Release Notes Report

Name: Matthew Rix

Date: 3/20/2023

1. **Overview**/General Remarks:
   1. Version Control Summary
      1. Describe the **benefits** of version control tools in organizing software development projects.

Version control tools such as Git and Bitbucket have revolutionized the way software development projects are organized and managed. Some benefits of using version control tools include tracking changes, backup generation, ease of experimentation in development, and collaboration (Metwalli 2020). Overall, version control tools provide a structured approach to software development, enabling developers to work more efficiently, collaborate effectively, and ensure the quality of the codebase.

* + 1. Describe how version control helped you **manage** your source code in developing the Calculator app.

The version control system implemented this week was very helpful in managing the changes I made. Unfortunately, due to technical difficulties with the Apporto app I had to migrate the code base from the U drive to the Z drive. Since the division function was added prior to the drive migration the initial commit in the local repo and the division branch were combined. However, this did not really affect my ability to effectively separate the commits of the different functions.

1. **Development:**
   1. Explain the **benefits** of **pushing local repositories** to remote shared repositories specific to expanding the functionality of repositories. Be sure to explain how **configuration and cloning** can be used to expand the functionality of repositories.

Pushing local repositories to remote shared repositories can bring several benefits to expanding the functionality of repositories. Some of those benefits include collaboration, backup and recovery, and accessibility. By pushing your local repository to a remote shared repository, you can collaborate with other team members on the same project. This allows multiple people to work on the same codebase simultaneously and track changes made by others. It also facilitates code reviews and helps ensure consistency in your coding conventions. Pushing your local repository also serves as a backup mechanism for your code. If you lose your local copy of the repository, you can always retrieve it from the remote repository. Additionally, if you accidentally delete or modify files in your local repository, you can recover them from the remote repository. In making your code available on a remote shared repository, you can access it from anywhere with an internet connection. This is particularly useful if you need to work on your code from different devices or locations.

To expand the functionality of repositories, configuration and cloning can be used to customize the behavior of software applications, including version control systems. By configuring your version control system, you can control how it interacts with your code and with other team members' code. For example, you can set up access controls to ensure that only authorized team members can modify specific files or that only specific members can approve branch merges with main. Cloning allows you to work independently on the code locally and push changes to the remote shared repository when you are ready (Sphitz 2022). Cloning is also useful for creating local copies of repositories that you may not have access to otherwise. For example, if you want to contribute to an open-source project, you can clone the project's repository to your local machine and make changes to it before submitting a pull request.

* 1. Describe your experience with branching and merging in your development process, being sure to discuss conflicts, blaming, stashing, and rebasing.

Throughout the last few weeks working with the calculator app code base, I used Git with a Bitbucket repository. I began by creating a local repository and doing an initial commit to the cloud repository. Unfortunately, at first the workspace I was using on the U drive wasn’t allowing me to connect to the remote repository. When I switched to the Z drive, I had already added a branch off of main for adding the division functionality which threw off some of the guides, but I still understood and was able to demonstrate the core concepts. One of those core concepts was conflicts which at first, I resolved manually by just clearing out the generated text, adding in the two lines of code and saving. I did this because I wasn’t sure if the move to the left tool would erase what was on the left. On my second go at resolving the conflicts after I rebased and reset, I used the tool and can see how that would be much more useful with large conflicts in the code. I didn’t do any stashing as I’m already familiar with it from where I work so I just did a commit without pushing it to the remote repository.

* 1. Specify how you utilized the **states of a source file,**such as committed, modified, and staged, in relation to the file status lifecycle.

In most of these tutorials I was able to use the different states of the files to complete the necessary work. For example, I had the Division.java file in a modified or ‘Dirty’ state in which the file has been modified since the last commit (Vogel 2023). When the instruction called for it I moved into the staged are in the Git Staging tab via the green plus icon and then committed when ready. During the point in which the Division.java file was dirty I was also able to modify, stage, and commit other files like the Power.java file. I did pay special attention though when staging that file not to include the Division.java file so as not to commit it before it was supposed to be committed.

References:

Metwalli, S. A. (2020, October 14). *Version control 101: Definition and benefits*. Medium. Retrieved March 11, 2023, from https://towardsdatascience.com/version-control-101-definition-and-benefits-6fd7ad49e5f1

Sphitz, A. (2022, September 4). *Git clone: How to clone A repo in git*. Initial Commit. Retrieved March 18, 2023, from https://initialcommit.com/blog/git-clone

Vogel, L. (2023, February 2). *Eclipse Git Tutorial*. vogella.com. Retrieved March 20, 2023, from https://www.vogella.com/tutorials/EclipseGit/article.html#git-integration-into-the-package-and-the-project-explorer