An Overlooked Ingredient to Successfully Implementing a Computerized Maintenance Management System (CMMS)

By Jennifer Ohl, MBA
Midwest Software Specialists, Inc.

Companies invest considerable resources to roll out a new Computerized Maintenance Management System (CMMS), expecting to receive the benefits these systems are designed to provide. While it seems as if a company has everything in place, an important factor that may determine the outcome of the project is often overlooked. This paper provides a case study comparing the implementation experiences of two similar companies, demonstrating how important this often overlooked factor can be.

Successfully implementing a new CMMS requires careful planning and resources. Factors include: a reasonable budget that doesn't result in squeezing a five day task into three days, adequate plant resources for the project, such as computers in a designated area or separate training room; an enthusiastic project manager with the requisite skills and authority and a knowledgeable, and dedicated consultant.

Companies believe this is all they need for a successful implementation. However, they may have overlooked another important factor.

In this paper we will examine two maintenance program roll-outs at companies which shared many similarities. The companies were in the same industry, the CMMS came from the same vendor, both budgets were adequate, project managers existed in each company (although one was more qualified than the other) and the same consultant was involved in both implementations. Results, however, were very different.

Introduction

It is important to first define a Computerized maintenance management System (CMMS) before one can discuss how best to implement it. A CMMS is an integrated system for managing all aspects of a maintenance operation. It is commonly referred to as an Enterprise Asset Manager, or EAM. In a nutshell, a CMMS is a computerized software package for maintenance. These programs track work to be performed on equipment via work orders, records on completed work orders, provide modules to perform preventative and predictive maintenance, help reduce critical part shortages and overall holding costs via inventory optimization and permit extensive reporting on activities.

Why do companies really need this type of software? Let's examine some statistics. Most organizations spend 10-12% of their total operations budget on asset maintenance. Assets refer to plant equipment and buildings. Total maintenance costs contribute between 4% and 14% of total production cost. Over the life of a building, operation and maintenance costs amount to seven times the original cost of the building. Unplanned maintenance (unscheduled maintenance) costs three to four times more than planned maintenance.

Benefits of CMMS

The following table lists reported benefits from successful implementations of CMMS.

Table 1. Benefit list.

Item	Method used	Results	Source
Maintenance labor cost	Improved scheduling	10-30% reduction	Gartner Study
Overtime cost	Improved scheduling	Up to 10% reduction	Grant Thornton
Production capacity	Implementation of preventive maintenance	Up to 40% increase	Grant Thornton
Storeroom inventory	Use of inventory functions	up to 25% reduction	Tompkins associates
Outdated/ obsolete inventory	Use of inventory functions	25-40% reduction	MRG inventory specialists
P.O. processing costs	Use of purchasing function	Up to 60% reduction	ARC report

In addition to the primary benefits listed in table 1, there are many additional benefits from having your maintenance operation computerized. For example, a CMMS system can be a very useful tool in earning and maintaining ISO certification. Instead of digging through cabinets of paper, you can quickly and easily generate relevant reports to demonstrate satisfactory equipment maintenance, availability of critical spare parts, availability of safety procedures, and adequate training for your employees.

However, CMMS are not plug and play! While CMMS is a very useful tool that provides many benefits, it requires consistency in data entry to obtain results. Data needs to be entered in a logical and easy to follow method to facilitate efficient system operation and reporting. In addition, someone needs to regularly run the reports and analyze the data. Contrary to popular belief, this will not happen by itself! Also, many people assume that once you implement a computerized system, you will immediately be able to operate with fewer people. This is seldom the case. As stated, someone from the operation, be it a clerk and/or a maintenance supervisor must enter the data in the CMMS, and this takes time. The program can be set up to streamline this process, but data entry always demands human resources. Clearly, the implementation of CMMS requires careful planning and success is not always guaranteed. According to Plant Services Magazine, between 35% and 50% of all CMMS implementations fail!

This paper focuses on potential reasons for failed implementations and how to avoid common pitfalls. Our staring point assumes that senior management provides the necessary resources (budget for the necessary hardware, software and training) and stands behind it. Both components are prerequisites and without them, success is unlikely.

Ingredients needed for successful CMMS implementation

As in cooking, ingredients are first assembled to assure that the end result matches the recipe. This section lists the needed ingredients to be assembled prior to implementation of the CMMS.

1. Pilot Testing.

Usually companies purchase software based on what they observe in product demonstrations. Once they roll out the software to all users, they often discover that there are some crucial functions the software does not perform, or performs inadequately to meet their needs. To avoid this, the software program can be tested in-house for a predetermined period of time. One week is usually enough time for pilot testing. It is preferred to have a consultant or a trainer who is an expert in the software being evaluated to ensure that the program is being used correctly and certain features of the program are not missed or overlooked. During the testing, make sure the appropriate people are present. Include one buyer, one maintenance supervisor and someone from stores or inventory. This may not be practical if expensive hardware is required to run the software. However, many current CMMS are Web-based, so the pilot test would not be onerous to set up.

