Pick the Right Programmer

How to select the right developer to build your custom application

By:
Avonelle Lovhaug
Avonelle@CodePoetrySoftware.com
www.CodePoetrySoftware.com

Last revised on: 6/26/2010

code poetry

custom software for your ^ business
astute

You've decided your organization needs some custom software. Great! The only challenge now is to select the right person to design and build your software.

If you are like many business people, you aren't overly knowledgeable about technology. That can make the process of hiring someone to write a custom application daunting. How do you select someone to do something you don't necessarily know how to do yourself?

Fortunately, it is not necessary to be a tech guru to hire a programmer. This article contains some selection criteria that will help you to select the right developer for your project. (Note: This document is targeted at organizations that are looking to hire solo practitioners, but it can also apply more broadly.)

1 **Someone you can talk to.** Sometimes business people are impressed with geeks who spout technical jargon. However, tech-bloviating is not a sign of genius. In fact, this may mean your candidate is not well-rounded, and will be difficult to communicate with in the future.

Also, all programmers are not introverts, so don't assume that this comes with the territory. You should hire someone who can converse with you without prompting. In fact, this is an absolute requirement if the programmer will also be designing your application, because they will need to be able to draw information from you and potential end users.

2 **Look for a professional.** I know it is tempting to hire your cousin's son Carl (fresh out of college) who thinks he can whip up your new web application in just a few days. After all, it will help him out, and he is available at popular prices. While it is possible this will not be a disaster, there are some reasons why this may not be as economical as you think. Experienced and talented software developers have learned the hard way the tricks and tools of the trade. They spend their days thinking about and writing code, and the more code they write, the better they get. Also, mature practitioners know about the critical tools that make a professional developer successful (and often aren't discussed in school):

> *When Carl installs a new version of your app that breaks an important feature for your customers, you'll be sorry if you can't revert to a previous version.*

- ✓ **Source code version control software.** Believe me – this is vital. When Carl installs a new version of your app that breaks an important feature for your customers, you'll be sorry if you can't revert to a previous version. In addition, this often serves as a back-up of the source code.

- ✓ **Issue tracking.** It is amazing how many details (big and small) you need to track for even the simplest applications. An issue tracking database allows the developer to track your requests for changes and bug reports, and lets you see the progress made. For very small applications you can use a spreadsheet or Word document, but you will quickly find that a database designed for this purpose is a much better alternative.

- **Code generation tools.** Experienced developers know that it is wasteful to write the same code over and over again. Code generation allows developers to write code that writes code, leading to more consistent source code and better use of their time.

- **Build management tools.** As applications get more complicated, creating a repeatable build also becomes more challenging. Seasoned developers know this and plan for it. They make sure enough time has been set aside in the schedule for developing a good build process, and they use build management tools to automate the process so that it is consistent.

- **Testing tools.** There is no substitute for real end-users testing your application. However, there are a lot of tools available to developers for streamlining and automating the testing process to minimize the time your end-users will need to spend testing.

3 **Fits your corporate culture.** Is your corporate culture laid back? Then selecting a developer who seems to be caffeine overloaded may not make sense. Will the application be used by people who are sticklers for details? Stay clear of the dude who misspelled his last company's name on his résumé. Your project has a much greater chance of success if you are working with an individual who has characteristics similar to your target user, or at least demonstrates an ability to adapt.

4 **Get the full picture.** When we had an addition built onto our house, we assumed the contractor would take care of everything. As it turns out, some things like staining and painting the interior were not included. This may also be true when working with a solo developer. Here are some things to check on so that you are comparing bids fairly:

- What software licenses will be required and who will pay for them?
- Who will be responsible for server setup (if required)?
- Are meetings with you and your users included or extra?
- Who will be hosting your app (if required), and how much with it cost?

If these items are not included in their bid, you'll need to figure out who will be responsible for these items and how.

5 **The right stuff.** One of the biggest challenges for business people who need to hire technology people is that the dizzying array of technology acronyms and words quickly becomes a sea of meaningless alphabet soup. Fortunately, understanding a few key things will help take you a long way in the process.

Here are the things that typically affect the language and technology selections:

- Operating systems (server and client). For example, the Visual Basic 6.0 runtime cannot be installed on Linux.
- *Developer preference.*

Yes, you read that right. The selection of technology and language is, for the most part, a function of the preference of the developer.

This doesn't mean that all languages and technologies are created equal. In fact, each has positive and negative attributes, and may work better or worse for your specific solution. But there are probably several options that could be successful.

Still, you need to ensure that the selected technology works on the target system's hardware and operating system, and is supportable in the future. Otherwise you may end up with a system that can only be supported by the original developer, because the selected technology is too obscure or old that no one uses it anymore.

> *The selection of technology and language is, for the most part, a function of the preference of the developer.*

To accomplish this, you need to know the answers to the following questions:

- ✓ What operating system will be installed on the computers of your application's users?
- ✓ If this is a web application or client-server application, what operating system will the server run (if known)?

Then, make sure the bids from prospective developers include information about the technology and language they will use. You will want assurances from the developer that the selected technology and language will work with the operating systems you have specified. And finally, you'll need to know how old the technology/language is. In this case, a Goldilocks answer is preferred: something neither too new nor too old. New technologies that were just released (or still in beta) are riskier than ones that have been around longer, because they haven't been proven in as many real-world scenarios. However, significantly older technologies may lose support from their vendor sooner, and may have fewer knowledgeable developers who can effectively maintain their apps.

A good solo practitioner should be able to explain their technology choices and assuage your concerns.

6 **Experience.** If you were selecting someone to build your dream home, you would certainly evaluate contractors on the basis of homes they had built previously. Similarly, you should evaluate prospective developers in part on their previous experience in building software similar to your project. While their experience certainly doesn't have to be exact, there are two areas worth considering:

- ✓ *Number of users.* Many developers don't have experience building high capacity applications. If your application requires this, then look for developers who have this expertise.
- ✓ *Type of users.* Some developers have only created applications for internal users. If your project will be used by customers, then they may not be a good fit. Look for a developer that has experience building applications for the type of users your project will have.

If you consider these selection criteria, you will increase your chances of a successful custom software project.