

CZ3002 - Advanced Software Engineering

Configuration Management

Faculty : Dr Althea Liang
School : School of Computer Science and Engineering
Email : qhliang@ntu.edu.sg
Office : N4-02c-107

Quick Review Sheet of Previous Lesson

In the last lesson:

- ▶ Software change is inevitable
 - ❖ Internal Reasons
 - ❖ External Reasons
- ▶ Change must be managed
 - ❖ Managing amount of changes
 - ❖ Managing potential impacts of changes
- ▶ Change Control Board (CCB) plays important roles in change management
- ▶ Change control process goes through a sequence of steps from identifying the controlled items and making a Change Request (CR) to closure of the CR

Lesson Objectives

At the end of the lesson, you should be able to, with the help of a number of well organised examples:

- ▶ Describe the purpose of software configuration management
- ▶ Describe the software configuration management functional areas and processes
- ▶ Explain software configuration item
- ▶ Explain the terms such as versions, variants, branch and tag used in software configuration management
- ▶ Apply version numbering in Concurrent Version System (CVS) and SVN



Preview: Example Organisation in this Lesson

Example	Slides
B1, B2	Baseline Tag
V1, V2, V3	Version Numbering in CVS and SVN Delta
R	Version Numbering in CVS and SVN Branching in CVS
E	Branching in SVN

Preview: Best Practice Organisation in this Lesson

Best Practices/Tools	Slides
Minimise the number of branches	Branching and Merging
Merge frequently	Branching and Merging
Teams to communicate likely conflicts	Branching and Merging
CVS, SVN	Version Numbering in CVS and SVN Branching in CVS Branching in SVN

The discipline of identifying the configuration of a system at discrete points in time for the purposes of systematically controlling changes to this configuration and maintaining the integrity and traceability of configuration throughout the system life-cycle.

Bersoff, Henderson and Siegel's, *Software Configuration Management - An Investment in Product Integrity*, (1980).

SCM Objectives

- ▶ Help organisations/ teams maintain consistency and control over what is produced.
- ▶ Ensure that:
 - ❖ Correct products are delivered (right versions).
 - ❖ Products satisfy the requirements (traceability).
 - ❖ Software and supporting materials (hardware, documents, training material, etc) are consistent (system integrity).

SCM Functional Areas

- ▶ Release Management
- ▶ Source Code Management a.k.a. Version Control
- ▶ Change Management
- ▶ Build Engineering
- ▶ Deployment
- ▶ Environment Configuration

