CMMS Failure Is Not An Option

Written by Blaine L. Pardoe, Enterprise Management Systems
Wednesday, 01 April 1998 18:29

Helpful information for getting a failing CMMS implementation projectback on track.

You spent a quarter of a million dollars and countless hours of productivity to purchase a computerized maintenance management system (CMMS). Good. Smart move. But it has been 6 months, and so far all you've gotten from the software is headaches.

Maintenance supervisors claim it takes too long to enter the information on the work orders. The person in charge of scheduling says the PM system is not like the old system and has refused to use it. The housekeeping manager claims the system is being used correctly but the reports are messed up, so it must be a system problem. Worse yet, upper management is asking when all of those big cost savings and paybacks are going to kick in.

Estimates indicate that anywhere from 40 to 60 percent of CMMS implementations fail for reasons from bad management to poor communications to resistance to change.

Here are some ideas that can help you get on top of a bombed or exploding implementation effort and prevent it from reaching the point that it is a total loss in the eyes of both your employees and management. If you are thinking about implementing a CMMS for the first time, this material may steer you away from possible stumbling blocks in your efforts.

Before examining guidelines on what to do, it is worth looking at what not to do, as outlined in the section "What Not To Do."

What to do

Now that you have an idea of what not to do, you can turn your attention to what can and will work for you. The following steps or processes can be followed depending on your situation and the degree of your implementation failure.

No matter what you need to do, you must begin with an analysis of what went wrong with your implementation before you can move ahead in fixing it. This is hard. You have to be brutally honest. More importantly, you need to look at the situation objectively.

The key to this kind of analysis is to constantly ask the question: "Is this a symptom or the real reason for failure?" In most cases there will be more than one reason that the implementation effort blew up. When someone says "the PM module didn't work," is he saying the software did not function as it was supposed to or that he just refused to change his old shoebox method of doing PMs? Again, was the reason given a symptom of the problem, or a problem all by itself?

These painful debriefing sessions need to be kept in focus. They are not gripe sessions or opportunities for finger pointing. Instead, they should be aimed at getting to why the implementation effort stopped or reversed. It is recommended that an outside facilitator be used to run these meetings to make sure that all parties have a chance to give their views or perspectives on the matters that are brought to the surface.

Often it will be found that the root cause was a lack of proper resources in an area. Make sure that as you define what went wrong, you determine if the appropriate resources were in place. Keep tabs as well on factors that you do not or cannot control in the implementation process-factors such as production schedules, management issues, etc. This information can help you organize your response to the root problems and get your implementation back on track.

Reorganize your attack

Failure is not an option. Knowing what went wrong allows you to make plans to correct things that went wrong or avoid trouble spots in the future.

The majority of CMMS implementations fail because of one key factor-people or managers are unwilling to change the way they do business. Many maintenance managers try to force the system to conform to the way they do business, which they, themselves, often do not understand-but they somehow expect a CMMS to understand it.

CMMSs are sophisticated, expensive tools. Many have functions and capabilities that allow them to minimize the amount of change you and your management will have to go through. But the reality of successful implementations is that people are going to have to change the processes and the way they work to fully leverage the power of the CMMS to save money and organize work. I once visited a production plant that was stalled on its third CMMS implementation in 3 years because the department was unwilling to change the numbering of its work orders to what the system could handle.

You must understand how you do your work before you can automate it. Take the time up front to analyze how you work. The CMMS reports are the key. Reports tell you what information is needed to conduct business. They tell you also what you need to get out of the system. A good reports analysis answers such questions as: What information are your people gathering presently when they do work (hours, materials, etc.)? How do they plan to manage using information from the system? Are they running reports now, and if so, how are they using them? If they are not running reports, what do they need to get from the system to allow them to manage their business? Is all the information they are looking at necessary?

Reviewing the information you need to obtain from the system tells you what you need to capture. There are data requirements that the CMMS itself must have to operate properly, such as required fields to process work orders. Once you have a full understanding of the minimum amount of data you have to capture, it is much easier to develop a plan for implementing various functions of the CMMS.

