Basic Web Development

#2 Basic Version Control

Lect Dr.Passakorn Phannachitta

passakorn.p@cmu.ac.th

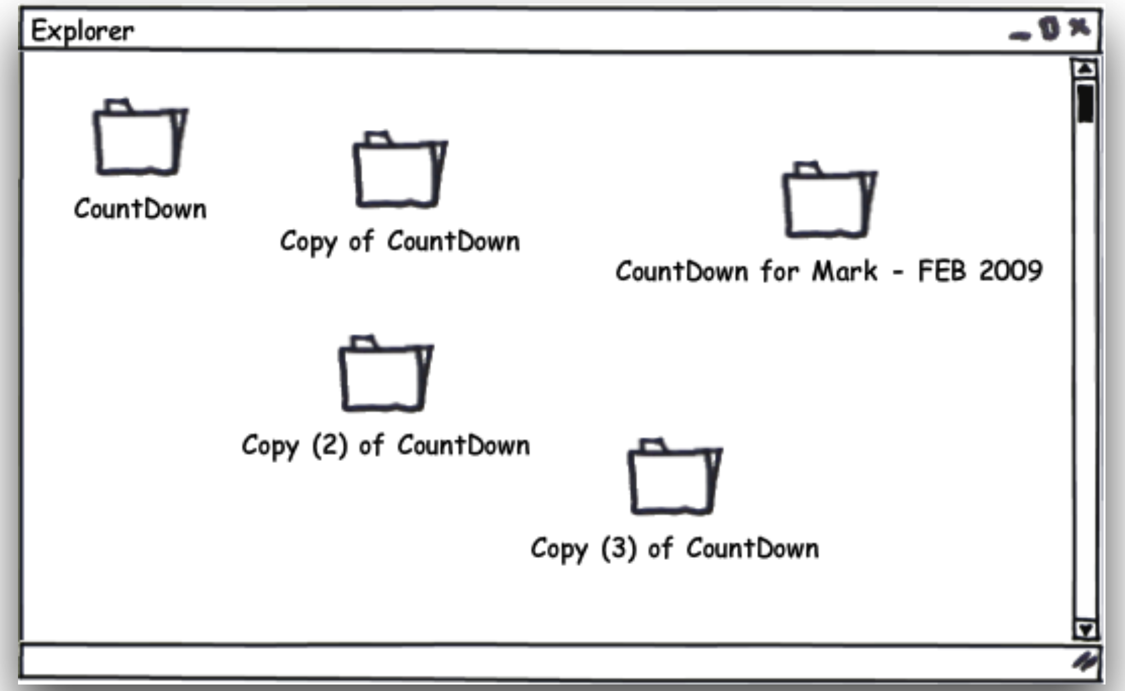
College of Arts, Media and Technology

Chiang Mai University, Chiangmai, Thailand

CAMT
College of Arts, Media and Technology
Chiang Mai University

Before Version Control

- File renaming
  - e.g., Draft1-Dec20.doc, Draft2-Dec28.doc
- New directory
  - /Version1, /Version1-1
- Zip file after a version
  - Jan10-Release1.0.zip

