Software Configuration Management

Lars Bendix
bendix@cs.lth.se

Department of Computer Science
Lund Institute of Technology
Sweden

Bologna, Italy, May 6th, 2004

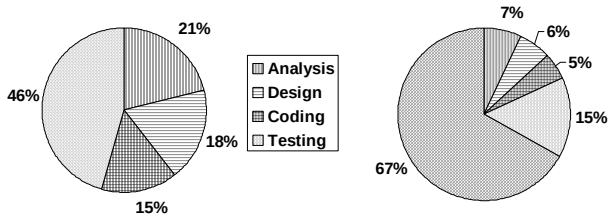
Part I: What is SCM - the problems

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 productivity by minimizing mistakes. (Babich)

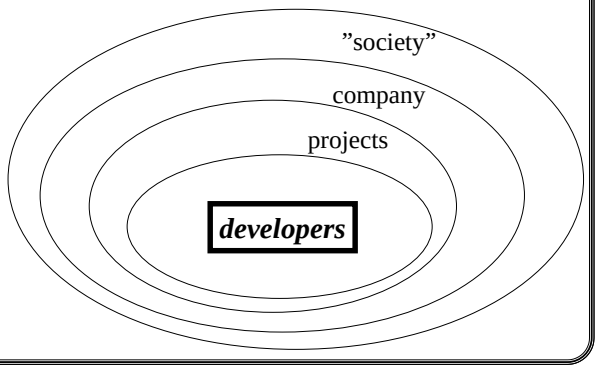
Life cycle costs



Different CM-roles

- users of CM
  - developers (ordinary/build meisters)
  - project leaders
  - customers
- designers of CM
  - processes
  - tools
- technical CM personnel

SCM perspectives



SCM examples

Canadian Telecom:
- 8 MLOC
- new release every six months
- 300-500 KLOC changed
- production time 8-11 months
- 1000+ programmers
- x variants

AT&T - Bell:
- software older than 20-25 years
- no documentation
- no source code

SCM examples

Synopsis:
- 17 MLOC
- 30+ production teams
- 10+ locations globally
- common code - product families

- 12+ hours to label a branch

- nightly and weekly builds

Other CM examples

- DoD torpedos
- Kasper's VW microbus
- Carlo's lemon marmalade
- Ole's crane hooks
- car engine tester software
- Concorde accident
- CPH metro station naming
- Citroën C3 fires
- DSB de-railing

Signs of a problem

- cannot meet deadlines
- cannot release multiple fixes
- do not know what went into a release
- developers fix the same bug
- do not know what has been tested
- have no visibility into work status
- year 2000 problems

Problems

Identification:
You should be able to identify the single components and configurations.
- This worked yesterday, what has happened?
- Do we have the latest version?
- I have already fixed this problem. Why is it still there?

Change tracking:
Helps in tracking which changes have been made to which modules and by whom, when and why.
- Has this problem been fixed?
- Who is responsible for this change?
- This change looks obvious - has it been tried before?

Problems

Software production:
Construction of a program involves pre-processing, compilation, linking, etc.
- I just corrected this, has something not been compiled?
- How was this binary produced?
- Did we do all the necessary steps in the right order?

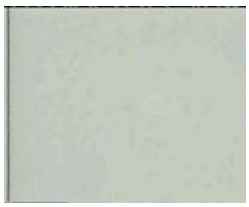
Concurrent updating:
The system should offer possibilities for concurrent changes to components.
- Why did my changes disappear?
- How do I get these changes into my version?
- Are our changes in conflict with each other?

Goals

- to be able to return to well-defined states
- to have an overview of the development history
- to give a model for the system architecture
- to show what depends on what
- to ensure the consistent generation of a system

- to save space
- to save time

Ferrari collaboration?



Three types of motivation

- paranoid
  - back-up,
  - afraid of disaster
- pragmatic
  - revert to old releases,
  - customisation,
  - builds,
  - debugging,
  - co-ordination
- business
  - company asset,
  - Intellectual Capital Report

Excuses for not using CM?

