

"Why I still believe in software configuration management"

A personal view by Nick Hawkins,
Telelogic

<mailto:nick.hawkins@telelogic.com>

A Personal View on Software Configuration Management

- After more than 5 years working as a process consultant for a leading vendor of SCM and having seen a number of implementations that realised "slightly less than their full potential" I still believe that SCM is a "good thing".
- Based on my experiences as a developer, (of several types of software over 12+ years), I hope to justify this view and illustrate what makes for a successful roll out and how to avoid an inappropriate solution.

Topics

- **Introduction**
- **My experience of CM as a developer.**
- **My experiences of CM as a Consultant**
 - Primarily with the Telelogic (formally Continuous) tool set
- **Summary**
- **Foundation Methodology**

Introduction

- **This are my personal views, they do not represent Telelogic corporate policies.**
- **This presentation focuses on the wider issues surrounding CM, it does not look in detail at tool specific problems.**

My Experience of CM as a Developer

Process Plant Design Group

- **Situation**
 - My “graduate” job as part of an engineering sponsorship with a large UK chemicals group.
 - Dealing with the specification and installation of complex control systems.
- **Lessons**
 - No software but established the basic principles of:
 - Defining the system components (CI's)
 - Knowing what state the system is in (configuration)

Medium Size Embedded Software Development

- **Situation**
 - Medium size company, small teams
 - Mixed development environment,
 - Every customer had a specific variant
 - Techniques
 - DEC operating system automatically versioned files.
 - Variants controlled by compile time switches and build file.
 - Backup tapes cut for each delivery
- **Lessons**
 - It worked for a small, close knit team
 - There was very little code maintenance required.

“Start Up” SCADA Vendor

(Supervisory control and data acquisition)

- **Situation**
 - Small company
 - (I was the 4th permanent employee, grew to 20)
 - Highly configured system
 - DEC operating system for version control, tapes for configuration management
- **Lessons**
 - Paid overtime makes a “tape based” CM solution expensive to run.
 - Affordable solutions that work for 4 people lead to problems for larger teams.

Large Engineering Company

- **Situation**
 - UNIX, C, official CM techniques too heavy to use (project manager would not allow it!)
 - Developed SCCS solution for my team (5 people) with full support of quality department
 - Propagated solution to 2 other teams
- **Lessons**
 - SCCS is free, but all systems cost time to develop
 - “Home grown” solutions do not transplant well

R+D for Major Project Management Tool Vendor

- **Situation**
 - Developing OO front end for legacy scheduling engine
 - Migrating (pre ANSI) C to C++ on PC, UNIX...
 - Failed to use PVCS, cut a CD-ROM each week.
 - Bug fixing using “home grown” CM solution.
- **Lessons**
 - Hidden costs of “Home grown” solutions
 - Difficult to teach to new staff.
 - Maintenance depends on key individuals
 - Value of appropriate solutions.

Consulting with Continuus

- **Situation**
 - Consultant in professional services group.
 - CM/Synergy used to maintain customisations
 - For my personal work I use a backed up drive.
 - (Note that the R&D group are full users of CM)
- **Lessons**
 - Use of appropriate technology
 - CM solution should match scope of business needs

My Experience of CM as a Consultant

A personal selection.

Banking System Development

- **Situation**
 - 30'ish developers, split over development & support.
 - One of my first major engagements
 - Up to 6 parallel variants in service
 - Initially very low tool confidence
- **Lessons**
 - Effective management support is VITAL
 - The tool may have “bugs” but people are even worse
 - Selling the solution is at least as important as getting the right solution
 - And much more important than perfecting the solution

Major German Bank

- **Situation**
 - Continuous adopted as the “proto” standard
 - Succeeded, client expanded from 30 seats to over 1200 worldwide
 - Mixed development tool set and CM knowledge
- **Lessons**
 - Importance of “ownership” for a CM solution
 - Use of CM to regularise sub-contracting arrangements
 - The high level of investment required for a “corporate” solution
 - Tool (and consulting) costs are secondary

UK Financial Institution

- **Situation**
 - Project to enhance configuration and release management of “back office” applications.
 - 3rd party (on site) development groups.
 - Roll out much delayed.
- **Lessons**
 - Interlinked commercial and political problems.
 - CM roll out requires both high level sponsor and an effective management champion.
 - Your CM solution is part of your development infrastructure, software QA is only part of the picture.

