

# Versioning

---



## Terms

---

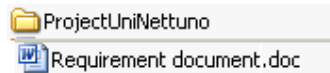
- Configuration item (CI)
- Version
- Configuration Management aggregate
- Configuration
- Baseline



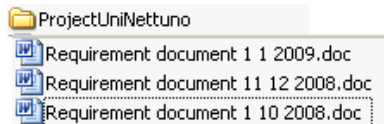
# Versioning

---

- No history



- Different names



- Tool capable of keeping track of versions
  - ♦ Same name
  - ♦ Different version name (ex 1.0 2.0 2.1 or 1,2,3,)
  - ♦ User decides when to change version (commit)
  - ♦ Always possible to recover a past version

**SOftEng**  
http://softeng.polito.it

---

# Versioning – CI

---

- CI, configuration item
- Element (file) under configuration control
  - ♦ Has a name and a version number
  - ♦ All its version numbers are kept
  - ♦ User decides to change version number with specific operation (commit)
  - ♦ It is possible to retrieve any version

**SOftEng**  
http://softeng.polito.it

---

## Configuration Item

---

- **Unit for the CM system**

a work product or piece of software that is treated as a single entity for the purpose of configuration management.

- ♦ **May correspond to one/more document(s), one/more programs**

- Simple example of CIs
  - Requirement document
  - Design document
  - Source code module

**SoftEng**  
<http://softeng.polito.it>

---

## Version

---

- **Instance of CI**

- ♦ Ex Req document 1.0
- ♦ Req document 1.1

**SoftEng**  
<http://softeng.polito.it>

---

## Version identification

---

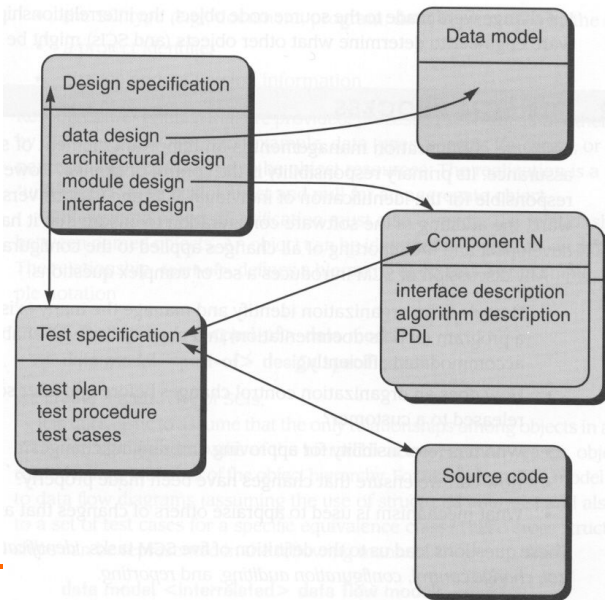
- Procedures for version identification should define an unambiguous way of identifying component versions
- basic techniques for component identification
  - ♦ Version numbering
  - ♦ Attribute-based identification

## Version numbering

---

- Simple naming scheme uses a linear derivation  
e.g. V1, V1.1, V1.2, V2.1, V2.2 etc.
- Actual derivation structure is a tree or a network rather than a sequence
- Names are not meaningful.
- Hierarchical naming scheme may be better

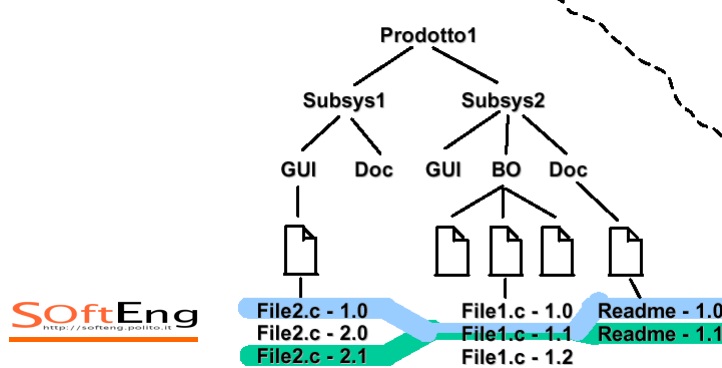
# Links between CIs



**SoftEng**  
http://softeng.polito.it

## Ex.

- ◆ System 1.0 (configuration 1.0)
  - File2.c 1.0 + File1.c 1.0 + Readme 1.0
- ◆ System 1.1 (configuration 1.1)
  - File2.c 1.0 + File1.c 1.1 + Readme 1.0

