



**MERANT**

**Principles of Effective Software  
Configuration Management:**

**Reducing the cost of SCM to near Zero**

**Bob Ventimiglia**

# Agenda

- Principles of Effective SCM
  - What is Software Configuration Management?
    - ◆ Some definitions and non-definitions
  - Principles & Characteristics of Effective SCM
  - Rule-Based, Closed-Loop Change Control
  - The Handwriting on the Wall
- Lockheed Martin Experience & Benefits
  - Overview of the C130 Family
  - C130J Software Development Process
  - General Features of the C130J Effective SCM Implementation
  - Features from an SCM Viewpoint
  - Benefits of the Implementation

# What is Software Configuration Management (SCM)?

## IEEE Definition

- "Configuration management is a discipline applying **technical** and **administrative** direction and **surveillance** to:
  - (a) Identify and document the functional and physical characteristics of **configuration items** (CIs)
  - (b) Audit the **configuration items** to verify conformance to specifications, interface control documents, and other contract requirements
  - (c) Control **Changes** to **configuration items** and their related **documentation**
  - (d) Record and report information needed to manage **configuration items** effectively, including the status of proposed changes and the implementation status of approved changes" IEEE 610.12-1990

# What is SCM? - Other Definitions

- **Configuration Management is the practice of handling changes systematically so that a system can maintain its integrity over time. Another name for it is “change control”. It includes techniques for evaluating proposed changes, tracking changes, and keeping copies of the system as it existed at various points in time.** Steve McConnell, “Code Complete”, Microsoft Press, 1993

**Incorrect Definition  
from an Outstanding Developer & Author**

# What is SCM? - more definitions

- Software Configuration Management is the **process** of identifying, organizing, controlling, and tracking both the decomposition and re-composition of: software structure, functionality, evolution, and teamwork. In short, SCM is the "**glue**" between software artifacts, features, changes, and team members; it forms **the ties that bind** them all together from **concept to delivery and beyond**. Brad Appleton, <http://www.enteract.com/~bradapp/acme/scm-defs.html#Appleton>.

- SCM is "a **disciplined** approach to **managing** the **evolution** of software products and the related development and maintenance practices" S. Dart, "Not ALL TOOLS are Created EQUAL", Application Development Trends, 1997

**Better Definitions**

# What is SCM? – A Better Definition

- “Configuration Management is a **management tool** that defines the product and controls changes to that product.”
- Mike goes on to say, “Configuration Management is a way to communicate. ...
  - If we **use** configuration management **properly**, we should **know** what we *are supposed to build or produce* and we should know at any point in time what we *are building or producing*. And, once we finish, we should know what we **have produced or built**. ...
  - The bottom line is – configuration management is an **effective, prudent, and logical function of project management** with a set of rules. ...
  - The **goal** is to set up the **simplest system** possible to achieve the desired result of control. If you can make the system so it’s **totally automatic and transparent**, then you have achieved a **perfect system**.”

“Principles of Configuration Management”, M. A. Daniels, 1985

# BOBEV's Definition of SCM, 2001

- Software Configuration Management is:

A **management tool** that applies an **engineering discipline** to manage the evolution of a software system from concept through to system retirement. The primary focus of this discipline is to ensure the **repeatability**, **traceability**, and **integrity** of the system being developed and produced. So that management knows what is supposed **to be** built, what **is being** built and what **has been** built.

**Watchwords of CM:**

**Repeatability, Traceability, and Integrity**

**All Actions determined by impact on the watchwords of CM**

# The Principles of Effective SCM

- **Proactive:**
  - Technical: a system engineering function
  - Define and Implement
- **Flexible:**
  - Satisfies the **needs** of **management** and **developers**
  - **Follow the Files** that need to be changed!
- **Integrated:**
  - Integral & Unobtrusive
    - ◆ 100% development team executed
- **Visible:**
  - **Embraces, facilitates and manages change(s)**
  - Manages **evolving** process model

