity of the project. The most important thing to know is that your developer has a process they can describe to you in detail, that you can understand it, and that the team actually follows it.

## 5. Giving the developer a big payment up front

If you encounter a developer who wants a big paycheck up front, take it as a red flag. The developer doesn't need to invest a lot of money in equipment in order to get started -- they simply have to start working.

If they're asking you for a lot of money up front, it's likely that the company or the developer is having cash flow issues -- which could mean that they won't be able to finish what they started. A solid company on good financial footing, meanwhile, will start work and invoice you on performance goals, or better, on work accomplished. With that method, you'll get to see what you've gotten before you pay for it.

One thing to note though: As your team works on architecture, it might be difficult to see or understand what they're doing. After that initial stage, however, you'll have a better ability to judge the work.

Hiring a software development team is a big step, and it should be clear by now that you have a lot to think about when you're hiring one. But with some knowledge of how to root out quality developers and separate them from the bad ones, you'll be destined for success.

### Testimonial

"I just wanted to send a note to express my appreciation of the job you did revamping our web site. It was an excellent job and your staff was very responsive and easy to work with. I look forward to working on additional projects with you and your team."

**Doug Hawkins**  
*Plastic Package*



**Squaretree** is a client-oriented company and focuses on ensuring that each professional assigned to a project possesses the exact skills and experience required.

We have a solution-oriented approach. Our commitment to results has earned the trust and respect of its clients, and has developed a solid reputation for expertise and experience in mission-critical application development.