Section 7.3
Configuration Management

1. Overview
2. Change management
3. Version management
4. System building
5. Release management

7.3.1 Overview

- What is configuration management?
  - deals with policies, processes, and tools for:
    - managing changes
    - tracking versions of components
  - essential for:
    - multi-developer projects
    - product support

Overview (cont.)

- Changes to software are necessary to:
  - fix bugs
  - add features
  - adapt to new platforms
- Each change creates a new version of the system
- Large amounts of information must be tracked

Overview (cont.)

- Policies and processes define how to:
  - record and process changes
  - decide which system components to change
  - manage different versions
  - distribute changes to client
- Tools
  - used to keep track of change proposals
  - store versions of system components
  - build system from stored components
  - track releases of system versions
- Configuration management is part of quality assurance

Overview (cont.)

- Configuration management activities
  - change management
    - tracking change requests
    - determining cost and impact of changes
    - deciding whether changes should be implemented
  - version management
    - tracking multiple versions of system components
    - making sure components by different developers don’t conflict

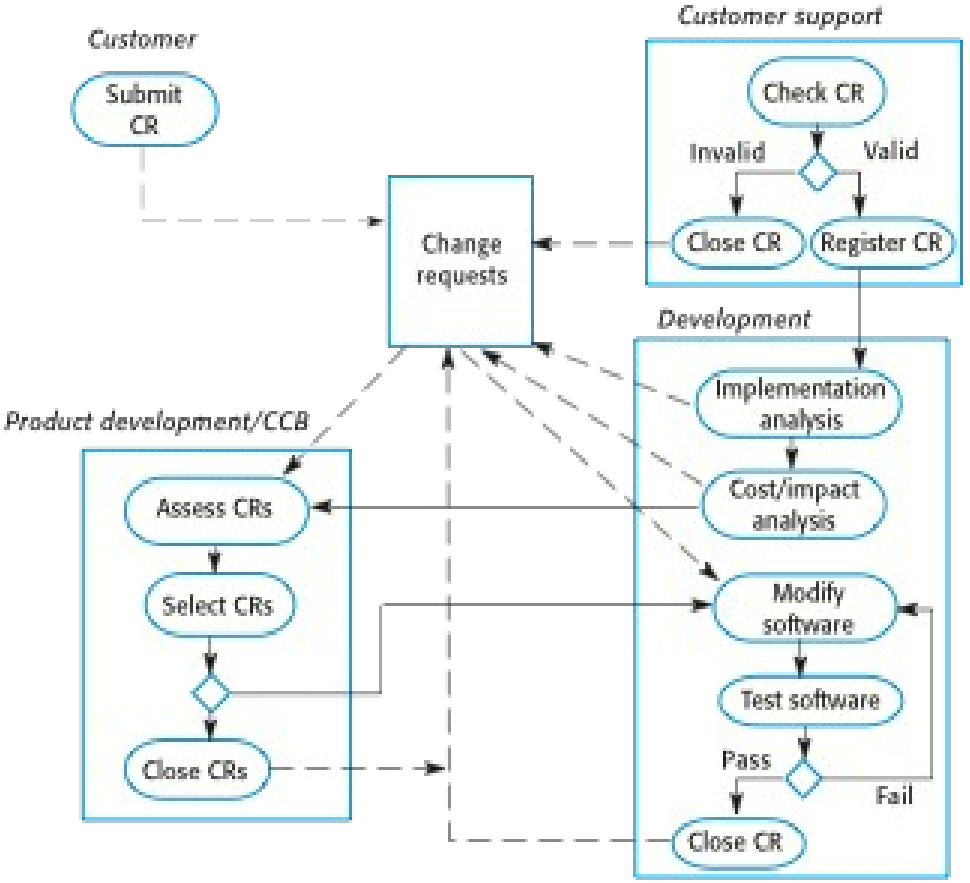
Overview (cont.)

