ISSN No. 0976-5697

Volume 5, No. 1, Jan-Feb 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Why to Go With Licensed Version Control Tool When Open Source Tool is There

Mansi Goel*[1], Priyanka Jain[2]
[1,2]Project Intern
ST Microelectronics Pvt Ltd.
Knowledge Park III Greater Noida, India
goelmansi28@gmail.com*, priyanka.jain2k5@gmail.com

Abstract: This Paper introduces why organizations should use licenced version control tool when open source tool (GIT) is available in the market. To purchase and maintain such tools, its hard for the small organizations, and now becoming a challenge for the large organizations too. For such kind of reasons and to make organizations cost effective, idea came for migration from vendor IBM Rational Clearcase to an open Source tool GIT/subversion. When all similar features are available in the open source tool GIT.

Keywords: Software configuration management, Clearcase, GIT, repository, cloning.

I. INTRODUCTION

Microsoft releases various versions of Windows. For each and every release , either old features are modified or new features are added. One release contain several programs .Each program compromises of several features and multiple developers contribute for the same program working on different features. Conflicts could occur, when two or more than two people work on same feature.

So, to maintain conflicts and track the changes Software Configuration Management Tool is required. In ST Microelectronics IBM Rational Clearcase is used.

Clearcase is software configuration version control tool. It is licensed version tool as well as its maintenance cost is too high. Why vendors should use IBM Rational Clearcase when same services and functionality is provided by Open Source Tool , GIT.

GIT is a free , distributed version control open source tool.[2]

There are three components in GIT which maintains and collaborate small as well as large projects.

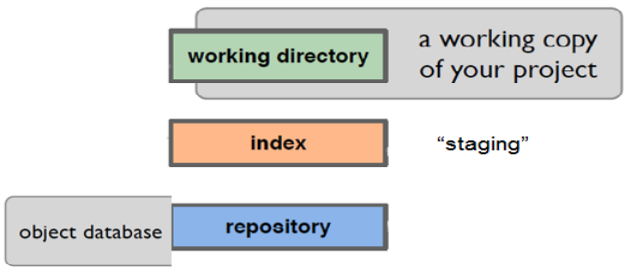

Figure: 1

- ***a. Working Directory*** – A Local copy of the repository is maintained in the local machine where actually user works.
- ***b. Index-*** Changes are temporary saved and yet to be commited.
- ***c. Repository-*** Central Folder is maintained for the project ,accessible to all the developers those belongs to their respective projects.[5]

Now our task is to migrate from IBM Rational Clearcase to Open source GIT . We prepared complete proof of concept for the migration.

A. Migration Proposal:

- ***a. Proposal1:*** Transfer everything from ClearCase to GIT including all versions and History
- ***b. Proposal 2:*** Transfer only Last version to GIT and ClearCase would be exist for older version reference with limited licenses for 1 year.
  - ***a) Time :*** Can take 6 month to 1 year
  - ***b) Effort*** : Avg 1 day (8 hrs) per Project/App
  - ***c) Risk :*** Minor
  - ***d) Resources :*** One trainee for 6 months - 1 year

We are going with the second proposal , because for first proposal lots of effort and time would be used. It would also create duplicity of data.

II. IMPLEMENTATION OF GIT

A. Getting Started:

Git can be installed on the local machine from the Web directly without any license and administration rights. With installation GIT GUI and GIT Bash is installed.[1]

- ***a) GIT GUI:*** Graphical Interface of GIT
- ***b) GIT Bash:*** Command prompt for GIT like Puttygen

a. Step1: Creating a Repository:

To Create a repository firstly maintain a folder for the project. Then Right Click on the folder and Click GIT Bash.

Figure: 2

Note: Create SSH Key through command ssh keygen for authentication.

Copy the ssh path URL into the Source Location and Target Directory would be any local path where we want to clone the Repository.

