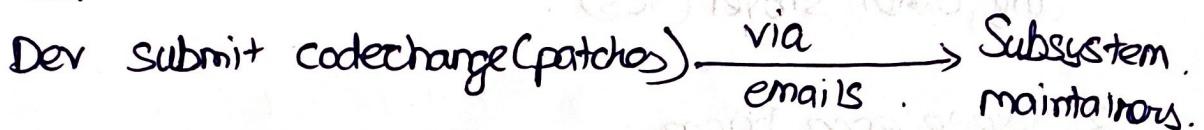


# Git

## Linux Kernel Community:-

- global group of developers, enthusiasts who collaborate to develop and improve the Linux kernel.
- relies on a distributed, collaborative model.

## Process:-



→ This process needs a robust version.

Control system (git.org mirror repository for distribution)

## Bitkeeper:-

- proprietary (proprietary), distributed version control system (DVCS).
- developed by BitMover Inc.
- designed to manage large, distributed software development projects efficiently.
- Better than CVS and SubVersion

## How Linux uses this:-

Linus Torvalds adopted Bitkeeper in 2002 for Linux Kernel Development. Bitmover provided a free of charge license for the open source community to use it.

## The conflict and the breakup

### ① License restriction:-

Developers using the tool for open source projects were not allowed to do contribution on ~~Linux Kernel development or to work on competing other VCS~~. ie do not use other shovel or not design any other shovel (VCS).

### ② Release engg Trigga :-

In 2005 Andrew Tridgell ~~reverse~~ engineered the Bitkeeper network protocols to create a functional open source client for it.

As there is a clear violation, BitMover CEO revoked the free license for Linux kernel dev community.

### ③ The Aftermath:-

→ Results in sudden and unexpected loss of their primary VCS.

→ Linus decided to write his own distribute, fast, open-source VCS

Result was git.

## Subversion:- Centralized VCS.

- single central repo stores all versions of project files.
- Dependent on server to commit changes.

### How it works:

→ Devs "check out" or get a copy of specific files they need from the central server to local machine.



Then they work on those files → To share changes with team.



They must commit their changes back to main server.

### Example

In Library - ~~a book~~ a book is taken. and write some notes on it and then return only the official copy for others. to see your changes. While the book is checked out; no one can modify the specific part.

### Drawbacks:-

→ Single point of failure

→ Req. internet connection.

→ Slower operations.

Eg:- Subversion (SVN) and CVS.

## Distributed VCS:-

- Every developer has a complete copy (or clone) of the entire project's repo.
- ↳ including its full ~~repository~~ history on their ~~single~~ local machine.
- There is no single master resource.

## How it works:-

Devs clone → They work and commit to local repo first.



To share changes.  
push the local commits to  
Github.

## Analogy:-

- A group of people has their own identical copy of entire book.
- They can make (changes) notes in their book anytime anywhere.
- When they meetup. they can swap notes/patches and merge their changes.

Eg: Git, Mercurial.