2. Review existing labor agreements.

This is even more important for companies with union environments. Job responsibilities may change as a result of the introduction of CMMS; as a result, labor agreements could be unknowingly violated.

3. Assemble a project team

Representatives from maintenance, purchasing, stores, production and Information Technology (IT) should be assembled to create the project team. Often companies assign mainly their IT personnel as the project teams. While including IT staff is important, the team should primarily consist of end users. They will be using the program on a day to day basis, not the IT staff. A project manager needs to be identified and this person should come from the maintenance group. The project manager has to ensure that the team works together towards the common goal of implementing the program, manage conflict, and be empowered to make binding decisions and enforce there implementation.

4. Reinforce behavior with incentives

The probability of meeting implementation objectives is increased with reinforcing behavior with positive incentives. Key members of the project team could receive a bonus for meeting measurable targets relevant to their role in the project. For example, the tool crib supervisor receives a bonus if there are no critical part shortages during a certain period of time. Incentives need to be distributed fairly. On one project, only the IT staff received a bonus once the system was up and running. This implied that there was inequality among the project team and it caused resentment and negative behavior. Everyone on the project team needs to be eligible for bonuses, not just the IT staff.

Also, timing is important. On the project mentioned above, the IT staff received their bonus for getting the program up and running, so naturally they moved onto their next project . Unfortunately, the end users were still learning the system and needed assistance, but the IT staff was no longer interested in helping them. Incentives can be strong motivators, but they need to be given out equitably and at the right time.

In addition to appropriately rewarding positive behavior, it is important to transfer knowledge during the project. Most likely a consultant(s) that work for the software vendor or for an independent consulting company will be hired to help implement the program. A consultant will orchestrate the process to make sure everyone is playing the same tune and help avoid common mistakes that tend to de-rail some implementations. Make sure an on-going transfer of knowledge from the consultant to the company is established. Many companies rely too heavily on these external resources, and one the implementation is complete, find themselves with inadequate knowledge to operate the program on an on-going basis. This could jeopardize the implementation.

A plan is need for the implementation and it must be in writing. Most implementation plans are typically written by consultants and they are based on input received from the project team. Many implementation plans are incomplete, they don't cover all areas of the implementation or go into sufficient detail. This happens because consultants have many other tasks and customers don't know what to expect in the implementation plan. An outcome of an incomplete plan is that parts of the system may not be implemented correctly. The next section details what should be included in the implementation plan, in ten distinct steps.

The CMMS Implementation Plan:

As in cooking, the order in which the process is implemented is important for a successful end result. A ten step CMMS implementation plan is described:

Step 1: Define Project Purpose and Benefits

The first step of the implementation plan is to define the purpose of the project and its benefits. Each area within the organization affected by the implementation should clearly understand the reasons for the implementation and why they should support it. For example, maintenance wants to reduce equipment downtime because each additional minute of uptime means more product out the door and more revenue. Purchasing wants easy access to history of items ordered and stores or inventory wants to know actual usage for stock items so they can order more effectively. If each group understand how the implementation can help they, they will be more willing to participate.

Step 2: Select Modules to Implement

Many implementations are multi phase, for example phase 1 is equipment and work orders, phase 2 is inventory and phase 3 is purchasing. Separating the project into phases makes it much more manageable and less overwhelming. Also,

implementing an entire program at once may not be feasible if there are many people to train- there may not be enough resources. The second step is to select and plan modules and the phases to be implemented.

Step 3. Create the implementation Schedule

A successful implementation requires a well planned schedule. The schedule is a list and description of all the tasks, their due dates and any deliverables for that task, such as written documentation. Here are some suggestions regarding creating the schedule:

- Break activities into measurable tasks

To complete each activity, specific tasks will need to be assigned to individual members of the project team. These tasks should be explicit and measurable. It must be possible to measure successful completion through some verifiable end product. Don't assume a task has been completed successfully because someone reported it as complete.

- Get agreement on target dates

When assigning tasks to project team members, ask them when they could complete an assignment, instead of just giving them a due date. Negotiate if necessary, but be sure to get agreement and hold the team member accountable for meeting the agreed upon date.

- Identify inter-dependant tasks:

Each phase of the project will consist of many activities, all requiring varying amounts of time to complete. Some activities can be scheduled concurrently; others can only be scheduled consecutively. By mapping out the interdependencies of these activities, the activities that must be completed on time in order to remain on schedule can be identified.

