Introduction to Software Configuration Management in 6-8 minutes

Lars Bendix
bendix@sneSCM.org

Scandinavian Network of Excellence in Software Configuration Management (sneSCM.org)

Department of Computer Science
Lund University
Sweden

Why SCM?

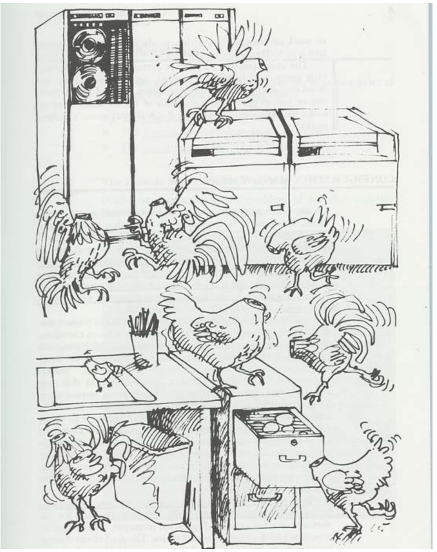
SCM is a set of techniques or tools to:

- manage the orderly development of a product
- co-ordinate the collaboration in a team
- make sure that we don’t make fools of ourselves

SCM is completely agnostic to:

- type of product
- type of development method

SCM provides service and support to everyone involved



Wayne Babich

Sometimes it is embarrassing to be a computer programmer. What other profession has such a remarkable rate of schedule and cost overrun and outright failure? [...]

Our failures are not of the individual contributors; most of us design, code and debug adequately or even well. Rather, the failure is one of coordination. Somehow we lack the ability to take 20 or 30 good programmers and meld them into a consistently productive team.

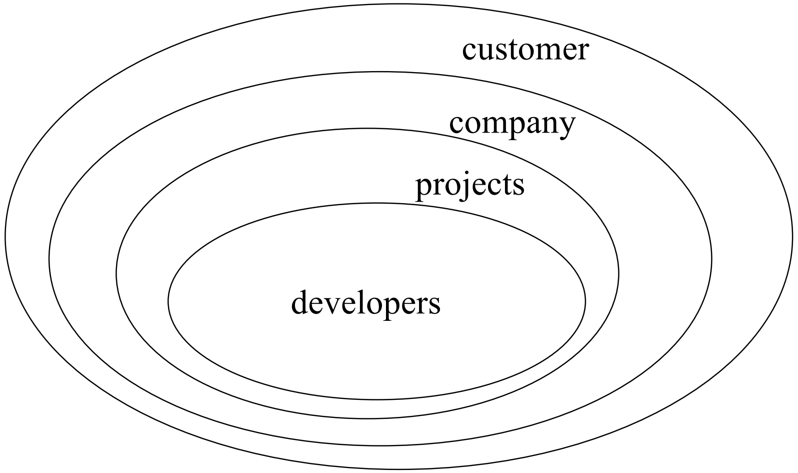
Wayne A. Babich, 1986

What is SCM?

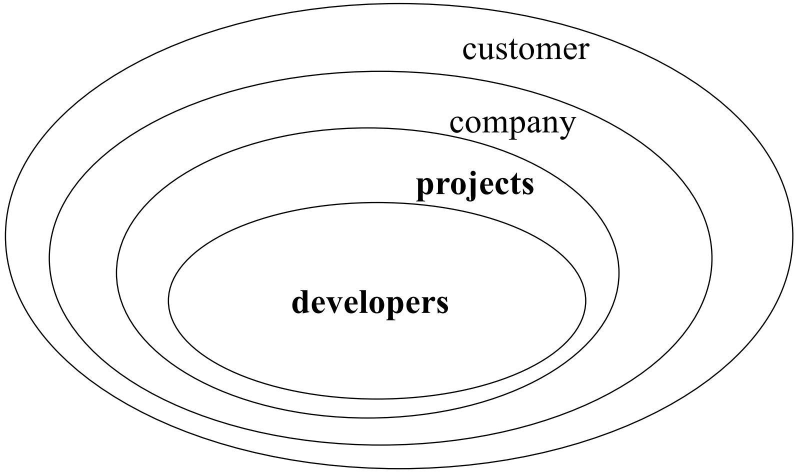
Software Configuration Management:
is the discipline of organising, controlling and managing the development and evolution of software systems. (IEEE, ISO,...)

