



This article has been contributed by Nancy Smyth, CQA; Sunburst Software Solutions, Inc. and Managing Editor IDNAC Newsletter **August, 2006**

## *The Evolution of the Role of the ProAdvisor: From Bookkeeping & QuickBooks Training to Working With, and Becoming, A Developer - Part III*

In the previous two articles I've discussed the evolution of the ProAdvisors' role. **Part I** – discovering the “need” for automation, then the “search” for that automation, and the time-consuming “work-a rounds” that we develop in order to obtain “some” automation. In **Part II** we delved into what it is like to work with a programmer (aka Developer) and my own evolution from Bookkeeper/ProAdvisor to Developer. If you've missed either article, they are available in the [Newsletter Archive](#).

This month I'd like to focus on just “what it takes” to create a QuickBooks compatible add-on, because it's not quite as simple as one would think. I know that I had no real clue as to all that was involved.....

### *What is involved in the creation of a QuickBooks compatible add-on..*

Previously I mentioned the amount of hours that we spent developing just one of our QuickBooks compatible add-ons. In case you missed that installment, a quick recap:

It took Ben and I each – 8 months of working 30-40 hours a week (we each had a full-time job, in addition to this project) to develop our **Certified Payroll Program**. We started the development process in August of 2000 and made our first sale in March 2001. That was approximately 36 weeks with 1080-1440 hours each, or a total of 2160 – 2880 hours without seeing one penny in revenue. Even at a modest rate of \$35.00 per hour, that is a total of \$75,600.00 to \$100,800.00 in “unbillable time.”

I'm sure many of you are asking “why did it take so long?” which is a legitimate question if you have never been involved in a project like this. I won't bore you with the details, just a few highlights.

#### **1. Choosing a programming language and development tools**

The programmer will have a “programming language” of choice, one that they've worked with and feel comfortable with. There are several different programming languages to choose from: C+, C++, Java, Microsoft Visual Basic, and Microsoft .NET to name just a few.

First and foremost on the list is the “specs” for the program or what the program has to accomplish when it is completed – this is the “black and white” end result.

You have to determine what information you can get from the QuickBooks data file, “where” that information comes from, and what “pieces of missing information” your program is going to have to hold and “how” your program will store it, how to “merge” the information for it's final format, and what you will use to create that final output. From this point you have to work backwards and thus begin the “fuzzy logic,” in which things don't always have a natural order.

For example; we knew that we could always get specific pieces of information from QuickBooks regardless of what state the customer resided in. We also knew that we would use Microsoft Excel and Word to produce the final certified payroll report and Statement of Compliance.

So we had our beginning and our end result, and these were the “easy” pieces of the process. There is so much information required on these types of reports that QuickBooks simply has no means to hold, in addition, we had 22 sets of “variables” to work with, and the “design” of the program became more defined.

Once the specs were more defined and the lists of “we can get this from QuickBooks” and “our program has to hold this additional information” in a linked record were complete; it was now time to really put that “fuzzy logic” to work and write the “code” to make the program do everything it needs to do. Does it sound simple yet, or far from it?

Most developers will get this far, and depending upon what the specs are for the program in question; they will begin asking questions such as:

- What do you think I can sell this product for?
- How many products do you think I can sell?
- How many of these products can a ProAdvisor sell?
- How can I, or another ProAdvisor, market this product?

These are just some of the questions that may be asked by a developer when they are approached to create a QuickBooks compatible application, and as a Bookkeepers/ProAdvisor we may not appreciate being asked these questions, but they are legitimate questions and frankly you should expect them. It is your obligation to answer them as honestly as possible when they ask you what you think the product could sell for, how many you think could be sold, how many products you could sell, and how the product could be marketed. Realize that the Developer is not looking for a “personal guarantee” from you; they are just seeking information before making any sort of commitment to a project.

Again I'll stress that you need to realize that developer's aren't going to necessarily “jump” at the chance to develop a product that they are not sure will have a resale value. It is important that everyone realizes the time and effort that goes into developing a product before the first sale is ever made. Answer their questions as honestly as possible without being offended (I know – tough one), and feel free to ask them questions ... communication from the beginning is key.

Ok, so suppose the Developer decides to proceed with the project, what's next?

