**VERSION CONTROL1**

1. **What is Version Control?**

**Definitions:** Version control is a way of recording changes to a file or set of files over time in order to recall specific versions later.

* As testers, you will most likely version control software source code though in reality you can do this with nearly any type of file on a computer.

* As collaborative project engineering team, you would certainly want to keep every version of code therefore it will be wise to invest in a version control system.

**Advantages**:

* It supports code reversion to a previous state.
* It supports change comparison over time.
* It is transparent and helps trace issues to when and by whom they were introduced.
* Helps recover lost or broken files.

1. **GIT**

* Git is an open source version control system (VCS)
* It is responsible for everything GitHub-related that happens locally on your computer.
* To use Git on the command line, you'll need to download, install, and configure Git on your computer.
* If you want to work with Git locally, but don't want to use the command line, you can instead download and install the [GitHub Desktop](https://desktop.github.com/) client.
* If you don't need to work with files locally, GitHub lets you complete many Git-related actions directly in the browser.

1. **Git Hub**

* GitHub Inc. is a web-based hosting service for version control using Git.
* It is mostly used for computer code.
* It offers plans for both private repositories and free accounts which are commonly used to host open-source software projects.

**Common bash commands**

* **Git clone -** Clones repository
* **Git pull** - Pulls content of repository from gtihub
* **Git checkout branch name -** checks out an existing branch
* **Git checkout -b “branch name” -** Creates a new branch which is a copy of existing branch on github
* **Git add..** - adds all local changes to local branch
* **Git add file name** - Adds specific file changes to local branch
* **Git status -** Displays the state of the working directory and staging area
* **Git log -** Displays all commits in a repository's history
* **Git commit -** Used to save your changes to the local repository
* **Git commit -m “commit message” -** Adds message to commit
* **Git push -** Update remote repository with local changes
* **Git push -u origin head -**