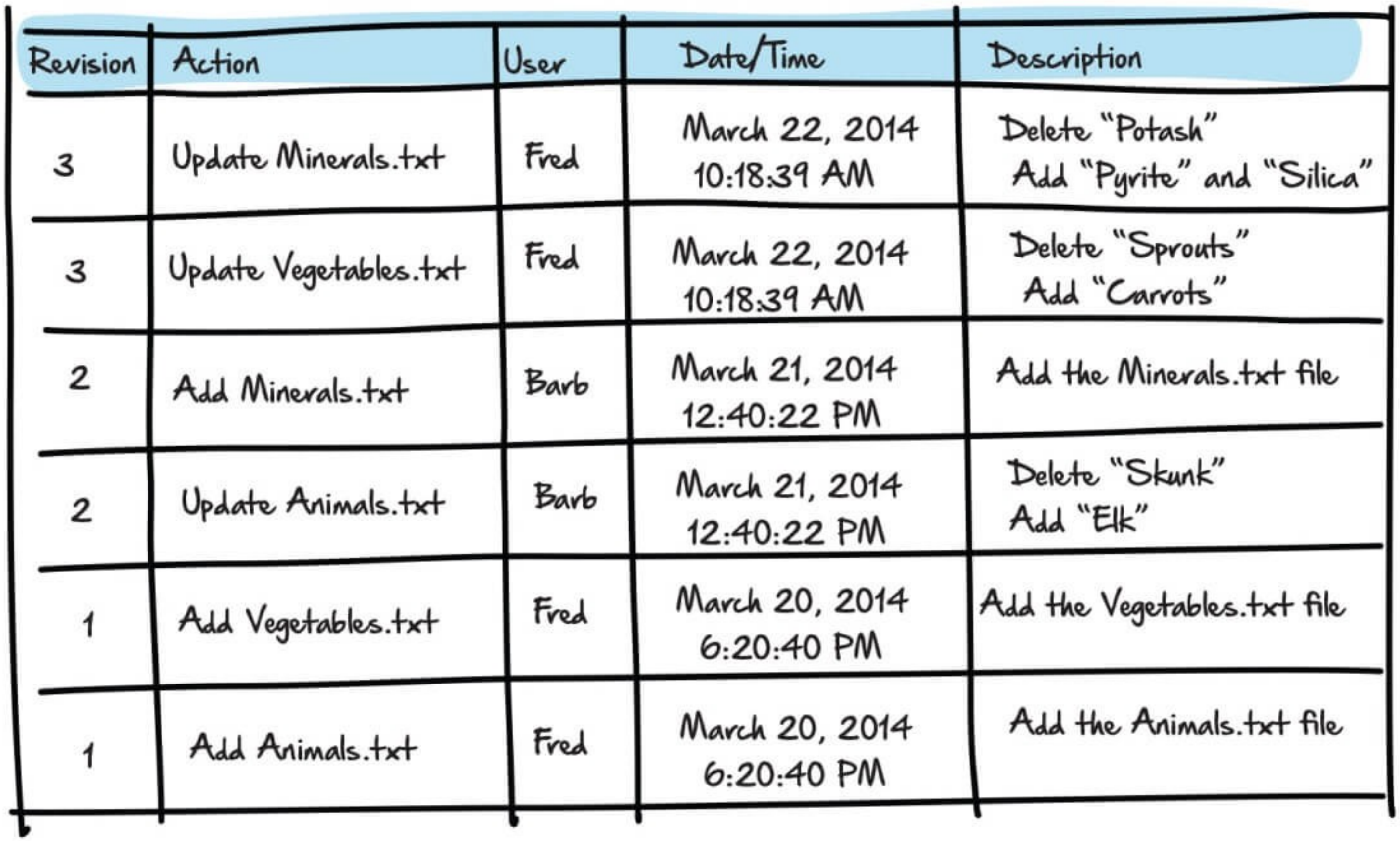


Difficult to track and review the changes when looking back what have been done

Basic Version Control

- A system or toolset that keeps tracking history of all the changes made in software projects.
  - Provide monitor access to the files
  - Every change made to the source is tracked
    - What is changed?
    - Who made it?
    - Why they made it?
    - References to the problem fixed

Basic Version Control



Revision	Action	User	Date/Time	Description
3	Update Minerals.txt	Fred	March 22, 2014 10:18:39 AM	Delete "Potash" Add "Pyrite" and "Silica"
3	Update Vegetables.txt	Fred	March 22, 2014 10:18:39 AM	Delete "Sprouts" Add "Carrots"
2	Add Minerals.txt	Barb	March 21, 2014 12:40:22 PM	Add the Minerals.txt file
2	Update Animals.txt	Barb	March 21, 2014 12:40:22 PM	Delete "Skunk" Add "Elk"
1	Add Vegetables.txt	Fred	March 20, 2014 6:20:40 PM	Add the Vegetables.txt file
1	Add Animals.txt	Fred	March 20, 2014 6:20:40 PM	Add the Animals.txt file

Ref: https://www.red-gate.com/simple-talk/sql/sql-development/core-database-source-control-concepts/

