

CWorks Implementation Guide

Implementing CWorks CMMS is no easy task, but if done correctly, CWorks may bring huge benefits. There are many implementation methods and this paper outline one of these methods.

To implement a solution such as CWorks, many steps are required due to the amount of information that can be collected. However trying to benefit from all of them at once is akin to trying to pass too many cookies through a small bottleneck. So we will break the implementation into various modules.

Setting Objectives

Before embarking on an implementation we have to be precise in the objectives. There are two main objectives that can be achieved by implementing CMMS, they are namely, reduce maintenance costs and increase equipment uptime. The table below shows how each of the objectives may be achieved.

Objective	Strategy	Tasks	CWorks modules
Reduce Maintenance Costs	Reduce labor costs	Control jobs Plan labor/materials	Work Order Location Employee Requester
	Reduce inventory storage costs	Analyze usage Analyze investment	Inventory
Increase Equipment Uptime	Reduce mean time to repair	Keep equipment data Keep location data Keep bill of material Keep employee info Keep contractor info Keep supplier info Manage spares stock Keep equipment history	Asset Location Employee Requester Work Order Inventory
	Increase mean time before failure	Analyze breakdown Schedule PM	Work Order Asset Location Preventive Maint.

Each of the strategy above can be implemented by itself. It is advised that the user chooses which objective and strategy that is most urgently needed and then implement it first. After that each of the other strategies can be implemented as needed. Implementing by objectives allows for the benefits of CWorks to be attained sooner. This is due to the fact that the implementation will be focussed and the tasks broken done into smaller doable chunks. Now let us look at the implementation tasks required.

Know the subject

After you have decided which objective, strategy, tasks and modules you want to implement, then scan through the user manual. There is no need to read everything but do get an understanding of each module that is to be implemented. It is imperative that it is understood how each module works.

Try the software

At this stage of the implementation, install CWorks. Installation steps are outlined in the user manual. Upon installation, two Cworks copies will be installed. One is the main CWorks and the other one is called CWorks Demo. Run Cworks Demo and you will find a Cworks that is filled with some example data. Look through the examples and create some dummy data on this database. Try different real-world examples on the modules you have selected to implement and this should give a better understanding of CWorks. At the end of this stage, the following need to be understood.

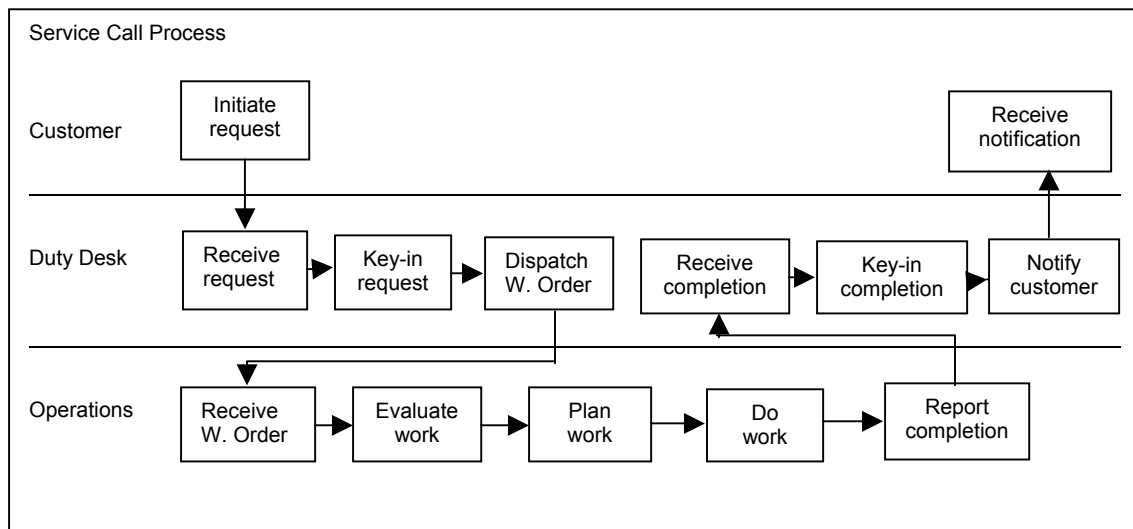
- What are the required fields
- What data need to be collected and when
- What information can be reported

Set benchmarks

Now go back to the objectives that have been set and do an audit of where the operations are currently. For example, if the objective is to reduce manpower costs then break down the costs into cost components such as overtime pay. Then calculate how much overtime is being paid currently. This figure will now be the benchmark against which improvement is to be targeted. Keep this list of benchmarks because it will be referred to throughout the use of CWorks as the basis of improvement works.