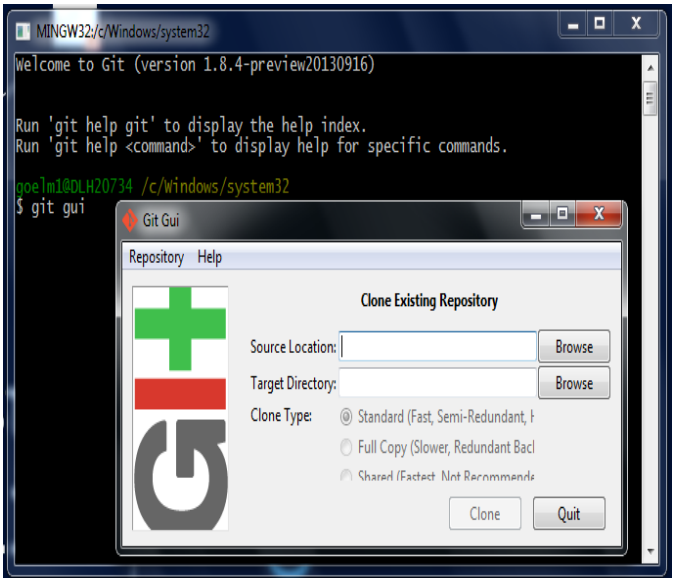

Figure: 3

b. Step2: Cloning:

Cloning is the beauty of GIT.

Organization would be maintaining a centralized directory consisting of all projects/Repositories. For working on the respective project ,developer have to copy the repository on his own local machine.

For this either we can use command <git clone> or we can use graphical interface to clone.

c. Step3: Workflow:

We have cloned a repository from the centralized location. Right click on the Folder to open GIT Bash Here.

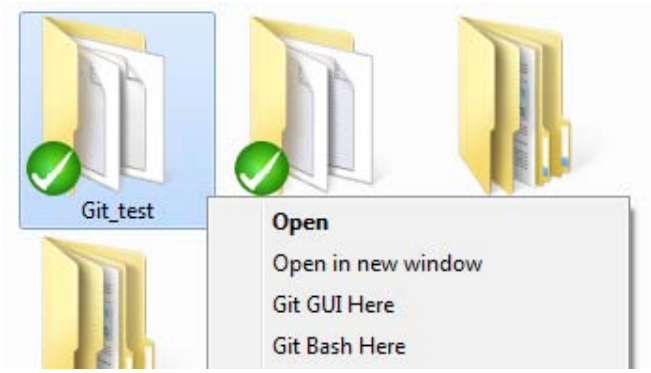

Figure: 4

a.) Create a File:

Can Create a file using vi editor e.g. vi demo_test.

b.) Add and Commit:

We added the file (demo_test) to the staging area using <git add .> and

Then save the content of this file using <git commit file name>.

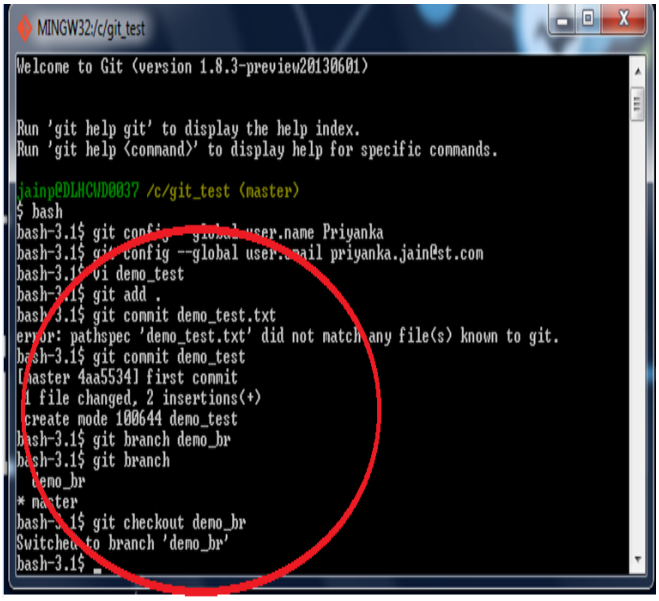

Figure: 5

c.) Branching:

Branch means a line of development. It is reference to the most recent commit on a branch.

