

Configuration Management

1

Configuration management

- ✧ Because software changes frequently, systems, can be thought of as a set of versions, each of which has to be maintained and managed.
- ✧ Versions implement proposals for change, corrections of faults, and adaptations for different hardware and operating systems.
- ✧ **Configuration management (CM)** is concerned with the policies, processes and tools for managing changing software systems.
- ✧ You need CM because it is easy to lose track of what changes and component versions have been incorporated into each system version.

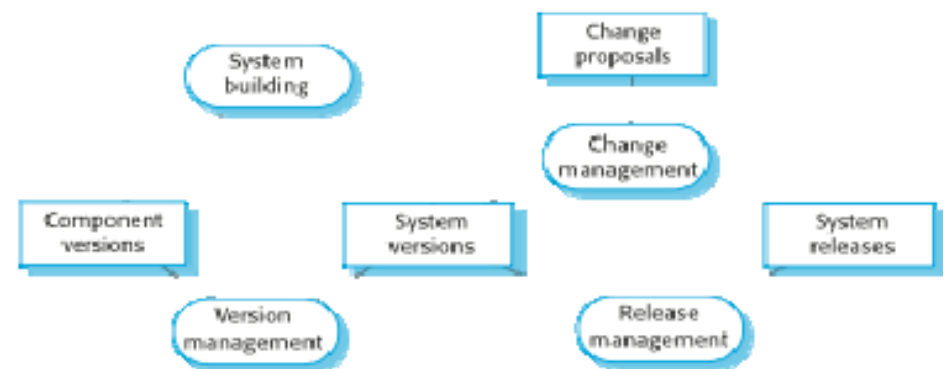
2

CM activities

- ✧ Change management
 - Keeping track of requests for changes, working out the costs and impact of changes, and deciding the changes
- ✧ Version management
 - Keeping track of the multiple versions of system components.
- ✧ System building
 - The process of assembling program components, data and libraries, then compiling these to create an executable system.
- ✧ Release management
 - Preparing software for external release and keeping track of the system versions that have been released.

3

CM activities



4

CM terminology

Term	Explanation
Configuration item or software configuration item (SCI) Configuration control	<u>Anything associated with a software project</u> (design, code, test data, document, etc.) <u>that has been placed under configuration control</u> . There are often different versions of a configuration item. Configuration items have a unique name. <u>The process of ensuring that versions of systems and components are recorded and maintained</u> so that changes are managed and all versions of components are identified and stored for the lifetime of the system.
Version	<u>An instance of a configuration item that differs</u> , in some way, <u>from other instances of that item</u> . <u>Versions always have a unique identifier</u> , which is often composed of the <u>configuration item name plus a version number</u> .
Baseline	<u>A baseline is a collection of component versions that make up a system</u> .
Codeline	<u>A codeline is a set of versions of a software component and other configuration items on which that component depends</u> .

5

CM terminology

Term	Explanation
Mainline	<u>A sequence of baselines representing different versions of a system</u> .
Release	<u>A version of a system that has been released to customers</u> (or other users in an organization) for use.
Workspace	A private work area where software can be modified without affecting other developers who may be using or modifying that software.
Branching	<u>The creation of a new codeline from a version in an existing codeline</u> . The new codeline and the existing codeline may then develop independently.
Merging	<u>The creation of a new version of a software component by merging separate versions in different codelines</u> . These codelines may have been created by a previous branch of one of the codelines involved.
System building	The creation of an executable system version by compiling and linking the appropriate versions of the components and libraries making up the system.

