Configuration management

10.1 Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information throughout its life. The CM process is widely used by military engineering organizations to manage complex systems, such as weapon systems, vehicles, and information systems. Outside the military, the CM process is also used with IT service management as defined by ITIL, resp. ISO/IEC 20000, and with other domain models in the civil engineering and other industrial engineering segments such as roads, bridges, canals, dams, and buildings.

Introduction

CM, when applied over the life cycle of a system, provides visibility and control of its performance, functional and physical attributes. CM verifies that a system performs as intended, and is identified and documented in sufficient detail to support its projected life cycle. The CM process facilitates orderly management of system information and system changes for such beneficial purposes as to revise capability; improve performance, reliability, or maintainability; extend life; reduce cost; reduce risk and liability; or correct defects. The relatively minimal cost of implementing CM is returned many fold in cost avoidance. The lack of CM, or its ineffectual implementation, can be very expensive and sometimes can have such catastrophic consequences such as failure of equipment or loss of life.[6]

CM emphasizes the functional relation between parts, subsystems, and systems for effectively controlling system change. It helps to verify that proposed changes are systematically considered to minimize adverse effects. Changes to the system are proposed, evaluated, and implemented using a standardized, systematic approach that ensures consistency, and proposed changes are evaluated in terms of their anticipated impact on the entire system. CM verifies that changes are carried out as prescribed and that documentation of items and systems reflects their true configuration. A complete CM program includes provisions for the storing, tracking, and updating of all system information on a component, subsystem, and system basis.[7]

A structured CM program ensures that documentation (e.g., requirements, design, test, and acceptance documentation) for items is accurate and consistent with the actual physical design of the item. In many cases, without CM, the documentation exists but is not consistent with the item itself. For this reason, engineers,

contractors, and management are frequently forced to develop documentation reflecting the actual status of the item before they can proceed with a change. This reverse engineering process is wasteful in terms of human and other resources and can be minimized or eliminated using CM.

History

Configuration Management originates in the United States DoD in the 1950s as a technical management discipline for hardware material items—and it is now a standard practice in virtually every industry. The CM process became its own technical discipline sometime in the late 1960s when the DoD developed a series of military standards called the "480 series" (i.e., MIL-STD-480 and MIL-STD-481) that were subsequently issued in the 1970s. In 1991, the "480 series" was consolidated into a single standard known as the MIL–STD–973 that was then replaced by MIL–HDBK–61 pursuant to a general DoD goal that reduced the number of military standards in favor of industry technical standards supported by standards developing organizations (SDO).[8] This marked the beginning of what has now evolved into the most widely distributed and accepted standard on CM, ANSI–EIA–649–1998.[9] Now widely adopted by numerous organizations and agencies, the CM discipline's concepts include systems engineering (SE), integrated logistics support (ILS), Capability Maturity Model Integration (CMMI), ISO 9000, Prince2 project management methodology, COBIT, Information Technology Infrastructure Library (ITIL), product lifecycle management, and application lifecycle management. Many of these functions and models have redefined CM from its traditional holistic approach to technical management. Some treat CM as being similar to a librarian activity, and break out change control or change management as a separate or stand alone discipline.

Overview

CM is the practice of handling changes systematically so that a system maintains its integrity over time. CM implements the policies, procedures, techniques, and tools that are required to manage, evaluate proposed changes, track the status of changes, and to maintain an inventory of system and support documents as the system changes. CM programs and plans provide technical and administrative direction to the development and implementation of the procedures, functions, services, tools, processes, and resources required to successfully develop and support a complex system. During system development, CM allows program management to track requirements throughout the life cycle through acceptance and operations and maintenance. As changes are inevitably made to the

requirements and design, they must be approved and documented, creating an accurate record of the system status. Ideally the CM process is applied throughout the system lifecycle.

The CM process for both hardware and software configuration items comprises five distinct disciplines as established in the MIL–HDBK–61A and ANSI/EIA-649. These disciplines are carried out as policies and procedures for establishing baselines and performing a standard change management process.

