

## Software Control For An APM System

**Matthew S. Aldrich, Project Engineer**

*Lea Elliott*, Inc., Transportation Consultants, 785 Market Street, 12<sup>th</sup> Floor, San Francisco, CA 94103, PH: 415-908-6450, FX: 415-908-6451, EM: maldrich@leae Elliott.com

### *Abstract*

An APM system, like many every day products, has software programs that enable it to function safely, efficiently, and consistently. These software programs require configuration control processes. Software configuration control processes are essential for an APM, as numerous software programs must be logged, version tracked, and tested for safety and correct operation. Software can be controlled on several operational levels: Source Code, final installable executables, full or partial system backups, proprietary development packages, and/or programmed hardware ready for installation. The actual requirement will depend on many factors.

This paper will explore the software control methods of version tracking, escrow accounts, and archiving for the implementation and operational phases of a system (software development will not be discussed in detail). The implications of each aspect of the software control process will be related to what system components benefit the most from their application. Selected industry standards for the software in APM systems and the software development industry in general are listed at the end of this paper.

## **1 Introduction**

APM systems of today have software forming an essential component of the operational system. Firmware, operating systems, drivers, compilers, editors, user interfaces, system backups, and other APM system elements require control to verify a system is ready for safe passenger service. The methodology used for version control differs for each level of an APM systems life. Initially, the developers must verify the correct software versions when updates are shipped. Later, APM system installers must verify that all of the software is the correct version and faultlessly installed. Finally, Owner or operators/maintainers must verify all new and updated equipment has the correct and up-to-date software installed.

Relevant software standards offer some solutions and means to set-up successful and complete software control processes. Often a software control process does not follow a simple standard recipe. Innovative methods or uses for tools maybe required to create a manageable software control process.

## 2 Definitions/Acronyms

**Baseline** – Software Development milestone where the software configuration is frozen. Further development will reference this configuration as its origin.

**V & V** – Verification and Validation. Software code and programs are verified to follow the specification requirements and performance is validated via module testing, lifetime test, system integration testing, and final safety testing.

**APM** – Automated People Mover

**ATO** – Automated Train Operating Software

**ATP** – Automated Train Protection Software

**ATS** – Automated Train Supervisory Software

**CASE** – Computer Aided Software Engineering

**COTSS** – Commercial-Off-The-Shelf-Software

**CRC** – Cyclical Redundancy Check, advanced Checksum used to verify the integrity of a software file

**LRC** – Lowest Replaceable Component

**MOTSS** – Modified-Off-The-Shelf-Software

**OEM** – Original Equipment Manufacturer

**OS** – Operating system of an electronic device

**PC** – Personal Computer built to industrial specifications

**PROM** - Programmable Read Only Memory

**SCM** – Software Configuration Management

## 3 APM System Components Requiring Control

This section provides a general list of software system components usually controlled in a software control system. Not only does a final installable software disk itself require control, other elements of the APM software system require control beyond retention of the latest versions. Future system development and/or troubleshooting depend on these elements. The retention of these items could save essential time and considerable money when issues arise during an APM system's operation. Fixes that require a return to the source code will not require complete reverse or forward engineering. Instead the original source code can be revised with complete and efficient software control.

Figure 3.1 depicts the various relationships between the components of an APM software system.