- Configuration management activities (cont.)
  - system building
    - assembling components, data, libraries
    - compiling and linking into executable system
  - release management
    - preparing software for external release
    - tracking released versions of system

7.3.2 Change Management

- System evolution needs to be managed
- Change management process ensures:
  - analysis of costs and benefits of proposed changes
  - prioritization of proposed changes
  - approval of important changes
  - tracking of system components to change
- Process comes into effect after system is deployed

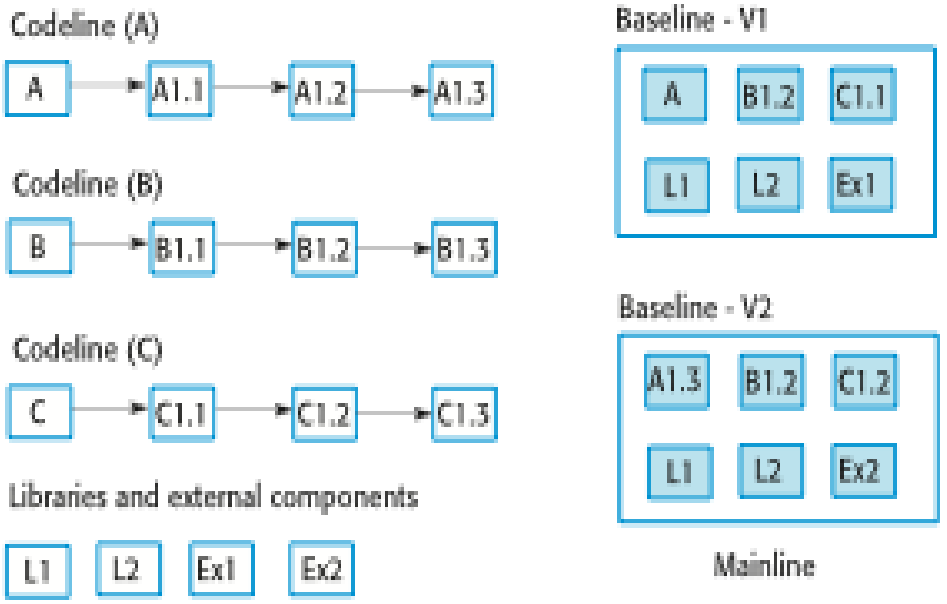
Change Management (cont.)



7.3.3 Version Management

- What is version management?
  - process of:
    - tracking different versions
    - managing codelines and baselines
  - ensures that changes by different developers don't conflict
- Codeline
  - sequence of source code versions
  - each derived from preceding one
- Baseline
  - defines a specific system (component and library versions used)

Version Management (cont.)



Version Management (cont.)

- Baseline used to define a system release
  - each client may have its own system version
  - may need to recreate a baseline long after release to client
    - when client site reports a bug

- Tools
  - called:
    - version management systems
    - version control systems
    - source code control systems
  - identify, store, control access to different component versions
  - many open source systems available
    - example: GitHub

Version Management (cont.)

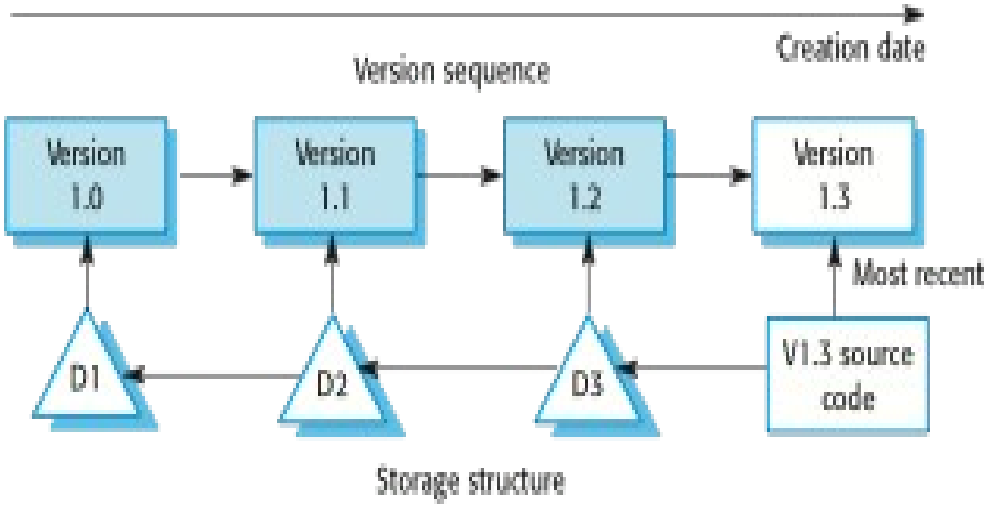
- Common features of version management systems
  - version and release identification
    - versions assigned identifiers when submitted
  - storage management
    - complete copy of each version usually not kept
    - only deltas (list of differences) between versions are stored
  - change history recording
    - components tagged with keywords describing changes

Version Management (cont.)

