Course: Software Configuration Management

Week 1: Introduction to SCM

Lecturer: Yimer Amedie (MSc.)

Addis Ababa Science and Technology University, Ethiopia

September, 2025

Course Description

- This course provides a comprehensive understanding of Software Configuration Management (SCM) as a discipline that supports all phases of the software life cycle.
- The course covers
  - Software configuration management fundamentals
  - Configuration Identification and status accounting
  - Configuration Control and Auditing
  - Release Management and Delivery

Course Goal

Figure 1: Concepts of SCM (Source: OpenAI, 2025)

- To equip students and learners with knowledge, tools, techniques, and practices
  - to manage, organize, and control software changes ensuring higher productivity, traceability, and quality in software projects with minimal mistakes.

Lecture - One

Introduction to

Software Configuration Management (SCM)

Figure 2: Concepts of SCM(Source: OpenAI, 2025)

Contents

Figure 3: Concepts of SCM (Source: OpenAI, 2025)

Learning Outcomes

After completing this lesson, you will be able to:

- Define configuration management and its core principles.
- Trace the high-level history of CM from its origins to modern software
- Identify key Software Objects and explain their importance.
- Describe the SCM Process and differentiate between Version Control and Change Control.
- Articulate the critical necessity of SCM in software development.

Introduction

Now a days, software is a life blood of any organizations.

- As software plays a big role in critical systems, development is becoming increasingly complex.
  - Size, sophistication, technology used
- Modern software must be reliable, scalable, and support diverse users, languages, and platforms
  - **Where even minor errors can have serious consequences**

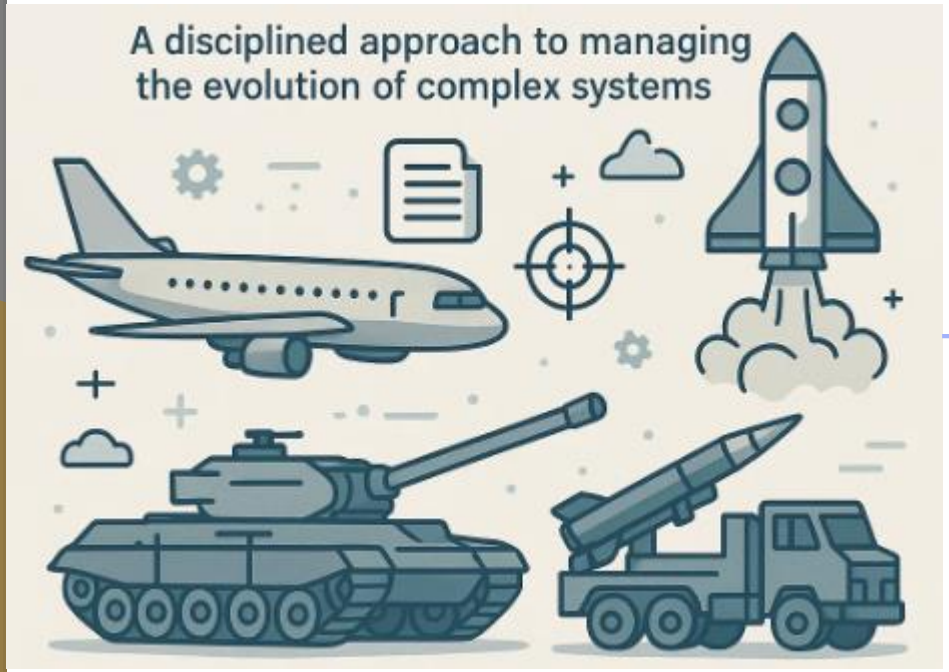
The Evolution of CM

CM has Its origin in the manufacturing industry (Leon, 2025)

- To manage design change and production
  - Due to the challenges for more complex and critical products.
    - Development spanned for many years,
    - Handled by more than one person
    - When control transferred from one person to another
      - Associated information lost

The Evolution of CM … (2)

Configuration Management



- **Aircraft**
- **Tank**
- **Spacecraft**
- **weapons systems.**

Figure 4: Complex systems like aircraft, tank and weapons(Source: OpenAI, 2025)

The Evolution of CM … (3)

- The Core CM Principles (From Hardware)
  - **Configuration Identification:**
    - What are the parts?
      - Naming, numbering.
  - **Configuration Control:**
    - How do we manage changes to the parts?
      - Formal change requests, boards.