- Hold regular status meetings

Meetings with the project team should be held every two to three weeks to review the status of the various tasks. This is a very effective way of ensuring that the project remains on schedule. The meetings will provide an opportunity to identify and resolve problems, and they will keep the pressure on team members to meet target dates. Issuing minutes of the status meetings to team members' management or supervision can reinforce this pressure.

Step 4: Establish User Access to the Program Modules

Once the order of the implementation has been determined, user access is determined. This step includes deciding which users have access to which data, and the type of access. In most programs, data tables can be made read or view only to allow users to view the data but not modify it. For example, maintenance technicians commonly have full access to work orders, and read only rights to equipment, inventory and work history. Database field names can be changed to

match users own terminology and/or hidden if they are not needed. Unnecessary fields can make programs appear cluttered. This also protects the data from someone inadvertently modifying or deleting it and will make the program more user friendly as each person will have access only to the areas needed.

Step 5: Collect and Enter Data

Once users access has been established, they will begin collecting and entering data. Data will be constantly added to the database as the program is used. However, things will run more smoothly if basic data, such as equipment, parts and employees is already listed in the database prior to going live. Global system codes, such as cost center, craft, reason for outage codes, work order, inventory and equipment types need to be determined, if they have not been already. Methods of data collection and entry need to be decided. Will data be imported directly into the program, or will it be manually entered? Usually some data has to be manually entered. How this will be done and who will do it needs to be decided. After the program is rolled out, a limited number of people should have permission to enter new data to reduce risk of duplicate data entry, deleting data accidentally and non-compliance with data entry standards.

Step 6: Create Standard Operating Procedures (SOP's)

Standard Operating Procedures (SOP's) are customized step-by-step instructions for the operation of the CMMS program. The program manuals provided by the software manufacture are often too generic for fulfilling this function. Examples of tasks that should have customized SOP's are: creating and completing a work order, searching for a part, checking out parts from a warehouse, and receiving items on a purchase order. SOP's should have screen shots with detailed text. These procedures force the project team members to get together and make decisions about how the software will be used, according to their business processes. After the go live, SOP's can be used as training manuals for new employees and for refresher course for staff who have had training.

Step 7: Document All Completed Tasks

No technical assignment should be considered complete until the results of that assignment have been documented in writing. While many technical people avoid documentation, this requirement must be insisted upon for several reasons. Documentation enables the Project Manager to verify that the task has been completed successfully, the results can be communicated to other team members, and it ensures that the task has been thought through in sufficient detail to withstand scrutiny.

It is also necessary to translate technical information into common English. It may seem, at times, as if everyone on the team is speaking a different language. The Project Manager must be able to recognize when a team member is spouting "technical information, and ensure that it is translated into common English.

Step 8: Training

Several different types of training is needed to implement CMMS software. Core team training is one type, and it involves assembling a group of end users representing all departments (maintenance, purchasing, and stores or inventory). This small group learns how to use the program. Another type is end user training, which is training for everyone who will use the program and should take place just prior to going live. Otherwise, the trainees will forget what they have learned and need to be re-trained. There should be a system in which to track the trainee's progress. For each job there are tasks that will be done using the software. The trainee completes each task while being supervised by the trainer. This will help ensure a knowledge transfer from the trainer to the trainee.

In many implementations, some users lack the basic skills needed to use a computer, so basic windows or other training needs to be done if necessary.

Other resources, such as computers and training facilities, need to be available and set up for training sessions. Networks, servers and desktop computers need to be configured to support the application. The IT staff must be reliable and available to provide assistance when called upon.

Step 9: Go-Live Preparation

Once all the necessary resources are in place, preparations can be made to go-live (roll out) the software to the end users. At this stage the data should be loaded and the program is given a "stress test" to simulate all user functions including printing. Interfaces to other programs should be tested, if there are any. This is acceptance testing, which is making sure the program is acceptable to roll out to end users. This is the time to work out any bugs in the software, or at least be aware of them before the end users find them! The project teams' credibility will be enhanced if they let the end users know about the issues, instead of the other way around. There may be a few functions that aren't working perfectly but if they are not significant, the go-live should not be delayed.

Step 10: Go-Live!

The final Step is Go-live. Going live marks a major milestone in an implementation project. It is the time when all users start using the program. Users usually have a lot of questions, there may be technical problems, including software errors, and there might be issues with interfaces. Hopefully some of these issues were handled during the go-live preparation, but the volume of so many users working in the program at the same time can cause difficulties, and it is very difficult to test for this in advance. Adequate support staff need to be available for the effort. If it is a multi-plant effort, experienced personnel from another plant could assist.

The next section of this paper discusses two CMMS implementation where the companies involved shared many similarities. They were in the same industry, the software package came from the same vendor, they had adequate budgets in place

with the backing of senior management, and the same consultant lead both projects. Results, however, were quite different.