The goal is to maximize [programmer] productivity by minimizing [co-ordination] mistakes. (Babich)

SCM perspectives



SCM perspectives



Co-ordination

Working in isolation:
- local dynamicity
- global stability
- problems:
  - double maintenance

Working in group:
- global dynamicity
- problems:
  - shared data
  - simultaneous update

SCM Hall of Fame

Wayne Babich, 1986:
Software Configuration Management – Coordination for Team Productivity

Team co-ordination problems:

- shared data
- double maintenance
- simultaneous update

Parallel work

Concurrency strategies:

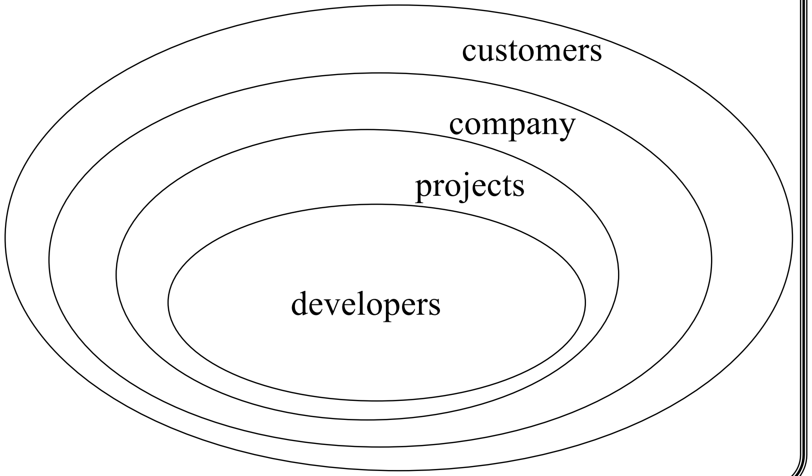
- locking
- copy-merge
- long transactions
- strict long transactions

Learning goals

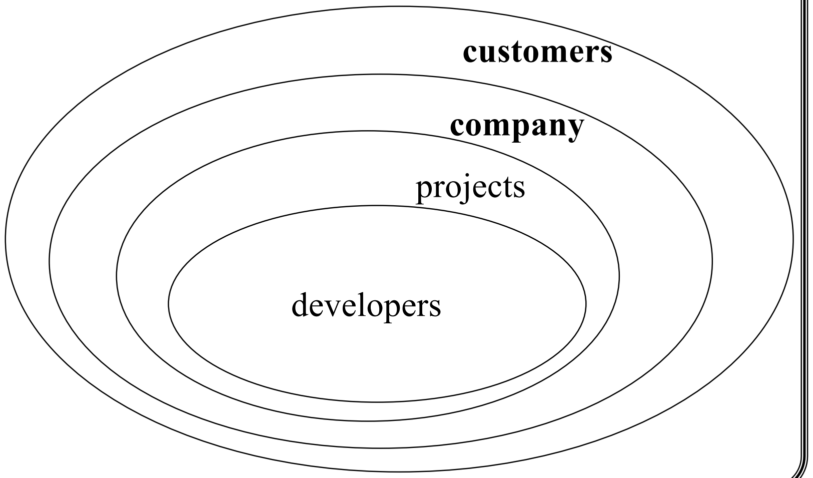
At this point the student should:

- understand co-ordination problems
  - shared data
  - double maintenance
  - simultaneous update
- know some central versioning concepts
  - workspace and repository
  - branch and merge
- know basic co-ordination strategies
  - locking, copy-merge, (strict) long transactions

SCM perspectives



SCM perspectives



customers

company

projects

developers

Definition of CM

Configuration management 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

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

Configuration Management – Overview

Configuration Identification

Configuration Control

Configuration Status Accounting

Configuration Audit

Configuration Management – Overview

Configuration Identification

Configuration Control

Configuration Status Accounting

Configuration Audit

Configuration Management – Overview

Configuration Identification

Configuration Control

Configuration Status Accounting

Configuration Audit

Learning goals

At this point the student:

- knows the traditional SCM activities
  - configuration identification (CI + CMDB)
  - configuration status accounting
  - configuration control (CR + CCB)
  - configuration audit
- knows some central SCM concepts and their use
  - baseline
  - traceability