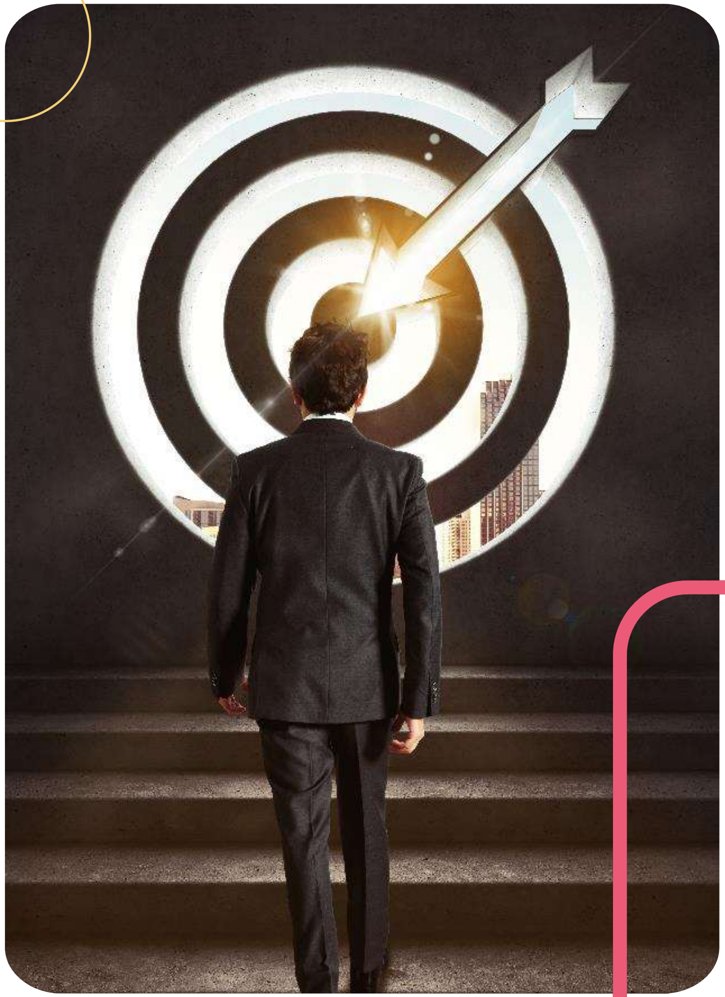
Diploma in

Computer Science

Lesson 7: Scaling Up



Objectives

- Explain how code is handled on a large scale
- Explore version control
- Outline the benefits of version control
- Discuss version control systems used in industry

Version control systems in software development

```
(this)
$($this.attr('data-targ
lace(/.*(?=#[^\s]+$)/, ''))
et.hasClass('carousel')) retu
s = $.extend({}, $target.data(
ndex = $this.attr('data-slide-
ndex) options.interval = false

($target, options)
```

Life before version control...

- 1000 lines of code used to be a lot
- Computers were hard to program so programs weren’t bigger than a few 100 lines

```
this)
$($this.attr('data-tar
lace(/.*(?=#[^\s]+$)/, ''))
et.hasClass('carousel')) retu
s = $.extend({}, $target.data(
ndex = $this.attr('data-slide-
ndex) options.interval = false

(target, options)
```

Life with version control...

- Software has several versions of source code
- One version released at a time while developers work
- Developers use version control for backtracking

Quick history of version control systems

- Book editions and specification revisions were early printed forms of version control
- Most elaborate systems used in software development today

Quick history of version control systems

- First popular version control system was SCCS
- Followed by Revision Control System
- Subversion and GIT are popular today

Version control, source control, source code management systems, and revision control systems are all the same thing.

Systems that keep track of files

- Record changes made to files throughout the development process
- Example: cloud services such as Google Drive

Systems that keep track of files

- Version control: Stand-alone applications
- Revision control: Built into software such as Microsoft Office, Windows, Mac
- Community collaboration with version control example: Wikipedia
- Content management system example: WordPress

Version control in software engineering…

- Class of systems that manage changes to a large collection of data or information
- Component of software configuration management

Identifying changes

- Changes carry some sort of identification: a letter, number, or both
- Each revision has a timestamp: date and time
- Provides functionality such as comparison, revision, reverts, and merges

Why do we need version control?