SCM - A Process Perspective

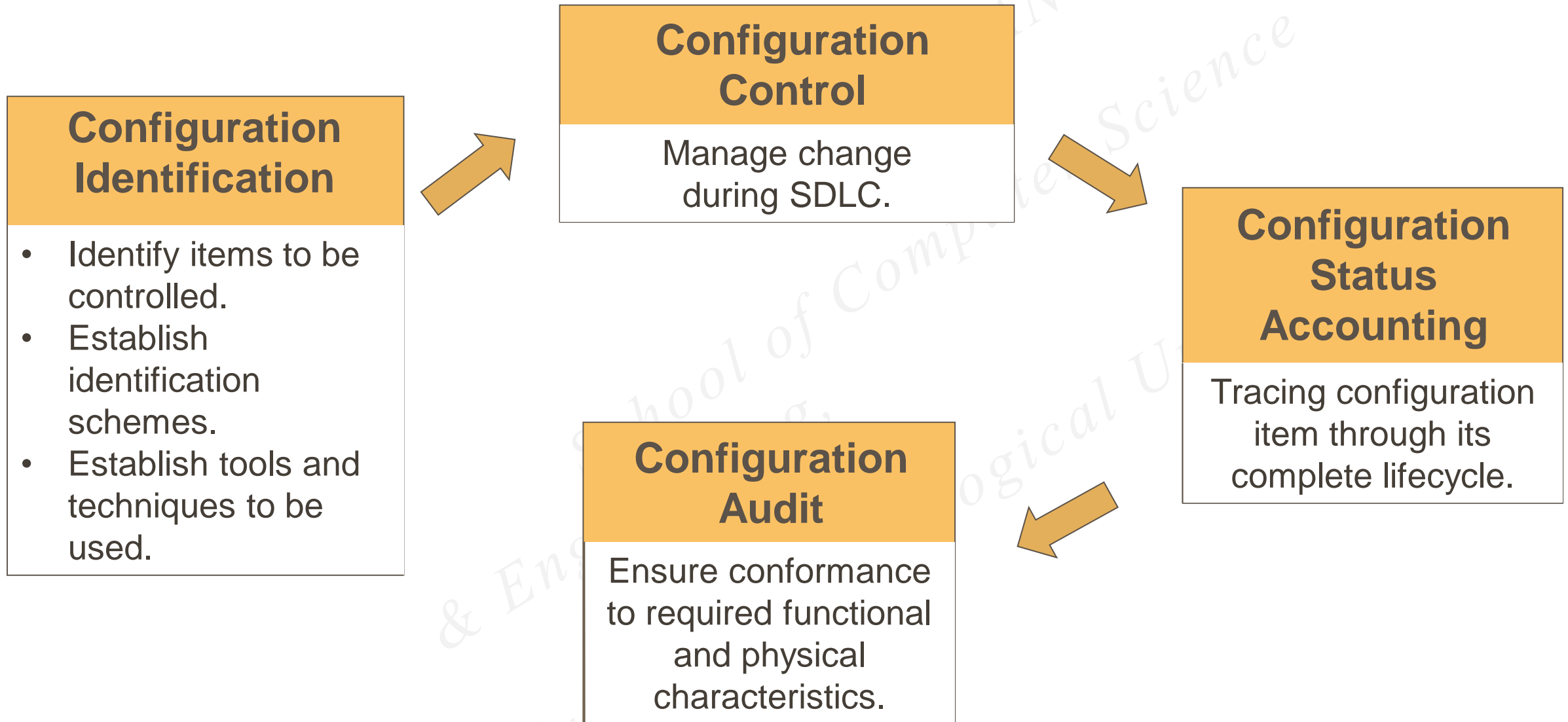
Software
Configuration
Identification

Software
Configuration
Control

Software
Configuration
**Status
Accounting**

Software
Configuration
Auditing

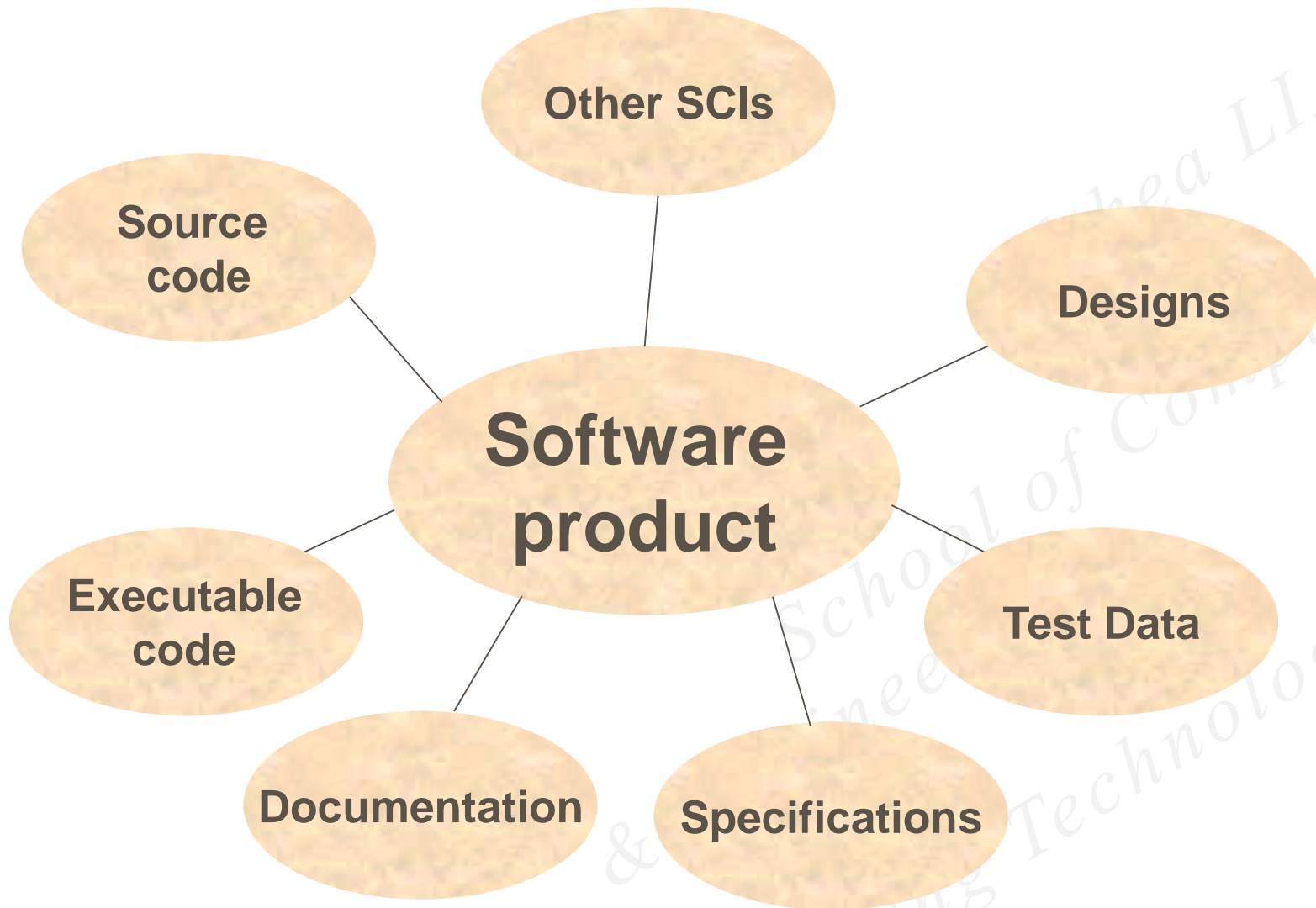
Steps in Process Perspective of SCM



Software Configuration Item (SCI)

- ▶ Configuration Identification
 - ❖ To identify software configuration items.
- ▶ In the context of SCM, a component or part of a software product is called a Software Configuration Item (SCI).
- ▶ Software Configuration Items may be:
 - ❖ Decomposed into further SCIs.
 - ❖ Modified to create new versions of the original SCIs.

Software Configuration Item



- Identifying and naming of SCIs is the first step of the SCM process.
