

Safeguarding Your Plant Automation Systems with Change Management

By Gary Gillespie



Contents

Introduction	3
Change Management and the Plant	3
Impact of Plant Activities on Versions of Program Logic.....	4
User Roles the Types of Changes.....	5
Types of Risks.....	6
Device Type Considerations	7
Key Features of a CMS.....	8
CMS Implementation	8
Additional Product Considerations.....	9
Regulatory Considerations	10
Conclusion	10
Bibliography	10
Existing Application Reference.....	11
About The Author	12

Abstract

The increased use of plant floor automation to achieve production goals has created a dependency on PLCs, PC control systems and programmable automation. These devices and their logic programs are costly to develop but vital to the running of the plant, and are viewed by most companies as corporate assets. It is incumbent on plant and corporate management to insure that proper safeguards are in place to protect and manage change in these assets. This paper examines the sources and types of changes that take place in plant automation environments, and the considerations and approaches necessary to safeguard your automation systems through the effective use of a Change Management System (CMS).

Introduction

Georgetown University defines change management as the complete set of processes employed to ensure that changes are implemented in a visible, controlled and orderly fashion.⁽¹⁾ Focus should be placed on several key elements:

- 1.Process
Must be a structured method that stays the same
- 2.Completeness
Must include all objects (person, place or thing)
- 3.Visibility
Without visibility, Change Management doesn't exist
- 4.Control
Who, what and where
- 5.Order
Changes may need a particular sequence to be effective

Change Management as it applies to plant automation focuses on the control systems that operate the production equipment. Change Management System (CMS) applications have matured over the past 20 years along with the increases in sophistication and capability of the automation devices and control software developed by automation vendors.

There are many elements of a CMS that make its function distinct from single-file-based source code control systems, with the most distinct differences being a suite of tools to manage a group of files (often referred to as a "project") as a single entity, and processes for detecting source code changes outside of source code control. In most automation devices the entire project file is necessary when managing revisions and identifying changes.

Change Management and the Plant

An automation Change Management System is a centralized system that manages changes to program logic for controls programs and devices such as PLCs, CNCs, HMIs, PC control systems, robots, drives and general automation programs. A typical small plant will have a few hundred programs that should be managed, while large plants will have several thousand. Over the life of a facility the investment in program logic alone represents a significant expenditure that should be preserved and optimized. In order to do this a CMS should have the following features:

- An archive of prior revisions of programs
- The ability to detect changes
- Tools for documenting changes and making them visible to users
- A historical record of who made the change, when, and from where it was made
- Secured user and workstation access

- Features for controlling editor operations mapped to user permissions
- Procedures for recovering from hardware failures
- Change notification

As automation devices have grown more complex and have incorporated more plant data in their operation, there is an increase in the need to make adjustments to variables and logic to continue smooth operation. These adjustments may be minor individually, but are directly linked to machine throughput and uptime. If the current device program and configuration are lost, and an old version of the device program must be used, the result is decreased machine performance, decreased quality and/or downtime. While this situation is costly enough, consider the ramifications to plant operation if there are no older versions of a lost program available and the program must be completely rewritten. This can and does happen, and the effects can significantly impact safety and plant throughput for months. These impacts added to the cost to re-rewrite, test and commission a single program are often greater than the cost to implement a plant-wide CMS product.

Consider the ramifications to plant operation if there are no older versions of a lost program available and the program must be completely rewritten.

Impact of Plant Activities on Versions of Program Logic

Each facility has a unique set of change types and frequencies that can affect a CMS strategy. A selected set of activities is outlined here to prompt further thought and highlight the need for a proper implementation of a CMS in order to achieve optimum results.

- **Nature and Frequency of Changes:** It is important to ensure that an adequate number of program copies are available (including the “as delivered” copy from the vendor/integrator) to ensure that changes can be classified and reviewed. Some changes represent true improvements, while others highlight a process problem or training issue that should be addressed by other means.
- **Process Enhancements:** If changes are made in the process that make prior versions of the program obsolete, these enhancements should be clearly identified so that users do not revert to an older version of a program to fix a new issue. Plant operating guidelines should identify when the deletion of prior programs is warranted, and which users will have this permission.
- **Unmanaged Changes:** Without a CMS the controls engineer would use the editor software on a workstation or laptop to make changes in a device. If multiple people make changes from multiple computers (as is especially common in multi-shift operations) the documentation of changes is often lost. Using a CMS to compare the program running in the device with the last recorded version, a facility can identify changes that were made outside of the CMS. Once the CMS is implemented and sufficient device networking is in place, edits outside the CMS should be discouraged. These unmanaged edits are often prohibited in more regulated facilities using a CMS.
- **Temporary Changes:** It is common to make a “temporary” change to a program to resume operation while a maintenance task is performed on a failed component. It is also common for these temporary bypasses to be forgotten, which can result in serious safety issues. A CMS is used to note these temporary changes and provide a means of easily restoring a prior version of the program once maintenance is complete.
- **Multi-Process or Recipe Operations:** In facilities that run different processes or recipes it is important to manage which version of a program is being updated. The creation of

specialized copies of programs to use as “master versions” for each of these processes can aid in managing them efficiently.

**The remainder of this White Paper is available by request to
Michelle.Meyer@MDTSoft.com. We hope you find it to be of use in
managing change and safeguarding your automation assets!**