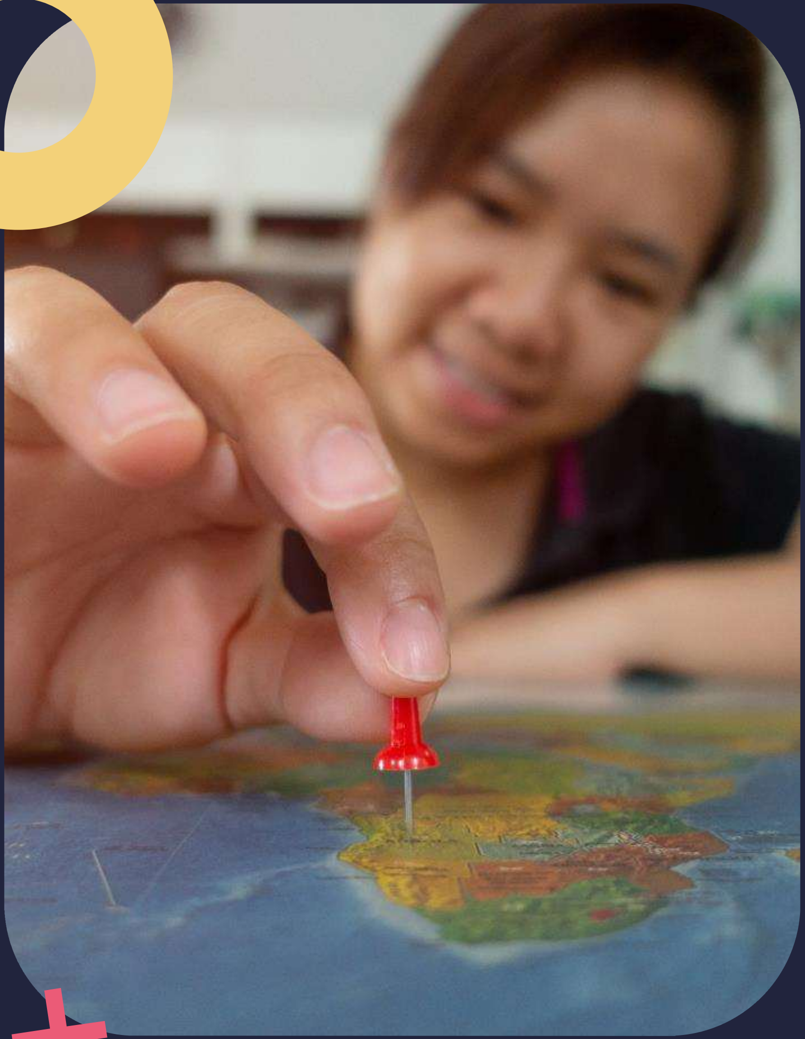
- Can easily revert to the last copy
- Provides backup when there are lots of reverts
- Allows software teams to work efficiently and speedily as team scales
- Each team member has the file history

With an efficient control system you can...

- Complete change history of files
- Work concurrently with other developers
- Branch and merge on large projects
- Track development

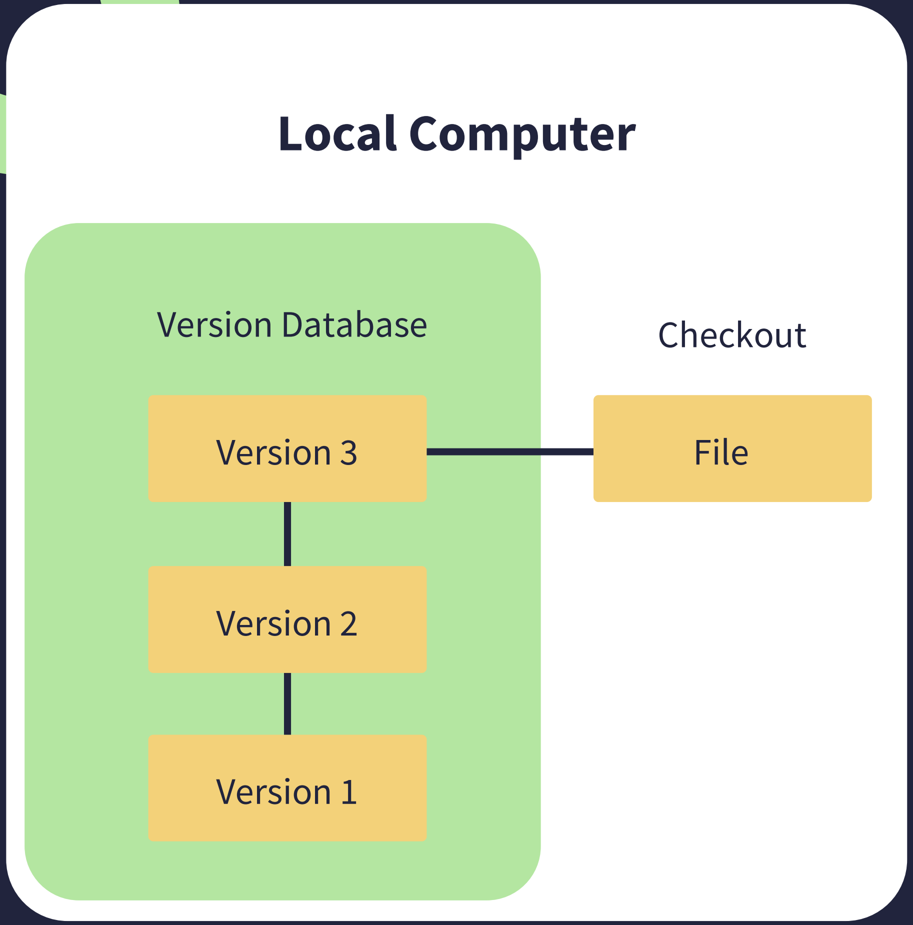
DID YOU KNOW?