We created a branch e.g. demo_br using <git branch demo_br>

By default it always points to Master.

To switch on any working branch <git checkout branch name>.

Note: Above figure is depicting the example of branch.

(a). We can also create a Tag(Label) on the branch to point out a specific commit because it references to an important step in your project development, use the Tag object using <git tag branch_name>.

d. Step4: Working With Remote:

a) PULL = Fetch + Merge:

To work on any File/Branch of the project on which others are working or have worked , fetch the data from the centralized location to the local machine. Or if local machine contains same file/ Branch and we want to update our data to work on the same File/ Branch we can merge the files.

Fetch operation to get changes made by another user using <git fetch –-all>

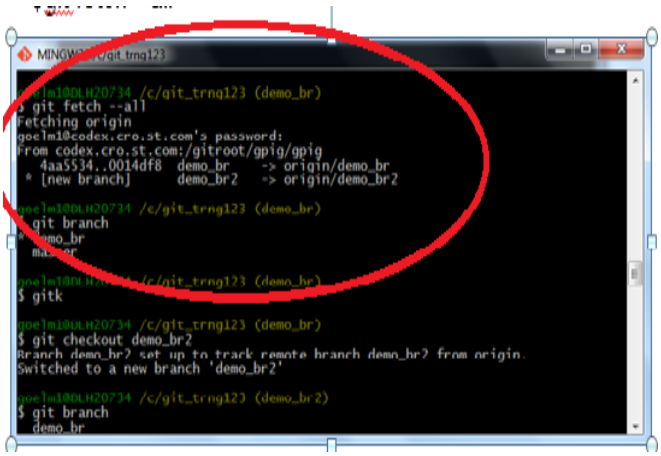

Figure: 6

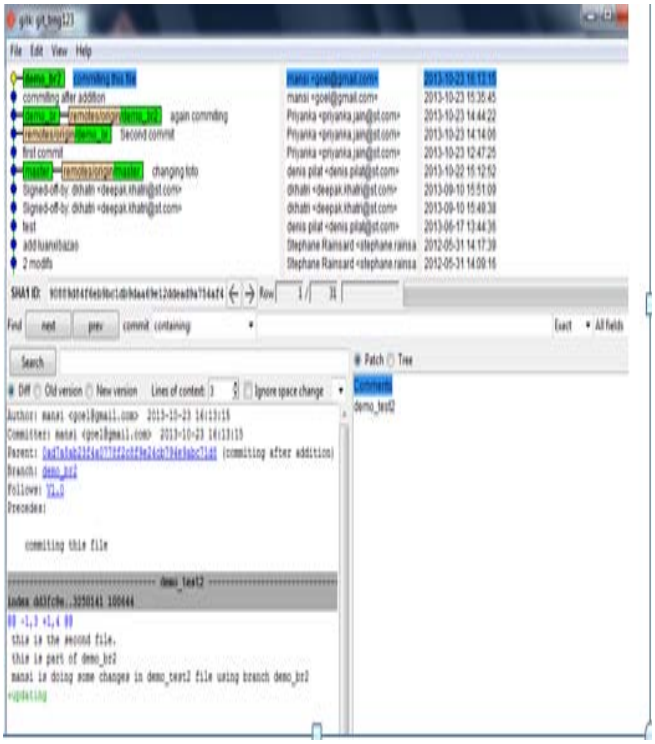

Figure: 7

We can also see the graphical representation of all the performed operations, using <gitk> on GIT Bash**.**

b). Merging:

To merge data, when wanted to work on the branch/File on which someone has worked.

