**VCS research report**

***Different VCS investigated -*** Rather than have only one single place for the full version history of the software as is common in once-popular version control systems like CVS or Subversion (also known as SVN), in Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.Unlike some version control software, Git is not fooled by the names of the files when determining what the storage and version history of the file tree should be, instead, Git focuses on the file content itself. After all, source code files are frequently renamed, split, and rearranged. The object format of Git's repository files uses a combination of delta encoding (storing content differences), compression and explicitly stores directory contents and version metadata objects.Git repository are secured with a cryptographically secure hashing algorithm called SHA1. This protects the code and the change history against both accidental and malicious change and ensures that the history is fully traceable.Some other version control systems have no protections against secret alteration at a later date. Git has been designed to support branching and tagging as first-class citizens (unlike SVN) and operations that affect branches and tags (such as merging or reverting) are also stored as part of the change history. Not all version control systems feature this level of tracking.

***Benefits of different VCS -*** Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.

***How the VCS I chose fits with the organisational requirements -*** Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. I chose Git over other VCS because of   
\*feature branches - Git branches are cheap and easy to merge.   
\*distributed development - each developer gets their own local repository, complete with a full history of commits.  
\*Pull requests - to keep the team in sync  
\*faster release

Installation process (including any issues encountered, and pre-installation factors)

Install the Xcode Command Line Tools and try to run git from the Terminal the very first time. If it is not installed already, you’ll be prompted to install it. Download and install it as instructed.

***What disruptions installation may cause***

***Configuration process*** - For making sure the VCS has my name as the author of commits, I use the following commands for the musoplan repository:  
$ git config user.name "Smita Jacob"  
$ git config user.email "smitajcb@gmail.com"

Bibliography:

Git-scm.com. 2021. *Git*. [online] Available at: <https://git-scm.com/> [Accessed 26 August 2021].

Atlassian. 2021. *Why Git | Atlassian Git Tutorial*. [online] Available at: <https://www.atlassian.com/git/tutorials/why-git> [Accessed 26 August 2021].

Atlassian. 2021. *Why Git | Atlassian Git Tutorial*. [online] Available at:  
https://www.atlassian.com/git/tutorials/what-is-version-control> [Accessed 26 August 2021].