Faculty of ECMS / School of Computer Science

Software Engineering & Project Configuration Management

Configuration Management

Lecture 9

Chapter 29 (24 in Edition 9)
in the course text book

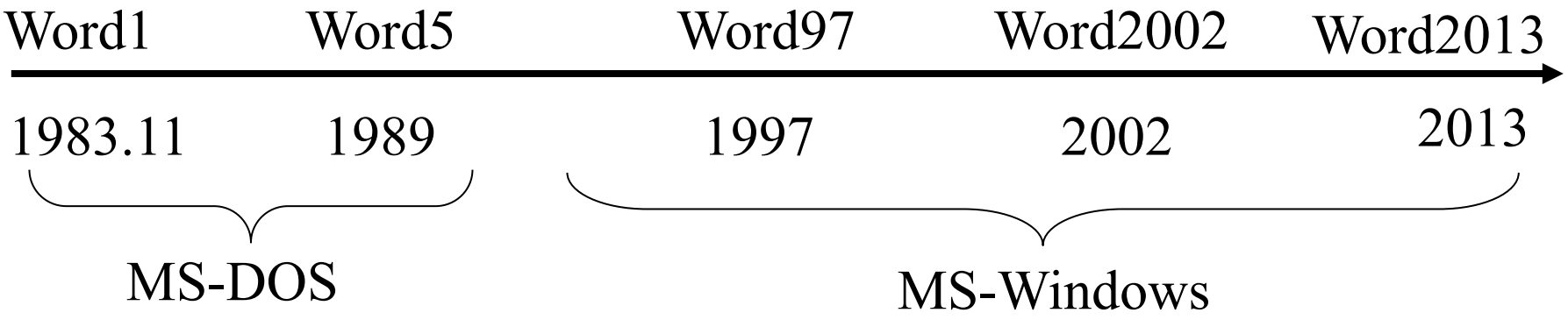
Outline

- Configuration Management: What and Why
- Fundamental CM Activities
  - Configuration management planning
  - Change management
  - Version and release management
  - System building
- CASE Tools for Configuration Management

Configuration Management: What and Why

- Anything that can tell you about the software (e.g., MS Windows)
  - New version, Release, Security patch, Bug (fix), etc

Let us look at the history of MS Word



30 years efforts, numerous changes, many versions

Configuration Management: What and Why (cont.)

- New versions of software systems are created as they change:
  - For different machines/OS/platforms;
  - Offering different functionality;
  - Tailored for particular user requirements.
- Configuration management is concerned with managing evolving software systems:
  - System change is a team activity (CM team);
  - CM aims to control the costs and effort involved in making changes to a system.

Configuration Management: What and Why (cont.)

- Involves the development and application of policies, procedures, standards, and tools to manage an evolving software product
- May be seen as part of a more general quality management process
- If no there is no configuration management
  - May waste time modifying the wrong version
  - May lost track of where the source code is stored
  - May deliver the wrong version to customers
  - May be fixing a bug that is irrelevant now
  - And so on

Configuration Management Planning

- All products of the software process may have to be managed
  - Specifications
  - Designs
  - Programs
  - Test data
  - User manuals
- Hundreds of separate documents are generated for a large software system
- The CM plan describes the standards and procedures that should be used for configuration management

The Configuration Management Plan

- CM planning should start during the early phases of the project
  - Same goes for quality management planning
- Defines the types of documents to be managed (and where) as well as a document naming scheme
- Defines who takes responsibility for the CM procedures
- Defines policies for change control and version management
- Defines the CM records which must be maintained