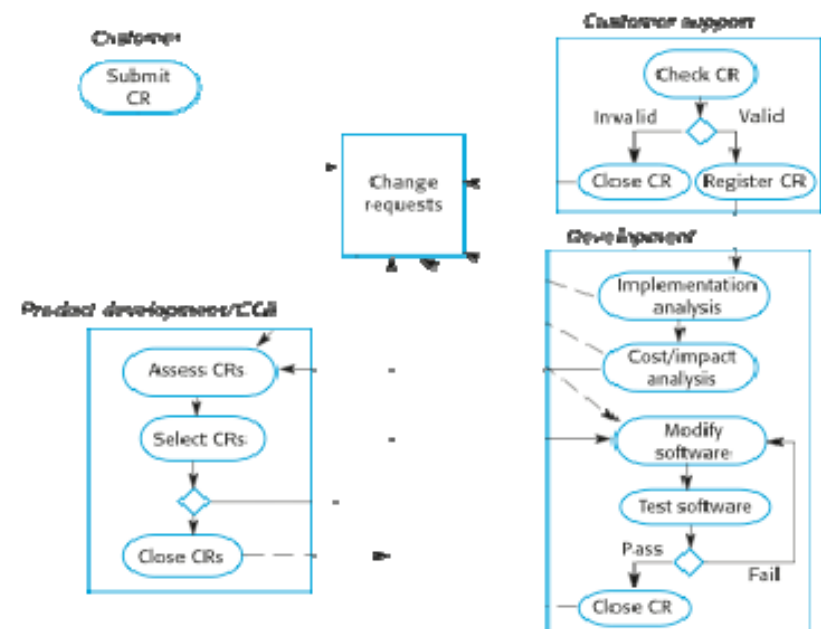
6

Change management

- Organizational needs and requirements change during the lifetime of a system, bugs have to be repaired and systems have to adapt to changes in their environment.
- Change management is intended to ensure that system evolution is a managed process and that priority is given to the most urgent and cost-effective changes.**
- The change management process is concerned with analyzing the costs and benefits of proposed changes, approving those changes that are worthwhile and tracking which components in the system have been changed.

7

The change management process



8

A partially completed change request form

Change Request Form

Project: SICSA/AppProcessing **Number:** 23/02
Change requester: I. Sommerville **Date:** 20/01/09
Requested change: The status of applicants (rejected, accepted, etc.) should be shown visually in the displayed list of applicants.
Change analyser: R. Looek **Analysis date:** 25/01/09
Components affected: ApplicantListDisplay, StatusUpdater
Associated components: StudentDatabase
Change assessment: Relatively simple to implement by changing the display colour according to status. A table must be added to relate status to colours. No changes to associated components are required.
Change priority: Medium
Change implementation:
Estimated effort: 2 hours
Date to SGA app. team: 28/01/09 **CCB decision date:** 30/01/09
Decision: Accept change. Change to be implemented in **Release 1.2**
Change implementer: **Date of change:**
Date submitted to QA: **QA decision:**
Date submitted to CM:
Comments:

9

Factors in change analysis

- ✧ 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

10

Change management and agile methods

- ✧ In some agile methods, customers are directly involved in change management.
- ✧ They propose a change to the requirements and work with the team to assess its impact and decide whether the change should take priority over the features planned for the next increment of the system.
- ✧ Changes to improve the software improvement are decided by the programmers working on the system.
- ✧ Refactoring, where the software is continually improved, is not seen as an overhead but as a necessary part of the development process.

11

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.