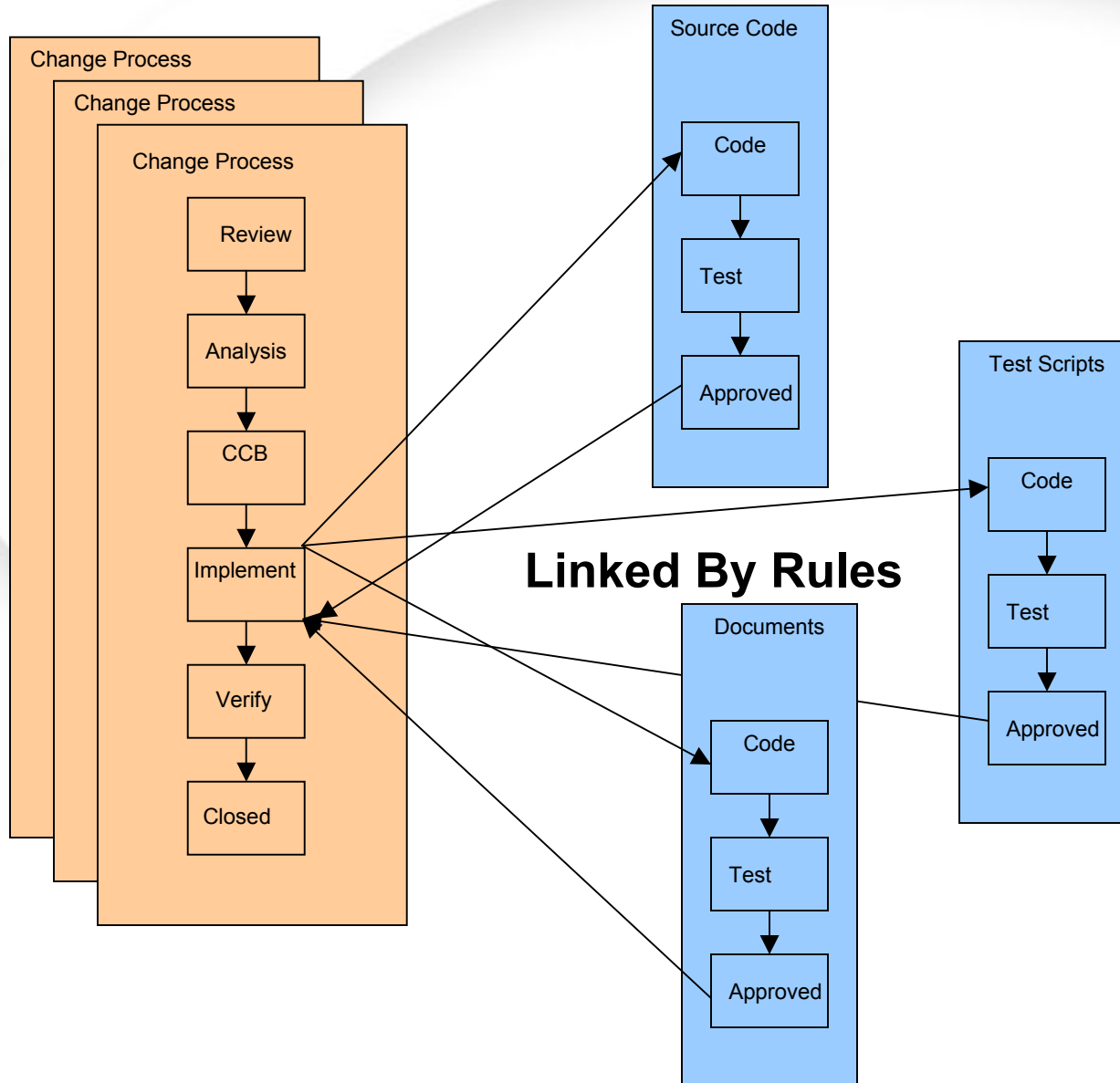
**Tool Based Software Lifecycle Management**

# Characteristics of Effective SCM

- Electronic Repository of **ALL** Software Development Artifacts
- ALL Artifacts are versioned following **specific lifecycles**
- **ALL** Artifacts carry **history** logs
  - Name, date, time stamp, status for every transaction
- GUI Based
- **Electronic enforcement** of business **objectives**