GitHub is the largest host of source code in the world.



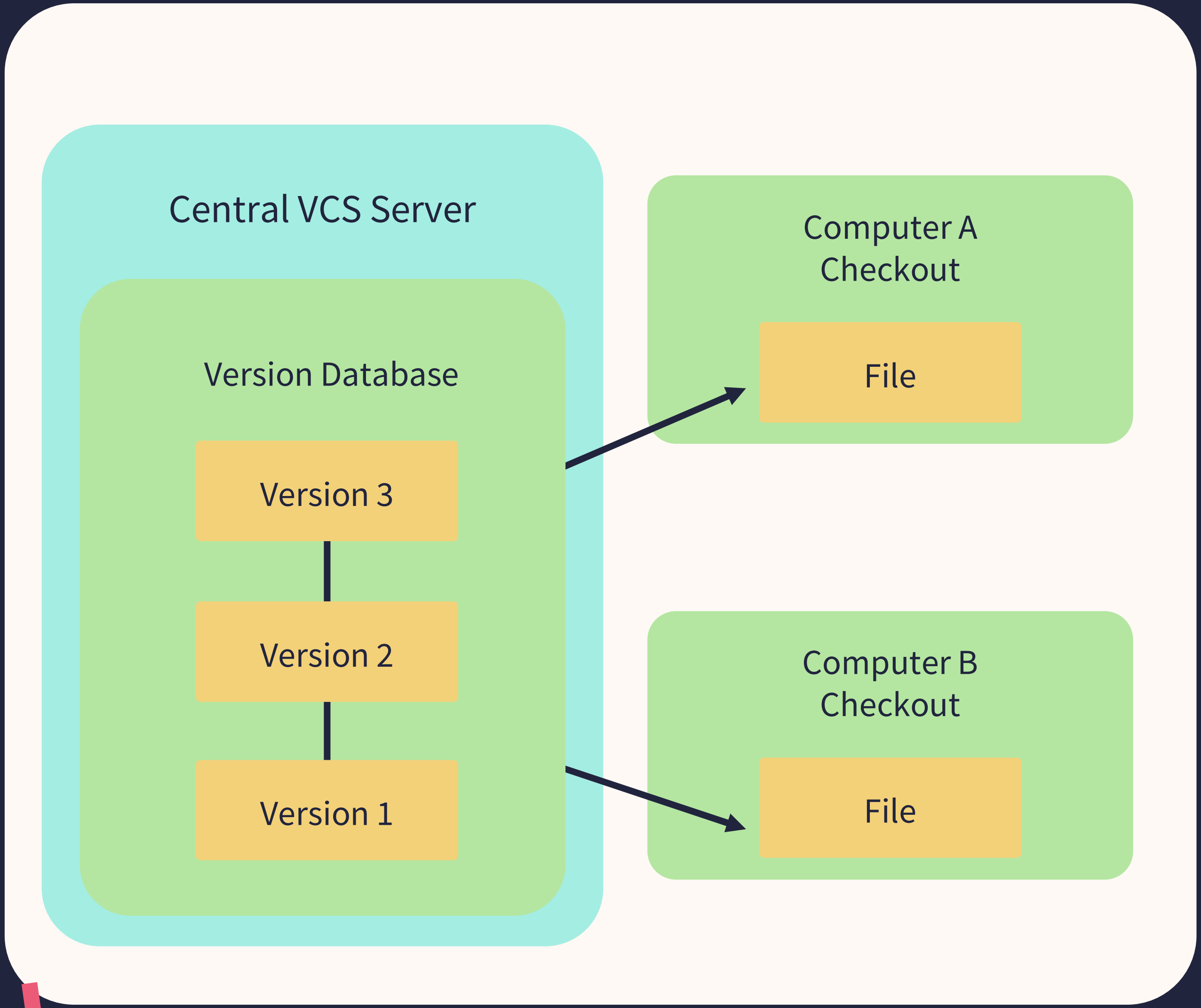
Types of version control systems

- Localised
- Centralised
- Distributed