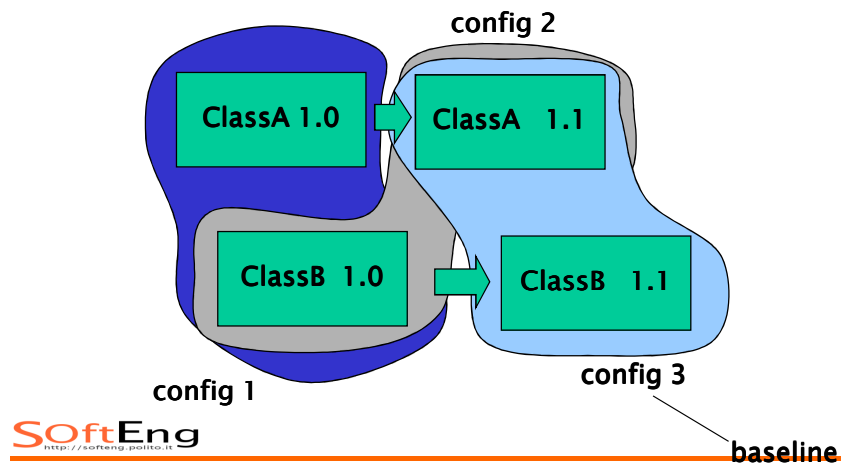


**SoftEng**  
http://softeng.polito.it

# Configuration

---

- Set of CIs, each in a specific version



- 
- Choices of CM system
    - ♦ What parts of software system become CIs
    - ♦ (not all documents may become CIs)
  - Changes to CI are subject to procedures defined by CM system
    - ♦ Typically, change must be approved and recorded
    - ♦ New version of CI must be generated

## Configuration

---

- Snapshot of software at certain time
  - ♦ Various CIs, each in a certain version
  - ♦ Same CI may appear in different configurations
  - ♦ Also configuration has version

## Baseline

---

- configuration in stable, frozen form
  - ♦ Not all configurations are baselines
  - ♦ Any further change / development will produce new version(s) of CI(s), will not modify baseline
- Types of baselines
  - ♦ Development – for internal use
  - ♦ Product – for delivery

## Derivation history

---

- Record of changes applied to a document or code component
- Should record, in outline, the change made, the rationale for the change, who made the change and when it was implemented
- May be included as a comment in code. If a standard prologue style is used for the derivation history, tools can process this automatically

**SoftEng**  
http://softeng.polito.it

---

## Component header info

---

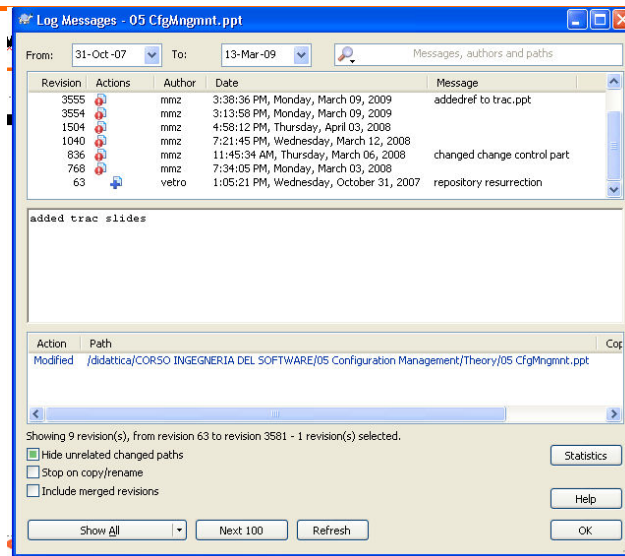
```
// PROTEUS project (ESPRIT 6087)
//
// PCL-TOOLS/EDIT/FORMS/DISPLAY/AST-INTERFACE
//
// Object: PCL-Tool-Desc
// Author: G. Dean
// Creation date: 10th November 1998
//
// © Lancaster University 1998
//
// Modification history
// Version      Modifier Date      Change      Reason
// 1.0   J. Jones   1/12/1998   Add header  Submitted to CM
// 1.1   G. Dean   9/4/1999   New field  Change req. R07/99
```

