

SOFTWARE EVALUATION

MARCH 2004 EJEMS

BY WILLIAM E. OTT

As most people in the EMS industry recognize, EMS is becoming a data driven business. There are a number of initiatives underway to define various aspects of what data points EMS data systems should include. This data once collected can be utilized for operational review, trending, mapping, quality assurance and improvement, and billing for reimbursement at the local level. Aggregating the same data points across multiple agencies allows for performance benchmarking and comparison. This process ultimate can be used when combined with patient outcome data to determine if EMS is a worthwhile endeavor from a funding standpoint. Currently there is no broad scale science to show that EMS works from a cost effectiveness perspective. As dollars become tighter and more is expected for less, the EMS industry will have to start to scientifically prove its worth to insurers and taxpayers.

So, if your service doesn't have some mechanism to electronically capture and maintain response and patient records, chances are that there will be the need to move in that direction in the near future. That will mean either developing your own software or buying into one of the existing packages on the market. Unless you are a large provider or exceptionally funded you will most likely find yourself

needing to evaluate and buy into one of the many EMS software packages now on the market.

EMS is a narrow and small vertical market from a software perspective and because there are no set national standards for EMS data collection, the large software companies such as Microsoft have stayed out of the EMS industry. There are several large vendors and many smaller software vendors within the EMS industry that all have fine products that fit certain needs. Describing full scale software projects including RFPs, time cycles, implementation time frames, training, cost per user, etc... is beyond the scope of an article but I wanted to give you some simple insight to evaluating software packages to help determine the best fit for your needs.

You wouldn't just take a used car salesman on his word that a car runs like a charm, right? Likewise, you can't rely on a software salesman's assurances that what he's offering isn't a lemon. It's up to you to put the system through its paces. If you follow these tips, you can do a lot to prevent yourself from unknowingly buying software that doesn't do what you need it to do, or worse, software that doesn't exist or isn't ready for customers to use, also known as "vaporware," an all too common slight-of-hand practiced by the software industry.

There are three ways you can evaluate software. You can pick one person in-house to do it, put together an internal team or hire an outside consultancy. If you

are buying software for your department and the budget is small, go ahead and give the job to one person. But if you are considering a large system that is critical to your agency as a whole you should use a team or an outside consultancy.



Should you go the single evaluator route, be sure to choose someone who knows what problems the software must solve, whether the software's users are tech-savvy and how the software needs to integrate with other systems in the department. Your evaluator can come from your department or from the IS department, as long as she understands these critical issues.

If you're looking at a larger software purchase and decide to form an evaluation team, you must have a technical person on it. She will know the key questions to ask about performance, integration with legacy systems and other technical issues. Make sure the team includes an employee who will use the system. Employees resist using software applications when their needs and concerns were not considered and addressed during the evaluation stage.

With software demos, the most important thing to remember is that what you see isn't always what you get. You have to ask questions to make sure that you are not seeing vaporware. Software salespeople avoid showing the functions and

features that don't work or exist even though they may show you the interface, but they know which on-screen icons to click and which ones to ignore.

Don't follow the salesperson's script—follow your own. Bring a checklist of the functions and features you need to see, and make sure that you see them during the demo. Tell the salesperson before the demo what features you want to see and in what order.

Don't let the salesperson spend time on a module that you are not going to use. And don't let the salesperson tell you what a key function does; ask to be shown what it does. If the salesperson rushes through a key feature, stop and ask to see the feature demonstrated again. If a feature or module doesn't work during the demo, watch out, chances are good that it won't work when you buy the software.

Given a choice between demonstrating an old version or a new version of software, a software salesperson will typically show an early version of the next release, called the beta release, if it has more features and doesn't have too many problems or bugs. That's fine for the salesperson but potentially bad for you if you make your decision based on software that isn't available yet.

Beware of the software crashing during the demo. The salesperson might shrug it off or try to blame it on Microsoft Windows, but this may be a sign of something

wrong with the software. If the software crashes with one salesperson using it, you have to question how it will perform when all of your people are using it. Additionally, make sure that what you're seeing is the base version of the product, not a customized version. Demos can be customized so much that you wouldn't recognize the real product. You want to see the system that you will actually get.

Create a checklist of key questions to ask each and every vendor, and keep the answers in electronic format, whether in a word processing document, a spreadsheet, or a database. [\[I have a free Excel Sheet designed for this I will share ... Drop me a line\]](#) Keeping a formal record of vendor responses to questions comes in handy when you want to compare vendors and their applications on a side-by-side basis.

Here are some of the questions your sales representative should be able to answer:

- How long has the company been around?
- Will the vendor be around for the long term?
- Does the vendor have a sizeable client base?

It's okay to ask for references. If you ask for three and the vendor gives you a two-page list, that's a good sign. If the vendor can't even provide three, you may

want to consider that a red flag.