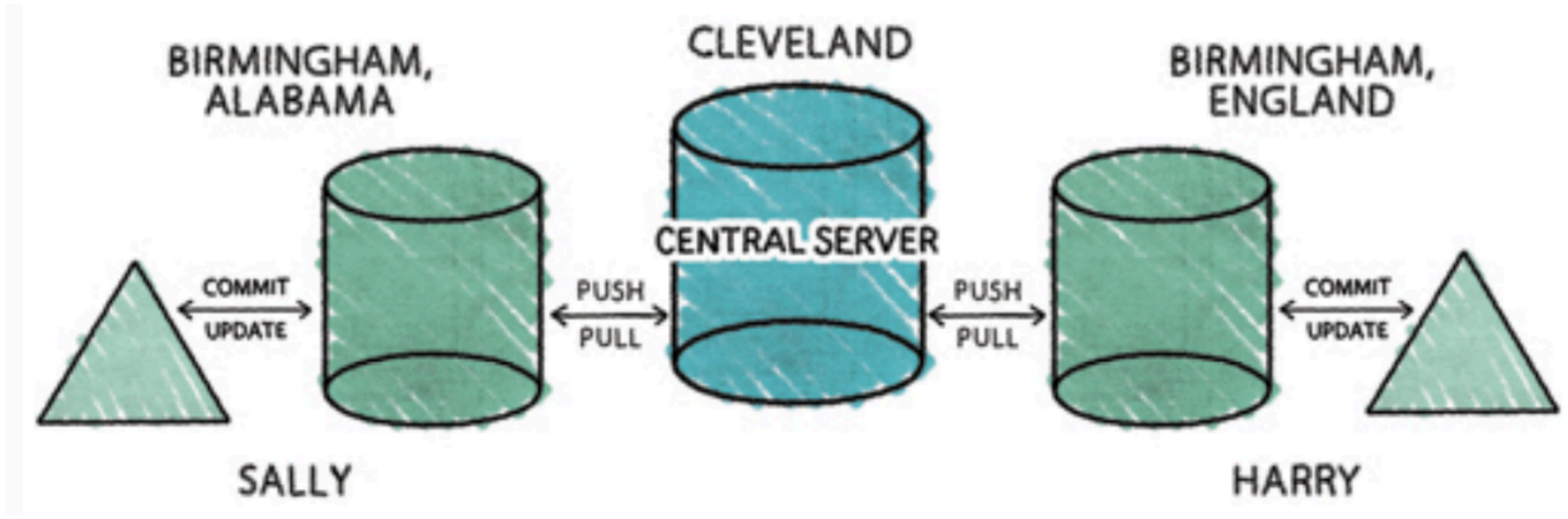
Basic Features

- A place to store your source code.
- A historical record of what have been done over time.
- A way for developers to work on separate tasks in parallel, merging their efforts later.
- A way for developers to work together without getting in each others' way.

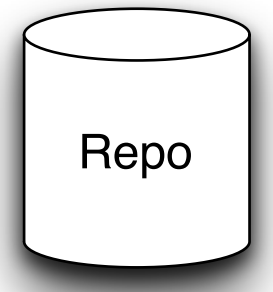
Why We should Use a VCS — Scenarios

- We made change to several code-based files, and realized that there were mistakes requiring us to reverse them back.
- Some codes are lost due to mistaken removal.
- We have to seriously maintain multiple versions of a product, and there are overlapping contents between versions.
- We need to share our code or let other people work on the codes.
- We want to experiment with some fancy features, but we do not want them interfered the currently working codes.

Operations


BIRMINGHAM, ALABAMA
CLEVELAND
BIRMINGHAM, ENGLAND
COMMIT
UPDATE
PUSH
PULL
CENTRAL SERVER
PUSH
PULL
COMMIT
UPDATE
SALLY
HARRY

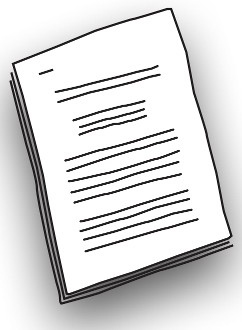
Operations

- Create 

  - Create a new, empty repository.
  - Repository has 3 dimensions
    - Directories — Tree structure which is the same as that of any file system
    - File Storage — Can be viewed as a simple network file system
    - Time — Every single version of the file being store in the repository