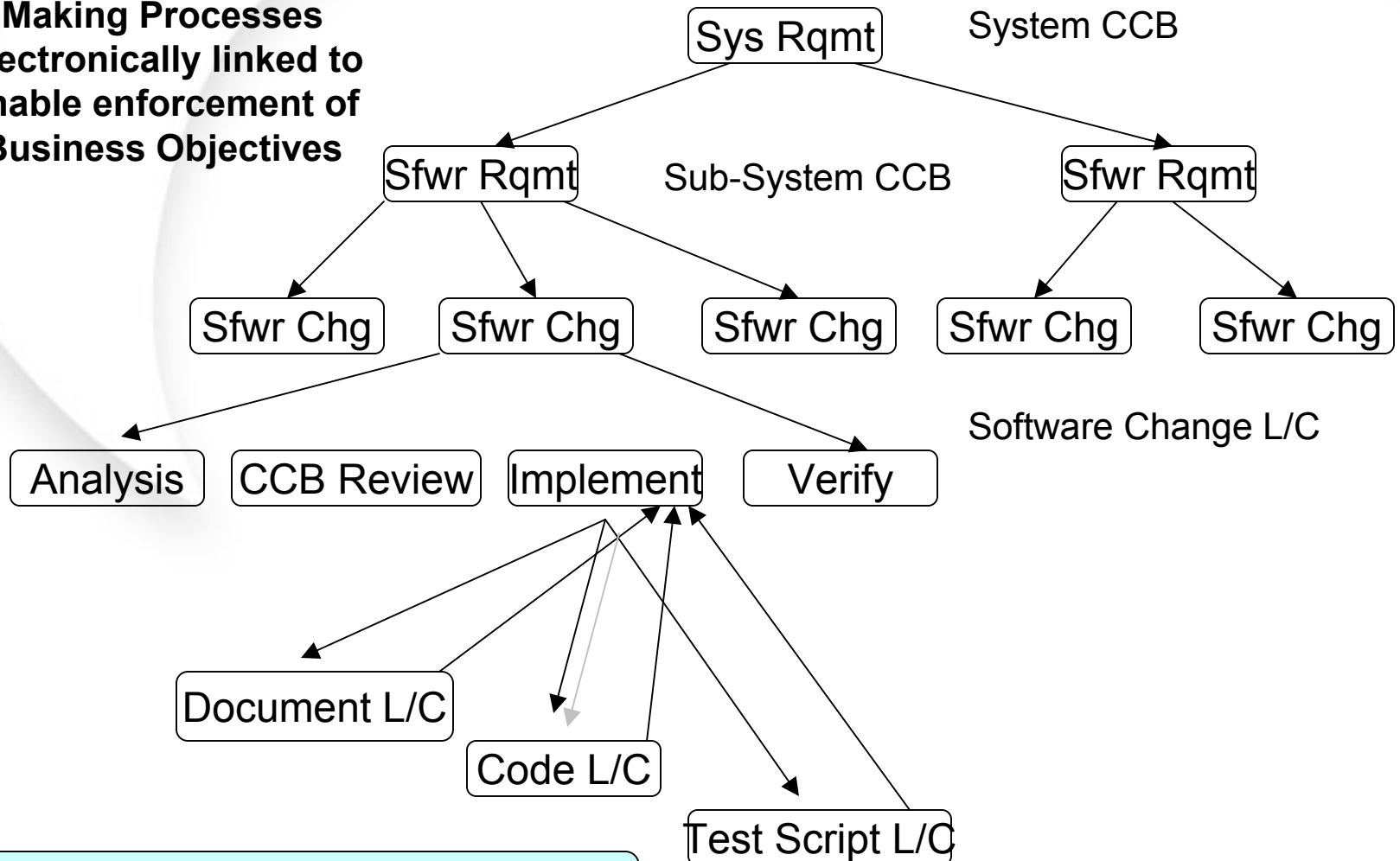
**Objective: 100% Development Team Executed SCM**