- Common features of version management systems (cont.)
  - independent development
    - component check-out and check-in functionality provided
  - project support
    - several projects with shared components can be managed together

Version Management (cont.)

- Storage management
  - major function of version management systems
  - reduces disk space required to maintain all versions
  - only most recent version is stored
  - earlier versions recreated from current version and deltas

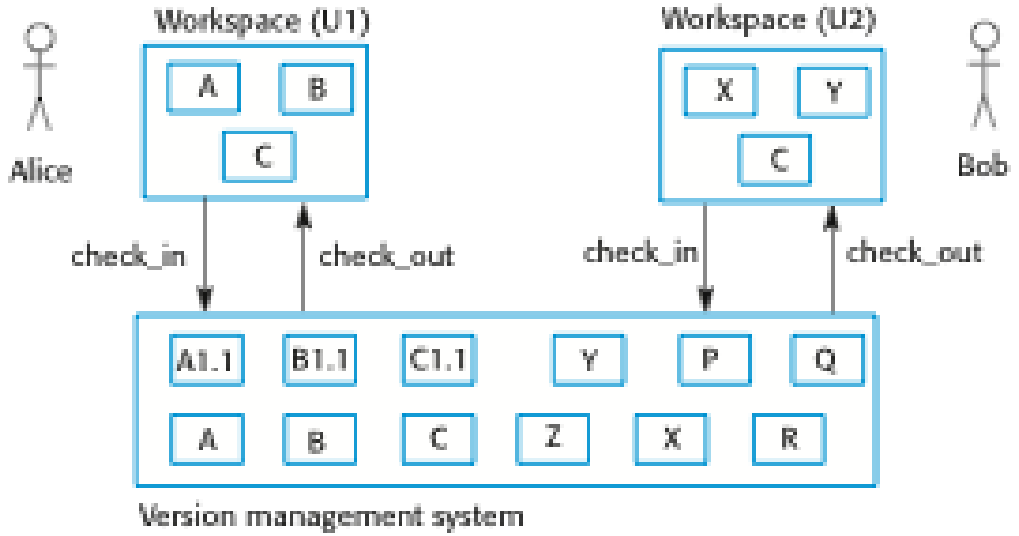


Version Management (cont.)

- Consists of:
  - public repository
    - contains all components
  - private workspace
    - developers check-out components into their workspace
    - changes are made in workspace
    - when finished, developers check-in components into repository
  - may result in multiple codeline branches
  - branches must be merged
    - automatically if different parts of the code were changed
    - manually if changes overlap

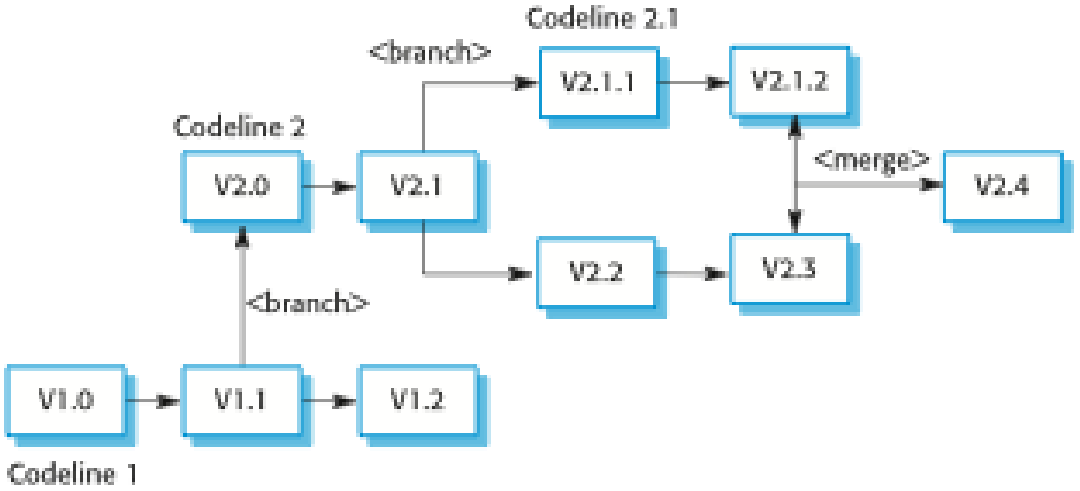
Version Management (cont.)

- Independent development:



Version Management (cont.)

