

UDENY = General Code Review?

Introduction to General

What is General?

- Source code Management (SCM) platform like GitHub
- version control system " Git
- provide code review and access control on code repos

Terminology:

change: represents a single commit that is under review. Each change is identified with a change-id

Patchset: patches on subsequent commits that are attempt to amend on existing change

Project Administration

Access Controls

Entity Review

Duration
Relevant to
object assigned

Example
rate / hour / month /

Access Category

What can or cannot
be done

Owner, push, Pull

Exclusion

If it person exclusion for
only this repository and
user / group

True or
False

User / Group
Access Rule

Who and what rule
is assigned

denying - Allow
approving - Deny

References

refs / * : all references in the project
 Use this to assign permissions to entire project

refs / tree / * : reference to push changes and change remote for
 eg: refs / tree / foo / head / * a particular destination branch

refs / changes / * : reference for each remote change and
 patchset

refs / head / * : reference for git branch

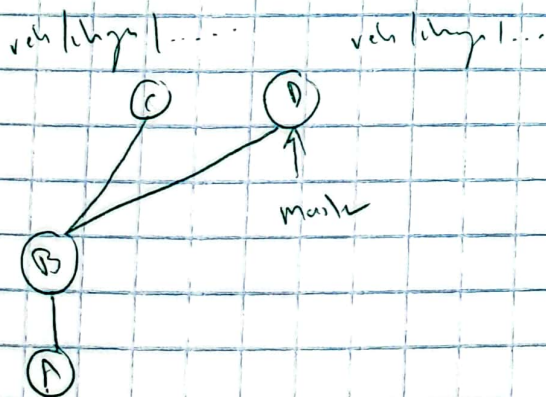
refs / tags / * : reference for git tags

refs / notes / * : where general configuration per project
 resides

Submit Types

- defines what client should do if the destination branch has
 changed state, meaning the head reference of the branch
 has changed due to merging or commit

Fast Forward → must ensure strategy that git will change by that
 only possible if the change being submitted to the same reference
 that the change was created from.



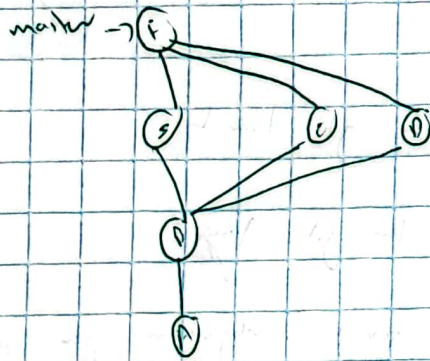
- Here we have patch set
 C and D, which are created
 from master branch reference
 B.

- Patchset approved and submitted
- merge will fast forward
 and new head reference
 of master branch will
 become D

- This option is only feasible if there is a low value of change to the repository as now has to release the change

Merge if necessary:

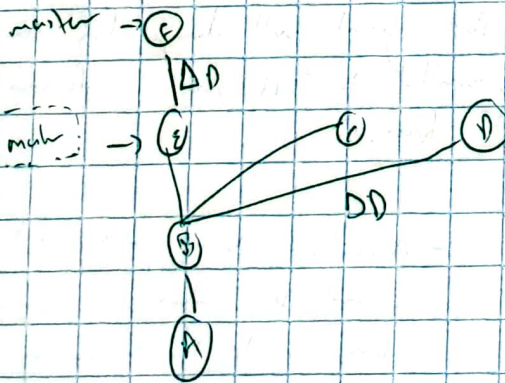
- only necessary if fast forward is not an option



- Here patch set D is submitted and a new merge F is created
- merge if necessary is the default and recommended option in Git

Rebase if necessary:

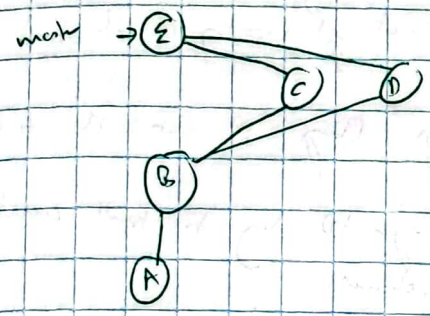
- if fast forward is not available, the rebase if necessary strategy will automatically rebase the patched and create a new patched
- and this patch is fast forwarded into the destination branch



- here diagram shows D is rebased and created
- F is the result of a fast forward into destination branch

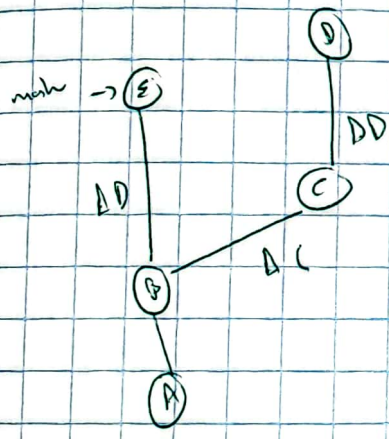
Rebase Always

- will change root if change into it despite fast forward being possible



- Here commit E is the result of rebasing D
- will change merge change into default branch, creating a new merge commit.

Cherry Pick



- Here change D has been applied but C rejected.
- Cherry picking will allow you to submit D despite being blocked by C

Pushing local to Commit

Commit - my hell:

- a shell script automatically commit on git commit
- must be pulled out placed in it but get reporting for each find
- get clear will need a globally merge change - 10 into it faster
- of it can't merge
- Commit was this change - 10 try to build commit
- account change - pull out releases