- CM Planning and Management: A formal document and plan to guide the CM program that includes items such as: Personnel; Responsibilities and Resources; Training requirements; Administrative meeting guidelines, including a definition of procedures and tools; baselining processes; Configuration control and Configuration status accounting; Naming conventions; Audits and Reviews; and Subcontractor/Vendor CM requirements.
- Configuration Identification (CI): Consists of setting and maintaining baselines, which define the system or subsystem architecture, components, and any developments at any point in time. It is the basis by which changes to any part of an information system are identified, documented, and later tracked through design, development, testing, and final delivery. CI incrementally establishes and maintains the definitive current basis for Configuration Status Accounting (CSA) of a system and its configuration items (CIs) throughout their lifecycle (development, production, deployment, and operational support) until disposal.
- Configuration Control: Includes the evaluation of all change requests and change proposals, and their subsequent approval or disapproval. It is the process of controlling modifications to the system's design, hardware, firmware, software, and documentation.
- Configuration Status Accounting: Includes the process of recording and reporting configuration item descriptions (e.g., hardware, software, firmware, etc.) and all departures from the baseline during design and production. In case of suspected problems, the verification of baseline configuration and approved modifications can be quickly determined.
- Configuration Verification and Audit: An independent review of hardware and software for the purpose of assessing compliance with established performance requirements, commercial and appropriate military standards, and functional, allocated, and product baselines. Configuration audits verify the system and subsystem configuration documentation complies with their

functional and physical performance characteristics before acceptance into an architectural baseline.

Software

The traditional software configuration management (SCM) process is looked upon by practitioners as the best solution to handling changes in software projects. It identifies the functional and physical attributes of software at various points in time, and performs systematic control of changes to the identified attributes for the purpose of maintaining software integrity and traceability throughout the software development life cycle.

The SCM process further defines the need to trace changes, and the ability to verify that the final delivered software has all of the planned enhancements that are supposed to be included in the release. It identifies four procedures that must be defined for each software project to ensure that a sound SCM process is implemented. They are:

1. Configuration identification
2. Configuration control
3. Configuration status accounting
4. Configuration audits

These terms and definitions change from standard to standard, but are essentially the same.

- Configuration identification is the process of identifying the attributes that define every aspect of a configuration item. A configuration item is a product (hardware and/or software) that has an end-user purpose. These attributes are recorded in configuration documentation and baselined. Baselining an attribute forces formal configuration change control processes to be effected in the event that these attributes are changed.
- Configuration change control is a set of processes and approval stages required to change a configuration item's attributes and to re-baseline them.
- Configuration status accounting is the ability to record and report on the configuration baselines associated with each configuration item at any moment of time.
- Configuration audits are broken into functional and physical configuration audits. They occur either at delivery or at the moment of effecting the change. A functional configuration audit ensures that functional and performance attributes of a configuration item are achieved, while a physical

configuration audit ensures that a configuration item is installed in accordance with the requirements of its detailed design documentation.

Configuration management database

The Information Technology Infrastructure Library, also known as ITIL, specifies the use of a Configuration management database (CMDB) as a means of achieving industry best practices for Configuration Management. CMDBs are used to track Configuration Items (CIs) and the dependencies between them, where CIs represent the things in an enterprise that are worth tracking and managing, such as but not limited to computers, software, software licenses, racks, network devices, storage, and even the components within such items.

The benefits of CMDBs include being able to perform functions like root cause analysis, impact analysis, change management, and current state assessment for future state strategy development.

Information assurance

For information assurance, CM can be defined as the management of security features and assurances through control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation throughout the life cycle of an information system.[10] CM for information assurance, sometimes referred to as **S**ecure **C**onfiguration **M**anagement, relies upon performance, functional, and physical attributes of IT platforms and products and their environments to determine the appropriate security features and assurances that are used to measure a system configuration state. For example, configuration requirements may be different for a network firewall that functions as part of an organization's Internet boundary versus one that functions as an internal local network firewall.

Maintenance systems

Configuration management is used to maintain an understanding of the status of complex assets with a view to maintaining the highest level of serviceability for the lowest cost. Specifically, it aims to ensure that operations are not disrupted due to the asset (or parts of the asset) overrunning limits of planned lifespan or below quality levels.

In the military, this type of activity is often classed as "mission readiness", and seeks to define which assets are available and for which type of mission; a classic

example is whether aircraft on board an aircraft carrier are equipped with bombs for ground support or missiles for defense.

