Introduction to Software Configuration Management Part II: The Company

Lars Bendix
bendix@sneSCM.org

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

*Department of Computer Science
Lund University
Sweden*

Learning goals

After this lecture the student will:
- know traditional SCM activities
- know central SCM concepts and their use

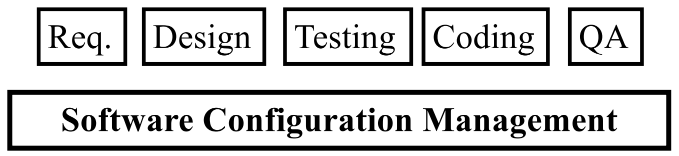
Different CM roles

Operational SCM (users & operators):
- developers (ordinary/wizards)
- project managers
- Quality Assurance
- company managers
- customers

Strategic SCM (designers):
- SCM processes
- SCM tools
- SCM improvement

Building on rock?

SCM is a CMM level 2 key process area



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

Traditional configuration management

Identification: The selection and handling of which artefacts that are important for creating the product.

Change Control: The controlling of changes to a configuration of a product and its artefacts.

Status accounting: The recording and reporting of the implementation of changes to a product and its artefacts.

Audit: The validation of configuration of a product for compliance with its definition.

Configuration Identification – I

Definition: the selection, designation and description of **configuration items**.

Purpose/goal: to capture, preserve and make available all the important things in a project – and to make sure that we have unique identification for these.

Configuration Identification – II

Terminology:
- artefacts
- configuration items (CIs)

Types of CIs:
- atomic CIs
- configurations
- (built) products

Configuration Identification – III

What to do with CIs:

- **kept safely**
- shared with others
- versioned

How to handle CIs:

- **unique naming**
- meta data
- structured storage

Configuration Identification – IV

Traceability:

- horisontal (versions)
- vertical (dependencies/relations)

Software Bill of Materials (SBoM):

- *what* went into the product
- *how* it was built

CMDB:

- items + information + traceability
- entities + attributes + relations

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

Configuration Status Accounting – I

Definition: the **recording and reporting of information** needed to manage and work on a project.

Purpose/goal: to allow people to easily get all the information that they need to carry out their work in an informed way.

Configuration Status Accounting – II

Information reporting:

- Status Accounting answers questions
- to answer questions **we need data**
- different people have different questions in different situations

Queries on the CMDB:

- standardized
- ad hoc

Sometimes we (still) make a report

Configuration Status Accounting – III

Recording of information:

- Status Accounting may require “updates” to the CMDB schema
- make sure that data is captured by the right person at the right time
- incorrect data is worse than no data
- data can/should be captured automatically by tools

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

Answers to Questions

Questions:

- Do we have the latest version?
- I have already fixed this problem. Why is it still there?
- I just corrected this, has something not been compiled?
- How was this binary produced?
- Has this problem been fixed?
- Who is responsible for this change?
- This change looks obvious - has it been tried before?

More questions:

- who made this change?
- when was it made?
- what changed?
- why did it change?

Configuration Change Control – I

Definition: the proposal, evaluation, coordination, approval or disapproval, and implementation of approved **changes to baselined CIs**.

Purpose/goal: to ensure that proposed changes are classified and evaluated and that approved changes are implemented, documented and verified.

Configuration Change Control – II

Change request (CR):

- problem report
- waiver
- requirement

- testers
- customers
- (sub-) contractors

Must contain **all** needed information about the change

Configuration Change Control – III

Change process:

CR => filtering process => Impact Analysis => CCB

Impact Analysis:

- time and resources
- costs and consequences

CCB => implementation => test => QA => closed

Project manager's "tool"

Configuration Change Control – IV

Change Control Board (CCB):

- meet on a regular basis
- CM prepares (CRs) before the meeting
- experts can be invited
- it is headed by a chairman
- the chairman has dictatorial power
- ad hoc meetings if needed

Outcomes:

- approved
- rejected
- deferred
- escalated

Configuration Audit – I

Definition: an independent evaluation of a (changed) CI to **ascertain compliance** to specifications, standards, contractual agreements and other criteria.

Purpose/goal:

- to verify that the CI matches the description in the specification and documentation.
- to ensure that work has been performed in the correct way, that is, in conformance with the development standards and guidelines.

Configuration Audit – II

It is a verification of:

- the product (CI) – does it conform?
- the process – did we follow the "rules"?

Before the product (CI) is accepted into the baseline

A Configuration Audit acts as a "**quality gate**"

CAs can be done at:

- various times
- various formality

Configuration Audit – III

A **baseline** is:

- something that we want to remain fixed – for some time
- something that is named so we can return to it
- something that has a specified (high?) quality
- when it is changed, it is done with much care

What can be baselined:

- requirements
- tests
- code
- …

Release management

Re-creation (and identification) is important:

- identification
- options
- tools

- tracing all relevant pieces and information
- baselining of the project/product
- software bill-of-material for a product
- version control for tools too

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

Business value

How does SCM translate to business value?

- faster development
- better quality
- greater reliability

Seven critical requirements:

- safety
- stability
- control
- auditability
- reproducibility
- traceability
- scalability

ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/pmpp/sevenkeys.pdf

Safety

SCM keeps project assets safe from:
- corruption
- unintentional damage
- unauthorised access
- other disasters
- even safe from changes!!!

SCM provides:
- secure access
- reliable recovery
- an easy way to rematerialize *past* configurations

It is the protection of key business assets.

Stability

True stability has some essential elements:
- guaranteed stable work areas
- individual control over introduction of new code
- the option to gradually update the work area

Introducing instability can cause a downward spiral.

Control

Find the delicate balance between:
- enforcing appropriate controls
- imposing bothersome constraints

Integrating contributions requires control over:
- who works on what
- how changes flow from developer to integration
- who can work where
- consider a “food chain” approach to development

Plan how you want to work - and enforce those rules.

Auditability

Auditability refers to the ability to answer:
- who, what, when and where
- about components or configurations
- tracks and records meta-data about changes

Common questions:
- did foo.java get changed?
- who made a change to this line in this version?
- were all bugs fixed for this patch?
- who did the fixes?

It allows you to query the data base to find answers easily.

Reproducibility

You must be able to quickly reproduce previous configurations:
- reproduce the exact configuration of the code
- and also corresponding versions of test cases
- including documentation and more
- and the right version of all tools used
- so you can roll back everything to any point in time

The tool must be able to:
- reproduce all the files in a build
- take into consideration the directory structure
- provide "name space" versioning

Can ensure that what is being built has also been tested.

Traceability

Comprehensive traceability involves the ability to:
- identify the version of a released product
- identify files and versions changed by a change request
- identify the change request that creates a certain version
- identify the configurations that contain a specific file

This allows us to:
- implement the "time machine"
- answer questions about the past
- modify the past?

Traceability tells you how you arrived at where you are.

Scalability

Scalability of SCM systems includes:
- configurable and functional when little is needed
- versatility to manage growth of the project
- ability to support (geographically) distributed teams
- handling of outsourced production
- reliability in the light of scaling

The tool should:
- scale to support large teams
- not impose burdens on small teams

Changing SCM tool is very painful for everyone involved.

Summarising

Effective SCM will:
- create a secure and predictable environment for working
- automate everyday build and versioning tasks
- provide quick access to file and versioning information
- be agile and robust to adapt to changing conditions
- provide managers with key information and data
- support end-to-end tracking of changes
- make it easy to do the right things
- make it hard to do the wrong things

It looks like it is worth the money.

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