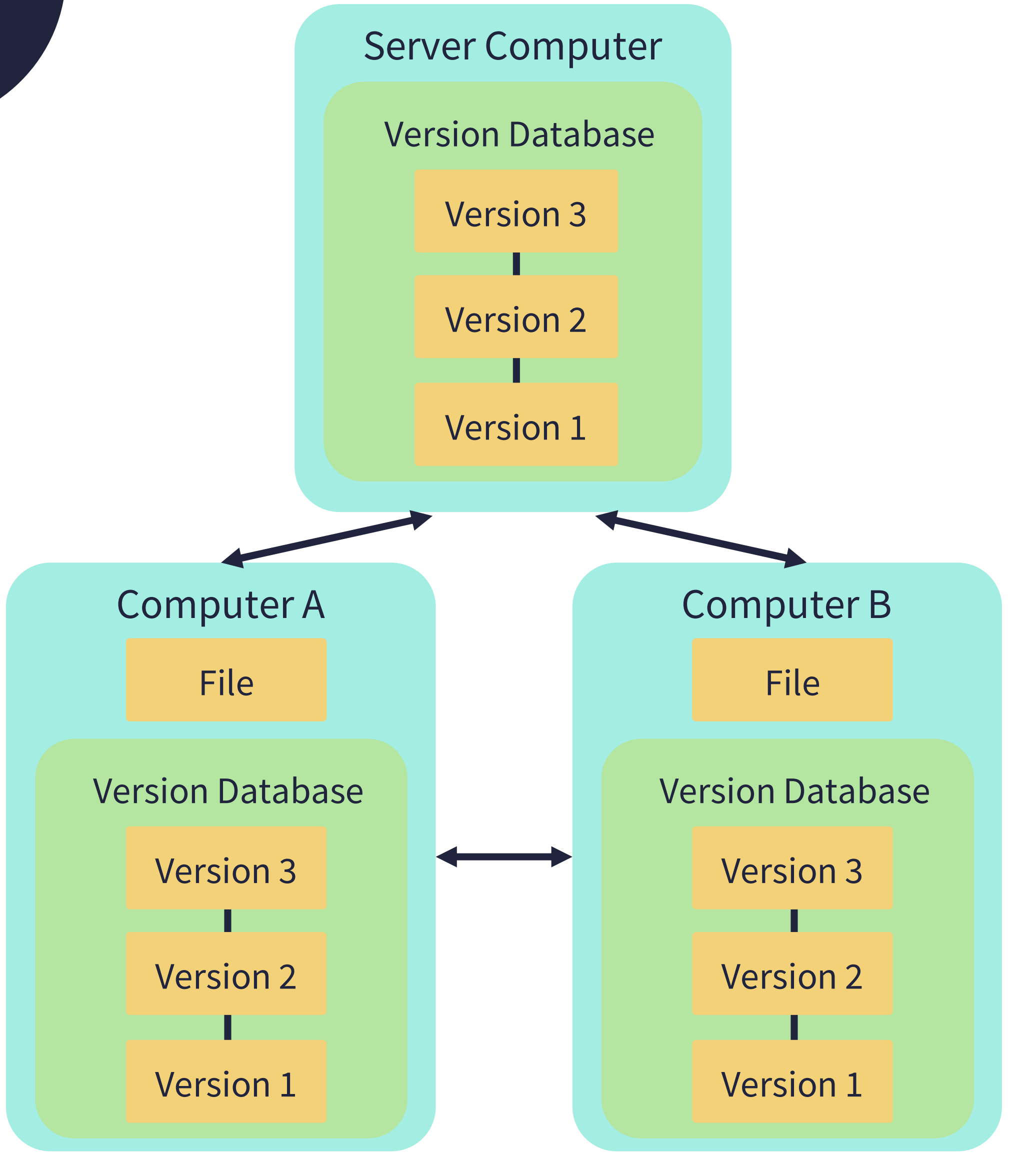
Localised version control systems

- Simplest of all version control systems and most common
- Data can be easily lost as you accidentally overwrite files