Before you invest a ton of money in software, hardware, and training for a system, you must ask yourself one question: What output do I expect from this system?

To answer that question, start by making a list of the titles of the reports you know you're going to need to run every day, every week, every month, every quarter, and every end-of-year.

Then, for every report title you identified, list the columns or fields you want to see on that report. Finally, ask yourself these questions:

- Assuming I configure the tables and enter the data correctly, does the application have built-in report templates for all of the reports I need?
- If the application doesn't have built-in templates for some of the reports I need, is there a utility that lets me customize my own reports?
- If there is a custom report-builder, how easy is it to use?

Don't fall in love with an application's front end and then find out it won't run on your current system. Start with these questions:

- What are the minimum hardware requirements for installing and running the program on the server? On workstations? Specifically, how much

available space is required on the disk drive, and how much RAM do I need for optimum performance?

- What are the minimum operating system requirements for the server and the workstations?
- Does my current infrastructure meet those minimum requirements?

I know it's an old-fashioned concept for most IT pros, but end users learn how to use software as much with lessons they can read as they do with the try-and-see-what-happens method. Before you buy, ask these questions:

- When I purchase the software, do I receive a printed manual at an extra charge?
- Or am I expected to use the online help feature to look up how to use the application and buy the print- or CD-based materials separately?
- Are explanations of error messages included in the documentation?
- Are the directions clear as to how someone uses the software effectively?
- Are directions enhanced by useful examples where appropriate?

Everything about your rights is in the agreement, but ask your representative:

- When I buy the application, do I get original copies of the software on CD or disk?
- Am I allowed to make backup copies of those disks?
- Can I upgrade or patch my software online?

I throw this one in on behalf of all those power users out there who don't have anyone they can call for tech support, and so they've learned to bail themselves out of tough situations by drilling down into online help. When you're testing the software, be sure to answer these questions:

- Is context-sensitive help available by pressing [F1]?
- Are mouse-over tips available for toolbar tools, buttons, and other program icons?
- Is the interface user-friendly enough to be used with a minimum of computer expertise?
- Is there a Web site I can visit for troubleshooting tips, lessons, or tech support?
- Does the software accurately evaluate and validate user input?
- When an entry is invalid, is it easy to redo the entry?

Some applications appear to process data and repaint screens with the speed of lightning—on the demo running on the sales rep's laptop. Be sure your prepurchase testing includes determining the answers to these questions:

- How long does it take to boot up when I launch the program?
- Are there any unacceptable delays in executing tasks?
- Does the software pause without explanation?

Though the location of the functionality may vary from interface to interface, self-respecting databases ought to provide favorable answers to these questions:

- Will the program allow me to export data to an external file type, such as a database, spreadsheet, or plain text (e.g., delimited and fixed-length), so I can manipulate that data for special reports?
- Will the program allow me to import data from external sources, such as databases, spreadsheets, or plain text (e.g., delimited and fixed-length), so I don't have to key my old data from scratch?

Here's a big surprise: Software that is complex enough to meet your needs often is complex enough to require end users to be formally trained to use that software efficiently. So don't be surprised when you discover that training, support, and upgrade costs may be as much as 25 to 60 percent of the cost of the software itself. You never know what options are available until you ask, though.

- Is any free training for my staff included in the purchase price of the software?
- Will you train my staff on site, or do I have to send them out for training?
- What kind of tech support do I receive for the purchase price of the software?
- Is there a toll-free number to reach tech support?

- What are the hours of operation of tech support?
- How many free calls can I make per week or per month?
- How responsive is the customer service?
- Are there any for-fee support options available, and what are the costs?
- Are there hidden charges?
- Is the software updated regularly and when?
- Are updates automatically snail-mailed or e-mailed?
- Will my sales/support representative inform me when minor patches or major upgrades are available?

Sometimes an application is too nice. It doesn't remind us when no one has run the backup utility for six months. Ask your vendor about what it can do for you in case of interruption of computer services.

- Does the software make it easy to perform manual backups? Schedule automatic backups?
- Does the company offer any free or for-fee disaster recovery support, in case I have a catastrophic hardware failure?
- How soon can the vendor provide on-site help or replacement equipment?

If you're looking for a set of formal criteria by which to evaluate the supportability of a commercial application, I hope this checklist helps you get started. The most important thing to remember is that, when you're dealing with someone who

wants you to write them a check in exchange for a product or service, its okay to ask a lot of questions. Good vendors won't mind answering.

I welcome your comments, criticisms, feedback, and ideas. You may contact me at ejems@cpcstech.com

William Ott is president and chief consultant of CPCS Technologies, a NC-based technology consultancy providing services to the public safety and defense communities. He's been involved in EMS since 1981, in field, education and administrative capacities