

Should you outsource your test software needs?

Patrick Kelly

LabVIEWexpert.com

February 5, 2002



Introduction

When it comes to Automated Test Equipment software or laboratory automation, you have many of alternatives. You could do any of the following:

1. Hire a full-time test engineer
2. Borrow an engineer from another department
3. Use a technician to manually test
4. Train a technician to write software
5. Hire an intern to write software
6. Outsource your test software requirements

Let's explore the pros and cons of each alternative.

1. Hire a full-time test engineer

If you have a lot of testing to do, and you think the time to design and develop the software is going to take over six months, then hiring someone full-time is a good alternative to consider.

Cost: The typical cost for a senior-level test engineer will be between \$80-\$120K per year in salary (depending on experience level). Benefits and overhead costs (including 401K, medical, payroll taxes, vacation and sick time, etc.) are typically about 15% of the salary. You should also include the associated company costs of hiring the individual. (These costs include such things as recruiting, interviewing, training, office equipment and software, etc.)

Pros: If you can find the right person and there are a lot of things to automate, this route can be very cost-effective. The engineer is available full-time to assist when things go wrong.

Cons: Test engineering is usually not the kind of activity that gets an engineer excited. Because of this, many engineers want to move out of the test department as soon as they can. Every company has a core competency, and most engineers want to be involved in the design of that core competency. For this reason, it is often hard to keep the best engineers in the test department. It is very common for an engineer to be promoted out of the test department at just the point when their training is starting to pay off. This means that many test departments are filled with a hodge-podge of partially written software modules, in which some pieces of code are well-written but many parts are spaghetti code.

2. Borrow an engineer from another department

Often, there is an engineer inside your company who could be a tremendous help on your project. If you have a small project that only needs a few hours of work per week (or maybe just a couple days of total development), then this option may be right for you.

Cost: In this case, you are working with another department and a current employee, so the cost is not something that you'll think about. However, if you want to benchmark the cost, you should use the same calculation in the above scenario of a full-time employee, and then pro-rate based on the length of the project.

Pros: If your project is not big, and you can find the right person inside the company, then this is an excellent option.

Cons: Often it is difficult to get the borrowed engineer's full focus on your project. And there may be political issues involved in getting the engineer. Also, if the project is expected to take more than a couple of days, or if it involves technology that the borrowed engineer is not an expert with, then this is not a good option. Finally, keep in mind the maintenance needs. Will the borrowed engineer be available to assist you in debugging the software?

3. Use a technician to manually test

Sometimes your tests just are not that complicated, and you don't need to test very often. Perhaps you already have a well-trained technician who could do the testing for you. If this describes your situation, then using a technician to manually test might be your best alternative.

Cost: In this case, you are working with a current employee, so the cost may not be something that you will think about. However, if you want to benchmark the cost, you should use the same calculation from **Alternative #1** of a full-time employee, and then pro-rate based on the length of the project. Include in the cost your time to train the technician and consider the scenario where the technician leaves the company or is out on vacation.

Pros: If the test doesn't take a long time, and you can find the right person inside the company, then this is an excellent option.

Cons: If the test is really this easy, then it is probably very easy to automate. Typically this scenario is the type that LabVIEWexpert.com could automate for less than \$2,000. If the costs associated with the technician are higher than this, then outsourcing may be an even better alternative than using the technician.

4. Train a technician to write software

This alternative is probably the worst idea of the six, but it is mentioned because many companies try it. In all of the companies where we have worked, this solution has never given the results that management had hoped. There are several simple reasons why (discussed in the cons).

Cost: In this case, you are working with a current employee, so the cost may not be something that you will think about. However, if you want to benchmark the cost, you should use the same calculation from **Alternative #1** of a full-time employee, and then pro-rate based on the length of the project. Include in the cost the time to train the technician and the time that the technician spends writing code. (Realistically, you should also expect that 95% of all code written will not be reusable. You should include the cost of having to rewrite code.)

Pros: This seems like a cheap solution. Unfortunately, this is a perfect example of “you get what you pay for”.

Cons: Software is not easy to write well. (That is why software engineers spend four years or more in college and several more years working on real projects before their code is well-written.) After you spend the money to train the technician, and once he or she writes the software, the result will often be spaghetti code which is not scalable. It is almost guaranteed that you will spend more time and money on this solution than you would if you outsourced.

5. Hire an intern to write software

This alternative at best is a little better than **Alternative #4**. It is also mentioned because many companies try it. This solution rarely gives the results that management hopes, and it is made worse by the fact that many interns don't return to the company after the internship.

Cost: You are hiring a college student, so the costs seem to be a bargain. You may be paying anywhere between \$12 and \$25 per hour. The hidden costs are in the supervisory time and the high probability that this student may not work for your company after the internship. Include in the cost the time to train the intern and all of the time spent by other employees supervising or mentoring. (Realistically, you should also expect that 50+% of all code written will not be reusable. You should include the cost of having to rewrite code.)

Pros: If you can find the right person, this solution can be a good bargain.