The Evolution of CM … (4)

- The Core CM Principles (From Hardware)
  - **Configuration Status Accounting:**
    - What is the state of the parts?
      - Tracking, reporting.
  - **Configuration Audits:**
    - Does the built product match the design?
      - Functional and Physical audits.
- **These principles are directly transferable to software.**

The Evolution of CM … (5)

The Software Crisis (1960s-1970s)

- **The Shift:** Software projects grew in size and complexity.
- **Problem:** Projects were chronically over budget, late, and buggy.
- **Realization:** Building software was like building a complex machine.
  - **It needed the same disciplined processes used in engineering.**
- **The Bridge:** The principles of hardware Configuration Management were applied to software.

The Evolution of CM … (6)

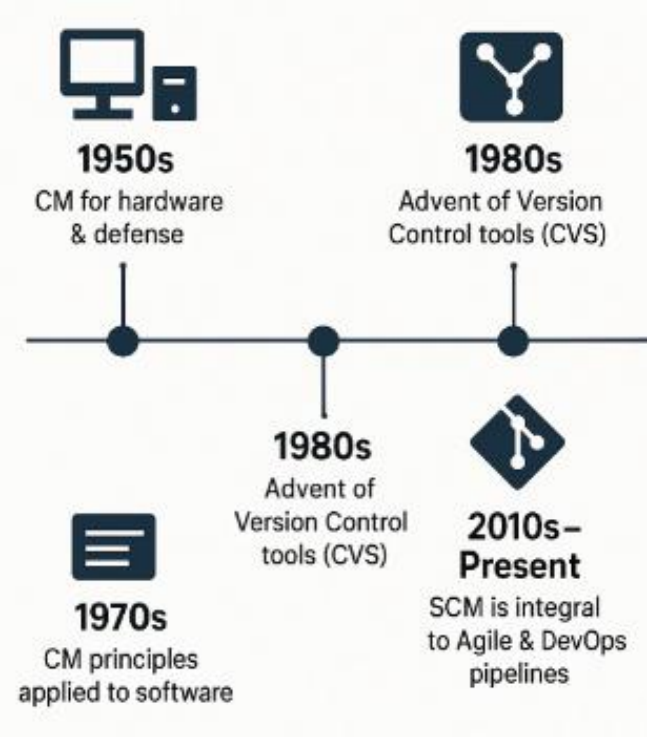
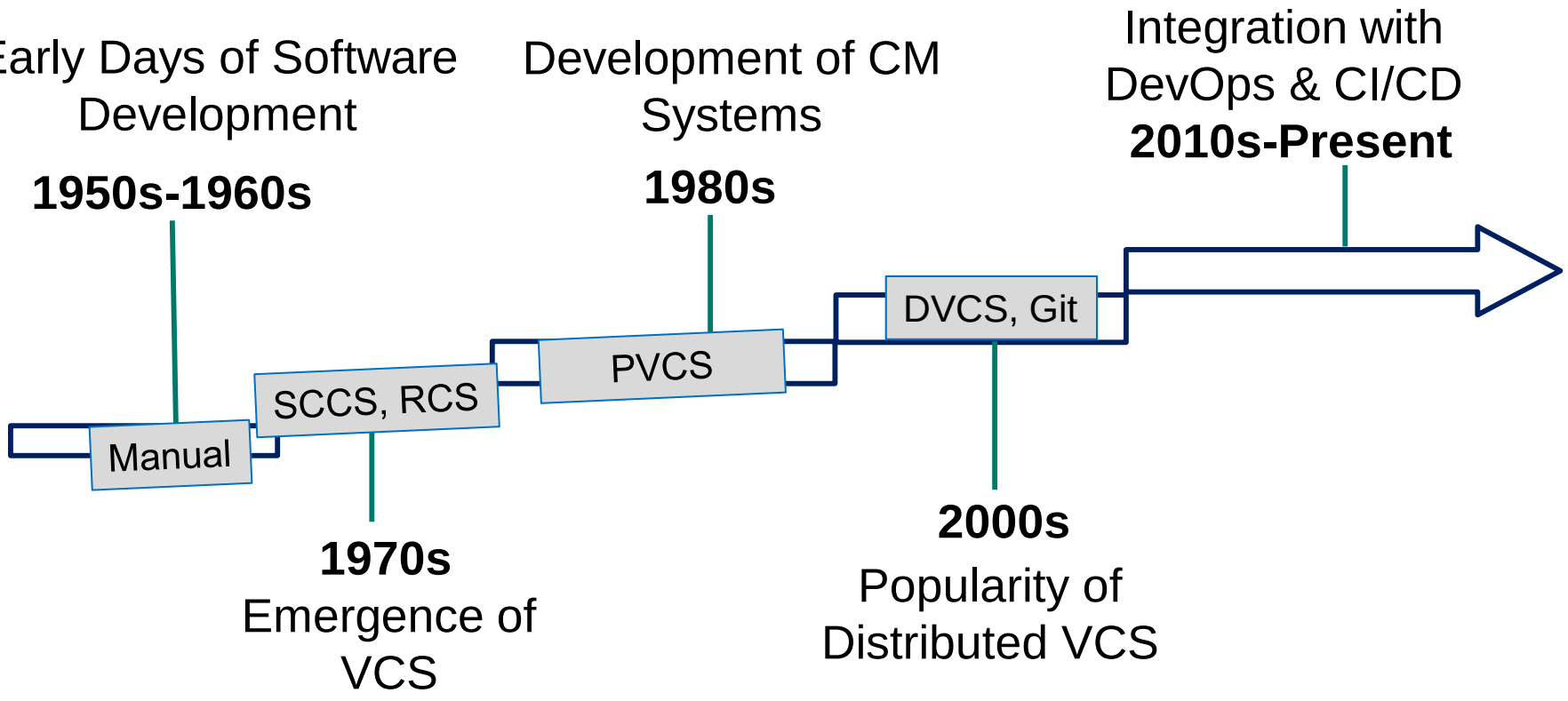