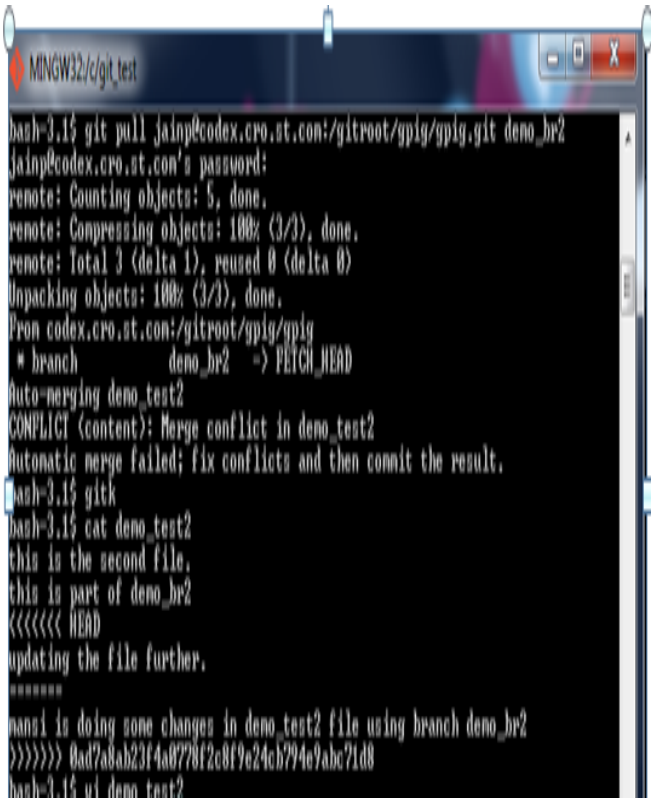

Figure: 8

Merging is also possible with merge tool manager , kdiff3 ,we can open a tool using a command on GIT bash < git merge tool -t kdiff3>[4]

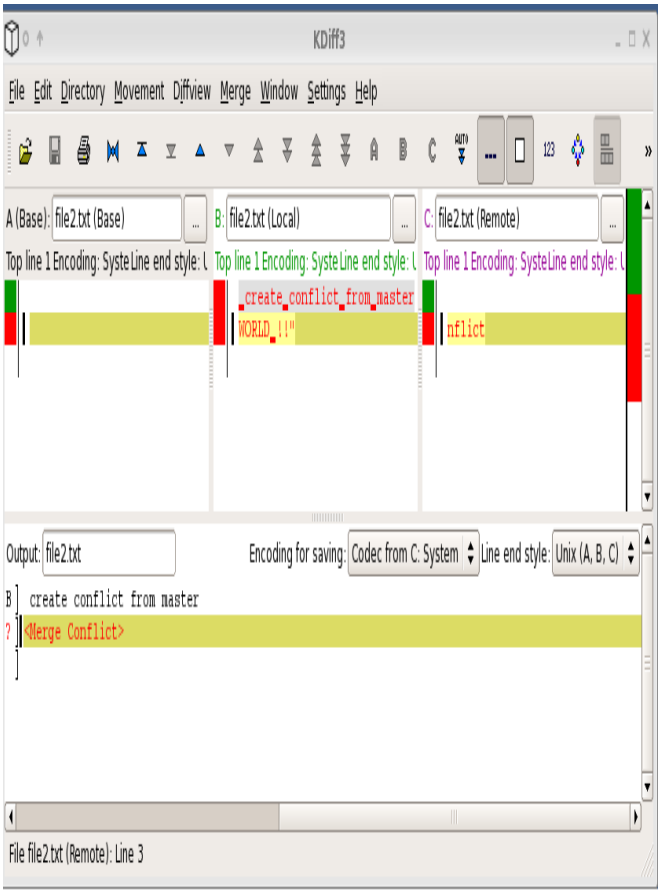

Figure: 9

(a). To Update the centralized repository after working on the local machine so, that other members can be able to work on the same branch/File, **PUSH** is performed. git push < remote> <branch>

