

# Ten Must-Have Service Software Features

*Don't overlook these critical attributes!*

If you're like most service managers, you're not happy with your computerized service system. Maybe it's an old fashioned DOS system or it only allows one person to use it at a time. There are missing reports that the author refuses to add and you cannot hire a local programmer to write them for you.

Maybe the system was acquired only three years ago and did everything you wanted but now that your business has doubled in size in terms of employees or products offered, the system can no longer handle your demands. Business information is much like a flowing river. You can't cross the same river twice because it is constantly changing. A software solution needs to adapt to changing business needs.

You can take comfort in the fact that you're not alone but that doesn't make the problem go away. You realize that you must replace your service system, again.

But this time, you're determined to get the right system. One that will meet today's needs and that will grow with you. Easier said than done. Even you don't know what the future holds so how can you properly evaluate new software?

The answer is easier than you realize. Make sure that any system that you consider possesses these ten features and the software will be able to grow with you.

1. ***The underlying computer language must be a viable product.*** This means that the system must operate under Windows but Linux is fast becoming more acceptable. Forget DOS. Forget CPM. Forget any proprietary operating system unique to a single manufacturer's computer hardware.

If you are running a very large operation, you probably need mainframe/mini-computer systems that operates under Unix. SQL is another requirement for larger organizations. Both of these solutions are not for everyone. The high cost of technical support required to keep them running is beyond the budget of most service organizations.

The programming language must also be viable. Forget old languages such as RPG, ALGOL, and others that reveal one's age and memory. Some languages that have been around for years are still viable because they have evolved with today's technologies. Visual Basic and COBOL are prime examples. The FoxPro language originated some 20 years ago but has been acquired and substantially enhanced by Microsoft. A viable language has a future.

Other languages might be considered viable but not appropriate for business systems because of the programming effort required to solve a given problem. The "C"

language is a good example. The “.NET” language has not yet proven itself so be very cautious when considering this type of system.

A viable language has thousands of independent programmers that can support your activities and write custom reports for you.

A good rule of thumb is to limit your options to computer languages published by industry leaders such as Microsoft. This doesn't mean that every Microsoft product is viable but chances are pretty good that the company will be around for a while.

2. ***The software solution must be an “open system.”*** In addition to being written in a viable language, the system structure must be well documented. This technical documentation goes beyond a pretty user manual. It provides a blue print for your private programmers and technically-adept end users as how the system is constructed.

This documentation includes data table structures, field definitions and usage, program functionality, and control file usage.

Truly open systems actually “invite” modification and provide guidelines for implementing them. For example, a system might provide tools for modifying reports or adding your own custom reports.

3. ***The system should not be specific to your industry.*** Your response to this key feature might be “say what?” Actually, there's a good reason for it.

Every segment of the service industry has a limited number of companies, such as yours. A software system written exclusively for that industry has a limited number of potential customers. For example, if there are 2,000 organizations in your industry a software solution would be limited to a sales volume of only around 500 systems because not everyone needs a new system, not everyone can afford that system, or they already employ a working solution.

Once the publisher of that industry-specific software approaches the magic saturation point, 500 systems in this example, revenues drop, staff is terminated, research and development stops, and your support disappears. Maybe the publisher disappears too.

If you pick a software solution that meets at least 80% of your processing requirements and targets multiple industry segments, the publisher's prospects of survival are much better. There are 3,000,000 service organizations in the United States alone which means that no single software solution will ever reach the saturation point.

4. ***The system must “speak” your language.*** This means that if the system refers to the repair staff as “technicians” and to you they're really “plumbers,” a mental translation must take place every time you use the system. The system might also refer to your customer work sites as “locations” or “offices” while you call them “work sites.”

Industry-specific systems might get the terminology correct but you should seriously consider generalized service software that let's you choose your own terminology.

You should also be able to tailor customer complaint codes and other reference tables so that they are meaningful to your operation.

Speaking your language also infers that your native tongue is supported. Many offices today are staffed with people of different ethnic backgrounds which means that in addition to English, many might also speak Spanish, French, Chinese, or even Turkish. To avoid mental translations and improve staff productivity, the system should be able to "speak" multiple languages.

5. ***Reporting must be flexible.*** You must have the ability to design your own work order form. You need the ability to add or delete fields from status reports. You should be able to add another reporting sequence or selection filter with little or no programming. New reports could be added to the system with little or no programming.
6. ***The system technically current.*** This includes such solutions as web-enabled customer inquiries, initiation of work orders by customers, e-mailing instructions to technicians, and even the ability to electronically send reports to customers or remote offices. Even if you don't need an Internet presence today, the ability of the system to provide these options infers a higher level of developer commitment and expertise.
7. ***Searchable service history.*** Have you ever racked your brain with a puzzling equipment failure that you just know that you were faced with a similar problem three years ago but don't remember who the customer was or which one of your staff worked on the problem?

The system should be able to perform key word searches of your prior service orders and support your research.

8. ***Data importing and exporting.*** Because no single system can meet 100% of your information requirements, it must have the ability to extract information for usage by other computer systems, such as Excel.

Furthermore, it must be able to "read" data from other systems. This would enable you to import industry price lists, data from an existing system, or purchased mailing lists.

9. ***Accounting system integration.*** No service system solution is an island unto itself. Life's too short to maintain two systems and enter the data twice. First, the cost of the labor will kill you and the inaccuracies of redundant data causes untold error conditions.

If your organization is larger than a one or two man shop, you need a formal accounting system. The service system under consideration should be capable of integrating with a well-established accounting system which infers that data is shared between the two

systems and redundancy is kept to a minimum. This integration should be in real time, as opposed to, a batch updating process.

The service system solutions should not include its own accounting system thus claiming to offer a complete enterprise solution. The challenge of providing both a feature-rich service and accounting system is beyond most software publishers.

In other words, leave the accounting software development to those who know it best and leave the service software development to those experts. In a large Unix environment, you may not have a choice when it comes to independent accounting systems.

10. ***Sales and marketing Support.*** The data in the service management system can be invaluable to your sales staff. For example, salesmen frequently need reference accounts to help close a sale. They should be able to produce a report that itemizes existing customers geographically close to the prospect.

The system should also provide “data mining” capabilities that enables salesmen to target existing customer purchase history thus enabling them to make knowledgeable suggestions for additional items that the customer might be interested in. The Internet site Amazon.com is a great example of this capability. Their software remembers your purchase history and compares it to other customers. When you purchase a particular book or CD, they’ll present you with suggestions for additional products that you’d probably be interested in.

#### ***About the author...***

Thomas J. Francl began his own computer service organization in 1984 and built it into one of the most innovative service companies in Los Angeles. His computer programming and systems design experience dates back more than 30 years.

Mr. Francl received a Masters in Business Administration from the University of Southern California and has held part time teaching positions at several large universities in Los Angeles. He has authored numerous magazine articles on the use computer technology in modern business and was the primary author of a hard cover book describing a useful budgeting tool for large computer departments. He is a member of the Institute of Management Accountants and holds management positions in several community organizations.

Mr. Francl is the president of South By Southwest and can be reached at either 760/751-1111 or [tomf@southbysouthwest.com](mailto:tomf@southbysouthwest.com).