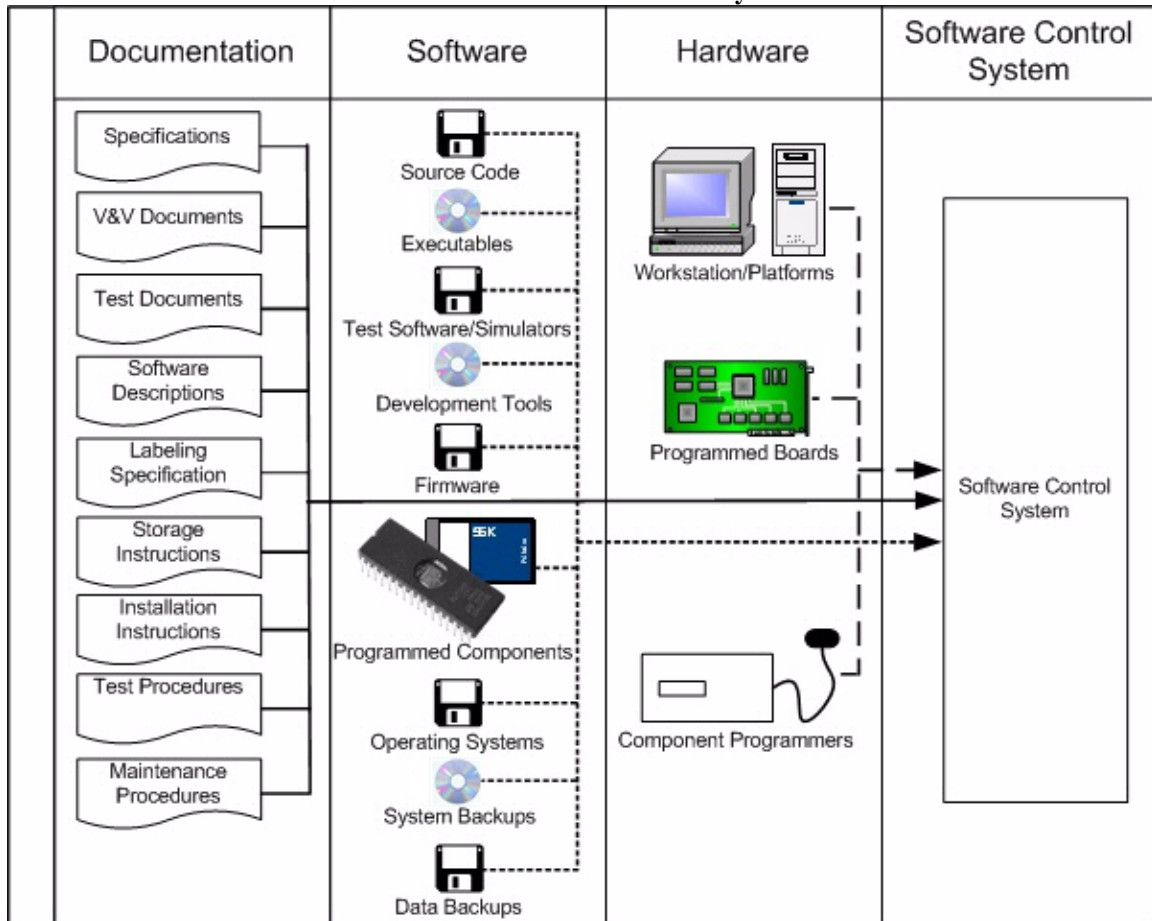


Figure 3.1 – Components of an APM Software System

### 3.1 Code Documentation

All final baseline code documentation, including specifications, test specifications, Verification and Validations (V & V) procedures and results, and software descriptions should be controlled and retained for the lifetime of the targeted equipment. The developer and owner should make the decision early in the process whether or not to retain obsolete version documentation indefinitely. If not, obsolete version documentation should be kept for typical duration of 6 months to a year.

### 3.2 Installation and Maintenance documentation

The Installation and Maintenance documentation include of software descriptions, labeling specifications, storage instructions, installation instructions, verification procedures, and maintenance procedures. During the operational life of an APM system, these are the most important documents to ensure system reliability and functionality during routine maintenance or repair. Inclusion of these documents in the software control system is a must. System installers and maintenance personnel require these documents to verify all the system software is installed and maintained correctly which improves system reliability and performance.

### **3.3 Source Code**

Most source code is proprietary and not directly available to APM customers except through an escrow account, if one exists. An APM supplier and end customer must negotiate early in the system development, what pieces of source code are essential for system stability, maintainability, and for possible upgrades in the future. The ATO, ATP, SCADA, ATS, availability reporting software, MMIS, and operating software source code latest baseline and released versions are essential and must be controlled and archived in an escrow account for an APM system. Software upgrades, troubleshooting, and system retrofits are much easier, cheaper, and last for shorter time durations when original source code can be used by software developers.

### **3.4 Test Beds/Simulations**

Test software, simulators, and system equipment used to test APM software must be documented, retained, and archived for future troubleshooting and/or upgrading of the system. In some cases where overhead expenses are too great to keep hardware items for future use, test configuration and specification documentation should be adequate to recreate any required test beds.

### **3.5 Development Tools**

Development tools must be documented and if possible archived for any APM system software. The tools may range from in-house function sets, CASE packages, Operating System/Development Environments, code compilers, to COTSS software packages. Also, an APM system supplier proprietary software tools must be archived for future system development or troubleshooting. APM suppliers and software developers may avoid some of the overhead costs of archiving the development tools with the use of portable code. Portable code allows a variety of development tools to be used to modify the resultant code. Software development and maintenance specifications must cover these requirements.