Figure 5: Timeline of SCM (Source: OpenAI, 2025)

Configuration management
from
hardware to Software

The Evolution of CM … (7)



The Evolution of CM … (8)

Therefore, the Birth of SCM is

- **1970s-1980s:** The term **SCM** emerges.
- **Goal:** Adapt traditional CM principles to the unique challenges of software.

Key Difference:

- Software "parts" (**Software Objects**) are digital and much easier to change, making control both harder and more critical.

Definition and Components

What is Software Configuration management (SCM)?

- SCM is the discipline of
  - identifying, organizing, controlling, tracking, and auditing the evolution of software artifacts throughout the development lifecycle to ensure integrity and traceability.

Definition and Components … (2)

What are the "Software Objects" we manage?

- CM doesn’t just manage source code.
  - It manages all software objects (configuration items – CIs) that comprises software system.
  - **Example:**
    - Documentations (Requirements, Design Docs, User Manuals)
    - Source code files (.java, .js, etc.)
    - Configuration files (.json, .properties, etc.)

Definition and Components … (3)

Components of SCM

Source Code & Executables

The core of the software, including all programming files and compiled applications.

Documentation & Design Models

Specifications, architectural diagrams, user manuals, and technical guides.

Data Files & Test Suites

Configuration files, databases, and comprehensive sets of tests to validate functionality.