**SoftEng**  
http://softeng.polito.it

---



# Derivation history – svn



SoftEng  
http://softeng.polito.it

# Svn – version identification

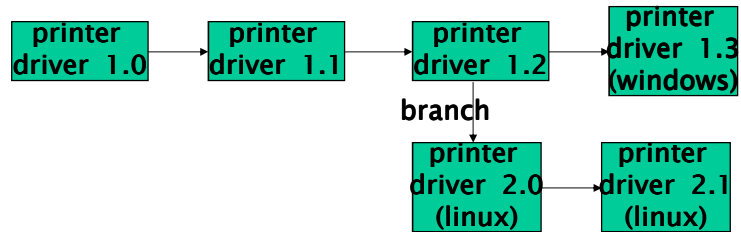
- In subversion a version is called → revision
- Each configuration has a new number
- Each element changes revision, even if has not been changed

revision#	1	2	3	4	5
	A	A	A'	A'	
		B	B	B'	B'

SoftEng  
http://softeng.polito.it

# Branches

---

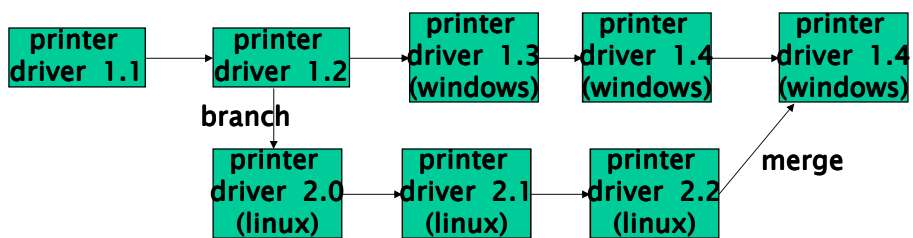


SOftEng  
<http://softeng.polito.it>

---

# Merge

---



SOftEng  
<http://softeng.polito.it>

---

## Change control

---

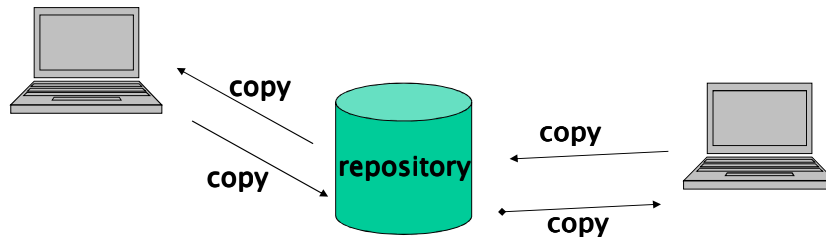
## Typical situation

---

- Team develops software
- Many people need to access parts of software
  - ◆ Common repository (shared folder), all can read/write documents/ programs