Centralised version control systems

- Attempts to remove risks of one machine and human element
- Developers can access files at a central location
- Provides degree of control
- Single point of failure can be a problem



Distributed version control systems

- Users have a local copy or ‘snapshot’ of the entire repository
- Allow for automatic management of branching and merging
- Developer does not need to worry about rudimentary operations
- Developer can work offline
- Each machine serves as a backup

Common terms in version control systems

Centralised source management models

- File locking
- Revision merging

File locking

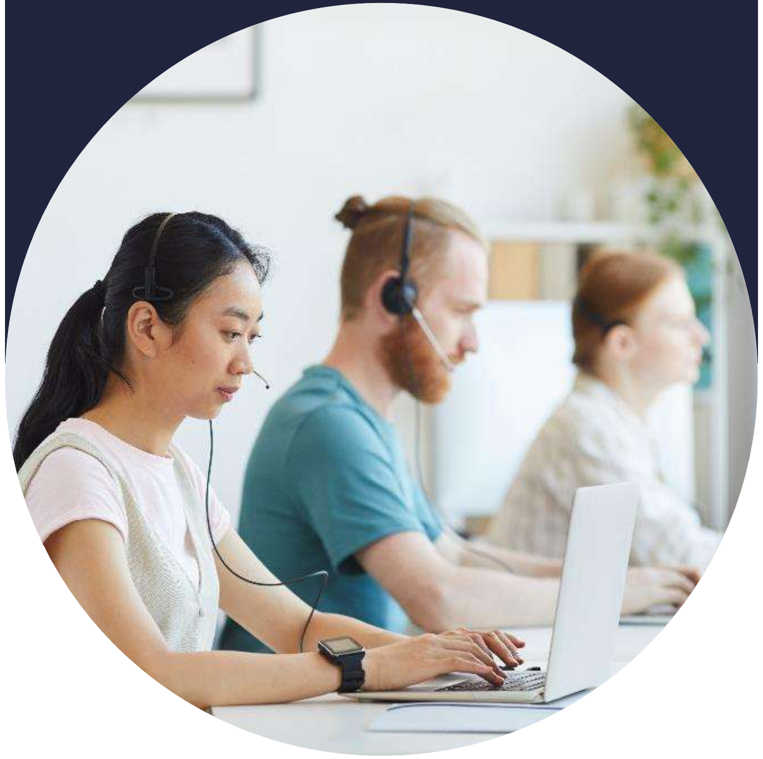
- Denies subsequent access requests
- Allows access to one request at a time
- When one developer ‘checks out’ a file others can read it

Advantages

- Can provide protection against difficult merge conflicts for significant changes
- Files have exclusive locks for the entire duration

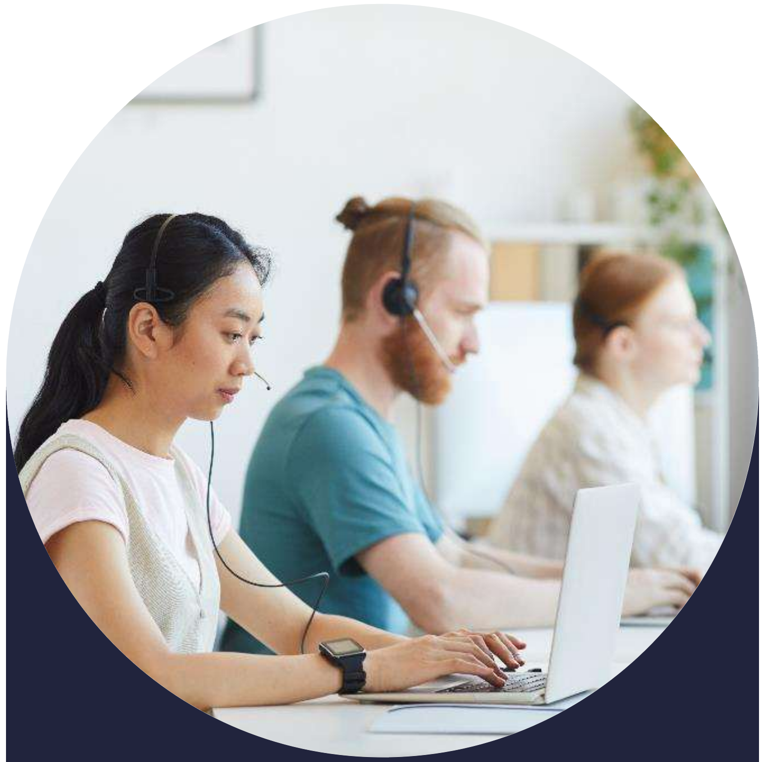
Disadvantages

- Other developers may be tempted to bypass the revision control software and change the files on their machines
- Might not be easy to see who has a file checked out



