REPORT

VERSION CONTROL SYSTEMS COMPARISON AND ANALYSIS

(VCS)

**Contents**

[**Introduction** 3](#_Toc80788220)

[**1.Version Control System** 3](#_Toc80788221)

[**1.1 Different types of VCS which are in use today:** 3](#_Toc80788222)

1.1.1Local VCS……………………………………………………………………………………………………………………….3

1.1.2Centralized VCS……………………………………………………………………………………………………………..3

1.2.3Distributed VCS……………………………………………………………………………………………………………..3

[**2.Popular Version Control Systems and its benefits.** 3](#_Toc80788223)

[**2.1 SVN** 3](#_Toc80788224)

[**2.2 Mercurial** 4](#_Toc80788225)

[**2.3 Git** 4](#_Toc80788226)

[**3. Selecting a VCS that fits the organisational requirements.** 5](#_Toc80788227)

[**4. Git Installation Process for Projects** 5](#_Toc80788228)

[**References** 6](#_Toc80788229)

# **Introduction**

VCS stands for Version Control System. In a software company, version control management is an inevitable part of every software project and its development team.

## **1.Version Control System**

First of all, what is VCS?

VCS is a system which manages the changes made in a file or set of files inside a project module. Every change made to the file is recorded as a new version. This helps to track the changes and progress of a software in development phase by the programmers.

### **1.1 Different types of VCS which are in use today:**

* Local Version Control System.
* Centralized Version Control System.
* Distributed Version Control System.

#### **1.1.1 Local Version Control System.**

Local version control tracks the changes in a file or set of files within the local computer. This method cannot be used in a larger project where multiple programmers involved.

**1.1.2 Centralized Version Control System.**

In this system, all the changes made in the file system is tracked under a centralized server. Hence, multiple programmers can access the file from the common server location and work on their copy. But the central server will have the main file and all the changes history.

**1.1.3 Distributed Version Control System.**

One of the widely used VCS type is Distributed Version Control as it provides the flexibility to access from a common server to a number of other users without affecting the main version. Here the users clone the entire repository to their local computer and make changes and then update under the main branch.

## **2.Popular Version Control Systems and its benefits.**

Many VCSs are in use today. Software companies choose their VCS according to the project requirements and companies’ policies. Some of them are described below:

### **2.1 SVN**

SVN is one of the VCS and term stands for Apache Subversion. It’s a centralized version control system. Its free and open source licenced one.

**Pros:**

* Offers cheaper branch operations.
* Wide variety of plugins for IDEs available.
* It’s a newer technology and its free.

**Cons:**

* Still has bugs
* Slower compared to other VCSs.
* Insufficient repository management commands.
* Branching and tagging takes a lot of disk space.

Its official website of SVN is:

<https://subversion.apache.org/>

### **2.2 Mercurial**

Mercurial is also a free and open-sourced VCS but it’s a distributed type of version control system**.** Mercurial is also a widely used VCS because it is easy to learn and use as its interface and command prompt is very simple. Also, it is customisable and well suited for projects in Python and PHP.

**Pros:**

* Easy to learn and free to use.
* Has better documentation.
* Ability to maintain a consistent user interface.

**Cons:**

* Does not have option to merge to parents
* Less out of the box power.
* Cannot distinguish between local and remote head.

The official website of Mercurial is:  
<https://www.mercurial-scm.org/>

### **2.3 Git**

Git is the distributed version control system, and it is also an open source licenced one. Git is popular for its speed and performance. Git is supported by all operating systems.

**Pros:**

* Very light, fast, safe, and high performance.
* Does not require server to use as we clone the repository to local.
* Cheaper branching operations.
* Easy branching and merging operations.
* Full history tree available offline.

**Cons:**

* A bit hard to learn and get used to it.
* Not optimal for single developers.
* Has only limited windows support.

The official website of Git is:  
<https://git-scm.com/>

## **3. Selecting a VCS that fits the organisational requirements.**

After analysing the pros and cons of popular version control systems used today, it is evident that, Git is the most efficient one in terms of usability, speed, and performance.

However, the chosen VCS must comply and meet the organisational requirements. For that, lets analyse if Git meets the Globex’s requirements for VCS.

* Git is a distributed version control system and has good online supporting systems.
* Better and effective branching system in Git is suitable for Globex’s requirement for branching.
* Git allows the users to enter meaningful messages with every commit action which helps to easily understand it later by other users when accessing the file.
* Git prompt has options to add author’s name and email address at each commit.
* Always Git repositories can be started with a Readme.md file which will be used for storing project details and description and no source code needed.
* Git can keep the commit scope to a single change.

Hence Git will be the selected VCS for this MUSOPLAN software project ss Git meets all the VCS requirements listed by Globex.

## **4. Git Installation Process for Projects**

**Step 1:**

* Go to **GitHub** website. <https://git-scm.com/>
* Create an account in your name.

**Step 2:**

* + Click on **Repositories** Tab and create a repository in the name of your project.
  + Give Project description(optional)
  + Click on the checkbox near **README.md** file while creating the repository.
  + On Summit, your GitHub repository is created with a **Main Branch** and readme.md file.

**Step 3:**

* Create a local **Project Folder** in your computer.

**Step 4:**

* Copy the GitHub repository link.

E.g.: https://github.com/yourName/MusoPlanProject.git

* Open Git Bash command folder from the local Project Folder in computer.
* Clone the GitHub project repository using Git command: **git clone**

**Step 5:**

* Change directory from local Project folder to the GitHub repository cloned as main in the local computer.

E.g.: cd GitHub Project Repository

Now the main branch is from GitHub Project repository.

**Step 6:**

* Now we can create the project related text files here.
* After creating or any update in the files, always do the git commands,

**Git add** and **Git commit**.

* Later the git commands like **git push** will add the files to the Git repository under main branch.

# **References**

* Rob Rawson 2021 Version Control Software Comparison: SVN, Git, Mercurial.

<https://biz30.timedoctor.com/git-mecurial-and-cvs-comparison-of-svn-software/>

* Kamaldeep Singh CEO at Zhoustify · Nov 13, 2020. Decisions about Git, Mercurial, and SVN (Subversion)

<https://stackshare.io/stackups/git-vs-mercurial-vs-svn>

* March 7,2018. Learn The Three Different Types of Version Control Systems.

<https://blog.eduonix.com/software-development/learn-three-types-version-control-systems/>

* <https://www.temok.com/blog/version-control-systems/>