# Rule-Based Closed-Loop Change Control



# Hierarchical Change Processes with Rule-Based Closed-Loop Change Control

Multiple Levels of Decision Making Processes Electronically linked to enable enforcement of Business Objectives



**Integrated Process Management**

# The Handwriting on the Wall

- Change Happens
  - They Keep Moving the Cheese
- Anticipate Change
  - Get Ready for the Cheese to Move
- Monitor Change
  - Smell the Cheese Often, So You Know When It Is Getting Old
- Adapt to Change Quickly
  - The Quicker You Let Go of Old Cheese, The Sooner You Can Enjoy New Cheese
- Change
  - Move with the Cheese
- Enjoy Change!
  - Savor the Adventure and the Taste of New Cheese!
- Be Ready to Quickly Change Again and Again
  - They Keep Moving the Cheese

(from “Who Moved My Cheese”, Spencer Johnson, M.D.)

# Effective Software Configuration Management a C-130J Software Development Success Story



**Bob Ventimiglia**  
**Team Coach**  
**Environment, Tools, and SCM**

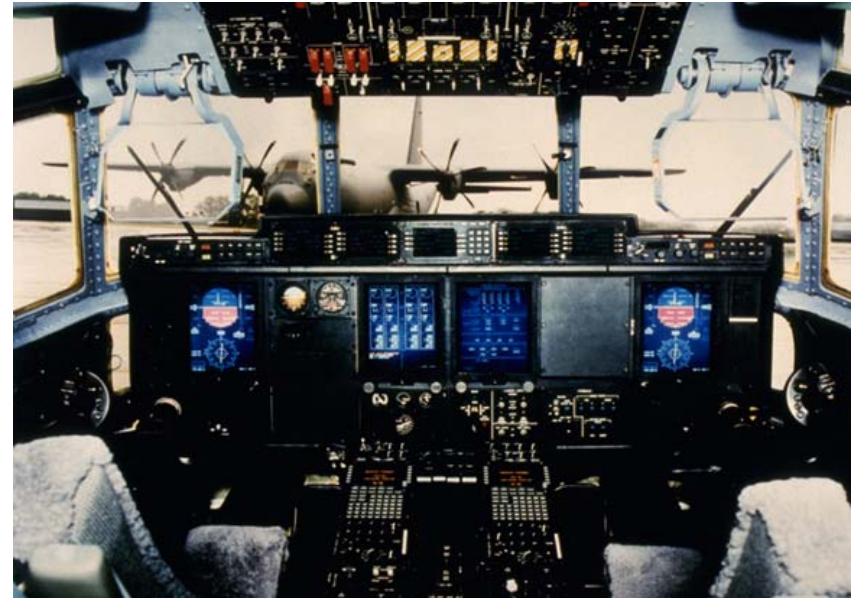
Contacting Me <mailto:bobev@bobev.com>; or [robert.ventimiglia@lmco.com](mailto:robert.ventimiglia@lmco.com)

# Overview of the C130 Aircraft

- First named “Hercules,” the C-130 has become a legend,
  - More than 2,100 C-130’s built since 1958
  - Purchased by over 60 nations in dozens of variations
- The C-130:
  - Carries troops, vehicles, and armaments into battle
  - Drops paratroopers and supplies
  - Serves as airborne and ground refuelers
  - Provides emergency and humanitarian relief (even acting as hospital ships)
  - Does airborne early warning and maritime surveillance (it even flies into hurricanes)

# Overview of the C-130J Aircraft

- The new C-130J looks like the original on the outside, but it is vastly improved:
  - 21% faster, 40% higher, 40% longer range
  - Reduced manpower (aircrew of 2 instead of 5), operating costs, support costs, lifecycle costs
  - A new propulsion system (29% more thrust with 15% more fuel efficiency)
  - Advanced avionics technology



