



FOR STUDENTS : ALL THE INGREDIENTS OF A GOOD ESSAY

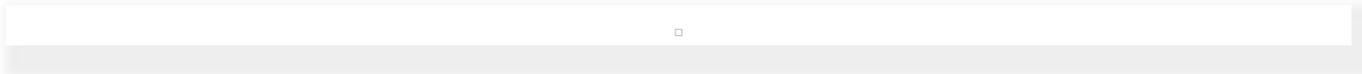
Menu



Essay: What is version control?

July 15, 2019 by Essay Sauce

Preview of page one of this free downloadable essay:



Essay details:

- **Subject area(s):** Information technology essays
- **Number of words:** 2028
- **Price:** Free download
- **File format:** PDF

Overall rating: **0** out of **5** based on 0 reviews.

500 word text preview of this essay:

The full version of this essay has 2028 words and is available to download in PDF format above.

A version control system or VCS is a software tool which allows you to manage multiple revisions of the same file, i.e source code files. Version control software helps keep track of modifications made to the file over time. Files are regularly pushed to repositories online as the users continue to update the file. This allows the user to preview earlier versions of the file if need be.

Below is a graphical representation of a version control system. The VCS allows multiple actors to cooperate and work collaboratively on files. Using a version control system users can push (upload) and pull (download) files from a repository as they continue to develop their files.

Issues software developers face with source-code management without a VCS

A common issue when dealing with code or information that is stored on a computer is handling the changes you make to the file. After making changes such as adding/deleting something in the file you may find later that you want to undo that particular change. At a very basic level this can be as simple as clicking undo in a word processor. However, at a more advanced level such as developing source code as a software developer, things aren't this simple and simply re-saving the file under a different name, i.e 'Project v2 FINAL.CPP' will not be sufficient. Creating duplicate files is a very common yet very inefficient way of handling different versions of a particular file, it can be an absolute nightmare for software developers who are continuously improving and updating their code. Using a version control system such as Git allows the user to know who made changes to a file (if working on code collaboratively), when the changes were made and also what changes were made to the file.

How the use of version control software can mitigate source-code management issues

When using a version control system rather than simply saving a document after you have made changes to it you commit it. Committing is a save-like feature where the version control system stores the particular version of the file at the time of committing. When you commit a file to a repository you have to define an accompanying message which typically states the reason for the change. However, when another commit is made rather than overwriting the existing file it simply stores the new version of the file while also keeping a copy of previous versions in its history. Therefore, the user can check the different changes that have been made to the file at a later date. And so version control systems can solve the issue of reviewing previous changes made to the file and allow one version of the file to be used at all times rather than having many duplicates of the file.

Version control software has become essential for collaboration of work. With a version control system everybody who wishes to work on the file can do so freely. Users can simply pull the latest version of the file from the repository and commit when necessary. The VCS will allow for merges of changes from different users into one common version of the file. When multiple people work together on the same file but have different tasks such as fixing bugs or adding new features to the code they can use different branches to work on. This means that each task can be worked on independently of each other, and then down the line the branches can be merged.

The screenshots below shows a very basic example of using version control software for source-code development. The first screenshot shows the original version of main.cpp while the second while shows the updated version of the file. The source code control system NTU uses is called WARREN, this particular source code control system highlights the changes made so they are clearly visible; red for what has been removed from the file and green for what has been added.

Using a version-control system – Git

Git is the world's most popular version control system, it is not only mature but also actively maintained. In this tutorial I will not only be covering how to get started with Git but I will also be going over its main uses/commands with the use of examples.

Installing Git

To get started with Git the first thing you need to do is install Git on your machine, Git is supported in most up-to-date unix based operating systems and can be installed using the command line (refer to Git's website for instructions on how to do so). Git can also be installed on windows machines, however in order to do so you will need to install their Git bash shell.

Getting started with Git – git config

Now that you have Git on your machine it is recommended that you do a couple of things to customise your Git environment. In your command line you can run a command – ‘git config’, this will allow you to set configuration variables which can change different aspects of Git. Using git config you can tell git who you are, it allows you to configure a name and email address to be later used with your commits. Furthermore, you can configure what default text editor will be used when you need to view/edit text using Git. An example of the use of this command is shown in the screenshot below.

Setting up a repository

Once you have installed Git and created the config file you can now begin to use Git. To start off using Git you want to create a repository for you to store your work in. You can do this by using the ‘git init [repository_name]’ command. Running this command will initialise an empty Git repository with the name you specified and create a new master branch. Below is a screenshot showing an example of this.