All is not lost if your computerized maintenance management system (CMMS) implementation effort fails. You do not have to scrap the software and your hard work thus far and call the whole process a loss. With the proper leadership, communication, and effort, even an implementation that is falling apart can be reversed and the rewards of a good system can still be achieved.

It is important to avoid blame and search the root cause of the failure. Then, the real data needs of the maintenance operation, as well as the CMMS software itself, must be reviewed as the basis for renewed effort.

"Chunking" your re-implementation effort
 A CMMS is a complicated software package in that it crosses a number of different business areas and fields of expertise. Planners, engineers, supervisors, technicians, purchasing agents, crib attendants, and others can all interact with the system and provide information it needs to make it a useful tool.

Given the size and scope of the software, often the best approach is to break up your implementation effort into bite-size chunks. This "chunking" of the implementation effort makes the software easier to implement, and, in the long run, increases the chances for success.

There is a tendency to look at chunking too broadly. When chunking, you need to break your chunks down into workable real elements within the system-not just modules or sets of features. For example, a common cry during implementation efforts by the maintenance staff is "just get the work order module up and running, we'll figure out the rest later." But it's not that easy. The key to this kind of sweeping statement is to look at two things. First, what defines "up and running;" second, what is needed to meet that definition.

The implementation team must determine what "up and running" means. Does it mean capturing labor and material information against a work order? Tracking equipment history? Does it mean that schedulers and planners can scope out the entire workload and balance the resources to on-line schedule work? Each element of these questions forces more detailed questions because a CMMS, no matter what CMMS you pick, requires certain bits of information to be in place just to fulfill these needs.

What does it take to meet those definitions? If the answer was yes to all the previous questions, your chunking efforts are numerous. Just to successfully implement work orders, you may have to have the building, grounds, and equipment databases fully populated. You will need to have the inventory put into the system as well just to capture material-plus the inventory system may require you to have vendor information populated. And do not forget the employees. You may have to enter them, their skills, available work hours, days of work, etc., into the system just to capture employee data. In those examples listed above, each is a chunk of information that has to be captured.

Finally, as you break your implementation effort into pieces, the team will need to make sure that for each chunk there is no required information that has to be in place first. For example, the equipment database may require that you have the vendor and buildings databases populated first. This helps you define the order in which you approach your chunking efforts.

Ownership of information

If your implementation effort is to be successful there is a hurdle that often has to be crossed-ownership of the data in the system. While far from tangible, this threshold is often what defines a successful or failed implementation effort.

Ownership of the information is when employees and managers are held accountable for the

data in the CMMS. Accountability is when maintenance management stops questioning the reports that come out of the system, and listening to the complaints that "the software or database must be screwed up," and begins to run the business using the information in the system. It is when business decisions are based on the figures from the CMMS.

Most CMMSs work well. They have been through quality assurance and, if they are used properly, the information in them is correct. All too often implementation efforts are undermined when management, fearful of a new software package, does not believe what it sees. How does this weakness manifest itself? Example: "Floyd on the night shift has been complaining for 3 years he's overworked, but the figures here show he's not. Obviously the system must be wrong."

The truth is that the employee probably *is* overworked but is not capturing that information in the system. This is not a personnel problem in most cases, but a process issue. Rather than refute the information in the reports, management needs to guide "Floyd" on how to use the system to reflect his status of being overworked. (In some cases, however, this may be a personnel problem if it is found that Floyd really is not overworked and the CMMS can now prove it.)

Importance of communication
 The key to a successful implementation effort is good communication. This does not mean that the day the software is installed management fires off a memo to everyone about it. Instead, communication is ongoing with the staff. It takes many forms, including e-mail, meetings, and perhaps a newsletter to keep everyone informed on where the implementation effort is and what is coming next.

Communication needs to be regular, consistent, and reinforcing. Regular means that the implementation efforts are passed to the right people in plenty of time for them to respond or act. Communication is consistent when the wording is always the same and the look and feel of the communication is always the same.

Reinforcing means that communication does not replace training and leadership, instead it substantiates it and supports it.