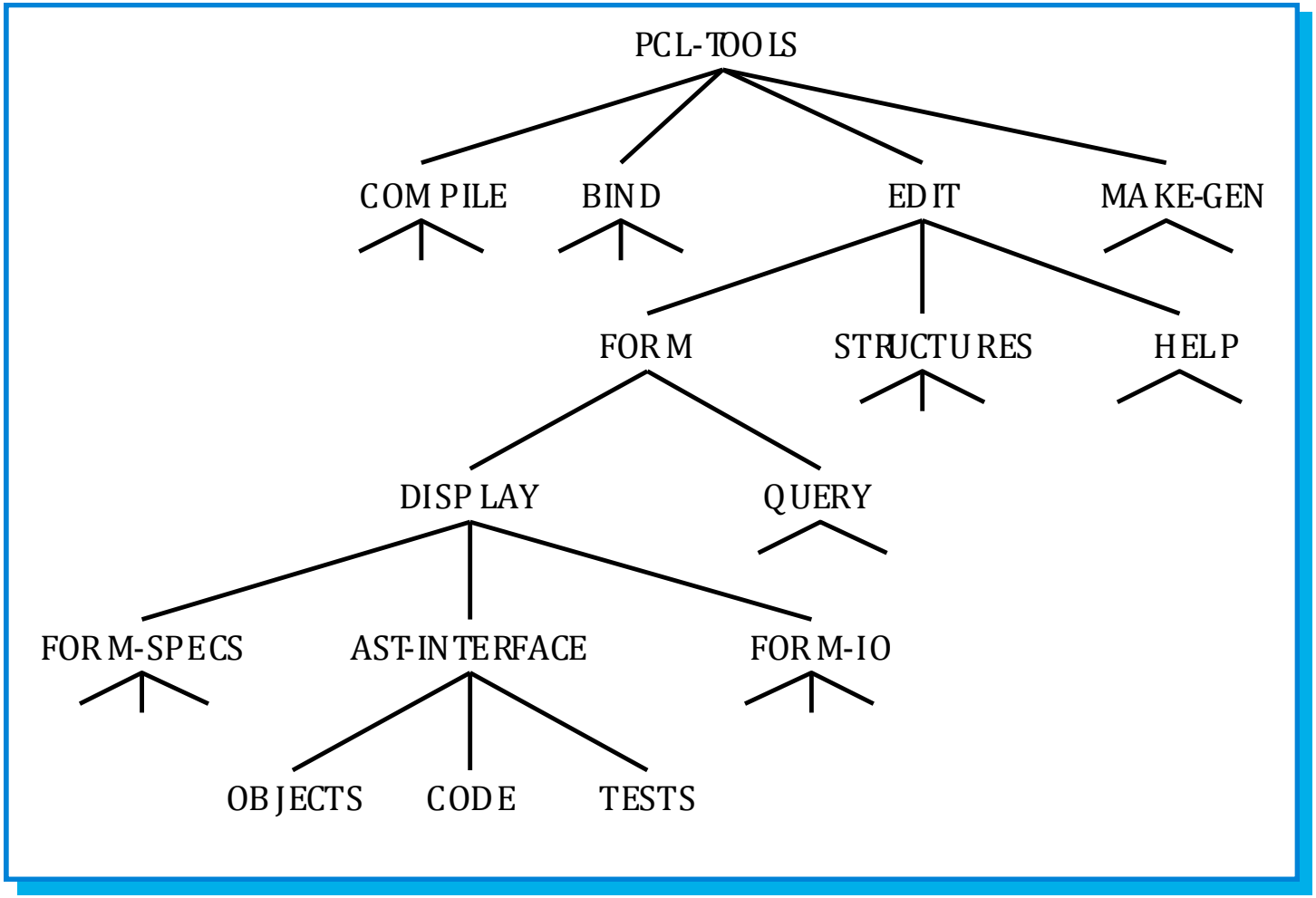
The Configuration Management Plan (Cont.)

- Describes the tools which should be used to assist the CM process and any limitations on their use
- Defines the process of how the tool is used
  - Not just what tool is
- Defines the CM database used to record configuration information
  - This does not imply the existence of a relational database! Maybe just files…
- May include information such as the CM of external software, process auditing, etc.