## **2. The project begins, the developer writes the code, and a “beta” version is released.**

Once the initial code has been written and compiled into a “Beta or Test” version there are hours of:

- Testing
- Debugging (finding errors)
- Fixing errors in the code
- More testing and debugging (to verify that the problems had been fixed)
- More fixing errors in the code
- Tweaking this and tweaking that

## **3. The initial testing is completed, but what about documentation?**

Whew! You might think, now we are done; right? Wrong! There's far more to do yet.

- There's documentation – manuals to teach people “how” to use the program.
- Is a printed manual enough? No, this is software...there **should be** “In-Program Help”.
- Is every QuickBooks user going to use QuickBooks to its fullest potential? No! So you need to document any QuickBooks required setup.
- The documentation needs to be tested to be sure it “makes sense” to the new or average QuickBooks user.
- Revisions, revisions, revisions.....

#### **4. Testing and documentation is complete, let's sell the product! But wait, there is still more things to take into consideration.....**

FINALLY, the testing and documentation are done, and you have a "finished product" to sell. Right?

**Wrong!** There are still many things to do and consider:

- Will user's read the manual? Chances are very good that they won't - so you need to provide "visual" training – perhaps through the use of PowerPoint Presentations or Flash Demonstrations.
- Will people want to purchase something before trying it out? Probably not; so you need to come up with a Free Trial or a Free Demonstration.
- Will they want to use their data file or have you provide one for them?
- Do you have the resources required to provide this type of information? Whether it be spending hours yourself to come up with these additional training materials or hiring someone to do them for you.
- Have you accounted for absolutely, positively, EVERY possible QuickBooks setup configuration? Probably not (and if someone tells you yes.....well, don't believe them....because we all know better!)
- Have you found absolutely positively EVERY bug in the program? Probably not, and again, if someone tells you yes.....don't believe them.
- Will your product install and work in the same manner on every possible type of computer configuration? Maybe, maybe not.....who can tell.
- How are you going to sell the product?
- How are you going to market the product?
- How are you going to protect your product (code) from being hacked (stolen) by someone else?
- Will you even be able to sell just one application?
- How many applications will you sell?
- Who is your target market and how do you reach them?
- How do I handle payments?
- Do I want to accept credit cards and pay the high fees?
- The list is just endless!

#### **5. Sales, Support, Bug Fixes, Enhancements, QuickBooks Compatibility Updates, Maintenance ...**

Ok, now the product is in a sellable format, perhaps you are still under the impression that you are finished? If so think again, just because a product is ready for the market doesn't mean that you are done.

- Someone has to handle the sales, including sales calls, order processing, and order fulfillment.
- Someone has to handle support, and that includes all the "bugs" that come from the various ways in which QuickBooks could be set up – or not set up, as the case might be.
- Debugging and error corrections lead to maintenance releases; but unlike with QuickBooks; end users are NOT going to be content to "wait" for these updates – they want them NOW or they want their money back!
- Smart developers will keep a list of product functionality requested by customers and release periodic product enhancements based on those customer requests; thoroughly testing and documenting the changes in-house and then releasing them in "beta" version to the customers who requested the enhancements for further testing, before making it a "live or final release" update.
- Each year when the new version of QuickBooks is released the developer must update his software to be compatible with that new version. The SDK, while backward compatible, is NOT forward compatible. Each year Intuit adds new functionality to the SDK, which smart developers will take advantage of – perhaps even adding more functionality to the product and releasing a "compatibility update" along with an "enhancement update". Again these are thoroughly tested in-house and new features are documented before the update is released.

- Then there is the questions of "maintenance"....does the developer make the user "upgrade" every year like Intuit, or do they charge an "Annual Maintenance Fee" to maintain the existing program?

While I haven't covered absolutely every step of the process, I'm sure by now you realize that the role of a developer is very different than the role of a ProAdvisor. The "risk" of creating a third party application is far greater, and perhaps even more time-consuming, than being an Advisor and supporting clients who use QuickBooks. Does this mean you shouldn't ask a Developer their opinion and seek out an automated solution? No Way! If no one ever asked, or ever pursued an idea, there wouldn't be over 400 add-on products available now. We would still be manipulating Excel spreadsheets to complete certified payroll (oh the thought!) and wishing there was a better way.

Developers of third party applications and ProAdvisors each have their own unique relationship with the end-user, with Intuit, and with each other..which will be the topic for next month's installment.

Until then.....

*Nancy*