## Change control – repository

---

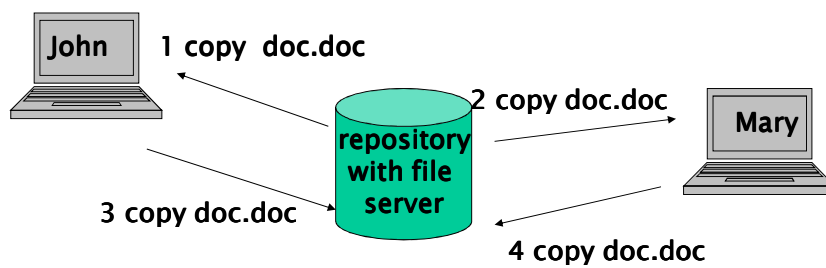


SoftEng  
<http://softeng.polito.it>

---

## Repository – file server

---



Changes by John are lost

SoftEng  
<http://softeng.polito.it>

---

# Change control

---

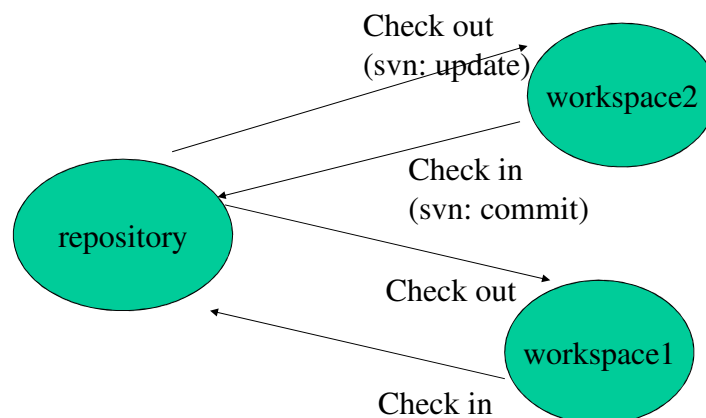
- Changes must be disciplined
  - ♦ Who controls
  - ♦ What is controlled
  - ♦ How control is implemented
- Approaches
  - ♦ Check in – check out model, Workspaces
  - ♦ CCB
    - On top of check in check out

**SOftEng**  
http://softeng.polito.it

---

# Workspace and check in/out

---



**SOftEng**  
http://softeng.polito.it

---

## Check-in check-out

---

- Check-out
  - ◆ Extraction of CI from repository
    - with goal of changing it or not
    - After checkout next users are notified
- Check-in
  - ◆ Insertion of CI under control

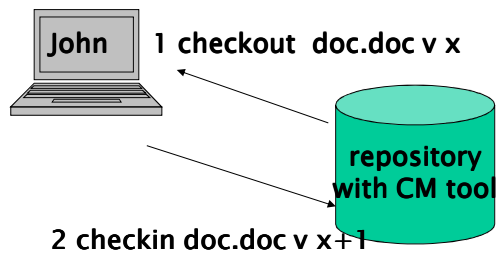
## Workspace

---

- 'Private' space where developer has full control

## Repository - check in checkout

---



**SoftEng**  
http://softeng.polito.it

---

## Checkin checkout vs. file system

---

### Check in /out

- CIs are in repository
- To rd/wr CI user needs to do check out
- After checkout next user knows that CI is used by someone else

### File system

- Files are in shared directory
- Any user can get copy of file, or work on original
- Users can work on copies of file without knowing that others are doing the same

**SoftEng**  
http://softeng.polito.it

---

## Check in/out – choices

---

- Who can do check in/out
- Checked-out CI is locked or not
  - ♦ If locked, one writer, many readers
    - One only can modify
- Checked-in CI increments version or not
  - ♦ If not, old version is lost

**SOftEng**  
<http://softeng.polito.it>

---

## Check in / check out – scenarios

---

- Lock modify unlock (or serialization)
  - ♦ One can change at a time
- Copy modify merge
  - ♦ Many change in parallel, then merge

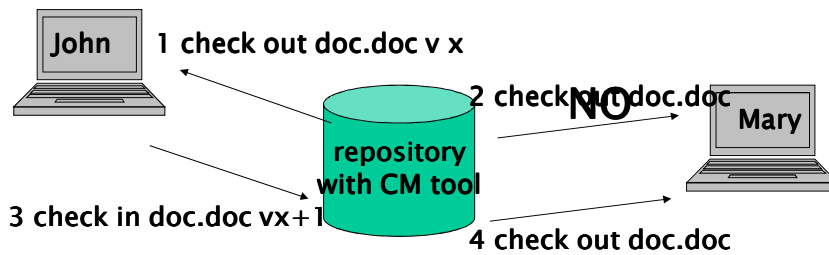
**SOftEng**  
<http://softeng.polito.it>