Operations

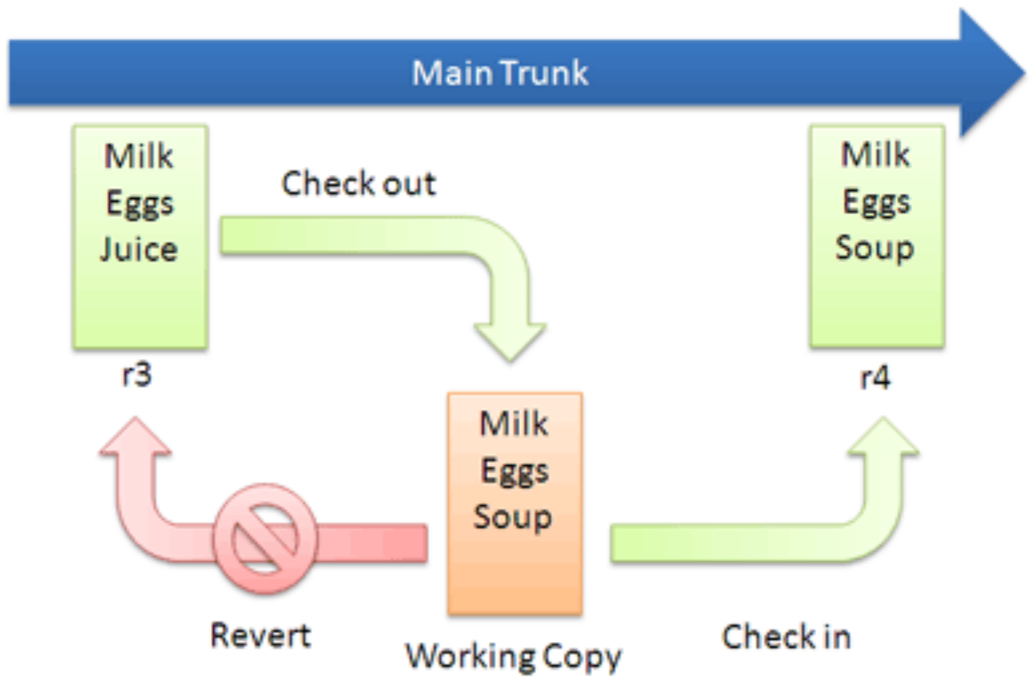
- **Clone**



  - Create a working copy.
  - Generally the repository is shared by the whole team.
  - But people do not modify it directly.
  - Rather, each individual developer works by using a working copy.
  - Allow one to work locally in their administrative area

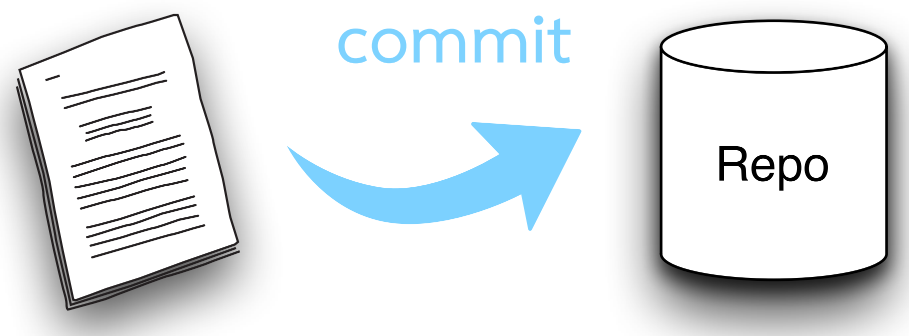
Operations

- Clone



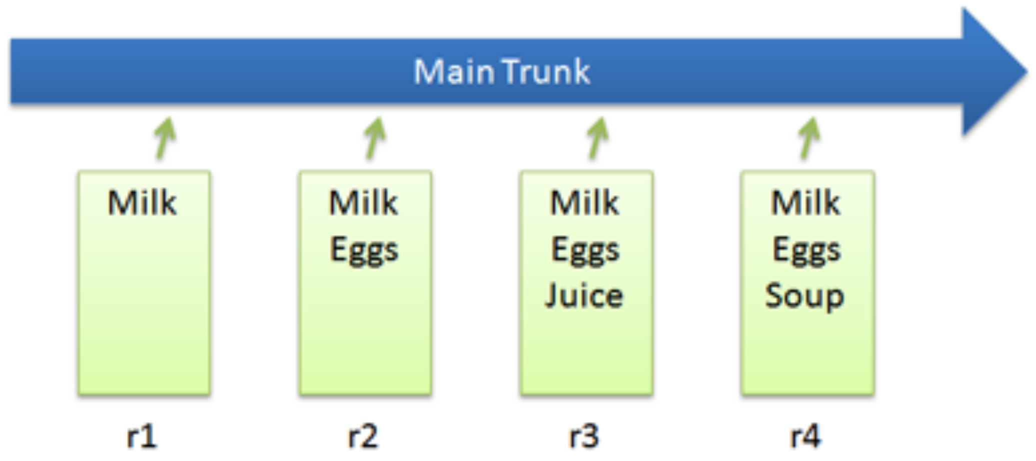
Operations

- Commit
  - Apply the modifications in the working copy to the repository.
  - Files to be uploaded to the repo is not the working copy version but what are changed from the master version being stored in the repo.
  - Comment messages should be sent along with the commit.



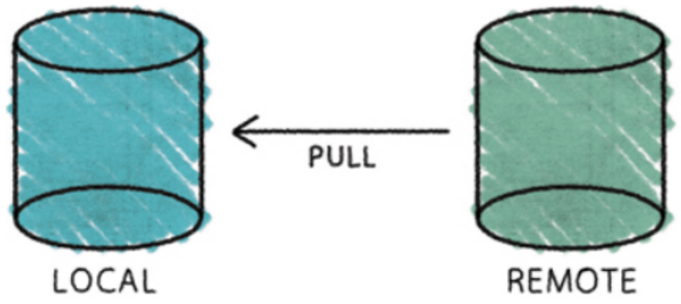
Operations

- Commit



Operations

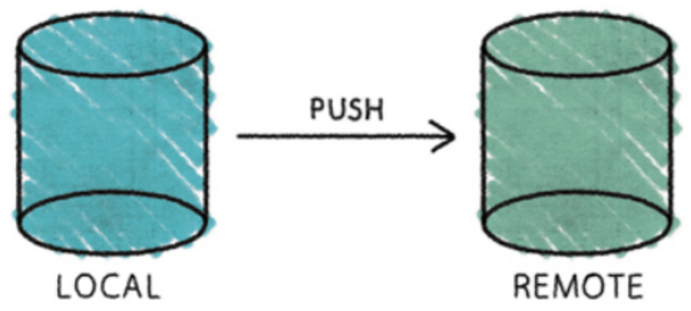
- Pull



  - Copy changes from a remote repository instance to a local one.
  - To synchronize between **two** repository instances.
    - Can be one to any other instances
  - Usually, the remote instance is the one from which the local was cloned.

