

Theme I

Software Processes

Software Configuration Management

Roadmap

- **Software Configuration Management**
 - **Software configuration management goals**
 - **SCM Activities**
 - **Configuration Management Plans**
 - **Configuration Management Systems**

Learning Goals

- What is the purpose of software configuration management?
- What activities does it consist of?
- How do you plan for configuration management?
- What tools are available to support it?
- How is configuration management handled in large projects, in practice?

Software Configuration Management

- SCM is the process of identifying, tracking, and storing all artifacts on a project.
- Each of these artifacts is referred to as a Configuration item (CI)
- Goals:
 - **Baseline Safety**
Ensure that new or changed CI's are safely stored in a repository and can be retrieved when necessary
 - **Overwrite Safety**
Ensure that engineer's changes to the same CI are applied correctly
 - **Reversion**
ensure ability to revert to earlier version
 - **Disaster Recovery**
retain backup copy in case of disaster

SCM Activities (I)

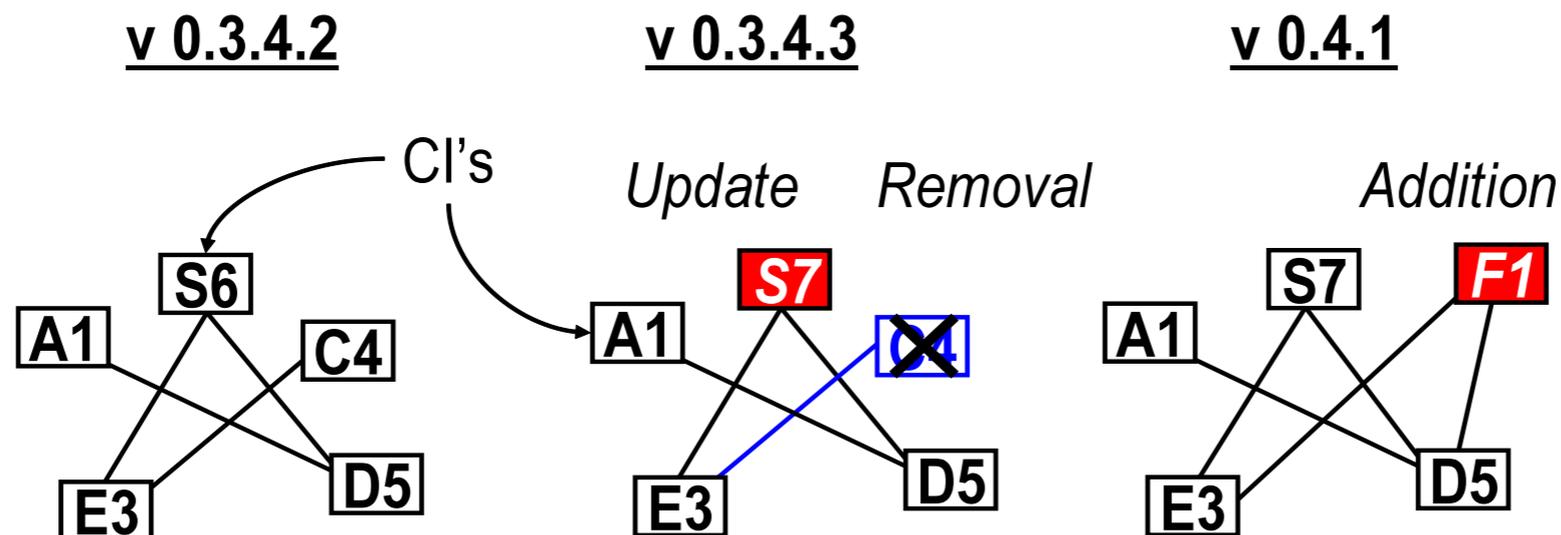
I. Configuration identification

- first step in CM is to identify the CI's,
- candidate CI's include:
 - source and object code
 - project specifications
 - user documentation
 - test plans and data
 - supporting software such as compilers, and editors
 - any artifact that will undergo modification or need to be retrieved at some time after its creation

SCM Activities (2)

2. Baselines

- an approved artifact
- individual CI or collections of CIs at key project milestones
- created by recording the version number of all the CIs at that time and applying a version label to uniquely identify it.
- are utilized for three primary reasons:
 - reproducibility
 - traceability
 - reporting



copyright owned by 2010 John Wiley and Sons

SCM Activities (3)

3. Change control

- = configuration control
- Set of activities to request, evaluate, approve or disapprove, and implement changes to baselined CI's
- Steps:
 - identification and documentation
 - analysis and evaluation
proposed changes are analyzed and evaluated for correctness, as well as the potential impact on rest of system
 - approval or disapproval
 - verification, implementation and release

SCM Activities (4)