- CM only applies to source code
- CM is not appropriate during development because we use rapid prototyping
- It's not that big a project
- You can't stop people from making a quick patch
- We lower our cost by using only minimum-wage persons on our CM staff because CM does not require much skill

Norwegian experiment

- defect reports down by 36 % (more defects found internally)
- defect fix time down by 6 %
- maintenance resource usage down by 33 %
- development resource usage up by 22 %
- less effort spent on: system build, deliveries and maintenance
- pay back time - 7 months

Hygiene at work

- In 17-1800 the mortality rate for women giving birth was as high as 50% in some hospitals.
- It was not uncommon practice to go directly from section of infected bodies to assisting women giving birth.
- In the 1840s the Hungarian Ignaz Simmelweiss from the birth clinic in Vienna discovered the benefits of disinfecting your hands before assisting the women.
- Despite clear empirical evidence Simmelweiss' ideas were rejected at the time.
- It was not until the 1870s when Pasteur and Lister had discovered the bacteria that hygiene was implemented.
- Today the mortality rate is less than 1 in 10.000 births.

Software hygiene?

Part II: What is SCM - the solutions

Building on sand?

CM is a CMM level 2 key process area

Req.	Design	Testing	Coding	QA

Software Configuration Management

Building a band!

The drummer is the heart-beat of a band

Voice	Guitar	Strings	Piano

Fills	Solos	Patterns	Bass

The little drummer boy

How does a programmer spend his time?

- 50% interacting with other team members
- 30% working alone
- 20% non-productive activities

Common heritage

- re-use things
- sharing things
- memory/history
- communication
- co-ordination

Co-ordination

Working in isolation:
- local dynamicity
- global stability

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

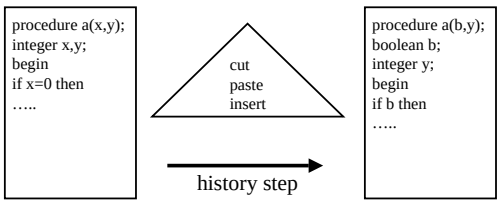
Problems of co-ordination

Double maintenance

Shared Data

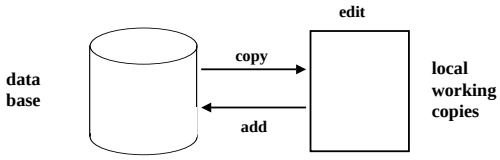
Simultaneous update

Modification



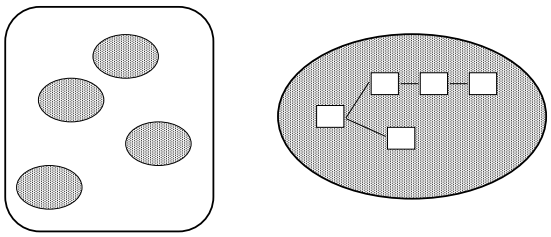
Model

Principle: components are **immutable**



Babich:
- shared data
- double maintenance
- simultaneous update

Concepts



Project data base

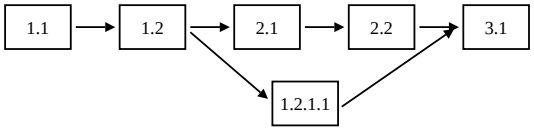
Version group

Algorithms and data structures

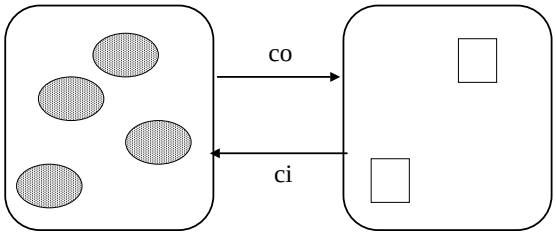
Structure:
- linear and tree (graph)

Visualisation of differences:
- diff

Merging of branches:
- merge



Workspace/repository



Project repository

Private workspace

Information

- give the history for a component
  - who / what / when / why
- data for metrics
  - hours spent, time spent
- information is represented by attributes
- we have two types of information
  - external
  - internal

Identification

We should be able to identify any version of a component, also when it is outside of the version control tool:

- as a text file
- as object code
- as a part of a configuration
- on paper

Baselines

A baseline is a reference configuration which is stable (over a certain period of time) relative to the work of the single programmers.

Ensures stability in relation to “the outside world”.

Allows safe dynamicity in the single programmers “private” work.

Acts as a communication channel for changes.

Spiderman vs. Rain-man

Traditional configuration management

Identification: The selection of documents that identify and define the baseline characteristics of an item.

Control: The controlling of changes to the configuration and its identification documents.

Status accounting: The recording and reporting of the implementation of changes to the configuration and its identification documents.