Revision merging

- Multiple developers edit the same file at the same time
- First developer to ‘check in’ changes to the central repository gets access
- Other users merge further changes into the central repository while preserving earlier changes

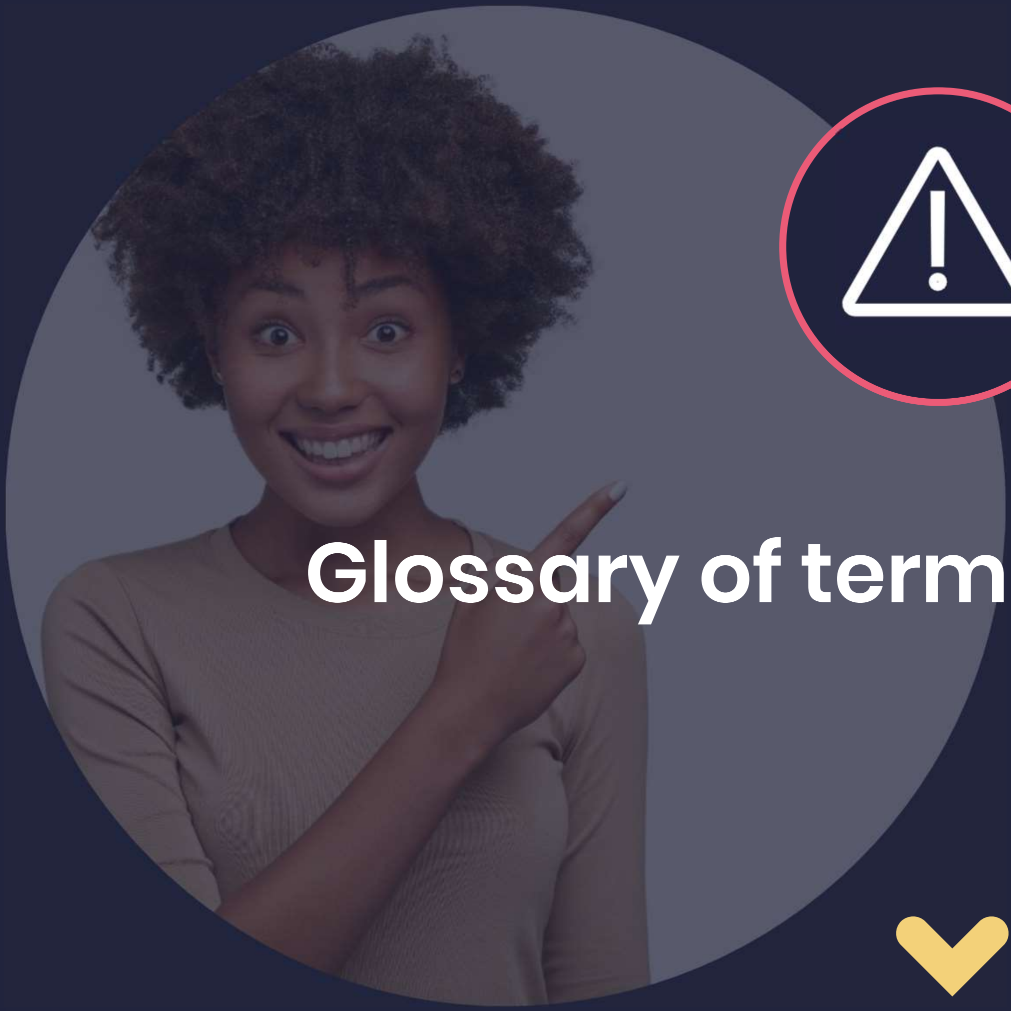


Revision merging

- Merging two files often requires the files to be simple text files
- Second developer checking in the code will need to take care with the merge
- Unless a specific merge plugin is available for the file types, it’s not simple
- Reserved edits can provide a different way of explicitly locking a file for exclusive write access