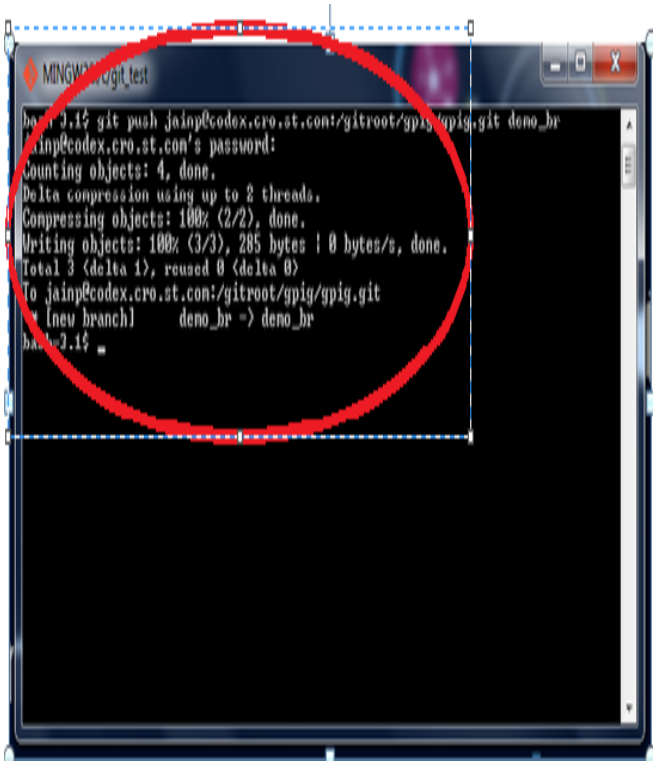

Figure: 10

Central repository is updated.

III. COMPARISION BETWEEN CLEARCASE AND GIT

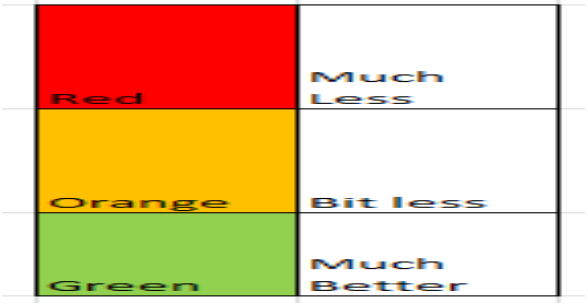

Figure: 11

	Issue\Tool:	Clear Case	GIT
2	Stability:	Usually stable, No loss of data. Sometimes, in unusual operations (e.g. deleting parent directory) some of the files might disappear and show up under Lost & Found.	The product is known to be very fast & Scalable.
3	Working off-line:	Allows working off-line except for Checkin and checkout.	Allows working Off -line .Git does nearly all of its operations without needing a network connection, including history viewing, difference viewing and commiting.
4	Recovery from failure, Atomic Commit	No support of atomic commit. Each file is checked in separately.	Supports Atomic Commit.Everything in Git is check-summed before it is stored and is then referred to by that checksum.This means it's impossible to change the contents of any file or directory without Git knowing about it
5	Performance	CC is known to be resource heavy tool on the server and disrupts the performance	GIT is very fast due to the fact that none of its operation depends on network latency.
6	Server Administration	Needs ongoing administrative attention	Very low administrative requirements.
7	Backup	There are ready made backup scripts but need to shut down the server for full backup.	Every repository is a backup of its remote.There are available backup scripts.
8	Operating System Support	Windows, Linux, Unix	Windows,Linux,POSIX, OS X.
9	Integration with Development tools	Good integration with Visual Studio .NET, Eclipse and with Windows Explorer	Integration is possible with Visual Studio and TFS.
10	Integration with Bug tracking systems	Possible with ClearQuest	Good integration exists with tools like Ticgit or BugTracker.Net.We can comment, add tags, manage the state of an issue, save views, and change to whom an issue is assigned.
11	Integration with Microsoft Office applications	Good integration with MS Office applications	Good integration with MS Office Application

Figure: 12

IV. CONCLUSION

GIT is fully distributed, offline work is possible, every clone is backup of the repository. Everything is very fast. IT is cost effective would help to increase business profit as well as user friendly. GIT is best open source tool to manage software configuration. We have modeled such structure in our organization to save the cost working effectively and efficiently.[2]

V. REFERENCES

[1]. http://msysgit.github.com/

[2]. http://git-scm.com/

[3]. http://git-scm.com/book/

[4]. http://gitref.org/

[5]. http://progit.org/