Operations

- Push



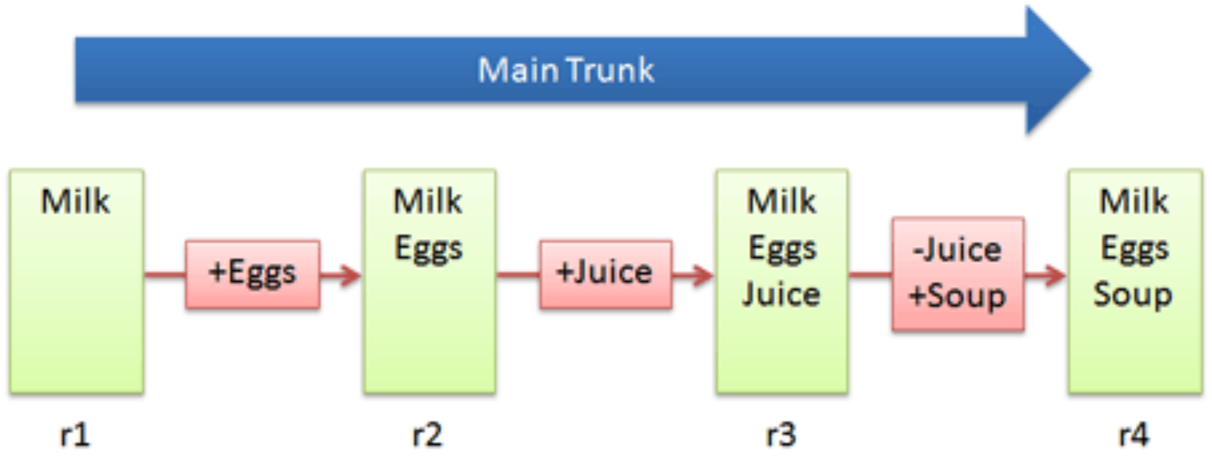
  - Copy changes from a local repository instance to a remote one.
  - To synchronize between **two** repository instances.
    - Can be one to any other instances
  - Usually, the remote instance is the one from which the local was cloned.

Operations

- Diff
    - Show the details of the modifications that have been made to the working copy.

Operations

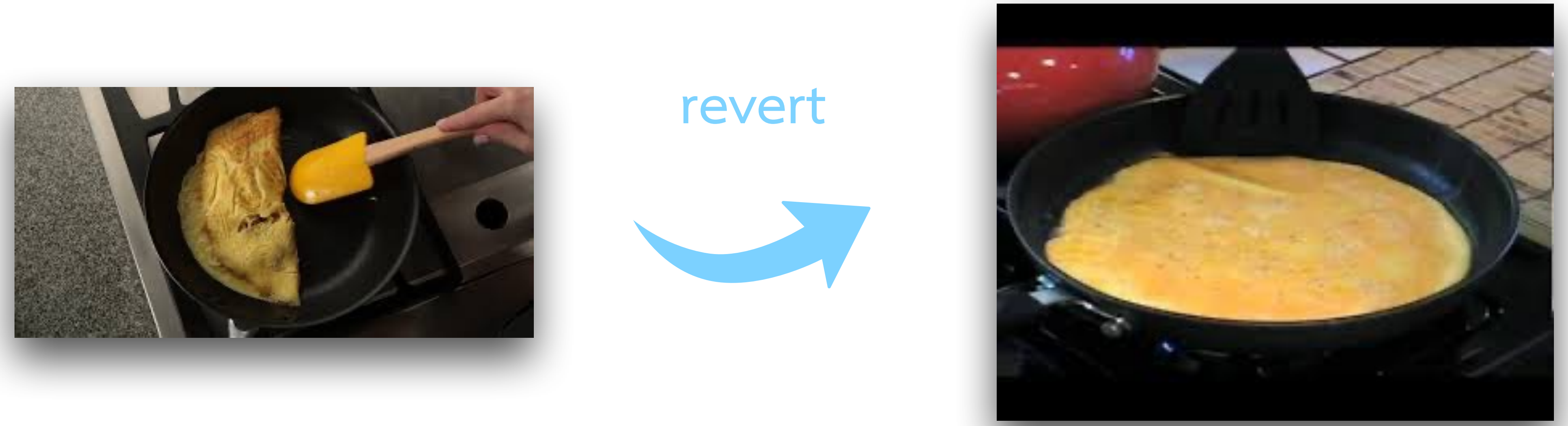
- Diff



Note diff(r1,r4) = +Eggs +Soup

Operations

- Revert
  - Undo modifications that have been made to the working copy to a version.