### **3.6 Executable software**

Executable software is installed in end devices within the APM System. Executables are essential to maintaining and/or installing an APM system. The media on which the software is stored must be logged, labeled, and stored in a safe yet accessible environment at the APM system locale. Precautions must be taken for this software because magnetic media, recordable CD-ROM and DVD media, and memory devices all have expected operational life, susceptibility to environmental damages, and handling limitations.

### **3.7 Firmware**

Firmware is machine executable code programmed or downloaded onto electronic hardware components (PROMs, processor boards, programmable logic, battery supported memory chips) of an APM system. This software may exist on a disk or in programmed components (see section 3.8) ready for installation into devices. Firmware software must be accompanied with any compilers, installers, or hardware required for installing the

firmware into the destination system. All devices, software, cabling, and installation procedures must be controlled and linked through the Software Control system.

### **3.8 Programmed Components**

Programmed components are composed of installable memory, electronic cards, or processors that contain firmware programs. In most APM systems, these are the Lowest Replaceable Components (LRC's) that maintenance and system suppliers install. These hardware devices require storage facilities that are more complex than the media used to store the software, i.e. static and dust-free storage. Some of these devices can also have shelf lives due to battery-powered components. As with all software, installation procedures for these devices must also be controlled and linked via software control.

### **3.9 Platforms**

Often-overlooked elements of an APM software system are hardware platforms. System software runs on these devices to perform numerous system operations. One such platform is the Personal Computer (PC). Today's APM systems often have the user interface and central control elements accessed by a PC. Software installed on PC's require configuration for the installed components of the PC and a specific OS.

In addition to software configuration control, the PC software and component obsolescence should also be considered during the software configuration control process. The obsolescence rate far outpaces the lifetime of a standard APM system. All APM system software requires specification of the platform and OS on which it will be installed. Platform requirements issues can occur as early as system installation. Moreover, the platform requirements must be defined for an APM system software programs early in the development cycle. The method of handling any platform obsolescence caused updates to the software should be negotiated on during project negotiations.

### **3.10 System Backups**

A possible maintenance process item, which would require software control, are system backups. System backups fall into two categories. The first category is data backups for the data the system creates during operation. The second backup category is a complete system backup of a fresh installation, with the most current software versions. These backups are used to quickly recover failed or electronic systems.

### **3.11 COTSS and MOTSS**

Commercial-Off-The-Shelf-Software (COTSS) development processes are not controlled by the system supplier. This type of software may have short lifetime availability, no support for future platforms (hardware/software), and/or inadequate documentation for use and system requirements. Many items fall under COTSS such as the OS for PC's, software developments suites (CASE), Software Configuration Management (SCM) programs, and other essential programs for an APM system.

The best practice for control of COTSS programs is to archive all the COTSS programs once installed so they may be reinstalled if required. The system supplier should limit using COTSS software to use on non-safety related equipment and equipment not essential to system operations. These recommendations will limit system downtime, software development times, and the cost for system troubleshooting or updates.

Modified-off-the-Shelf Software (MOTSS) has all of the same issues faced with COTSS with the added factor of tracking the specific modified code separately. This tracking should also include the off-the-shelf software version compatibility listing.

### **3.12 Obsolete Parts/Limited Lifetime Components**

Obsolescence affects every electronic portion of an APM system. The hardware and its installed software will become obsolete at some point in time. The best way to avoid obsolescence is for system providers to develop proprietary electronics. Thus the system supplier maintains full control over the availability of the system components. With this control, the system supplier can also verify the APM system software will function on each proprietary device. This is not always cost effective for the APM subsystems which are not directly related to control and safety of APM subsystems. For the non-critical systems it is more cost effective to utilize commercially available hardware as much as possible. These non-critical systems can be updated as necessary without serious impacts to system operations.

## 4 Possible Control Methods

This section discusses possible software control solutions. A common contributing factor in each solution is documentation. If a process is accurately documented, all software media and all hardware are labeled, then each user at any level or stage of the APM life cycle can determine the correct version of the software, the installation procedure, and the maintenance procedure.

The table below cross-references the software components mentioned in section 3 with the possible control methods of this section. Many control methods are listed multiple times, since multiple methods may work for several components.