- Usually this is done on a project basis – relevant SCIs are associated with each major software product from the SLC process.

Baselines

- ▶ A set of configuration items formally designated and fixed at a specific point in time.

BEFORE

establishment of baseline

Changes can be made quickly and informally.

AFTER

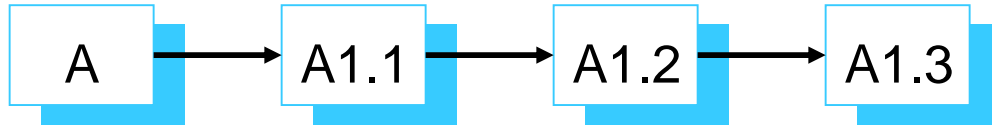
establishment of baseline

Changes to the SCI can only be made under a formal change control procedure.

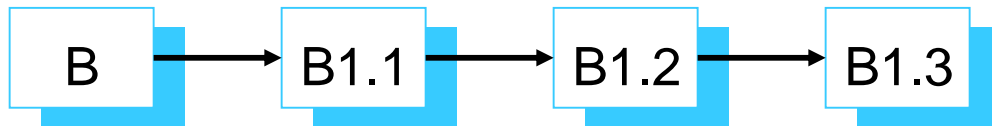
- ▶ Typically, a baseline with a SCI will be established once a deliverable form of the SCI has been completed or after the formal technical review of a SCI. Subsequent baselines may be established for specific purposes, e.g. testing.