CMMS Failure Is Not An Option

Written by Blaine L. Pardoe, Enterprise Management Systems
Wednesday, 01 April 1998 18:29

There is a thought with many maintenance managers in failed implementations that "my people don't need to know anything about the system. We're a small shop. If I tell them to do it, they'll do it; they don't need to know what we're doing or why." These are obviously fallacies. Size of a shop only governs how you communicate. The front line maintenance workers can make or break an implementation effort. If they think the CMMS data are going to be used for downsizing decisions or for performance appraisals, guess what-they will derail the implementation effort. It is important to tell them truthfully what the CMMS information is going to be used for and their role in the implementation effort.

Use outside help when necessary
 CMMS consulting is becoming a very popular way to make or repair your implementation efforts. If it is done for the right reasons, and properly supported, bringing in a "hired gun" to jump-start your implementation can work effectively. There are some tips and hints of what you should do and avoid, however, that can make this process productive and effective.

As you have probably gathered, successful CMMS implementation requires some knowledge of the CMMS, but it also requires a good understanding of the business and how work is done (work flow processes). You should look for the same characteristics in any consultant. He should understand the maintenance function and at the same time know enough about a CMMS to ensure that the implementation process can be broken down effectively into manageable work packages.

The best application of an outside consultant is to assist with developing the plan. Remember, ownership of system information is vital. The key then is to not put the consultant in charge of the implementation. Instead, the consultant can take on the role of trouble-shooter, working through process issues and managing the planning portion of the implementation effort while the company-leader of the implementation actually leads the effort.

Should you hire someone from the company that sold you the CMMS? Yes, if the person is the right individual. Computer skills are important in a consultant, but not nearly as vital as his people and communication skills. Be careful in working with your CMMS vendor that you are getting someone who knows CMMS, but someone who knows the business more. Key question: "How many successful CMMS implementations have you put together?" There is a big difference between how many systems a vendor consultant has installed versus how many he has implemented. Bottom line: nothing is more frustrating for a maintenance manager than sitting down with a computer technician who does not understand why the second shift does not have time to close work orders in the -system.

CMMS Failure Is Not An Option

Written by Blaine L. Pardoe, Enterprise Management Systems
Wednesday, 01 April 1998 18:29

All is not lost if your CMMS implementation effort fails. You don't have to scrap the software and your hard work thus far and call the whole thing a loss. With the proper leadership, communication and effort, even an implementation that is falling apart can be reversed and the rewards of a good system can still be achieved. **MT**

Blaine L. Pardoe, author of Cubicle Warfare: Self-Defense Strategies for Today's Hypercompetit ive Workplace (Prima Publishing, 1997), is a partner at Enterprise Management *Systems, 8033 Sudley Road, Suite 155, Manassas, VA 22110; (703) 331-3749;* -e-mail pardoe@emcs.com.

What Not To Do

Before examining guidelines on what to do when implementing a CMMS, it is worthwhile looking at what not to do.

These examples are based on experiences with actions that simply do not work for most managers in most companies.

Do not hire a consultant to run the whole thing
 Consultants will tell you that they should run the entire implementation effort. Why? The truth is that for an implementation effort to succeed, your own people need to take ownership of it. That's not to say you can't bring in outside help; in fact, in some cases that help is critical. However, do not get sold on the concept of someone outside your company being a miracle worker.

Do not blame the implementation manager
 If an implementation fails, seeking a scapegoat will not solve anything. If it fails, it is the fault of everyone involved with the project. More often than not, the fault lies not with those in charge of implementation, but rather those in management that stood in its way or derailed it in the first place.

Do not threaten the supervisors and staff

CMMS Failure Is Not An Option

Written by Blaine L. Pardoe, Enterprise Management Systems
Wednesday, 01 April 1998 18:29

Having lived through one of these situations myself, I can vouch for the fact that if you threaten the supervisors and staff with their jobs, they will indeed use the system. However, they may not use it right or care about the data that they put in it. In the short term you will get them to use the CMMS but at the same time you will force them to hate it, and you.

Do not buy a new CMMS
The logic behind this is akin to: "When I hit myself in the head with a hammer it hurts. If I switch to a wrench, it might not hurt." Often when the implementation of a CMMS goes bad, everyone blames the software. Usually, the real problem is that the people who use the software are resistant to change. Bringing in new software is simply a chance for everyone to repeat the disaster that messed up the last implementation.