---



## Lock modify unlock

---



First check out locks the file  
No other checkouts are allowed until check in

SoftEng  
<http://softeng.polito.it>

## Problems – lock

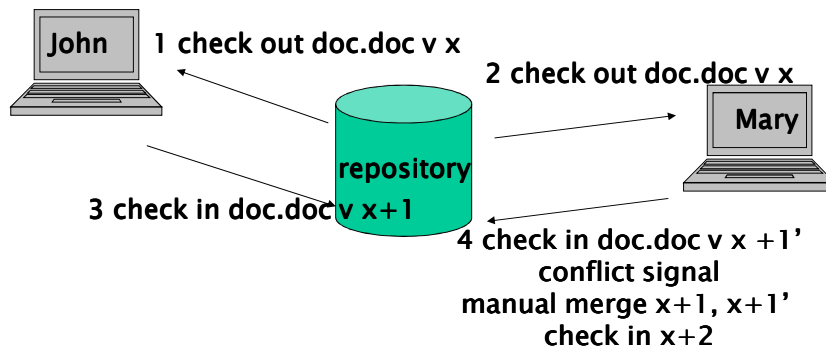
---

- Locker forgets to unlock
- No parallel work, delays

SoftEng  
<http://softeng.polito.it>

## Copy modify merge

---



**SoftEng** <http://softeng.polito.it>  
The second check in signals conflict  
User has to do a manual merge of x+1 and x+1' in x+2

---

## CM Tools

---

- CVS
- Subversion
- Clearcase
- Bitkeeper
- ..

**SoftEng** <http://softeng.polito.it>

---

## CCB

---

- Configuration Control Board
  - ◆ Authorizes changes to a baseline
    - Corrective maintenance
  - ◆ Defines what will be in next baseline
    - Perfective maintenance

## Configuration control board

---

- Changes should be reviewed by an external group who decide whether or not they are cost-effective from a strategic and organizational viewpoint rather than a technical viewpoint
- Should be independent of project responsible for system. The group is sometimes called a change control board
- May include representatives from client and contractor staff

## Change procedure

---

```
Request change by completing a change request form
Analyze change request
if change is valid then
  Assess how change might be implemented
  Assess change cost
  Submit request to change control board
if change is accepted then
  repeat
    make changes to software
    submit changed software for quality approval
  until software quality is adequate
  create new system version
else
  reject change request
else
  reject change request
```

**SoftEng**  
<http://softeng.polito.it>

---

## Change request form

---

- Definition of change request form is part of the CM planning process
- Records change required, suggestor of change, reason why change was suggested and urgency of change (from requestor of the change)
- Records change evaluation, impact analysis, change cost and recommendations (System maintenance staff)

**SoftEng**  
<http://softeng.polito.it>

---

## Change request form

---

Change Request Form	
<b>Project:</b> Proteus/PCL-Tools	<b>Number:</b> 23/94
<b>Change requester:</b> I. Sommerville	<b>Date:</b> 1/12/98
<b>Requested change:</b> When a component is selected from the structure, display the name of the file where it is stored.	
<b>Change analyser:</b> G. Dean	<b>Analysis date:</b> 10/12/98
<b>Components affected:</b> Display-Icon.Select, Display-Icon.Display	
<b>Associated components:</b> FileTable	
<b>Change assessment:</b> Relatively simple to implement as a file name table is available. Requires the design and implementation of a display field. No changes to associated components are required.	
<b>Change priority:</b> Low	
<b>Change implementation:</b>	
<b>Estimated effort:</b> 0.5 days	
<b>Date to CCB:</b> 15/12/98	<b>CCB decision date:</b> 1/2/99
<b>CCB decision:</b> Accept change. Change to be implemented in Release 2.1.	
<b>Change implementor:</b>	<b>Date of change:</b>
<b>Date submitted to QA:</b>	<b>QA decision:</b>
<b>Date submitted to CM:</b>	
<b>Comments</b>	

**SOftE**  
http://softeng.it

## Tools to support change process

---

- Trac, Jira, Bugzilla, ..
  