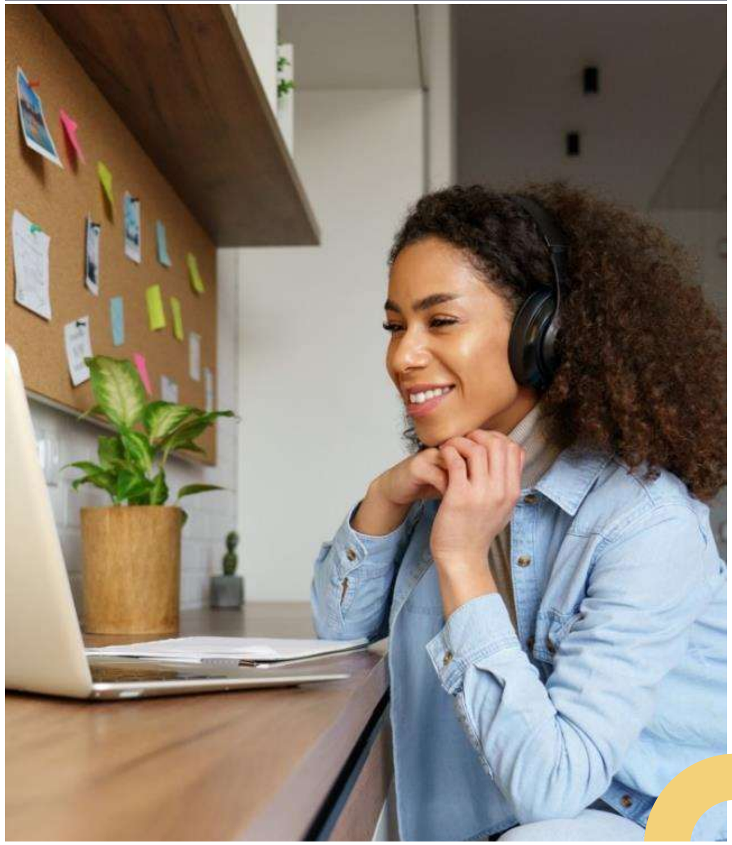
Baselines, labels, and tags

- Refer to the process of identifying a snapshot or the record of the snapshot
- Only one of the terms is used in documentation
- When both are used, *label* and *tag* usually refer to the mechanism while *baseline* indicates the increased significance



Glossary of terms used in VCS

Glossary of terms



- Baseline: An approved revision of a document
- Atomic operations: System is left in a consistent state even if the operation is interrupted
- Branch: Two copies of files develop independently of each other
- Change: Specific modification to a document under version control
- Change list: Set of changes made in a single commit