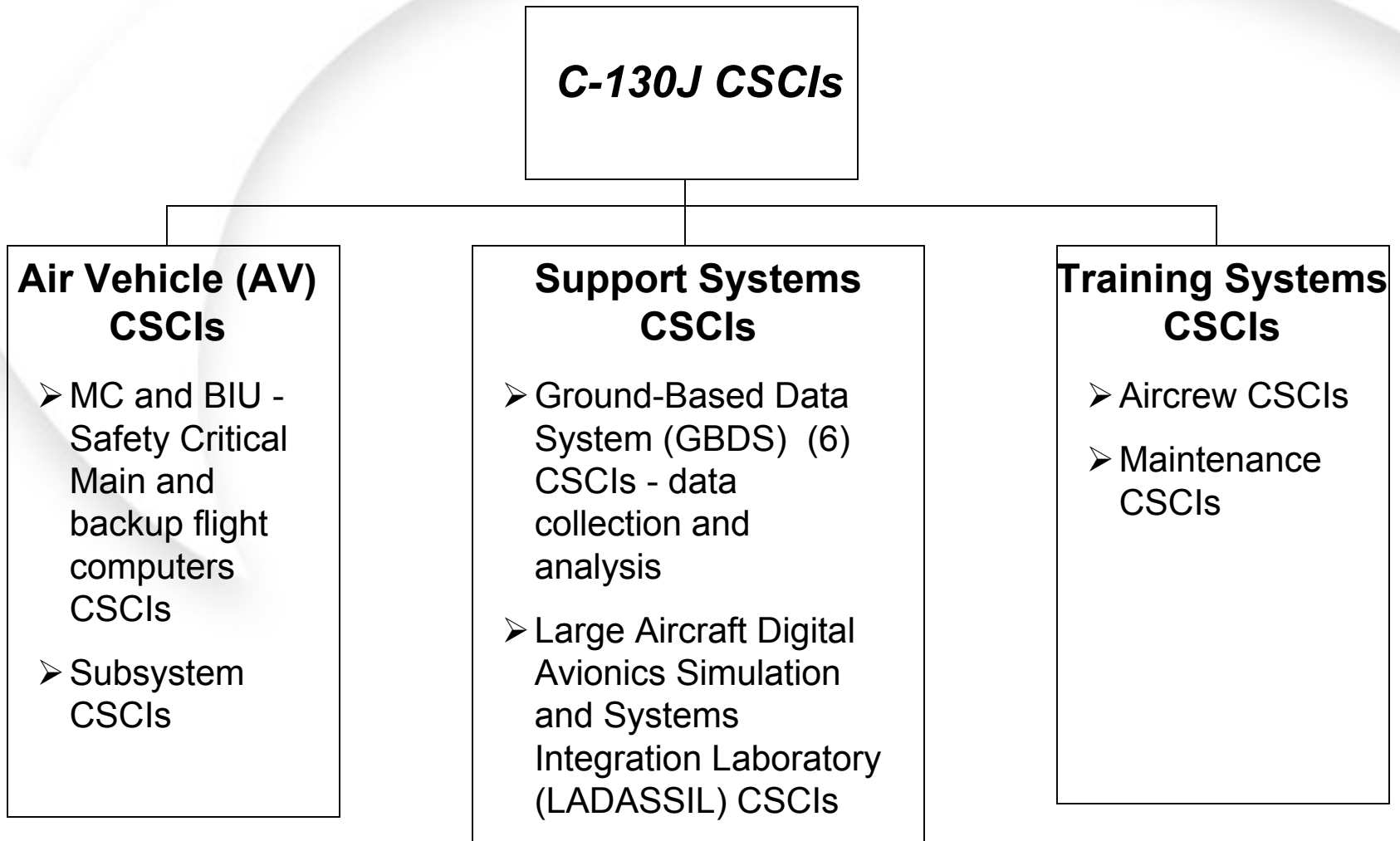
**The State-of-the-Art Cockpit of the C-130J**

# Classes of Aircraft Software

- Block 2 - the 382J Aircraft
  - The base class upon which the C-130J is based
  - Developed to FAA DO-178B standard
  - Received FAA Type Certification Aug 98
- Block 3 - the basic C-130J Aircraft
  - Inherits from the 382J Aircraft
- Block 4 - the Variants of the basic C-130J Aircraft
  - Unique versions of the C-130J modified for several customers, including, but not limited to:
    - ◆ United States Air Force (2 Variants)
    - ◆ Royal Air Force - United Kingdom (2 interim variants plus final)
    - ◆ Royal Australian Air Force

