Version Control system:

**Terminologies of VCS:**

Basic setup:

Repository : (repos) – the database storing the files

Server: the computer storing repo

Client : the computer connecting to repo

Working set/working copy: your local directory of files, where you make changes.

Main : the primary location for code in the repo. Think of code as a family tree.  
---- the trunk is the main line.

Uses of VCS:

1. Made a change to code, realized it was a mistake and wanted to revert back.?
2. Lost code or had backup that too old ?
3. Had to maintain multiple version of a product ?
4. Wanted to see the difference between two (or more) versions of your code ?
5. Wanted to prove that a particular change bore or fixed a piece of code?
6. Wanted to review the history of some code?
7. Wanted to submit a change to some else code ?
8. Wanted to share your code, or let other people work on your code?
9. Wanted to see how much work is being done, and where, when and by whom?
10. Wanted to experiment with a new feature without interfering with working code?

Basic Action of VCS:

1)add/push : put a file into the repo for the repo for the first time, i.e, begin tracking it with version control

2) revision: what version a file is on (v1, v2,v3….etc)

3) head :the latest revision in the repo

4)checkout/pull/fetch : Download a file from repo

5) check in/ push: upload a file to the repository (if it has changed ), the file gets a new version number, and people can “check out” the latest one.

6)Check in message: a short message describing what was change.

7) change log/history : a list of changes made of a file since it was created.

8)update/sync : synchronize your files with the latest from the repo, this lets you grab the latest revisions of the files .

9) revert : throw away your local changes and reload the latest version from the repository.

Git advanced Actions:

Branch: Create a separate copy of a file/folder for private use ( bug fixing, testing, etc..)

Diff/change/Delta : finding the difference between two files. Useful for seeing what changed between revision.

Merge ( or patch) : Apply the changes from one file to another, to bring it up-to-date.  
for example : you can merge feature branch from one branch into another ( at Microsoft, this was called Reverse integrate and forward integrate)

Conflict : when pending changes to a file contradict each other (both changes cannot be applied ) .

Resolve: fixing the change that contradict each other and checking in the correct version.

Locking : Taking control of a files so nobody else can edit it until you unlock it. Some version systems use this to avoid conflict.

Git Workflow :

1. Define a strategy for committing, merging and promoting changes to a repo ( central repo)
2. Structured delivery of work, increase efficiencies and reducing errors.
3. Combines various branching strategies.
4. Introduces opportunities for code reviews and protected branches.

Git branching strategies :

1)Store the source code related to some code changes that will male code base, separately from the existing code base.

Using the branching, we can quickly maintain the separate line, to work on new source code in separate area (by branching )

Long Running branching :

Branching that remain open and receive regular merges ( Stories history little line of development often the trunk or master holds code for release.( official release )

Develop Branches : which allows integrated or changes in process , which bundle of feature branches are ready & discuss they would be integrated into long running devlop branch   
most of workloads typically or both branches  
  
Master

Devlop (feature ):

Feature Branching Strategy ;

Master

Devlop

Feature

Feature

Or

Master

Hotfix

Devlop

Feature

Or

Or

Master – contains a copy of the current production code ( tightly controlled )

Hotfix – for production bugs ( for master branch )

Release – develop contains enough release, release branch created

Devlop-- develop branch is used by everyone to build feature branch

Feature --- once feature is complete , this feature branch is merged into develop branch

Feature—same as above.