Baseline (Examples B1, B2)

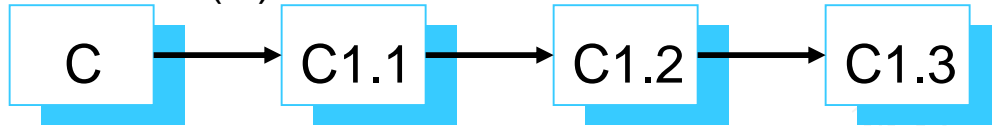
Codeline (A)



Codeline (B)



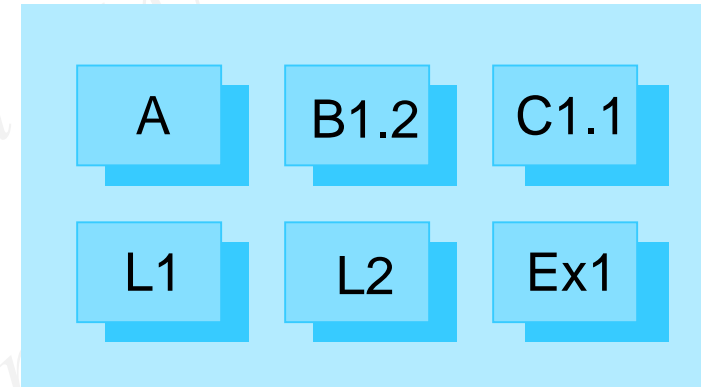
Codeline (C)



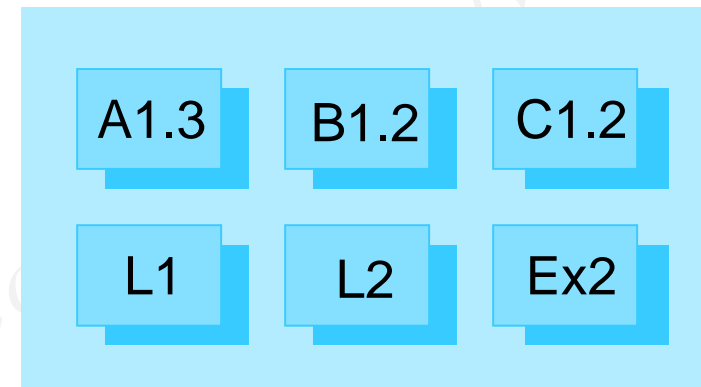
Libraries and External Components



Baseline – B1



Baseline – B2

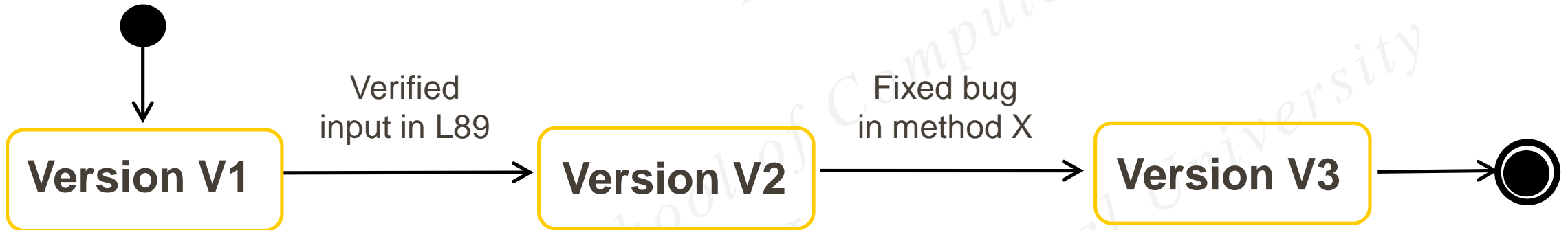


Mainline

Versions v.s. Variants (Examples V1, V2, V3)

► Version

- ❖ Used to indicate state of a configuration item at a well-defined point in time. Example shown below.

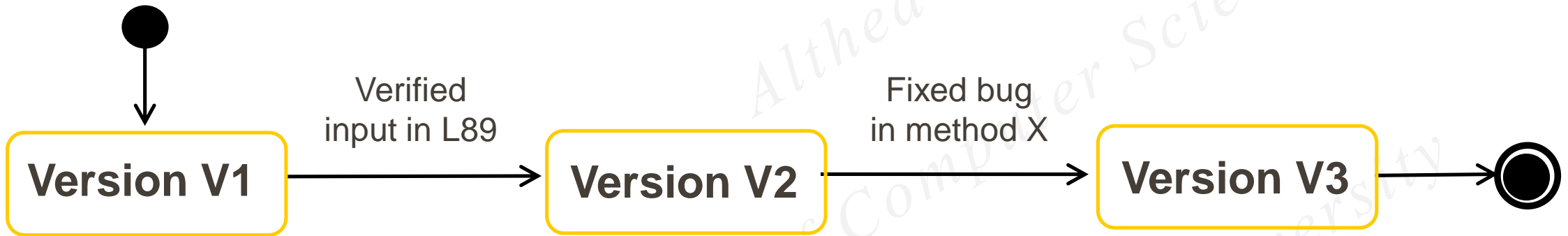


► Variants

- ❖ Collectively refer to versions that coexist (typically to support different hardware, operating systems or customer requirements).