Configuration Item Identification

- Large projects typically produce hundreds of documents which must be uniquely identified
- Some of these documents must be maintained for the lifetime of the software
- A document naming scheme should be defined so that related documents have related names.
- A hierarchical scheme with multi-level names is probably the most flexible approach

Configuration Item Identification Example



Change Management

- Software systems are subject to continual change requests
  - From users
  - From developers
  - From market forces

- Change management is concerned with the managing of these changes and ensuring that they are implemented in the most cost-effective way

The Change Management 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 the software
            Submit software changes for quality approval
        until software quality is adequate
        Create new system version
    else
        Reject change request
else
    Reject change request

Change Request Form

- The definition of a change request form is part of the CM planning process.
- This form records the change proposed, requestor of change, the reason why change was suggested and the urgency of change (from requestor of the change).
- It also records change evaluation, impact analysis, change cost and recommendations

Change Request Form Example

Change Request Form

Project: HEMP **Number:** 007/13
Change requester: Jarad Anderson **Date:** 20/05/13
Requested change: When a new computer is selected from the catalog, display the name of the school who owns the device.

Change analyser: Simon/Anthony **Analysis date:** 22/05/13
Components affected: Display-Icon.Select, Display-Icon.Display

Associated components: FileT able

Change assessment: 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.

Change priority: Low

Change implementation:
Estimated effort: 0.5 days

Date to CCB: 22/05/13 **CCB decision date:** 23/05/13
CCB decision: Accept change. Change to be implemented in Release 1.1.
Change implementor: Scott/James/Luke **Date of change:** 28/05/13
Date submitted to QA: **QA decision:**
Date submitted to CM:
Comments

Change Control Board (CCB)

- 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.
  - The consequences of not making the change
  - The benefits of the change
  - The number of users affected by the change
  - The costs of making the change
  - The product release cycle
- Should be independent of project responsible for system. The group is sometimes called a *change control board* (CCB).
- CCB may include representatives from client and contractor staff (usually a formal requirement for military projects)

Version and Release Management

- Invent identification scheme for system versions
- Plan when new system version is to be produced
- Ensure that version management procedures and tools are properly applied
- Plan and distribute new system releases

Some Terminologies: versions/variants/releases

- Version - An instance of a system which is functionally distinct in some way from other system instances (e.g., MS Word97, Word2007)

- Variant - An instance of a system which is functionally identical but non-functionally distinct from other instances of a system (e.g., MS Word10 for Windows; for Mac OS)