Audit: The checking of an item for compliance with the configuration identification.

Configuration identification

Ensures that we have unique identification of all parts and products:
- configuration items - atomic and composed (what)
- "storage" structure
- naming/numbering scheme (how)
- how are versions, variants and releases handled
- create a safe repository for configuration items

A plan should describe:
- which parts are configuration items
- the naming scheme to be used
- which tool to use

Configuration identification has to be carried out before project start.

Configuration control

This activity handles change requests and how they are controlled and documented from birth through to implementation or rejection.
- define change request formats and the change process
- define and enforce access rules for CIs
- define levels of control
- change control board (CCB)

A plan should describe:
- composition of change control board
- standard agenda for the CCB
- documents and process

Contributes to the objectives of traceability and accountability.

Change Control Board

Consists of representatives from both supplier and customer. Experts can be called in / drawn upon.

Its tasks are:
- to prioritise change requests
- to document change requests
- analyse and estimate impact of change
- approve changes
- follow the implementation status

The board must approve all changes **before** they are implemented

Status accounting

Recording and reporting of information:
- listing of approved identifications
- status of proposed changes
- implementation status of approved changes
- status of the project
- accounting information
- development history for single items

A plan should describe:
- what information is needed
- how it will be collected and processed

The information should be available - and useful - to everyone, to keep them informed of the status on a day-to-day basis.

Auditing

The process of verifying that all required items have been produced:
- that testing has been successfully performed
- that versions agree with specified requirements
- that technical documentation is complete
- that all change requests have been resolved
- that the product can be shipped (and installed)

A plan should describe:
- how audits will be conducted
- the authority of those conducting the audit
- who will approved the end result (baseline)

The configuration audit establishes the product baseline.

Release management

Re-creation is important:
- identification
- options
- tools

- baselining of a project
- bill-of-material for a product
- version control for tools too

Everything that has been in contact with a release must be preserved.

Part III: What is SCM - and more….

Why worry about CM ?

CM from a developer's perspective is about:

- supporting the individual person's work
- improving collaboration among people on a team
- communicating project results with others

Metaphors

Of the four levels of CM we focus on:
- the individual level
- the project level

The metaphors for these two levels:
- "The Study" - personal workspace
- "A Construction Site" - collaborative work
- "The Library" - collect, share, and use knowledge

Construction Site Metaphor

Imagine yourself at a construction site …

– feel the busy atmosphere
– see the different parts and trades
– notice how everything is in process
– recognise the schedule and budget pressure

Relate this to collaborative work on a project

Construction Site

The collaboration aspect of CM:
- to integrate the individual person's work in the system in a safe and efficient manner
- to allow people to work in parallel
- to be able to co-ordinate

It all boils down to
- "Putting-it-all-together"

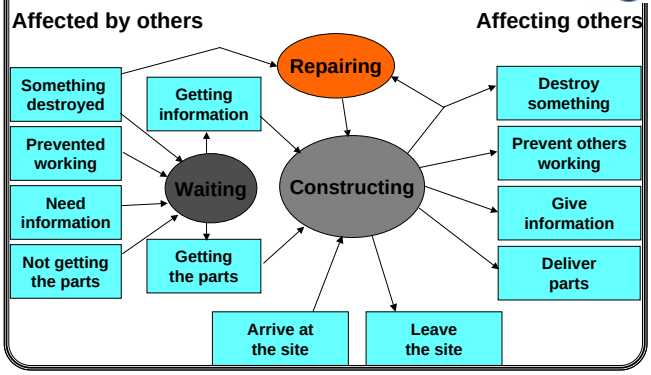
We need mutually agreed rules of good conduct in order to succeed

Construction Site, Main Principles

To achieve the common goal you should:

- get hold of the right parts
- get all the parts you need
- not be delayed by others
- not delay others
- not destroy the work of others

Construction Site, Activity Model



Theory Pertaining to the Construction Site

Purpose of the construction site:

- integrating parts to a whole
- awareness
- co-ordination
- communication

The Study Metaphor

- Imagine yourself in your own private room …
  - feel the inspiration
  - see the different things around you
  - you have collected everything you need
  - and you can get something done effectively
- Relate this to your work on an assignment