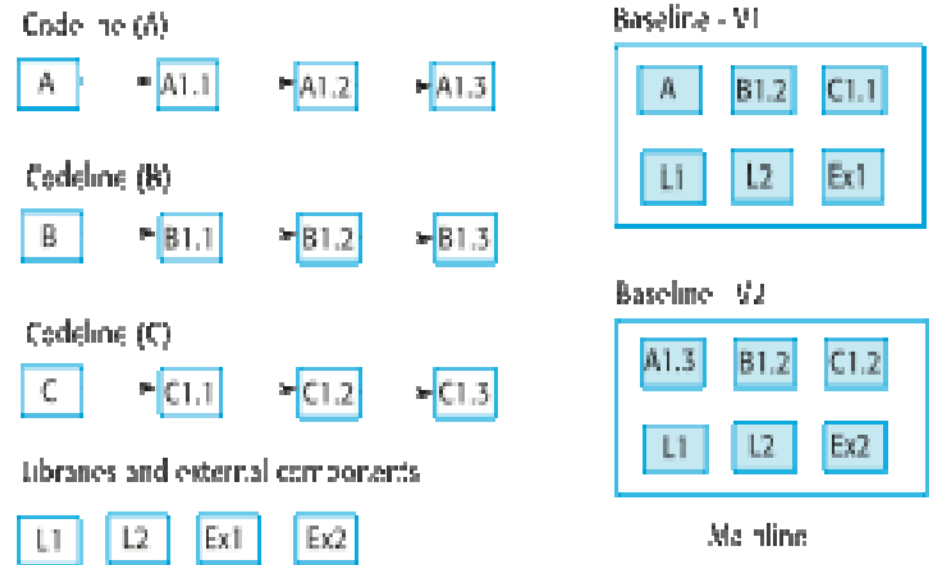
12

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.**

13

Codelines and baselines

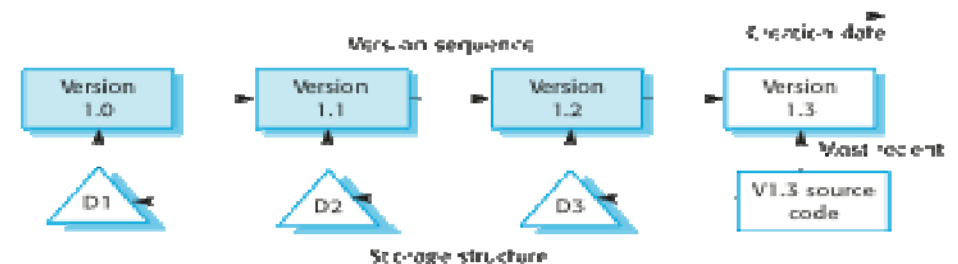


14

Version management systems

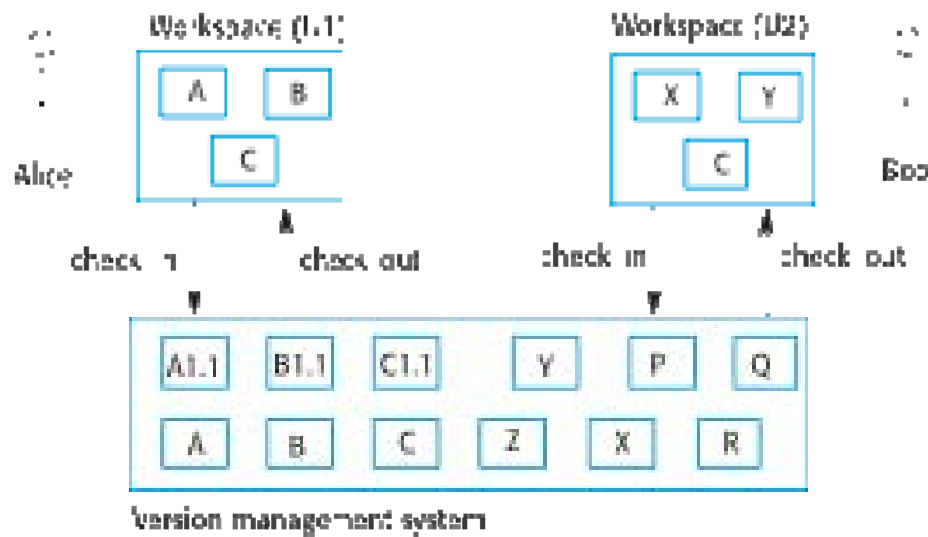
- ✧ **Version and release identification:** Managed versions are assigned identifiers
- ✧ **Storage management:** To reduce the storage space required by multiple versions of components that differ only slightly,
- ✧ **Change history recording:** changes made to the code of a system or component are recorded.
- ✧ **Independent development:** ensures that changes made to a component by different developers do not interfere.
- ✧ **Project support:** support the development of several projects, which share components.

Storage management using deltas



16

Check-in and check-out from a version repository



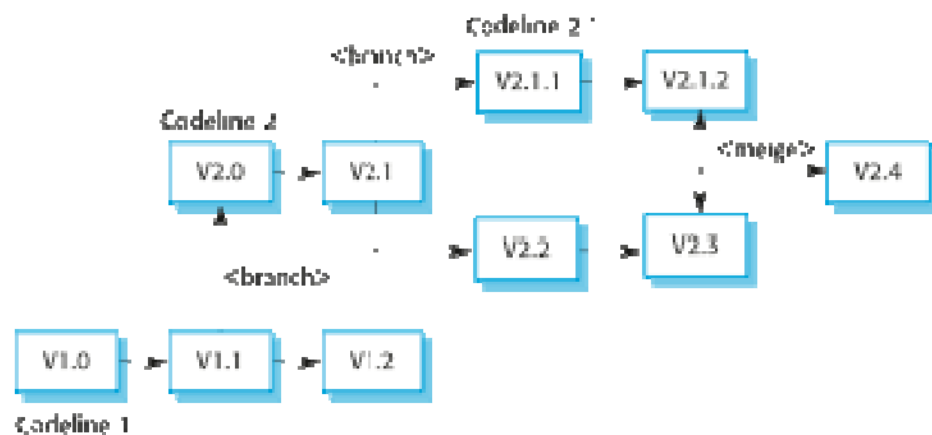
17

Codeline branches

- ✧ Rather than a linear sequence of versions that reflect changes to the component over time, there may be several independent sequences.
 - This is normal in system development, where different developers work independently on different versions of the source code and so change it in different ways.
- ✧ **At some stage, it may be necessary to merge codeline branches to create a new version of a component** that includes all changes that have been made.
 - If the changes made involve different parts of the code, the component versions may be merged automatically by combining the deltas that apply to the code.