Delta (Example V1, V2, V3 continued)

- Difference between one revision and other.

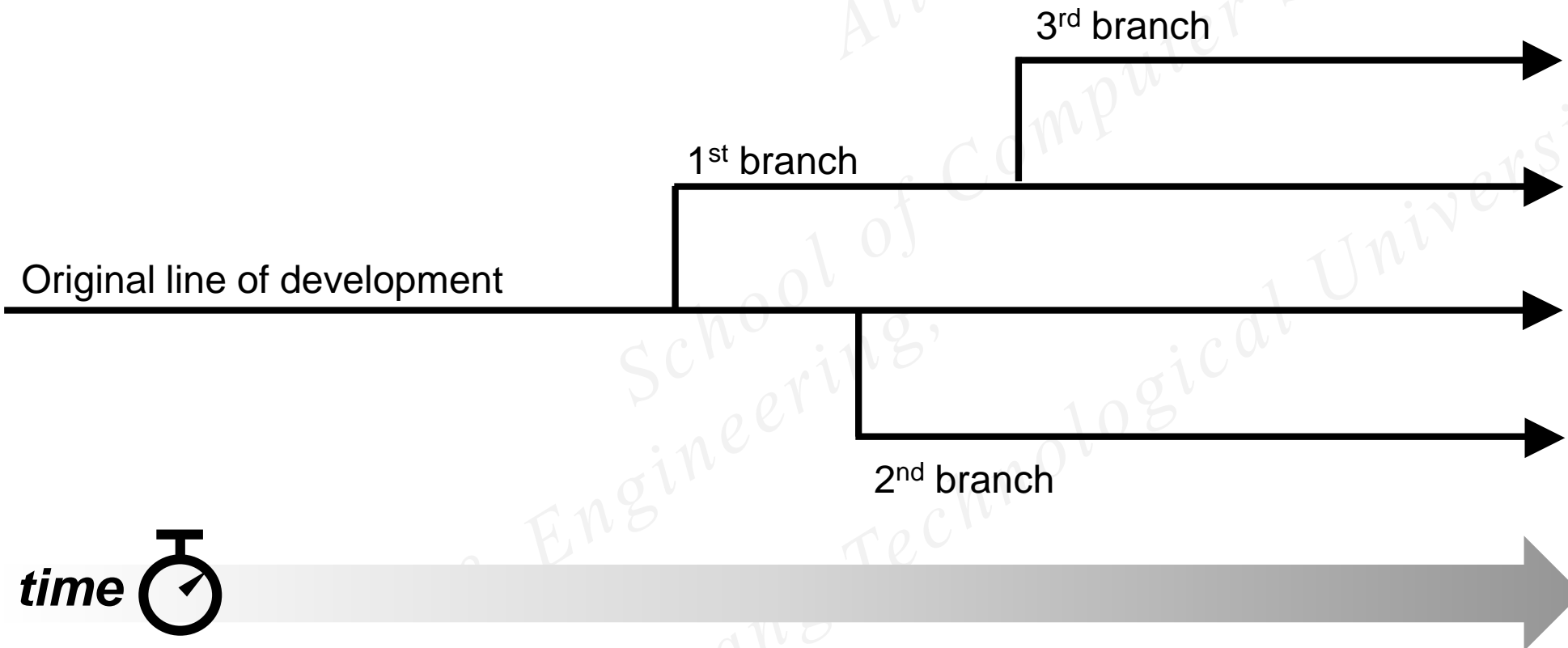


Delta between Version V3 and Version V1 consists of:

- Bug fix in method X
- Code to verify input in L89

Branch

- ▶ A line of development that exists independently of another line, yet still shares a common history, and can be merged in the future.