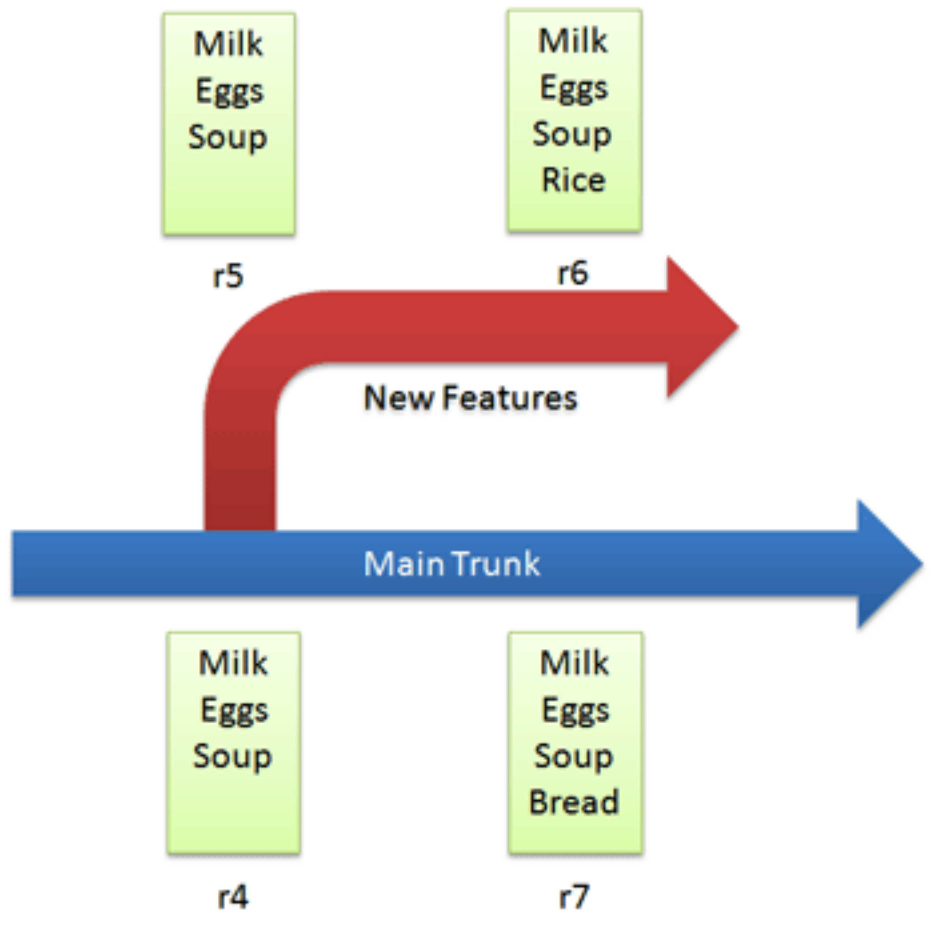


Operations

- Branch
  - The branch operation is what you use when you want your development process to fork off into two different directions.
  - For example, when you release version 3.0, you might want to create a branch so that development of 4.0 features can be kept separate from 3.0.x bug-fixes.

