Sometimes the ‘cheap’ option in a software project can end up being a lot more expensive, says Tim Fellows

Comment

“if you ask me...”

HOW TO AVOID GETTING SUCKED INTO THE BLACK HOLE OF SOFTWARE DEVELOPMENT

I WORK in software development for process control and test. In the last nine months I have encountered the same worrying scenario four times. A company has a new project involving a relatively complex system. They decide to get the software written by a programmer (either an employee or a contractor) who has only a little experience. It takes a long time to complete and they have to draft in extra help from another programmer or two. Everyone is under a lot of stress but eventually it gets completed.

In one instance the stress resulted in a manager suffering serious health problems that have subsided now that the project is over. In another, the company has decided to stop making systems entirely because it got so bad. In addition, I suspect that as we progress a lot of support will be required for poorly written code.

Software development can easily become a black hole like this, with serious consequences. A poor programmer can easily take five or ten times as long as a good one – and produce a worse result at the end. The good programmer has, by experience, figured out techniques that work well and save a lot of time and heartache.

Now, I know software is a bit of a dark art to most people. The programmer starts talking about how his or her code works and your eyes glaze over, right? For this reason, and because the program will always be one that has never been written before, it is hard for managers to know whether the programmers are working at a good rate or not.

There is also a temptation that can afflict contractors and sometimes employees as well – the desire to spin the project out a little longer. This may keep the contractor in work until their next contract begins. But there are some who still like to work to a fixed price. There are huge advantages with this in that it gives you the security that it will not be the never-ending project. Although it means defining the work in some detail at the start, which can be frustrating for people who want to dive in, it focuses the mind and can avoid a lot of unforeseen problems and lost time later. Can every project be defined like this at the start? No, but even if not, it is usually possible to define a significant portion of the work and get a fixed-price quote for that, then, at the end of that phase, define the next and get another fixed-price quote and so on until the work is finished.

Are there any disadvantages of fixed-price projects? There can sometimes be a danger that the software provider turns out to be incompetent or awkward in some way, but you are stuck with them because you have placed an order with them. Software providers can also charge high prices for changes to the specification – a bit like equipment suppliers charging a lot for spares. There are several ways to avoid these problems. Firstly, at the start, you should discuss the work in detail with the company concerned. It will quickly become clear whether they grasp what you are trying to do and whether they care about the detail. You will also get a feel about whether they will be easy to work with.

Make sure that you talk with the actual programmers rather than someone who is just a front for the company. If, after this, you are still not sure, you should start them off on a small project or a small chunk of a larger project.

Another consideration – particularly if it is going to be a big program – is what will happen if the programmer becomes unavailable for some reason. This is significantly more likely if the programmer is an employee of your company, simply because employees move on from time to time. If there is a change of personnel, there is a significant learning curve for a new programmer when taking over someone else’s program. Your best bet is to try to find someone who you sense would like to support you long-term.

Tim Fellows has programmed full time in LabVIEW for nearly 20 years and is a certified LabVIEW developer. He works as a consultant, mostly for manufacturing companies, and can be found at www.livewires-automation.co.uk