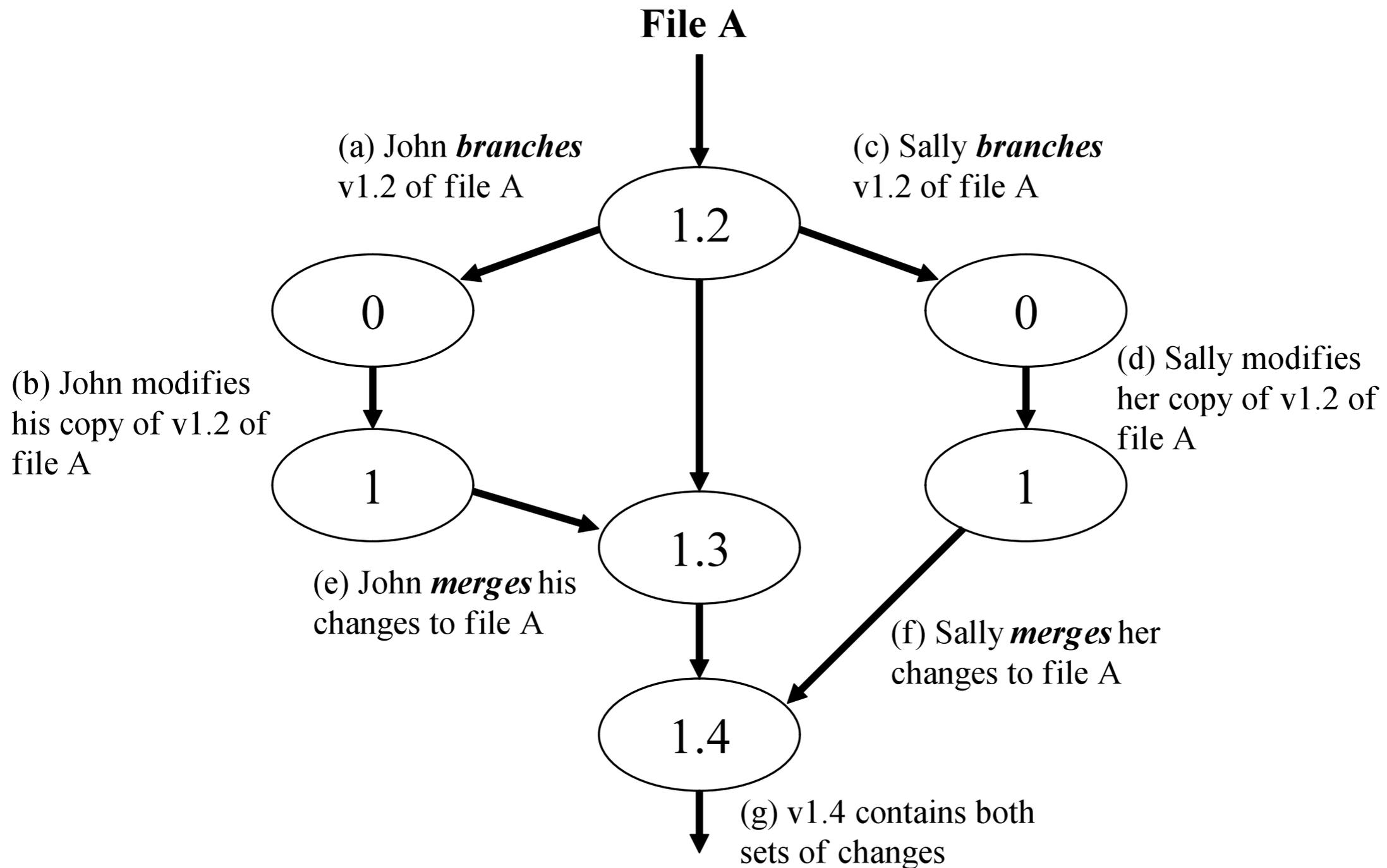
4. Version control

management and storage of CIs as they are created and modified throughout the SDLC.

Version control supports:

- Repository = centralised database stores all artifacts and keeps track of versions
- Checkout/Checkin
- Branching and Merging
working on the same set of files concurrently (branching) and correctly applying changes to common files.
- Builds
- Version Labeling

Example Branching and Merging



SCM Activities (5)

5. Configuration audits

- **audit:** *independent examination of a software product, software process, or a set of software processes performed by a third party to assess compliance with specifications, standards, contractual agreements, or other criteria.*
- Goals of a configuration audit:
 - Verify proper procedures are followed, such as formal technical reviews
 - Verify SCM policies, such as those defined by change control, are followed
 - Determine whether a software baseline is comprised of the correct configuration items

SCM Activities (6)

6. Configuration status reporting

- Extraction, arrangement and formation of configuration reports
 - Name and version of CI's
 - Approval history of changed CI's
 - Software release contents and comparison between releases
 - Number of changes per CI
 - Average time taken to change a CI

7. Release management and delivery

Policies to ensure that once software and documentation is released it is archived safely and reliably, and can always be retrieved for future use

CM Plans

IEEE 828-2005 SCMP Table of Contents

- 3.1 Introduction
- 3.2 **SCM management**
 - 3.2.1 Organization
 - 3.2.2 SCM responsibilities
 - 3.2.3 Applicable policies, directives & procedures
 - 3.2.4 Management of the SCM process
- 3.3 **SCM activities**
 - 3.3.1 Configuration identification
 - 3.3.1.1 Identifying configuration items
 - 3.3.1.2 Naming configuration items
 - 3.3.1.3 Acquiring configuration items

- 3.3.2 Configuration control
 - 3.3.2.1 Requesting changes
 - 3.3.2.2 Evaluating changes
 - 3.3.2.3 Approving or disapproving changes
 - 3.3.2.4 Implementing changes
- 3.3.3 Configuration status accounting
- 3.3.4 Configuration evaluation & reviews
- 3.3.5 Interface control
- 3.3.6 Subcontractor / vendor control
- 3.3.7 Release management and delivery
- 3.4. **SCM schedules**
- 3.5. **SCM resources**
- 3.6. **SCM plan maintenance**

copyright owned by 2010 John Wiley and Sons

CM Systems

- Concurrent Version System (CVS)
- Subversion (SVN)
- Git
- Monticello (Smalltalk)
- Mercurial
- Microsoft SourceSafe
- Rational Clearcase

Case Study: CM in Eclipse

- Two classes of Eclipse users:
 - read and/or modify Eclipse code but do not have write access to the CVS repository
 - have write permission for the Eclipse source code and can modify and update the CVS repository. (committers)
- Version Numbering