18

Branching and merging



19

System building

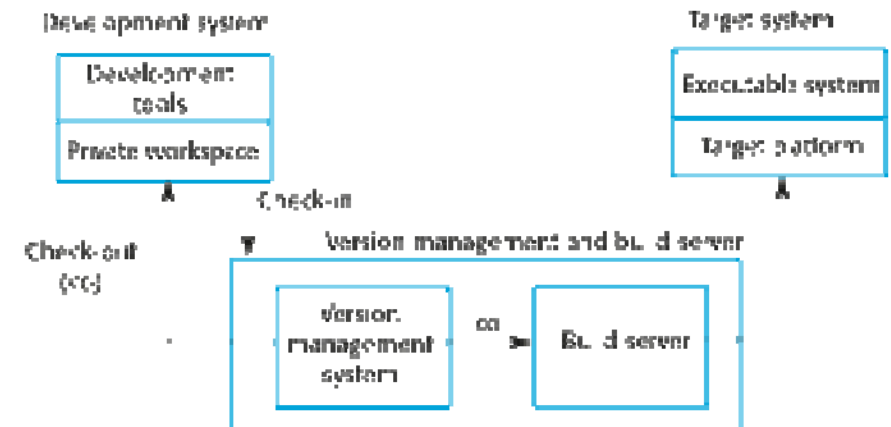
- ✧ System building is the process of creating a complete, executable system by compiling and linking the system components, external libraries, configuration files, etc.
- ✧ System building tools and version management tools must communicate as the build process involves checking out component versions from the repository managed by the version management system.
- ✧ The configuration description used to identify a baseline is also used by the system building tool.

20

Build platforms

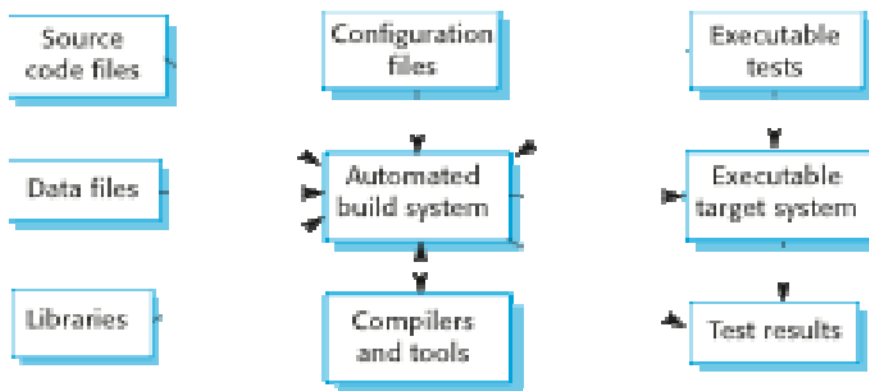
- ✧ The **development system**, which includes **development tools such as compilers, source code editors**, etc.
 - Developers check out code from the version management system into a private workspace before making changes to the system.
- ✧ The **build server**, which is used to **build definitive, executable versions of the system**.
 - Developers check-in code to the version management system before it is built. The system build may rely on external **libraries that are not included in the version management system**.
- ✧ The **target environment**, which is the **platform on which the system executes**.

Development, build, and target platforms



22

System building



23

Release management

- ✧ A **system release** is a **version of a software system that is distributed to customers**.
- ✧ For mass market software, it is usually possible to identify **two types of release: major releases which deliver significant new functionality**, and **minor releases, which repair bugs and fix customer problems** that have been reported.
- ✧ For custom software or software product lines, releases of the system may have to be produced for each customer and individual customers may be running several different releases of the system at the same time.

24

Release tracking

- ✧ In the event of a problem, it may be necessary to reproduce exactly the software that has been delivered to a particular customer.
- ✧ When a system release is produced, it must be documented to ensure that it can be re-created exactly in the future.
- ✧ This is particularly important for customized, long-lifetime embedded systems, such as those that control complex machines.
 - Customers may use a single release of these systems for many years and may require specific changes to a particular software system long after its original release date.

25

Release reproduction

- ✧ To document a release, you have to record the specific versions of the source code components that were used to create the executable code.
- ✧ You must keep copies of the source code files, corresponding executables and all data and configuration files.
- ✧ You should also record the versions of the operating system, libraries, compilers and other tools used to build the software.

26

Release components

- ✧ As well as the the executable code of the system, a release may also include:
 - configuration files defining how the release should be configured for particular installations;
 - data files, such as files of error messages, that are needed for successful system operation;
 - an installation program that is used to help install the system on target hardware;
 - electronic and paper documentation describing the system;
 - packaging and associated publicity that have been designed for that release.

27

Factors influencing system release planning

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. Minor system faults may be repaired by issuing patches (usually 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	This 'law' suggests that if you add a lot of new functionality to a system; you will also introduce bugs that will limit the amount of functionality that may be included in the next release. Therefore, a system release with significant new functionality may have to be followed by a release that focuses on repairing problems and improving performance.

28

Factors influencing system release planning

Factor	Description
Competition	For mass-market software, a new system release may be necessary because a competing product has introduced new features and market share may be lost if these are not provided to existing customers.
Marketing requirements	The marketing department of an organization may have made a commitment for releases to be available at a particular date.
Customer change proposals	For custom 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.