I'll Just Write a Software Program to Do That . . .

***Dave Hunt**, Contributing Editor*

How many times have you said or heard those words? Is your reaction, Oh joy! Or something less cheerful?

We often look for ways to automate repetitive tasks, fill a gap in the commercial code we are using, or capture a new capability that we developed on our last project. Early codes such as Modal-Plus and Cosmic Nastran, and even the first "mini" computers, had built-in programming languages to make their operation more effective. Current engineering software continues to be extensible through programming interfaces. Many of us learned a programming language in college and have since delighted in writing our own programs to make our engineering work more efficient.

A next logical step is to share our creations with our fellow engineers. We might ask the in-house programming guru to write us a program, for example, to calculate ply failure margins for the composite analysis we are doing. But what happens when the operating system changes and the programmer is no longer around? Or when everyone makes their own "improvements" to the code, resulting in multiple versions, and a bug is found? Who decides what quality assurance checks are needed and what kind of documentation standards are to be met? Which version is the "master" version? Do we incorporate everyone's improvements? If so, how? And there is potential for disaster when someone thinks it would be a good idea to make the program available to customers and even charge them to acquire it.

Having lived through all of these and other scenarios, I want to offer my perspective on how to make software development choices that will be positive and valuable in the long term. We'll start with the premise that someone has a software program idea or new capability that they think will benefit others in their group or company. ATA Engineering's IMAT program is a good example to illustrate.

IMAT began as an idea from one of our test engineers. He simply wanted a translator that could take the data into MATLAB®, where he could manipulate it in ways that were not possible within the existing commercial test software product, and then take it back to the software product for further analysis without losing any information in the process. At the time, MATLAB 5.0 had just come out with some major new functionality that made it an attractive platform for what he wanted to accomplish. MATLAB was becoming more widespread and was a relatively easy language to learn by nonprogrammers. It had robust math, graphing, and graphical user interface capabilities, and it also interfaced with C and FORTRAN. The commercial test package we were using had a FORTRAN and C programming interface to access its proprietary data file formats, so all of the pieces were in place. Thus, in 1997 IMAT was born.

As IMAT was taking shape, we thought that since we found it useful, others might as well. Since our company at that time (SDRC) was primarily in the software business, we decided to see if we could make IMAT available through our standard sales channels and release it as a commercial product. Of course, this took us down the path of having to learn and manage all of the additional aspects of making software commercial, including licensing, documentation, on-line help, a graphical user interface, support, and delivery.

The first customer version was released in August 1998 after being used internally for about a year. However, just months after IMAT was starting to be used (and before it was released externally), the engineer who first developed it left the company. We were then faced with the dilemma of what to do with IMAT. We had already committed to releasing it commercially. We decided to transition development responsibilities to another engineer who had been involved in a small portion of the IMAT development effort. Then, less than two years later, our group spun off as a small engineering services company (ATA Engineering), leaving the commercial software infrastructure of our former company behind.

Through the years, IMAT has expanded from simply being a test-data translator to and from MATLAB to having a wide range of built-in functions, including orthogonality, translation of analysis data from Nastran, ABAQUS®, and FEMAP, an alias-free polyreference curve fitter, and graphical utilities for performing pre- and post-modal-test tasks and signal processing from the time to the frequency domain. As ATA used the tool on projects to solve specific customer needs, more and more capabilities were added. The software was capturing innovations developed during many of our most challenging projects.

Through all of this, ATA has learned a lot about how to develop software to support our services. Because we had decided to make IMAT a commercial product, we were committed to the extra effort required for "commercial-grade" software over and above that for "home-grown" internal solutions. For example, commercial-grade software documentation standards are higher. We also had to figure out who was going to support users when they called, provide product direction and planning, and handle bug tracking and enhancement requests. Not only did we have to consider installers and training, but we also had to plan how to deliver the software to our users and notify them of new releases.

Beyond the actual creation of the software, we also needed to implement a licensing mechanism. Then we had to sell it, take orders, send invoices, track users, renew the maintenance, keep customers informed of updates, and more. Then there's a whole other topic – "let's see if we can make money selling it." We will save that discussion for another time.

For an engineering services company, all of this is a lot to take on. Has it been worth it? For IMAT, we feel it has been. IMAT has provided a capture mechanism for ATA and our customers for the advances made in each project that require a new software capability. As ATA has grown and expanded, we have learned that even internally, we need this commercialization process to allow each of our engineers to build upon ATA's accomplishments and broad experience as well as deliver new methods for each new project.

As the number of ATA engineers and IMAT customers increased, so did the number of innovative software solutions that we needed to make accessible. IMAT provided a better capture and distribution mechanism than the old way of the software author showing every person how to use his tools. Therefore, we found that even our internal needs required the commercial level of documentation, support, and delivery.

We have taken the lessons we learned from making IMAT commercial grade and applied it to other solutions. We have made the conscious decision to take the best internally developed solutions and bring them up to the standard of commercial software. Even though it is a lot of work, it is worth the effort. The ability to effectively disseminate these software solutions easily throughout the company and make them available to interested customers is one key to our success.

As a company with a number of regional offices and a growing staff, good documentation is very important to us. New users need to be able to get up to speed as much on their own as possible. We also need to be able to support these products internally as well as externally. We have systems in place for beta testing, bug tracking, and enhancement requests. In the end, we concluded that all of these steps are also needed to make effective tools within the company.

So next time you say, "I'll just write a program for that," challenge yourself to add, "and write a user's guide, on-line help, graphical user interface, and version control for it." We have found that these extra steps go a long way in building powerful tools that all of our engineers can more easily share and utilize. SV

The author can be reached at: dave.hunt@ata-engineering.com.