Operations

- Branch

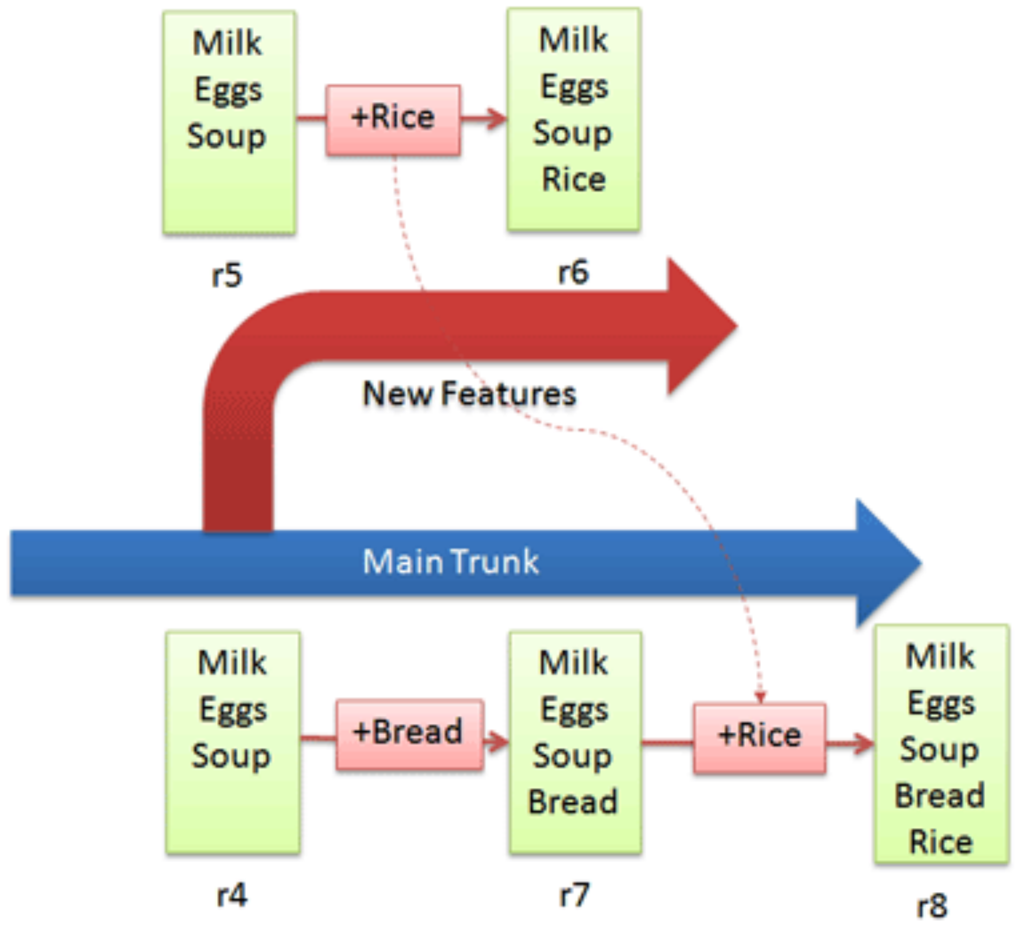


Operations

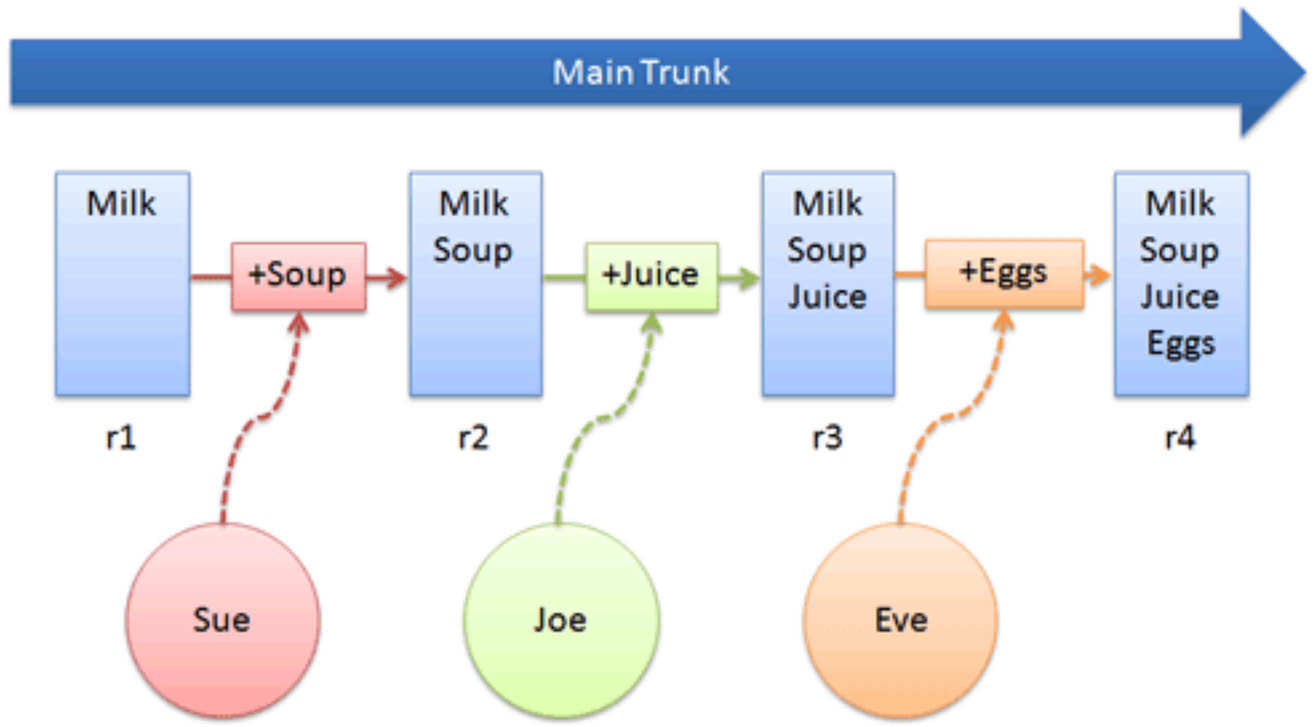
- Merge
  - Typically when you have used branch to enable your development to diverge, you later want it to converge again, at least partially.
  - For example, if you created a branch for 2.0.x bug-fixes, you probably want those bug fixes to happen in the main line of development as well.
  - Modern tools aim at making this merge feature as much automatically as possible.