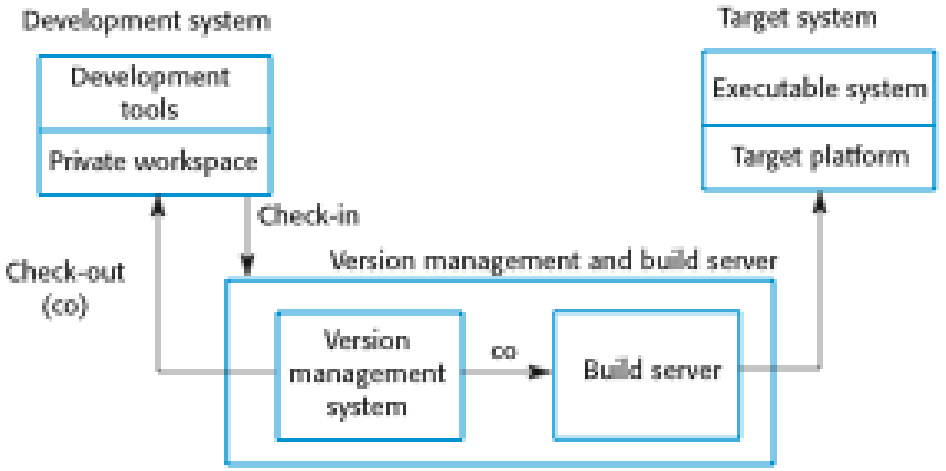
- Independent development (cont.):

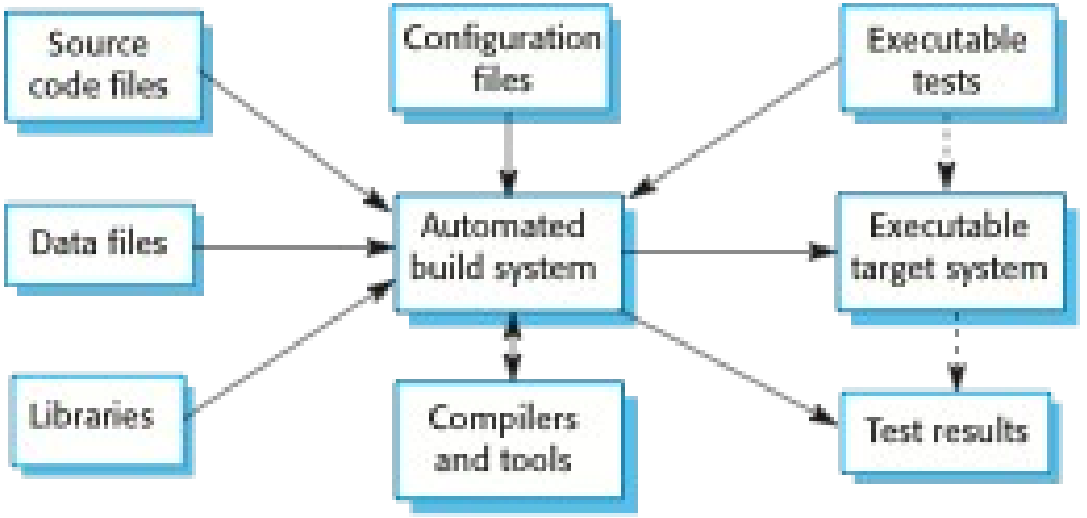


7.3.4 System Building

- What is system building?
  - process of creating executable system by compiling and linking all components
    - internal, external, libraries
- Build server requires:
  - configuration files to indicate baseline
  - version management system to check-out source code
  - compilers and tools for target environment
- Sanity check should be conducted on each build

System Building (cont.)





7.3.5 Release Management

- What is a system release?
  - version of system that is distributed to clients
    - major release: significant new functionality
    - minor release: bug fixes

- What does a system release include?
  - executable code for the system
  - configuration files
  - data files
  - installation program
  - documentation

Release Management (cont.)

- Managing releases is very complex
  - customized releases may be deployed to each client
  - every client may be running different releases at different sites

- Client support nightmare!
  - many different releases must be supported at once
  - client releases must be tracked
    - bugs must be reproduced by support team on exact same release

Software Management Recap

- What we learned:
  - understand basic project management activities
  - understand the different software development processes
    - waterfall, V-model, spiral, agile
  - understand the need for software management processes and tools, including:
    - configuration management
    - version management
    - system building
    - release management