- See Trac demo
  - ◆ For trac: change → ticket
  - ◆ Usr demo pwd demo
- <http://www.hosted-projects.com/trac/TracDemo/Demo>

**SOftEng**  
http://softeng.polito.it

# Trac – create ticket

## Create New Ticket

Short summary:

Display name position

Type: **enhancement**

Full description (you may use [WikiFormatting](#) here):

**B I A** [img] [link] [code]

When a component is selected from the structure, display the name of the file where it is stored.  
 "Relatively simple to implement as a file name table is available. Requires the design and implementation of a display field. No changes to associated components are required"

Ticket Properties

Priority: **low** | Milestone: **2.1**  
 Component: **component1** | Version: **2.1**  
 Severity: **critical** | Keywords: **file table**  
 Assign to:  | Cc: **I. Sommerville, G. Dean**

I have files to attach to this ticket

**SoftEng**  
<http://softeng.polito.it>

# Trac – see all (active) tickets

Use Demo/Demo to log in as the admin user



Integrated SCM & Project Management

logged in as Admin | Logout | Settings | Help/Docs | About Trac

Home | Tickets | Roadmap | Errors/Source | **View Tickets** | New Ticket | Search | Admin

## {1} Active Tickets (14 matches)

- List all active tickets by priority.
- Color each row based on priority.
- If a ticket has been accepted, a "\*" is appended after the owner's name

Ticket	Summary	Component	Version	Milestone	Type	Owner	Created
#5	None	component2		None	enhancement	None	02/20/07
#8	None	component2	2.0	None	enhancement	None	02/20/07
#3	None	component2	2.0	None	task	None	02/20/07
#9	bla	Test1		None	task	Yo4iz	03/09/09
#2	None	component1	1.0		task	None	02/20/07
#4	pacifica high school	component3	1.2	2.1	task	None	02/20/07
#11	teste	component1	2.0	2.1	task	anonymous *	03/10/09
#13	Error en cálculo del ETJ	Docs	1.5	2.1	defect	Carlos	03/10/09
#10	teste	Docs	2.0	2.1	defect	somebody	03/09/09
#16	ORA test1	rFAC	2.0	2.1	defect	anonymous	03/10/09
#12	Short summary	Docs	2.0	2.1	task	somebody	03/10/09
#14	ad asdi adf	component1	2.0	2.1	task	Pelle	03/10/09
#1	crayola marker	component1	1.1	3.0	defect	demo *	02/20/07
#15	Display name position	component1		2.1	enhancement	somebody	03/10/09

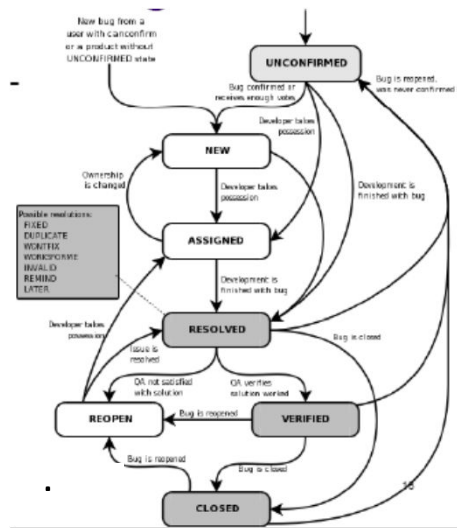
Note: See [TracReports](#) for help on using and creating reports.

**SoftEng**  
<http://softeng.polito.it>

# Trac - open ticket

SoftEng  
<http://softeng.polito.it>

# Lifecycle for change (bug)



SoftEng  
<http://softeng.polito.it>