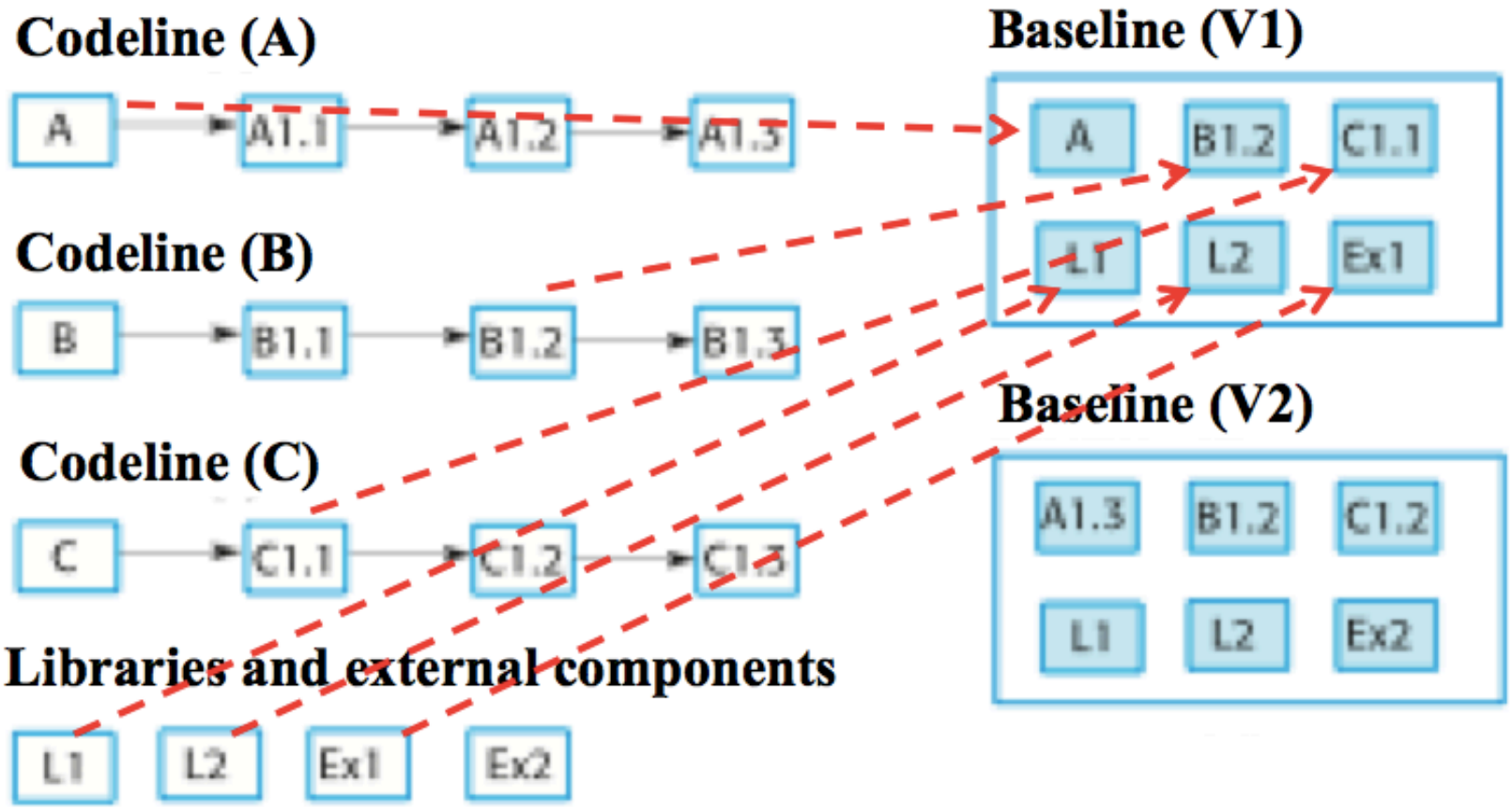
- Release - An instance of a system which is distributed to users outside of the development team (e.g., Word 2007 for Windows Vista, released 30/01/2007)

Some Terminologies: Codelines and baselines

- A codeline is a sequence of versions of source code with later versions in the sequence derived from earlier versions.
- Codelines normally apply to components of systems so that there are different versions of each component.

- A baseline is a definition of a specific system.
- The baseline therefore specifies the component versions that are included in the system plus a specification of the libraries used, configuration files, etc.

Codelines and Baselines



Version Management

- Version management (VM) is the process of keeping track of different versions of software components or configuration items and the systems in which these components are used.
- It also involves ensuring that changes made by different developers to these versions do not interfere with each other.
- Therefore version management can be thought of as the process of managing codelines and baselines.