**Concurrent, Parallel Development of ALL Variants during 1997-98  
While Achieving SEI Level 3 and ISO 9003 Certifications!**

# C-130J CSCI Hierarchy



*There are more than 50 Air Vehicle CSCIs for each Block or Block 4 Variant.*

# Features of of the C130J Effective SCM Implementation

- Software Development Process managed using an automated, process driven, rule-based, closed-loop change control, tool - PVCS/Dimensions (nee PCMS)
  - ◆ Achieving a 90% development team executed SCM environment
  - ◆ Integrated electronic development and change processes
- Electronic enforcement of business rules
  - ◆ No unauthorized changes
  - ◆ No closure of incomplete changes
  - ◆ No loss of visibility into change authorization hierarchy
- Facilitated concurrent, parallel development and test of more than 7 variations of 8 software configuration items
- 1-2 hr, 1:1 User Training when account created
  - ◆ an early brainwashing to facilitate paradigm shift

**Accommodates, embraces, facilitates, and leverages change!**

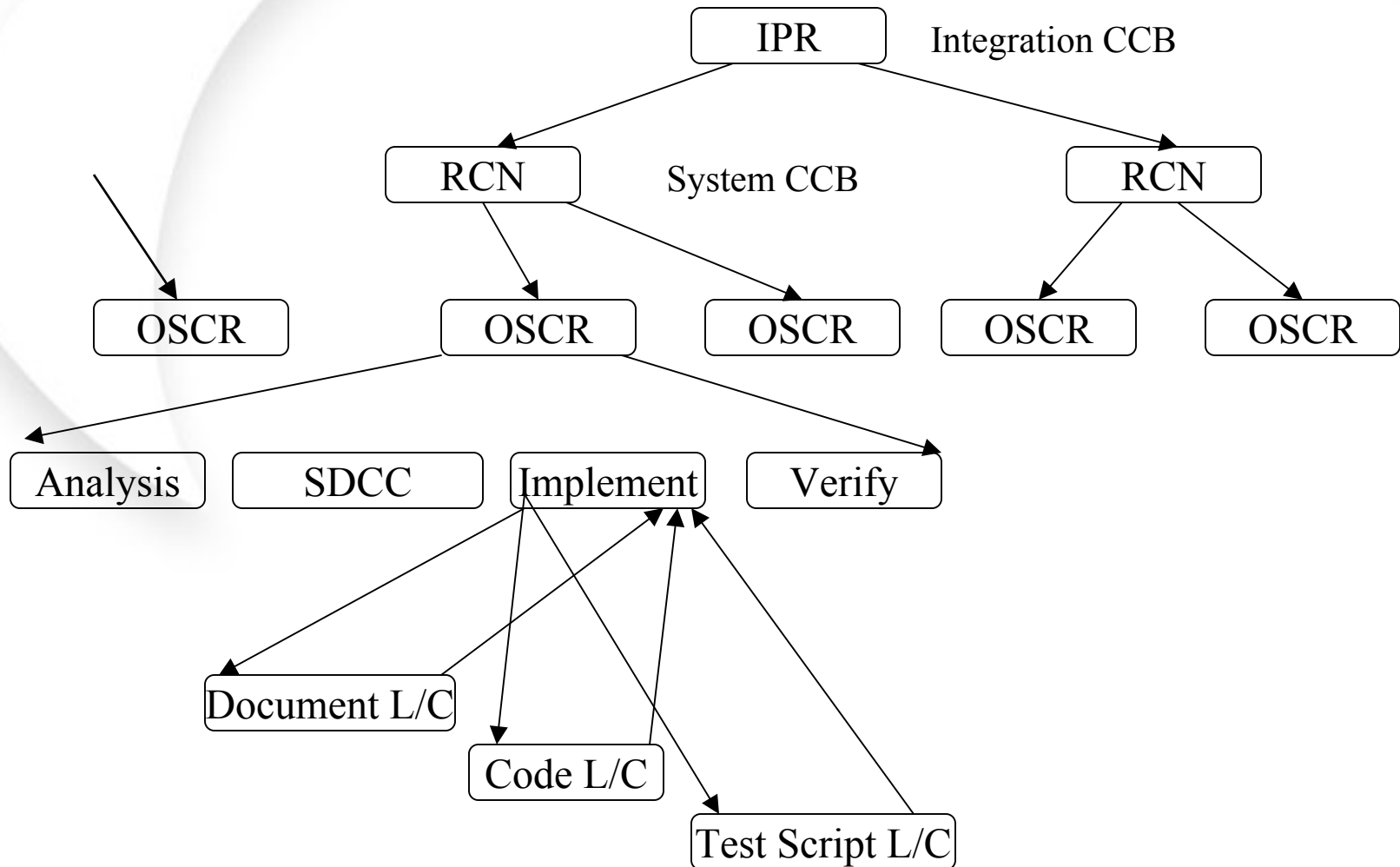
## Features of of Implementation, cont.