Cloning a repository

If a project has already been setup in a Git repository then the git clone command is usually what is used to retrieve a copy of the repository to be stored and worked on locally. To clone a repository you simply write ‘git clone <repository URL>’, once the repository has been cloned you can then work on it in your local branch. Below is an example of using the git clone command using a URL from a repo on Github.

Pulling from a remote repository

Using the ‘git pull’ command essentially pulls the changes from a remote repository and integrates those changes into your current local branch. As using git pull merges the changes in the remote repository into your local repository you can encounter something called a merge conflict. I will briefly cover merges and merge conflicts later in the tutorial. Below is a screenshot of me using the git pull command on a project repository I am currently working on.

Status of a repository

Using the git status command tells us a few pieces of important information. Firstly, it tells the user what branch you are currently on, which is very important especially if you are working with multiple branches. Furthermore, it also tells you’re tracked and untracked files, which means files that have and haven’t been staged (staging is the step before the commit process, covered later in the tutorial). Below is an example of using the git status command with the repository we recently created. I saved ‘test_commit.rtf’ in the new repo and so it is coming up as untracked, this is because we have not added the file to the staging area yet.

Staging a file for a commit

Staging a file simply means preparing it for a commit. You stage a file by using the command ‘git add [file_name]’. You must stage a file in order to commit it, however, if you stage a file but no longer wish to commit it then you can unstage the file following the instructions in the command line. Below is a screenshot showing of the status of the repo following the git add, as you can see test_commit.rtf has moved from ‘untracked files’ to ‘changes to be committed’.

Committing a file

Committing a file essentially records the changes you have made to the repository, once the file has been committed it is ready to be pushed to the remote repository. You commit a file by using the command ‘git commit -m[commit message]’. When you commit a file you can specify a message to accompany the commit

as to why you are committing the file, usually a small message containing what has been changed. This message is especially important when looking at past versions of source code to see what has been changed along the way. Below is a screenshot showing the process of committing a file.

Pushing a file to a remote repository

Once you have staged and committed the files you wish to upload to the remote repository you are then ready to push those files. Pushing is essentially transferring the commits in your local repository to the remote repository. When using the 'git push' command you need to be careful as it can potentially overwrite changes made by others. Below is a screenshot of the process of staging, committing and pushing a file to a remote repository.

Branches

Git branches are effectively independent lines of development. Branches help isolate your work from other users who are also working on the same project. Branches are essential for splitting up tasks, you will always want to create a new branch to encapsulate your changes, branches can later be merged into a single branch. You can create a new branch using the git command 'git branch [branch_name]', you can list the current branches in the repository by using 'git branch', with the branch that you are currently on highlighted with an asterisk. If you wish to switch to a different branch you use the command 'git checkout [branch_name]'. Below is an example of me using the commands mentioned above, from listing the branches to creating a branch to then switching to that branch.

Merging

The git merge command lets you merge the different branches that have been created in the repository into one single branch. As I mentioned in the branches section of this tutorial, many branches are created for different tasks so they can stay independent of one another and so merging is essential to combine all the branches into one. Git merge is used to merge two branches together, you use the command 'git merge [branch_name]' to merge a different branch into your active branch. In the example below I merged the 'test' branch into my master branch.

However, merging isn't always this simple, sometimes you can run into something called a merge conflict. Merge conflicts usually come about when the merging of the two branches has changes in the same file, often Git can work this out on its own, however there are times when it needs the users help. This is usually the case when the same line in the file has been changed on both branches, Git simply does not know which version of the file is correct and so there is a merge conflict which you as the user will then have to fix.

What are continuous-integration servers?

Continuous integration is a software development practice used in many coding projects. An automation verification system is setup to run a series of tests when there is a new merge into the repository. Therefore, each time source code is updated and pushed to the remote repository the user will know if there any issues/bugs with the new updated code. Developers use this so they do not have to hunt down an issue with the code after a very large merge, testing the code each step of the way helps narrow down any potential issues.

Why are continuous-integration servers used in software development?

About Essay Sauce

EssaySauce.com is a completely free resource to help students research their academic work and learn from

great essays!

View all posts by Essay Sauce

...(download the rest of the essay above)

About this essay:

This essay was submitted to us by a student in order to help you with your studies.

If you use part of this page in your own work, you need to provide a citation, as follows:

Essay Sauce, *What is version control?*. Available from: <<https://www.essaysauce.com/information-technology-essays/what-is-version-control/>> [Accessed 15-07-19].

Review this essay:

Please note that the above text is only a preview of this essay. The full essay has 2028 words and can be downloaded free in PDF format, using the link above.