Develop, design, and document process

CWorks is a tool. The tool will not bring any benefit without being used. So we need to develop work processes to use CWorks. A process is a set of linked activities that take an input and transform it into an output. Also each activity should not only be defined but also assigned as a responsibility to someone. An example of a process for service call processing is displayed below.



It needs to be clarified that CWorks may be seen as a collection of forms, thus in terms of CWorks usage, we are working to fill forms. For each form or document that needs to be filled with information we need to create a work process. Therefore, to identify how many processes need to be designed, we just need to know how many different forms or documents that is needed.

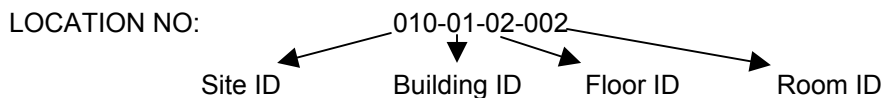
All processes should be documented and distributed to the workforce.

Develop, design, and document coding structure

At this stage, all the fields that are required should be known. For each of the field that has an item number such as asset no. Or location no., they will need a standard coding structure. This is to ensure that all the codes will have the same length and will be named the same way. This is to ensure uniqueness and will allow for easy sorting. For example, if two assets are of the same type, perhaps the type of the asset such as HVAC may prefix asset numbers like so – HVAC001.

It is important that all codes should not contain references to information that may change such as location names for assets that are movable. Also important to note is that, the level of coding must be appropriate. If for example, one is a fleet manager and all work is referenced to the automobiles in the fleet, then there is no need to assign asset numbers to the parts of the automobile. The parts should be given a part number and not an asset number.

An example of a code is as below.



Having a coding structure such as the example above will not only give the information to the technician to find the work site sooner but also allow the manager to sort the information in a more structured manner. Also the coding example ensures uniqueness of the record for easy querying.

The coding structures that are developed should be documented and distributed to the workforce.

Data collection and input

Once the coding structure has been documented, data collection of all the master data of the modules to be implemented should then commenced. The screens for these data are all available in the Masters module. Do remember that only data for the modules to be implemented are to be collected. If the objective is to implement inventory, then there is no need to collect requester data at this time.

It is advisable that this time is used also to do any improvement works so that any operational issues will not delay the CWorks implementation. For example, while collecting nameplate information for equipment, the equipment may also be cleaned at that time and any issues remedied. This will reduce demand for work during the initial stages of CWorks and thus allow for full concentration of the workforce on the change management at hand.

Apart from the equipment issues, people issues could also be remedied at this time. Getting employee data may easily be taken from the payroll but it would be more effective if each of the employees be interviewed and asked for their data. This would allow for current data to be collected and also allow for discussions on the change management such as process issues to take place.

As and when the master data are collected, it may also be keyed-in concurrently into the Master module.

Training

Once Cworks master data has been collected and entered. All processes should now be compiled into training notes or presentation slides and training should commence. Trainees should be divided into three groups namely data entry, operations and management.

Data entry should be trained on the forms that they have to key-in and know where the data is going to come from, where is it going and when. If the data entry is also a helpdesk, service quality issues may also be taught. Finally all processes that involve the data entry should be informed to them.

Unless operations personnel are required to key-in the data into CWorks themselves, there is no need to teach them CWorks. However they would need to be informed on the processes that affects them and assigned the proper responsibility. This will ensure that data would flow smoothly from the data creators (operations) to the data keeper (data entry). As some say, garbage in, garbage out, it is imperative that operations understand how and when data should be delivered. If failure codes for example are defined, then operations should use them rather than just recording "Work done" in the work order.

For management, they need to be run through the various screens to understand what data is being collected and also understand what reports is available. This training time is also the time to negotiate report delivery methods and times so that management will have the output that they require.

Test run

It is not advisable to run all the elements of CWorks at one go. Do small test runs by small locations first to ensure that everything is working. Test runs are also good to justify that the system is working and thus increase buy-in and reduce resistance to the change.

Go live

Once all tests are successfully completed, the whole site can then participate in the system and go live.

Performance monitoring

The success of the system depends greatly on a control feedback loop that is closed. This feedback loop is the reports read by management. Management should compare reports available to them against the benchmarks identified previously and assign resources as needed to achieve improvements on those benchmarks. Regularly reading the reports and actioning on them will also ensure that data integrity is achieved.

Summary

This paper outlines one method in implementing CWorks. There are other methods, which may or not be feasible for a particular site. However, the components of this guide should help in identifying the tasks required in reaping the benefit from a well-implemented Cworks.

Abdul Rani Achmed is the CEO of Clueword Dotcom Sdn Bhd, the producer of CWorks CMMS. He can be contacted at rani@clueword.com