Challenges Without SCM

- Code overwrites and conflicts
- Loss of earlier versions
- Difficult bug tracking
- Unclear ownership of changes
- Chaos in large teams

The Cost of Not Having SCM

- **Integration Hell:**
  - The nightmare of merging weeks of divergent work.
- **Lost Work:**
  - Accidental deletions or overwrites with no backup.
- **Endless Debugging:**
  - Inability to isolate which change caused a bug.

The Cost of Not Having SCM … (2)

- **Team Conflict:**
  - Developers constantly breaking each other's code.
- **Business Risk:**
  - Inability to deliver stable software, losing customer trust.

Necessity & Consequences

Why we need SCM?

Uncontrolled Changes

Lead to confusion, errors, and project delays, undermining productivity.

Increased Mistakes

Uncoordinated alterations can introduce bugs and inconsistencies.

Productivity Loss

Time wasted on rework and debugging due to poor change management.

Necessity & Consequences … (2)

Why we need SCM?

SCM's Solution

Maximises productivity by minimising mistakes, fostering a streamlined workflow.

- Manages complexity of modern systems
- Protects from accidental loss
- Enables controlled evolution
- Helps meet compliance and standards

Benefits of SCM

Some of the benefits of SCM are (Leon, 2025)

- Improved organizational competitiveness;
- Better customer service and improved customer goodwill;
- Better return on investment;
- Improved management control over software development activities;

Benefits of SCM … (2)

- Improved software development productivity;
- Easier handling of software complexity;
- Improved security;
- Higher software reuse;
- Lower software maintenance costs;

Benefits of SCM … (3)

- Better quality assurance;
- Reduction of defects and bugs;
- Faster problem identification and bug fixes;
- Process-dependent development rather than person-dependent development;
- Assurance that the correct system was built.

The Role of SCM in Software Engineering

- Ensures project stability and control
- Supports teamwork and collaboration
- Provides audit trail for accountability
- Improves quality and reliability

Goals and Objectives of SCM

- Version control
- Change control
- Configuration identification
- Auditing
- Status accounting

Goals and Objectives of SCM … (2)

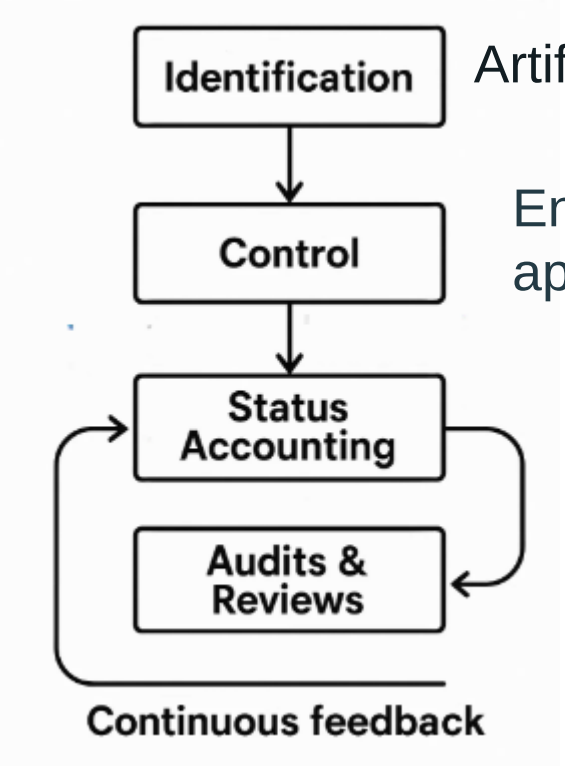
Activity:

- Imagine a team working without a SCM Process:
  - **Version Control is absent:** Developers overwrite each other's changes to shared Software Objects
  - **Change Control is absent:** No one knows why a change was made, leading to bugs and instability.
  - **Result:** Chaos, wasted time, and an unreliable product. This highlights the direct need for Configuration Management.