The first company will be referred to as company A. Company A is Steel producer in the Midwest. Prior to this implementation, they had a different CMMS and customized it so it would match their paper system. This meant they could not take advantage of software upgrades and patches made by that vendor. The software was not intuitive and was difficult to use. It was DOS based and if the exact keystroke that was required was not known, the user was lost. Sufficient training was not provided and the users struggled with the software, eventually abandoning it.

Company A then implemented a second maintenance software package (the one being discussed here) and were determined to learn from history and get it right this time. They handled things very differently. Customization to the software was not permitted. The person assigned to the project manager role was an engineer with superb organizational and computer skills. Company A set up a large training room with state of the art equipment and class sizes were deliberately small. Training manuals were written, tested and revised based on user feedback. Implementation was done slowly, and in stages. There were four maintenance supervisors who would become the primary users of the software and they reported directly to the project manager. The supervisors were part of the project implementation team. With all of this, total success seemed very likely.

One of company A's maintenance supervisors refused to participate in the implementation.
This supervisor was responsible for one of the plant's four mills, roughly a quarter of the plant reported to him. Although company A's Project Manager had the authority to make this reluctant supervisor use the program, he did not exercise his authority. The maintenance supervisor and everyone in his group did not learn how to use the software. Consequently, others in the other departments followed their example. This resulted in sporadic usage of the software and only a partial implementation.

The second company, Company B, is another steel producer. Company B had been tracking their maintenance on paper and unlike Company A, had no experience in implementing and using a CMMS. That was one disadvantage. Compared to company A, they did not have a large training room or any training room. Instead, a few computers were set up in an office and the trainees squeezed together. It was not possible to use common training tools such as a projector, flip chart, or even a white board in this cramped area. Company B had a smaller staff than Company A, resulting in a compressed training schedule due to lack of substitute staff for the training attendees.

To summarize, Company B had zero CMMS implementation experience, had cramped and inadequate training facilities and less time was spent on training due to lack of substitutes for the trainees.

Although Company B's assigned project manager was able to devote almost all of his time to the implementation, his organizational and computer skills were not as

strong as the project manager from Company A. Company B's project manager was a peer, not a supervisor to the maintenance supervisors who would become the primary users of the software. He was given the authority to oversee the implementation. In contrast to Company A, Company B's project manager used his authority to make sure everyone implemented the program appropriately. There were some reluctant users at Company B, which are found in most implementation projects. Company B's project manager did what he needed to do to accomplish 100% utilization of the new program.

Company B now has and fully uses an electronic system for both work orders and purchasing, phases 1 and 2 of the implementation. They followed up with phase 3- electronic work requests and phase 4- barcode for inventory and purchasing.

Company A is using their program for work orders only, and approximately three quarters of the plant is using the program.

To summarize, each company began their implementation as follows: Company A had previous experience implementing a CMMS, they enjoyed superior training resources and a
technically skilled Project Manager. Unfortunately, the project manager did not make sure that all users participated in the implementation effort.

Company B had no previous experience implementing or using a CMMS. They had limited training resources and their Project Manager had fewer technical skills that Company A's Project Manager, yet he firmly ensured that everyone participated in the CMMS.

The results are that Company A is using only a small fraction of their program's capabilities even though they spent considerable time and resources to implement the entire program.
Consequently, their reports are incomplete and they didn't get a good return on their investment and did not achieve their goals. In contrast, Company B is using all the functions of their program, as well as additional functions that were implemented in phases 3 and 4. Company A's reports are reliable, they enjoyed a high return on their investment and have exceeded their goals.

With every implementation, there are reluctant users who try to avoid working with a new system. Either they are afraid of the new technology, they don't want to make the effort to learn the new program, or they feel they don't have time to learn the new system. When these people are encountered, they need to be told the benefits of the new system as they relate to their jobs, receive additional training, be provided more instructions, receive guidance while they perform a process in the software, until they are comfortable. All of these can be done, but eventually the time may come when they need to be told firmly "you'll do this because it is your job, and if you don't, we will find someone else who will."

If end users are permitted to avoid using the new system, the success of the implementation at best will be limited. Even if some staff use the program, all the data will not be recorded and reports will be incomplete or wrong.

The overlooked ingredient to a successful implementation is full participation of all end users.

About the Author
Jennifer Ohl is a maintenance and reliability consultant based in Chicago and Miami. She was Regional Manager of Consulting Services for a leading maintenance software manufacturer and subsequently founded her own company, Midwest Software Specialists in 2001. For twelve years, Jennifer has been helping companies implement maintenance systems that reduce costs and increase profits. Jennifer has an MBA in Finance and Operations and a BA in Business.