Operations

- Merge

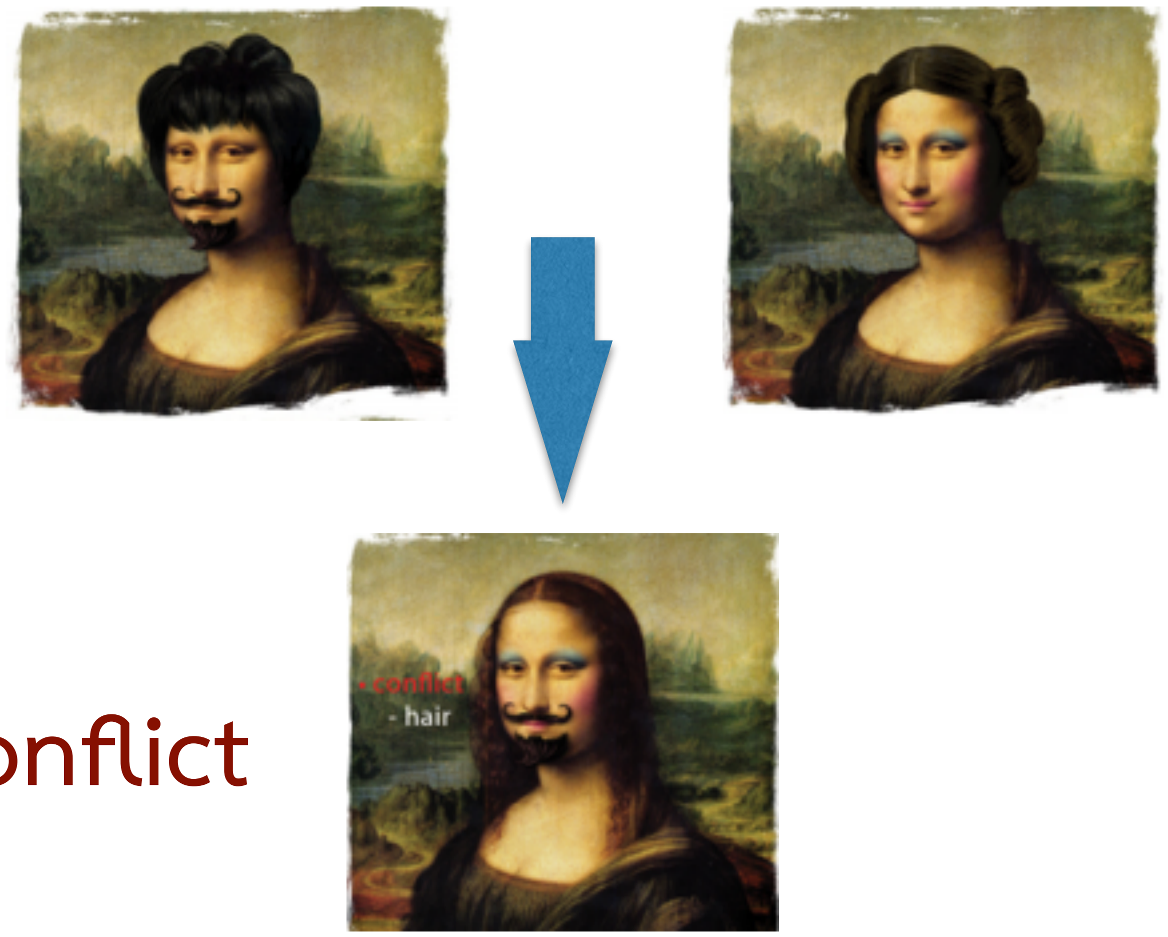


Multi Users


Main Trunk
Milk
r1
+Soup
Milk Soup
r2
+Juice
Milk Soup Juice
r3
+Eggs
Milk Soup Juice Eggs
r4
Sue
Joe
Eve

Operations

- Resolve
  - Handle conflicts resulting from a merge using human intervention.



Conflict

Question Times