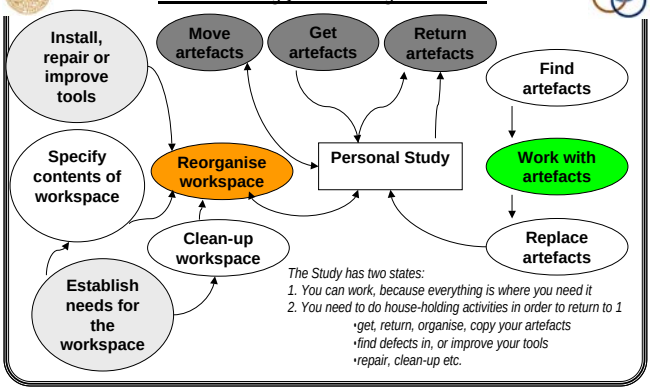
The Study

- The individual aspect of CM
  - to work on an assignment in a safe and efficient manner
  - to synchronise with the ”world” when needed
  - organise the work and environment to fit individual needs
- It all boils down to
  - ”Getting-your-work-done”
- You need mutually agreed rules of good conduct in order to succeed

The Study, Main Principles

- To achieve your individual goal you should
  - be able to maintain an overview of the whole system
  - establish and maintain well-defined interfaces to others
  - be able to get and return artefacts
  - be able to take artefacts with you to another place
  - be able to customise the environment
  - be able to work the way you prefer
  - be able to work uninterrupted

The Study, Activity Model



CM Theory Pertaining to the Study

Purpose: Study vs. Construction site

- long transactions
- parallel work
- the study
  - creation
  - doing the work
  - termination

Roles

- developers
- testers
- quality assurance
- release people
- help desk personnel
- maintainers

The Library Metaphor

- Imagine yourself in a large library
  - see the wealth of knowledge around you
  - feel how everything is neatly organised and tidy
  - enjoy the easy access and assistance
- Relate this to the knowledge on your project

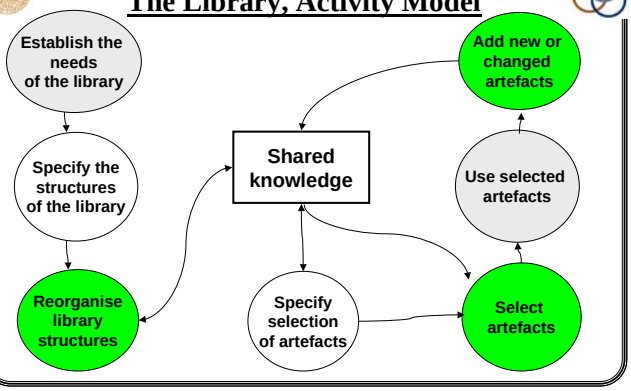
The Library

- The knowledge aspect of CM
  - to collect and store all knowledge
  - to work with consistent parts of the knowledge
  - to create and evolve the structure of the knowledge
- It all boils down to
  - ”Collecting, sharing, and using knowledge”
- We need mutually agreed rules of good conduct in order to succeed

The Library, Main Principles

- To manage the project’s knowledge you should
  - be able to store all type of artefacts produced over time
    - source code, generated products, documents
  - handle several issues of artefacts
  - maintain dependencies between artefacts
  - enable selection of a consistent sets of artefacts
    - builds, baselines, configurations, releases
  - enable different exchanges of artefacts via the library
    - private, group, project, public
  - manage concurrent use of artefacts
  - organise and reorganise the knowledge and structure

The Library, Activity Model



CM Theory Pertaining to the Library

- Library structure
- Purpose of the Library
  - identification
  - preservation
  - history
  - exchange
- Availability

History

- Information
  - who
  - what
  - when
  - why
  - status
- Change logs
- Time machines

Black Boxes

- Three inexplainable jet-crashes in 1953/4
- Australian chemist David Warren got the idea
- Purpose to avoid having to “reconstruct” events
- Recorded pilot conversation for four hours
- Additional recording of instruments’ showings
- Resistance for fear of “control”
- Two great crashes in Aus in ‘67 led to its introduction

Why should CM be automated?

- simple tasks
- repetitive

- accuracy
- precision
- discipline

- saves space
- saves time

Benefits of CM

- Insurance against accidents
- Help in debugging
- Support for co-ordination
- Company assets
- Intellectual Capital Report

Configuration management promotes consistency, productivity and quality

Final words

This lecture has been about how SCM should be done under ideal conditions - but real life is never ideal, so be prepared to improvise.

Remember that SCM is not an end in itself - but only a means to help you arrive at your goal(s).