Scope of SCM in the Software Lifecycle

- Requirements management
- Source code control
- Documentation control
- Build and release management
- Maintenance and support

The SCM Process Overview



Artifacts we want to control

Ensuring that any change request is analyzed, approved, and implemented correctly

Records what has changed

Verify compliance

Ensures that lessons learned improve the next cycle

Key Roles in SCM

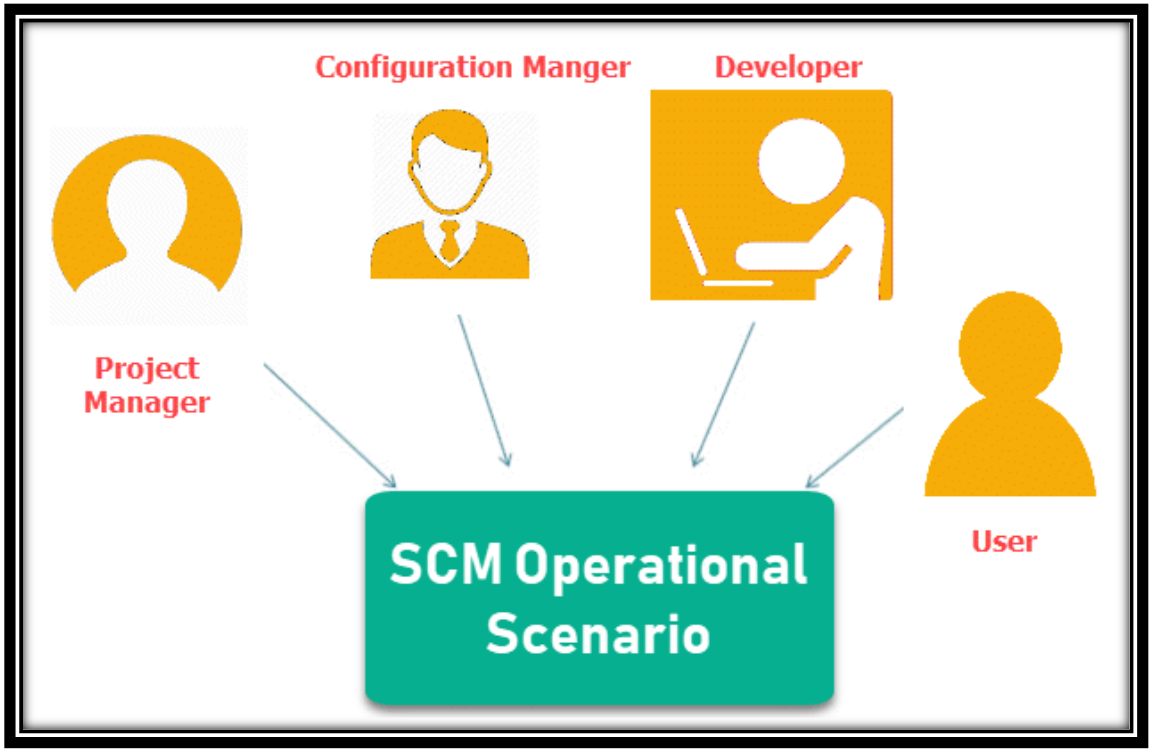


Figure 6: key participants in SCM (Source: Bennett, 2024)

Key Roles in SCM … (2)

Project Manager

- Oversees development progress, monitors status, and ensures all deadlines are met.

Configuration Manager

- Enforces SCM policies, manages change requests, and meticulously controls code versions.

Key Roles in SCM … (3)

Developer/Software Engineers

- Collaborate efficiently, merge their changes, and diligently maintain comprehensive documentation.

User/Customers

- Formally request changes and report any bugs or issues through established channels.