Tag (Example B1, B2 continued)

- ▶ A “snapshot” of a project in time.

time
↓

FileA	FileB	FileC	L1	L2
A1	B1	C1	L1	L2
A1.1	B1.1	C1.1		
	B1.2	C1.2		
		C1.3		
		C1.4		

- ▶ Tag “Blue” groups FileA (A1.1), FileB (B1.2), FileC (C1.4), FileL1 (L1), FileL2 (L2).

Version Numbering in CVS and SVN (Example R)

- ▶ In CVS, each version of a file has a unique revision number.

```

+-----+   +-----+   +-----+   +-----+   +-----+
! 1.1 !----! 1.2 !----! 1.3 !----! 1.4 !----! 1.5 !
+-----+   +-----+   +-----+   +-----+   +-----+

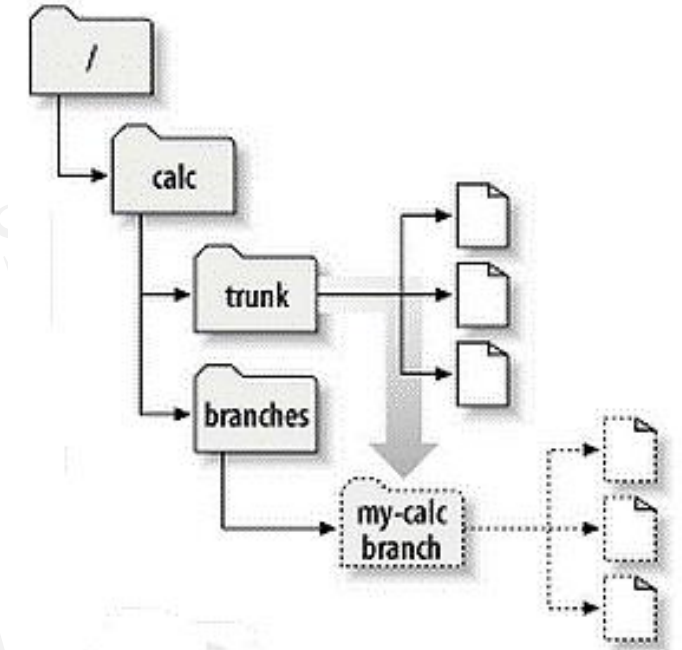
```

- ▶ SVN uses “Global Revision Numbers”
 - ❖ SVN's revision numbers apply to entire trees, not individual files.
 - ❖ Revision **N** represents the state of the repository file system after the **N**th commit.



Branching in SVN (Example E)

- ▶ SVN makes a copy of the directory.
- ▶ To work on the branch, check out a new working copy.



```
$ svn checkout http://svn.example.com/repos/calc/branches/my-calc-branch
A my-calc-branch/Makefile
A my-calc-branch/integer.c
A my-calc-branch/button.c
Checked out revision 341.
```

Branching and Merging - Best Practices

- ▶ Minimise the number of branches
- ▶ Merge frequently
- ▶ Teams to communicate likely conflicts

Althea LIANG
School of Computer Science
& Engineering,
Nanyang Technological University

- ▶ Compliance to the standard: IEEE 828 Standard for Software Configuration Management Plan.
- ▶ Main Sections of the Software Configuration Management Plan
 - ❖ Introduction
 - ❖ SCM Management
 - CM Organisational Structure (personnel, resources, projects supported, etc)
 - ❖ SCM Activities
 - CI identification, control, audit, reporting, including vendor control
 - ❖ SCM Schedules
 - ❖ SCM Resources
 - ❖ SCM Plan Maintenance

Summary

At the end of the lesson, you should be able to, with the help of a number of well-organised examples:

- ▶ Describe the purpose of software configuration management
- ▶ Describe the software configuration management functional areas and processes
Explain software configuration item
- ▶ Explain the terms such as versions, variants, branch and tag used in software configuration management
- ▶ Apply version numbering in CVS and SVN

**Special Thanks to Kydon
during the TEL Efforts of the
Lecture**

End of Configuration Management

Faculty : Dr Althea Liang
School : School of Computer Science and Engineering
Email : qhliang@ntu.edu.sg
Office : N4-02c-107