Introduction to
Configuration Management

COMP3100/3500 Lecture

Ian Barnes

Wednesday 28th April 2004
What is it?

A set of practices and techniques for:

- keeping track of all documents, source code and other files for a project (called configuration items);
- managing changes to them;
- keeping backups of them; and
- maintaining traceability.

An alternative definition: “Control of system differences to minimise risk and error.”
What is it? (2)

- CMM key process area for moving from Level 1 (chaotic) to Level 2 (repeatable)
- **Not** just about the maintenance phase
- **Not** just about selection of tools: choosing a tool (like Aegis or CVS) is not a configuration management plan. (But tools are very important and can help a lot.)
Four CM Questions: can you answer them?

1. Show me the current version of document X, with a list of all changes since date Y.

2. Show me the latest executable, with a list of all outstanding problems and all features not yet implemented.

3. Show me a listing of module Z, version W.

4. When was change A made to configuration item B, who made it, who approved it, and why was it made?

Example: Ariane 5 crash

- ESA rocket for launching satellites
- Crashed 40 seconds after first launch in 1996, cost about US$500 million
- SRI = Inertial Reference System: a bunch of gyroscopes and stuff, attached to a computer
- Software reused from Ariane 4
- Was only needed *before* launch
Ariane 5 (cont’d)

- After launch, a variable went out of range, the overflow caused a system crash.
- Exception handling had been disabled to improve performance.
- The team *proved* it couldn’t go out of range.
- But the proof was only valid for Ariane 4, *not* Ariane 5.

**This was a failure of requirements traceability, and thus a failure of configuration management.**
Configuration Items

- requirements
- design modules
- program code
- tests
- documentation
- makefiles and other configuration files
- scripts
- anything else
Versions and Revisions

- **Revisions** are ordered in time (like in RCS)
- **Variations** (or variants) are different versions that co-exist in time (e.g. for different hardware, o/s, options etc)
- **Versions** includes both
- In general you have revisions of variations.
- CM gets messy when you have to change several variants of a component and make sure they all work correctly.
Configurations

- (Narrow definition) A configuration is a record of the exact version of each module that went into a compiled system.
- For software maintenance, you have to be able to recreate every possible configuration.
- In some situations, the configuration needs to include all details of the development environment (compiler version, library versions etc).
- e.g. DCS staff vs. student systems
Change Management

• For large projects, need a formal change management process.

• Change requests registered with Change Control Board

• Need to make a case for each change

• Examine costs vs. benefits

• Effect on schedule

• When s/w changes, may also need to change requirements, design, documentation etc.
Change control questions

- Is this change really necessary?
- Are there different ways to do it?
- Which has lowest risk?
- What modules are affected?
- What requirements could be affected?
- What are the risks?
- What are the benefits? Costs?
- How long will it take? How much will it cost?
What is a baseline?

• A controlled (working?) standard version of the software, from which modifications are made.

• Once tested and approved, changes are checked in and become part of the new baseline.

• The process for accepting changes may be quite formal, requiring approval by the CCB or a supervisor/manager, CM person...

• Baseline vs. Development system
CM Plans

- IEEE Std 828-1998
- Don’t follow it stupidly – it is not a recipe
- It is very general – covers every possibility
- Do use it as a checklist
- Do tailor it to the needs of your project
- Do read it together with something more helpful e.g. ANSI/IEEE Std 1042-1987 (Guide to SCM).
Components of a CM Plan

1. **What** will be managed? (List and organise all configuration items.)

2. **Who** is responsible for what? (Roles, tasks)

3. **How** will this be done? (Processes for registering CIs, for change requests)

4. **What** records will be kept? (All configurations, all changes...)

5. **What** tools and resources will be needed? (Software, database, people, time, money.)
Configuration Databases

Probably only for large or complex projects:

- What customer has what configuration?
- What are the system requirements for each configuration?
- What versions exist? Dates?
- What versions, customers, (requirements?) will be affected by changing module X? (or by changing version Y of module X?)
- What change requests are outstanding?
- What faults exist in version X?
Configuration Databases (2)

- Remember the principle of storing information once only

- Can the Config DB be linked by software to the version management system?

- If not, need strong procedures for keeping it up to date

- An out of date database is useless
Steve McConnell on CM

• Former MicroSoft project rescue team leader

• He says: Biggest CM problem is *overcontrol.* (Probably not true in COMP3100)

• Best way to reduce road toll is stop driving...

• 1 person, 1 week project? RCS and a notebook is enough

• 50 person, 5 year project? Need serious SCM with CCB, Config database, formal procedures, regular configuration audits and reviews etc
So, what do you need for COMP3100/3500?

- Depends on your project
- 6 people, 8 months (part-time)
- Somewhere between the two extremes
- Probably at least:
  - Version control
  - Semi-formal change control
  - Regular backups
McConnell’s Checklist

General

• Is your plan designed to help the team produce software, and minimise overhead?

• Do you avoid overcontrolling the project?

• Do you group change requests?

• Do you estimate the effect of each proposed change?

• Do you look back at requirements when you see lots of big changes?
McConnell’s Checklist (2)

Tools

• Do you use version-control software to facilitate configuration management?

• Do you use version control software to assist with teamwork/co-ordination?

• Do you use a build tool to manage compilation efficiently and reliably?
McConnell’s Checklist (3)

Backup

• Do you back up all project materials regularly?
• Are backups transferred off-site regularly?
• Is everything backed up, including source code, documents, notes, infrastructure etc?
• Have you tested your backup recovery procedure?
Release management

- Need to plan how you will get the software to clients/customers
  - Executables? (Means a different configuration for each target platform)
  - Java jar files? (Portable)
  - Source plus build instructions?
- Also data files, install scripts, documentation
- Record all details in configuration database: enough to recreate exactly
- What about upgrades/new releases?
One strategy: daily build & test

- Common industry strategy
  - Deadline for changes each day
  - Full build of new version
  - Full run of all regression tests
  - Defects recorded and sent back to developers

- Alternatively, make a full test run part of the check-in process for every change to the baseline – it doesn’t get in if it doesn’t work. (Aegis does this.)
Don’t duplicate information

- Store each item of information in one place only (e.g. a requirement)
- If it has to appear in more than one file, use scripts or build tool to replicate it automatically.
- Instead of replicating (in documents) consider the possibility of hyperlinking
- This helps with traceability – each design module has hyperlinks to the relevant requirements (and vice versa)
Tools

- Version control: RCS, CVS
- Bug tracking: Bugzilla, Issuezilla
- Build management: Make, Ant
- Change control: Aegis
- SourceForge
- CASE tools
- IDEs
- Wiki for documentation?
Further reading

- Humphrey, *Introduction to the Team Software Process*, Appendix B.


- IEEE Std 828-1998 CM Plans

- ANSI-IEEE Std 1042-1987 Guide to CM

- McConnell, *Code Complete* (or others)
Case studies for discussion

• COMP2100 course materials
  – Central CVS repository
  – Build script for web pages
  – Big makefile
  – Local and published versions of web site
  – Problems with updates via rsync

• Tim Wilson-Brown’s team