Section	Software System Component	Control Method
3.1	Specifications	Archive Baselines
3.1	V&V Documents	Archive Baselines
3.1	Test Documents	Archive Procedures and Results
3.1	Software Descriptions	Archive, Site Storage (if required)
3.2	Labeling Specification	Archive, Site Storage
3.2	Storage Instructions	Archive, Site Storage
3.2	Installation Instructions	Archive, Site Storage
3.2	Test Procedures	Archive, Site Storage
3.2	Maintenance Procedures	Archive, Site Storage
3.3	Source Code	Escrow, Pseudo-Code, Site Store Hardcopy
3.6	Executables	Archive, Site Storage, Master Load Programs
3.4	Test Software/Simulators	Document or Store Components
3.5	Development Tools	Archive
3.7	Firmware	Site Storage, Archives, Master Load Programs
3.8	Programmed Components	Site Storage, Archive, Master Load Programs
3.9	Operating Systems	Site Storage, Archive
3.10	System Backups	Site Storage, Archive
3.10	Data Backups	Site Storage, Archive
3.9	Workstation/Platforms	Document
3.8	Programmed Boards	Site Storage
3.8	Programming Hardware	Site Storage, Document Procedures

**Table 4.1 – Software Components and Control Methods**

#### **4.1 Software System List**

An inventory of all APM software system components should be included in the documentation provided by the system supplier. This list may be a simple spreadsheet, a more complex database, COTSS logging program (often SCM function of a CASE program), or as component entries in the maintenance parts inventory system. This list should include version, release dates, the storage media, documentation name and revision, and storage location.

#### **4.2 Software Configuration Management (SCM) Software**

SCM software can be an integral part of the software development, system installation, and even the system maintenance organization software system control. SCM programs record all versions of the software, record current baselines, provide the latest released versions, and even allow users to reserve exclusive use of a software program. CASE software suites often have an SCM component. SCM software may not work as well for physical copies of software, since primarily the SCM software locks and stores software code versions on its resident computer system. SCM software used with physical copies, such as an APM system site, should be coupled with a specific software control procedure.

#### **4.3 Escrow**

Development documents, source code, and test bed tools for proprietary software, COTSS, MOTSS, OEM software, and OS software are usually not available to an APM system owner or maintainer. These items are always retained by the developing organization. All of the controllable software components developed for the APM system installer should be placed in Escrow for the life of the system. This process protects both the system provider and the end-user. If the provider were to either be sold and/or cease to exist, the end-user will still have access to the essential software tools for troubleshooting or upgrade. If the software developer has a catastrophic event that destroys the software source items, the escrow company will still retain the released baseline for each product or component escrowed. Escrow fees should be negotiated into any APM installation and/or maintenance contract.

#### **4.4 Hardcopies of Code/Pseudo-Code**

Another way the system provider may alleviate some of the risk to the end-user proprietary software risk is to provide a pseudo-code listing or a hard copy of the listing. These items will allow the user and/or the operation and maintenance organization a great analysis tool to resolve system issues. The system may be recreated from these items as well, but for a much higher cost if the original software source items are not available.

#### **4.5 Archiving**

Archiving of system software is the system supplier's responsibility. The software media and documentation should be stored in adequate facilities to ensure indefinite retention of all the released baselines. Media should be verified it will last indefinitely or the lifetime of the APM system.

#### **4.6 Site Storage**

Site storage falls under the responsibility of the system owner and/or the maintenance organization. The APM software components stored at the APM site can be anything from electronic media to programmed components. All site stored items must be clearly labeled with all the pertinent information required to identify the item as the correct version. Not only does the software or programmed components require clear and accessible storage, the required documentation to install and maintain the system component requires the same type of storage. The documentation and the actual device carrying the software must be clearly cross-referenced.

#### **4.7 Maintenance**

APM System software maintenance encompasses software updates, bug documentation, and procedures to make software updates. The entire software maintenance process must include all of the organizations that use, develop, or install the software. Much of the software control process falls under this umbrella.

Software maintenance has 5 phases that match the system development phases: the specification/definition/design phase, the development phase, the module and integrated testing, the installation phase and the operational phase. In each phase different criteria are placed on the software process for updates, control, and operation.