- Electronic repository for **ALL** software work products, change documents and their development history
- Role Based
  - Development and change processes are 90% development team executed
  - Provides fine grained access control
- Integrated, multi-level change/version management
  - Integration Problem Report (IPR)
  - Requirements Change Notice (RCN)
  - Online Software Change Request (OSCR)
- Baselines automatically created from change requests

**Passed Unannounced FAA SCM Audit in 45 minutes**

# C130J Hierarchical Change Process

## •Electronic Enforcement of Business Objectives



# SCM Viewpoint - Identification

- Enter the filename once
  - revisions & history automatically captured
  - only one physical copy of any revision exists in the repository
- Worksets and Named Branches for each variation facilitates concurrent, parallel development
- Current Configured Build Process
  - **Revised Baseline from Change Requests**
  - **Automated merge**
  - **Automated Release** to an SCM controlled directory
  - SCM executes the build script

# SCM Viewpoint - Change

- Change Document ID number assigned by system
- Electronic capture of change process
  - CCB Meetings an exception - few meetings
  - Participants enter their own comments
- Electronic enforcement of business rules
- Electronically linked to development process
- Online Software Change Request (OSCR)
  - Attributes by role to prevent premature promotion
  - Linked to IPR's, RCN's, and files to prevent inadvertent closure

**No Triggers, No Scripts**

# SCM Viewpoint: Status Accounting

- Accurate, current, complete, and real-time for **ALL** work products
- Records created as a **free** by product of developer interaction with the system
  - Documents execution of the development process
  - Name, Date, time stamp automatically captured
  - No separate record collection effort
- Focus is on value added extraction of information from the database
  - > 100 reports used to manage development, build, and flight test integration processes
  - Requirements volatility
  - Frequency of file changes

# SCM Viewpoint - Audit

- Process: Online, interactive review and query of object development history
- Configuration: Online, interactive review of baseline contents; Online review of comparison between baselines and releases

***Can you say Database Query?***

# SCM Viewpoint - Other

- Release records produced as by product of build procedures
- Process Change Requests (PCR)
  - Controlled by SEPG
  - Collecting SEI Level 5 artifacts!
- Software Build Requests (SBR)
  - Replaced manual creation & approval process
- Flight test removable memory media (RMM - PCMCIA card) production for loading of software on flight test aircraft
  - Replaced manual, less formal process
  - Provides traceability for software loaded on flight test aircraft
  - Tracks RMM by serial numbers
- Supplier Software Receipt and loading on flight test aircraft
  - Provides traceability of verification process
  - Captures electronic signature of responsible engineer

# Benefits Achieved

- SCM staff < 50% of classical requirements
- Reduction in span of change process - no meeting delays
- Improvements in Flight test integration and management processes
- Improved build and flight test management processes
- Improved media production process
- Improved communications:
  - Electronic/Interactive “top-down,look-down” and “bottoms-up, look-up” visibility into all levels of change review and approval
  - No place to hide

**Twice Passed Unannounced Customer SCM Audits**



**MERANT**

**Principles of Effective Software  
Configuration Management:**

**Reducing the cost of SCM to near Zero**

**Bob Ventimiglia**