“Legacy” Financial System Provider

- **Situation**
 - System more than a decade old
 - Obsolescent development tool set, difficult to manage.
 - Large installed base, many customisations
 - System architecture “blurred”.
 - Senior management initiative
- **Lessons**
 - New customer support contracts to ease CM problems.
 - Underestimated time to transition
 - After new process had been defined
 - Underestimated change management (people) issues
 - Hard for existing QA/CM staff to adapt to new techniques

European Based Semiconductor Manufacturer

- **Situation**
 - Large, worldwide organisation.
 - Many projects split across sites
 - Developed “packaged” CM solution
 - Process, training and standardised installation.
- **Lessons**
 - Maintain strong control over distributed CM processes
 - Need to balance training
 - Tool and process
 - Managers and developers
 - Problems of planning/resourcing large implementations

South African ISV

- **Situation**
 - Pre-Sales opportunity
 - Excellent fit of tool capabilities & customer needs
 - Users wiling for change
- **Lessons**
 - “Practical” cost of ownership
 - South African currency is weak
 - Required consultancy would be very expensive
 - (It is possible to adjust tool prices to suit local markets.)
 - **Must scope requirements and match to business**
 - Needed a “big” solution, could only afford a “small” solution.

CM “Cookbook”

- **Situation**
 - A set of CM “process patterns” that my colleagues and I have developed to meet real life situations.
 - Direct support now being incorporated in CM/Synergy
- **Lessons**
 - The value of modular CM solutions - no need for “one size fits all” corporate standards.
 - The strength of known solution “templates” that can be adapted to individual situations.

Summary

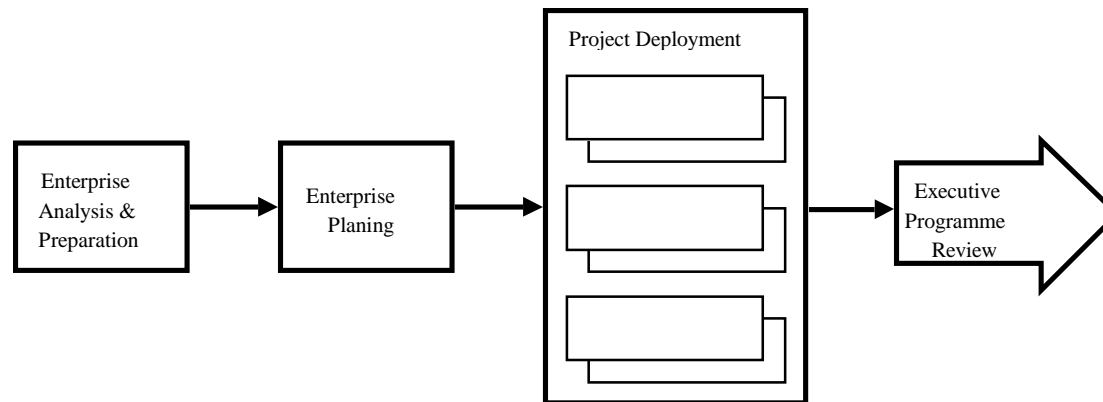
Summary

- **Requirements and Goals**
 - Funding and scope should match
 - Management Commitment
 - Clear Ownership
- **People are a problem**
 - “Soft” change management issues
- **Planning**
 - Investigation Phase
 - Roll out

Foundation Methodology

Telelogic's approach to planning a major CM implementation.

Foundation Methodology



- Split up overall roll out into:
 - Enterprise level roll out
 - Individual project deployments
- Project Deployments categorised by
 - Project “purpose” (e.g. development, maintenance, deployment),
 - Management complexity (e.g. multi-component, cross site).
 - Development technology (e.g. C++, Oracle, VB),

Individual Project Deployments

- **Analysis**
 - Confirm goals, expectations and strategic needs, assess feasibility, develop rough Project Plan.
- **Planning**
 - Identify Team, identify all variables, assess impact, design solution, develop Project Deployment Plan.
- **Pre-Deployment**
 - Prepare system for deployment.
- **Deployment**
 - Train end users, deploy system, and provide end user support.
- **Assessment**
 - Gain acceptance for the completion of the project.