A specific software maintenance process must be in place as early as the software specification phases. The procedure governing the process should be adapted to each individual project as required by the end-user, Operations and Maintenance organization, and the system provider. The software control processes outlined in this paper should be integral parts of the maintenance process set forth.

#### **4.8 Master Load Programs**

Master load Programs compare the versions of all software modules against a master table before allowing the system to operate. These programs also run a CRC check to verify the integrity of all software modules prior to system operation. If either one of these checks do not match for any of the modules, the program aborts loading the other system software programs preventing system operations.

### **5 Review/conclusion**

Software control is a complex process that if done right will improve reliability, safety, performance, maintenance, interoperability, and supportability of an APM system. CASE and SCM suites aid development of system software. By controlling the development of the system software, the end-product software is correctly documented and meets all the system requirements. The installation effort of an APM system software benefits from software control. The system installers always know the correct version of software to install and test. They also know the software is meant for the installed hardware in the current system configuration. Installation procedures are also referenced in the software control system.

Operations and maintenance personnel need to know what software and version has to be installed in what system. When troubleshooting is required, the O & M providers will have the required documentation and information to aid the investigation and possible repair of the anomaly.

If a system ever requires retrofit or upgrade, having the source code, source documentation, test beds, test procedures, test results, original programming tools, and system specifications to a great extent improve the affordability and efficiency of implementation. Without any one of these essential pieces of software control system components, the entire retrofit process becomes more expensive and longer in duration.

Software control is a process that should be addressed in the very early specifications and negotiations for an APM system and carried through to the end of the system's life. In the process of creating and keeping a system safe, efficient, and consistently available to passengers, APM system software control plays an indispensable role.

## **6 Relevant standards**

These are industry standards an organization can reference when specifying a software control system.

### **6.1.1 IEEE 610.12-1990**

IEEE Standard Glossary of Software Engineering Terminology. This standard provides the basis of consistent terminology for communication related to software.

### **6.1.2 IEEE 730-1998**

IEEE Standard for Software Quality Assurance Plans. Defines the basic requirements of a Software Quality Assurance Plan (SQAP) and how an organization should develop the SQAP. The standard also defines the basic required documents for specifying and documenting a software system.

### **6.1.3 IEEE 828-1998**

IEEE Standard for Software Configuration Management Plans. Important standard for guiding an organization to the minimum details in a Software Configuration Management (SCM) Plan.

### **6.1.4 IEEE 830-1998**

IEEE Recommended Practice for Software Requirements Specifications. Standard that describes the process of documenting the required design elements of developed software, Commercial-of-the-Shelf Software (COTSS), or other software essential for a product.

#### 6.1.5 IEEE 1012-1998

IEEE Standard for Software Verification and Validation. Defines the process of determining whether or not a developed piece of software meets the Software requirements. Verification and Validation (V & V) is a process that should be revisited for every software change made to a system.

#### 6.1.6 IEEE 1012a-1998

IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997. Essential for verifying an organization will produce documents that comply with IEEE-1012 and IEEE/EIA-12207.1.

#### 6.1.7 IEEE 1028-1997

IEEE Standard for Software Reviews. Important standard for software system auditors and quality organizations. Defines the processes used to meet IEEE-830.

#### 6.1.8 IEEE 1042-1987

IEEE Guide to Software Configuration management. This standard lays out the planning and implementation of a SCM program to comply with IEEE-828.

#### 6.1.9 IEEE 1228-1994 (R2002)

IEEE Standard for Software Safety Plans. This standard applies to safety-critical software in a system. A system compliant with this standard develops the software and evaluates the safety-critical software elements within the over-all system safety process.

#### 6.1.10 ASCE 21-96

ASCE Automated People Mover Standards Part 1. Section 3 details the safety requirements for an APM system. This standard encompasses system software in the system safety process including Verification and Validation.

#### 6.1.11 AREMA Communication and Signals Manual Section 17.5 - Quality Principles- Recommended Configuration Management Program for Electronic/Software Based Products Used in Vital Signal Applications (R2000)

Document recommends a methodology of managing and documenting the configuration of products, which are vital to the safety, and reliability of a system during the development and life cycle.

#### 6.1.12 AREMA Communications and Signals Manual Section 17.6 – Quality Principles Recommended Guidelines for Conducting Quality and Safety Audits of Electronic/Software Based Products Used in Vital Signal Applications.

Explains the guidelines for auditing organization conduction an audit of Electronic and/or Software systems in Vital Signal equipment.