Operating System configuration management

Configuration management can be used to maintain OS configuration files.[11] Example systems include Quattor, CFEngine, Bcfg2, Puppet, and Chef.

A theory of configuration maintenance was worked out by Mark Burgess,[12][13][14] with a practical implementation on present day computer systems in the software CFEngine able to perform real time repair as well as preventive maintenance.

Preventive maintenance

Understanding the "as is" state of an asset and its major components is an essential element in preventive maintenance as used in maintenance, repair, and overhaul and enterprise asset management systems.

Complex assets such as aircraft, ships, industrial machinery etc. depend on many different components being serviceable. This serviceability is often defined in terms of the amount of usage the component has had since it was new, since fitted, since repaired, the amount of use it has had over its life and several other limiting factors. Understanding how near the end of their life each of these components is has been a major undertaking involving labor-intensive record keeping until recent developments in software.

Predictive maintenance

Many types of component use electronic sensors to capture data which provides live condition monitoring. This data is analyzed on board or at a remote location by computer to evaluate its current serviceability and increasingly its likely future state using algorithms which predict potential future failures based on previous examples of failure through field experience and modeling. This is the basis for "predictive maintenance".

Availability of accurate and timely data is essential in order for CM to provide operational value and a lack of this can often be a limiting factor. Capturing and disseminating the operating data to the various support organizations is becoming an industry in itself.

The consumers of this data have grown more numerous and complex with the growth of programs offered by original equipment manufacturers (OEMs). These are designed to offer operators guaranteed availability and make the picture more complex with the operator managing the asset but the OEM taking on the liability to ensure its serviceability. In such a situation, individual components within an asset may communicate directly to an analysis center provided by the OEM or an independent analyst.

Standards

- ANSI/EIA-649-1998 National Consensus Standard for Configuration Management
- EIA-649-A 2004 National Consensus Standard for Configuration Management
- TechAmerica/ANSI EIA-649-B 2011 Configuration Management Standard
- ISO 10007:2003 Quality management systems - Guidelines for configuration management
- Federal Standard 1037C
- GEIA Standard 836-2002 Configuration Management Data Exchange and Interoperability
- IEEE 829 Standard for Software Test Documentation
- MIL-STD-973 Configuration Management (cancelled on 20 September 2000)
- STANAG 4159 NATO Materiel Configuration Management Policy and Procedures for Multinational Joint Projects
- STANAG 4427 Introduction of Allied Configuration Management Publications (ACMPs)
- ACMP 2100 (DRAFT) NATO Contractual Configuration Management Requirements
- CMMI CMMI for Development, Version 1.2 Configuration Management
- CMII-100E CMII Standard for Enterprise Configuration Management
- Extended List of Configuration Management & Related Standards

Guidelines

- 828-2012 Currently active IEEE Standard which supersedes/supports older ones.
- MIL-HDBK-61A Configuration Management Guidance 7 February 2001
- 10007 Quality management - Guidelines for configuration management
- ACMP 2009 (DRAFT) NATO Guidance on Configuration Management

- GEIA-HB-649 - Implementation Guide for Configuration Management
- ANSI/EIA-649-1998 National Consensus Standard for Configuration Management
- EIA-836 Consensus Standard for Configuration Management Data Exchange and Interoperability
- ANSI/EIA-632-1998 Processes for Engineering a System
- MIL-STD-3046 (ARMY) Interim Standard on Configuration Management, 6 March 2013

Construction

More recently configuration management has been applied to large construction projects which can often be very complex and have a huge amount of details and changes that need to be documented. Construction agencies such as the Federal Highway Administration have used configuration management for their infrastructure projects.[15] There are construction-based configuration management tools that aim to document change orders and RFIs in order to ensure a project stays on schedule and on budget. These programs can also store information to aid in the maintenance and modification of the infrastructure when it is completed. One such application, ccsNet, was tested in a case study funded by the Federal Transportation Administration (FTA) in which the efficacy of configuration management was measured through comparing the approximately 80% complete construction of the Los Angeles County Metropolitan Transit Agency (LACMTA) 1st and 2nd segments of the Red Line, a $5.3 billion rail construction project. This study yielded results indicating a benefit to using configuration management on projects of this nature.[16]