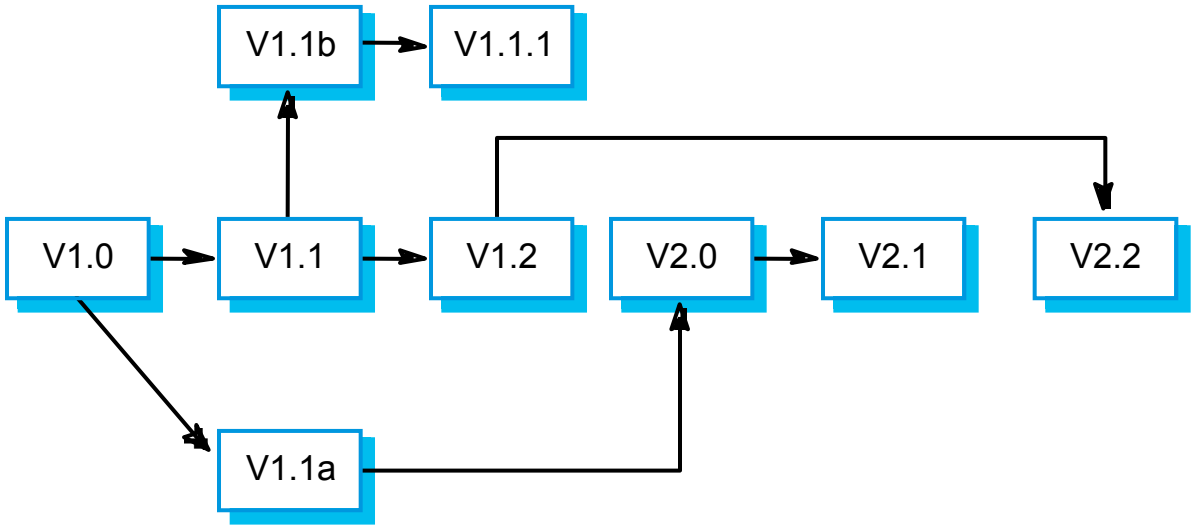
Version Identification

- 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
  - V1, V1.1, V1.2, V2.1, V2.2 etc.
- The actual derivation structure is a tree or a network rather than a sequence
- Names are not meaningful (extra information needed)



Release Management

- A system release is a version of the system that is distributed to customers

- Release management includes deciding when to issue a system version as a release; managing the process of creating the release; documenting the release so that re-creation is possible

System Releases

- Not just a set of executable programs
- May also include
  - Configuration files defining how the release is configured for a particular installation
  - Data files needed for system operation
  - An installation program or shell script to install the system on target hardware
  - Electronic and paper documentation
  - Packaging and associated publicity
  - Description of changes/known issues
- Systems are now normally released on optical disks (e.g., CD, DVD) or as downloadable installation files from the Web

Release Problems

- Customer may not want a new release of the system
  - They may be happy with their current system as the new version may provide unwanted functionality

- Release management should not assume that all previous releases have been accepted. All files required for a release should be re-created when a new release is installed
  - Tradeoff between efficient release deployment and less complicated release migration process

Release Decision Making

- Preparing and distributing a system release is an expensive process

- When to release: Factors such as the technical quality of the system, competition, marketing requirements and customer change requests should all influence the decision of when to issue a new system release

- Too frequent or too infrequent release is not good

System Release Strategies

Factor	Description
Technical quality of the system	If serious system faults are reported which affect the way in which many customers use the system, it may be necessary to issue a fault repair release. However, minor system faults may be repaired by issuing patches (often distributed over the Internet) that can be applied to the current release of the system.
Platform changes	You may have to create a new release of a software application when a new version of the operating system platform is released.
Lehman's fifth law (see Chapter 21)	This suggests that the increment of functionality that is included in each release is approximately constant. Therefore, if there has been a system release with significant new functionality, then it may have to be followed by a repair release.
Competition	A new system release may be necessary because a competing product is available.
Marketing requirements	The marketing department of an organisation may have made a commitment for releases to be available at a particular date.
Customer change proposals	For customised systems, customers may have made and paid for a specific set of system change proposals and they expect a system release as soon as these have been implemented.

System Building

- The process of compiling and linking software components into an executable system
- Different systems are built from different combinations of components
- This process is now always supported by automated tools that are driven by ‘build scripts’ (e.g., make, Ant)

CASE Tools for Configuration Management

- CM processes are standardised and involve applying pre-defined procedures
- Large amounts of data must be managed
- CASE tool support for CM is therefore essential
- Mature CASE tools to support configuration management are available ranging from stand-alone tools to integrated CM workbenches

Configuration Management Workbenches

- Open workbenches
  - Tools for each stage in the CM process are integrated through organisational procedures and scripts. Gives flexibility in tool selection
  - E.g., Bugzilla for bug tracking; CVS for version management

- Integrated workbenches
  - Provide whole-process, integrated support for configuration management. More tightly integrated tools so easier to use. However, the cost is less flexibility in the tools used
  - E.g., ClearQuest

Key Points Revisit

- Configuration management is the management of system change to software products
- A formal document naming scheme should be established and documents should be managed in a database
- The configuration database should record information about changes and change requests
- A consistent scheme of version identification should be established using version numbers, attributes or change sets
- System releases include executable code, data, configuration files and documentation.
- System building involves assembling components into a system.
- CASE tools are available to support all CM activities