Cons: Most often, the software written will be spaghetti code, which is not scalable. If the intern doesn't come back to work for your company, someone else will have to maintain the code. If the intern does come back, he will often work in a different department, meaning that any future improvements to the test software will be made by someone else. In either case, if the code is not reusable and scalable, you'll spend more money rewriting the software.

6. Outsource your test software requirements

The final alternative is to bring in an expert from the outside. You get to benefit from the experience of the consultant, without paying the costs to have him/her on staff full-time. When the project is done, you no longer have to pay the consultant; however, you have the benefit of being able to use him/her for maintenance or upgrades in the future.

Cost: If you are hiring an expert, expect to pay between \$75 and \$200 per hour. Depending on the contract, you will either pay a fixed price for the whole project or pay hourly. The consultant should give you a detailed proposal with approximate costs. (NOTE: The price per hour does not always indicate the quality of the consultant, but it can be a good indicator.)

Pros: If you can find the right company to use for outsourcing, this solution is very often the best solution. You only pay for the project expense, and don't have the liabilities of a full-time person. You get maintenance, proper documentation, and expert advice.

Cons: The hardest part of outsourcing is choosing the right company. Many outsourced projects don't work out (the result of either picking the wrong outsourcing partner or not managing the outsourcing partner well). See the "About Us" section on LabVIEWexpert.com for a discussion of the pros and cons of different outsourcing options.

A Note on Creative Work

Study after study has shown that the best software engineers are between 10 to 100 times more efficient than average software engineers. This is due to the fact that software engineering is both a rigorous discipline requiring many years of training, and it is a creative endeavor, requiring imagination and abstract thinking. If you believe the studies, and the idea of "10 times more efficient", then let us explore the implications.

Imagine that you have an assembly line, and you have ten workers on the line. Your best worker may be 25% more efficient than your worst. Perhaps he or she is as high as 50% more efficient; rarely will there be a higher difference than this.

Now imagine that you have two software engineers. One is excellent and is ten times (1000%) more efficient than the other engineer. When you consider their salaries, the difference may be 25% or even as high as 50%. Never will the excellent engineer make 1000% more than the average engineer.

Keeping this in mind, the biggest benefit of outsourcing is that when you find the right company, you will be working with engineers that are 1000% more efficient than your other alternatives. The good news is that you won't pay 1000% more. You might even pay less.

Final Word

At LabVIEWexpert.com, we have worked with over 200 companies in San Diego county. We have seen outsourcing work wonders, and we have heard customer stories of outsourcing failures. When all is said and done, the reasons for outsourcing success are the same as those for employee success: great people. We at LabVIEWexpert.com are proud to say that we have never had a customer who was not satisfied at the end of a project. We pride ourselves on our list of references, and that we can refer a prospective customer to 100% of our previous customers.

The mission of LabVIEWexpert.com is to be the source for Instrument Automation Software Design that provides the best value (among all alternatives) for your project. We will not have the cheapest rate per hour; however, the total cost of using us will often be less than all other alternatives. Sometimes we are not the best alternative. Often, we are. Only by assessing your current situation and the alternatives available to you will you be able to determine whether LabVIEWexpert.com is the right choice.

What follows are three short case studies of outsourcing success.

Case Study 1: A startup Biomedical company

The company needed to develop a prototype diagnostic instrument. They needed it quickly and on a tight budget. LabVIEWexpert.com worked with the customer and developed a software package that far exceeded their expectations. Approximate cost was \$50K, spread out over a 12-month period. (Hiring a full-time person for the same budget would be equivalent to a \$35K salary.)

Case Study 2: A major Telecom company

The company had an important need for scalable software, optimized for test time. They needed a senior software engineer with a strong background in test automation. The engineer had to understand cell phone technology, signal processing, telecom GPIB instruments, and be excellent at software architecture. The project would ramp up and ramp down quickly, and there was a need for long-term maintenance.

LabVIEWexpert.com was the perfect choice for both cost and experience. Finding the right person was critical; however, the project didn't justify a full-time position.

LabVIEWexpert.com gave the customer a cost-effective solution, solving the short term needs of developing the software, and the long term needs of maintenance on the software.

Case Study 3: A mid-sized Industrial Equipment company

The company had a two-man-month project that involved lots of thermocouples, pressure transducers, and other equipment. They decided to outsource and initially picked a consultant who was not right for the job. The project was several months past its deadline when LabVIEWexpert.com was called in to help.

We worked closely with the customer to define everything, from the GUI screens to the details of what the software needed to do. Within two days, we had a well-defined

functional specification, and began design and integration. We finished the project within three weeks (two weeks ahead of schedule) and dazzled the customer.

Summary

When it comes to Automated Test Equipment software or laboratory automation, you have many of alternatives. The one you pick depends on your particular situation. We at LabVIEWexpert.com know that outsourcing is a good alternative, but it is not always the best. If you determine that outsourcing your test development is right for you, then we hope that you will consider our services.



Email: Patrick@LabVIEWexpert.com

Copyright 2002 LabVIEWexpert.com. All rights reserved.
Product and company names listed are trademarks or trade names of their respective companies.