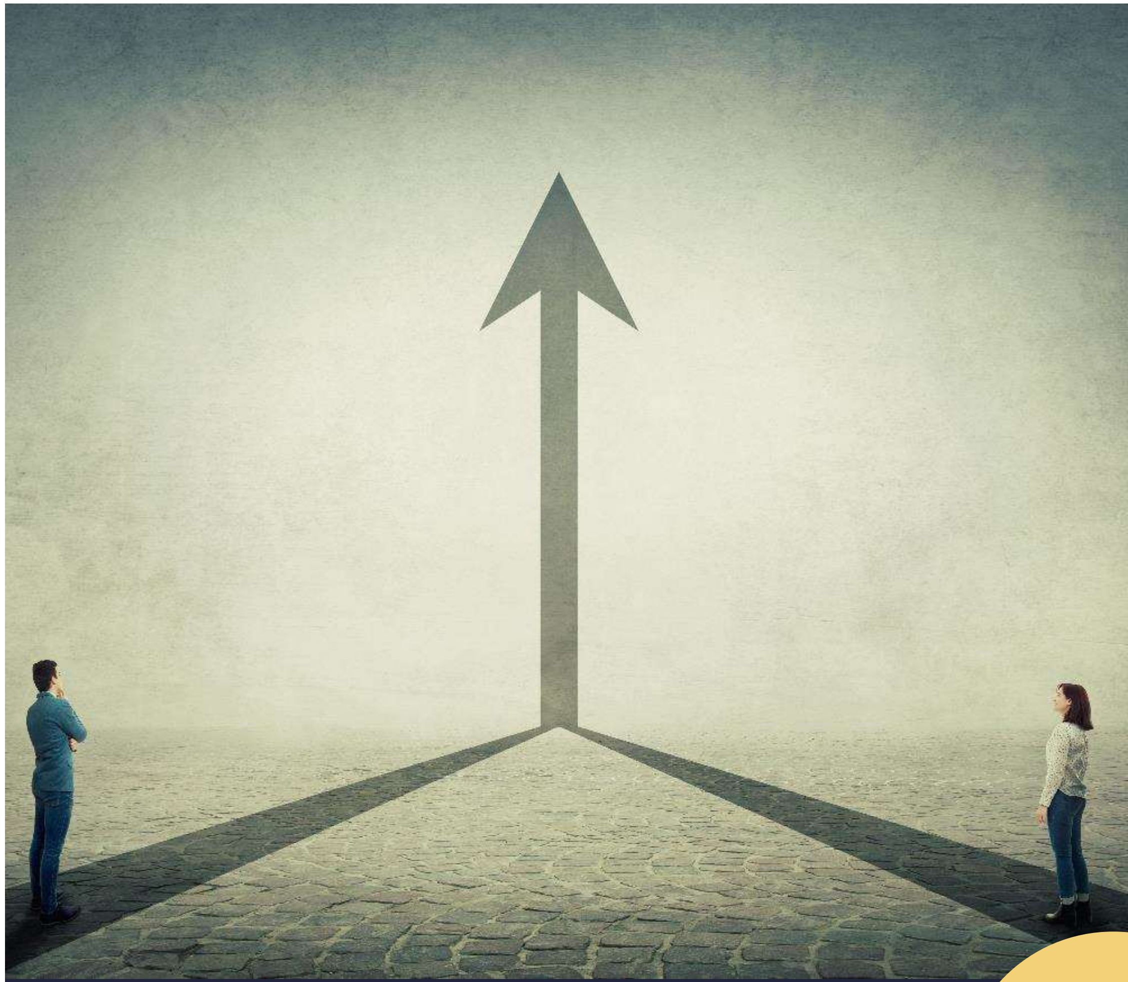
Glossary of terms

- Check out: Create a local working copy from the repository
- Clone: Creating a repository containing the revisions from another repository
- Commit (noun): Modification that is applied to the repository
- Commit (verb): Write or merge the changes in the working copy back to the repository
- Conflict: Different parties make changes to the same document that cannot be reconciled

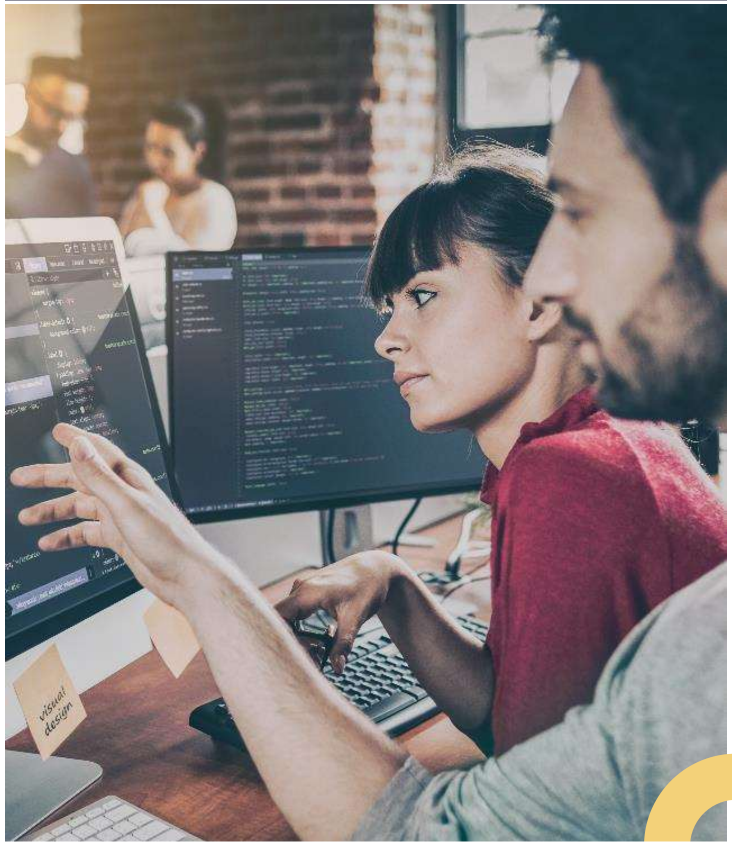
Glossary of terms

Merge: Two sets of changes are applied to a file

- A user updates or syncs their working copy with changes made, and checked into the repository, by other users.
- A user checks in files that have been updated since the last check out, resulting in an automatic merge
- A branch is created, the code in the files is independently edited, and the updated branch is later put together
- A set of files is branched, a pre-existing problem is fixed in one branch then merged into the other branch



Glossary of terms



- Promote: Copying file content from a less controlled location into a more controlled location
- Pull-push: Copy revisions from one repository into another - pull→receiving, push→source
- Pull request: Asking others to merge their ‘pushed’ changes
- Repository: Where current and historical data of files are stored
- Resolve: User intervention in a conflict