Name *	<input type="text"/>
Email	<input type="text"/>
Rating *	☆☆☆☆☆
Comments (optional)	<input type="text"/>
<input type="submit" value="Submit"/>	

Latest reviews:

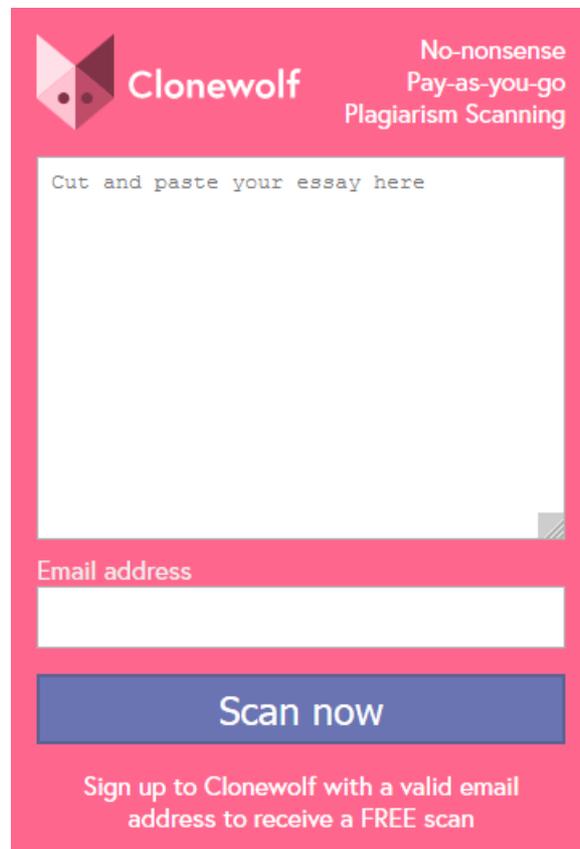
Information technology essays

- < Shakespeare's King Richard III and Al Pachino's Looking for Richard
- > Interpretive Practice Worksheet: Galatians 5:19-23

Search for student essays:

About EssaySauce, the student essay site:

EssaySauce.com is a free resource for students, providing thousands of example essays to help them complete their college and university coursework. Students can use our free essays as examples to write their own.



The screenshot shows the Clonewolf plagiarism scanning interface. At the top left is the Clonewolf logo, a stylized wolf head in shades of pink and purple. To its right, the text reads "Clonewolf" in a bold, sans-serif font, followed by "No-nonsense Pay-as-you-go Plagiarism Scanning" in a smaller font. Below the logo and text is a large white text area with the placeholder text "Cut and paste your essay here". Underneath the text area is a pink input field labeled "Email address". Below the email field is a blue button with the text "Scan now". At the bottom of the interface, there is a pink banner with the text "Sign up to Clonewolf with a valid email address to receive a FREE scan".

Latest student essays:

Teaching styles

How do God and humans relate?

16th century to 17th century education

Phonics lessons

AI Classifiers

The History of Public Education in the US

Federalism

Physician Assisted Suicide (PAS)

A person's habitus

How we care for our clothes

Student essay categories:

Accounting essays

Architecture essays

Business essays

Computer science essays

Criminology essays

Economics essays

Education essays

Engineering essays

English language essays

English literature essays

Environmental studies essays

Finance essays

Geography essays

Health essays

History essays

Hospitality and tourism essays

Human rights essays

Information technology essays

International Relations

Law essays

Leadership essays

Linguistics essays

Literature essays

Management essays

Marketing essays

Media essays

Medicine essays

Miscellaneous essays

Music Essays

Philosophy essays

Photography and arts essays

Politics essays

Project management essays
Psychology essays
Religious studies and Theology essays
Science essays
Social work essays
Sociology essays
Uncategorized
Zoology essays

Average review:

Overall rating: **0** out of **5** based on 0 reviews.

Q: Is EssaySauce.com free?

Yes! EssaySauce.com is a completely free resource for students. You can view our **terms of use** here.

Why use Essay Sauce?

The brightest students know that the best way to learn is by example! EssaySauce.com has thousands of great essay examples for students to use as inspiration when writing their own essays.

Is Essay Sauce completely free?

Yes! EssaySauce.com is a completely free resource for students. You can view our **terms of use** here.

Help! I found my essay!

All of our essays are donated in exchange for a free plagiarism scan on one of our partner sites. However, despite displaying clear terms on our sites, sometimes users scan work that is not their own and this can result in content being uploaded that should not have been. [Find out what to do if this happens here.](#)