Knowledge Check

1. **Which of the following is the BEST definition of Software Configuration Management?**
   A. A tool for writing software code.
   B. A discipline for controlling change to software assets.
   C. A methodology for testing software.
   D. A programming language.
2. **The ability to revert your codebase to the state it was in yesterday is primarily a benefit of which SCM concept?**
   A. Build Management
   B. Release Management
   C. Version Control
   D. Change Control

Knowledge Check … (2)

3. **A formal process for approving and tracking a modification request (e.g., a new feature or a bug fix) is known as:**
   A. Change Control
   B. Version Control
   C. Data Control
   D. Build Management
4. **Which of the following is not the result of not having SCM?**
   A. Chaos
   B. Frustration
   C. wasted time
   D. a broken product.
5. **Why is Configuration Management important in software development?**

Knowledge Check - Answer

1. **B**
2. **C**
3. **A**
4. All are common results of not having SCM
5. There are different reasons
    - The nature of software products, projects, and development teams;
    - The increased complexity of the software systems;
    - The increased demand for software;
    - The changing nature of software and the need for change management.

Summary

- Configuration Management is the structured SCM Process for managing the evolution of software.
- The need for SCM is
  - To manage complexity and change in order to reduce confusion, eliminate redundancy, and enforce discipline, thereby significantly increasing productivity and product quality.

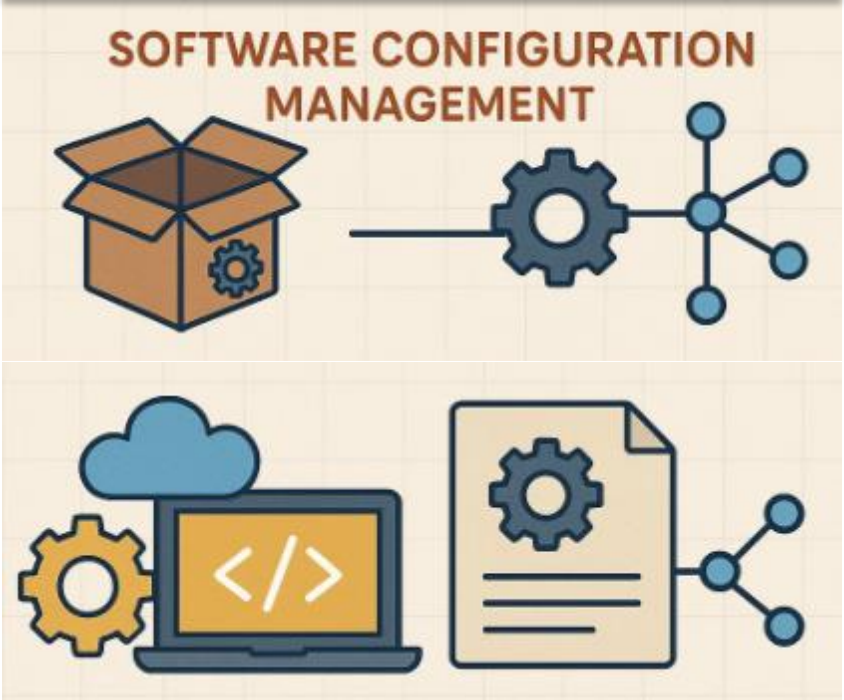
Summary

- The SCM Process is implemented through interconnected activities. Two of the most fundamental are:
  - **Version Control**: The foundation. The practice of tracking and managing changes to Software Objects over time.
    - **The "What" and "When".**
  - **Change Control:** The governance layer. The formal process for requesting, reviewing, approving, and implementing changes to Software Objects.
    - **The "Why" and "Who approved it".**

References

1. Bennett, L. (2024, August 13). Software Configuration Management in Software Engineering. Retrieved September 4, 2025, from guru99.com: https://www.guru99.com/software-configuration-management-tutorial.html
2. Leon, A. (2015). Software Configuration Management Handbook (3rd ed.). Norwood: Artechhouse. Retrieved September 4, 2025.
3. OpenAI. (2025, September 4). SCM history and concepts [AI-generated image]. ChatGPT (Sora). https://chat.openai.com/
4. OpenAI. (2025, September 6). Complex systems in defense and aerospace [AI-generated image]. ChatGPT (Sora). https://chat.openai.com/

Thank You!



NEXT LECTURE

Core SCM Process Activities