

Master's Thesis

City College of New York

David Leonard

Date: May 2, 2016

Contents

1	Introduction	2
2	Version Control Systems	2
3	Collaboration	2
3.1	Building Roadmaps	2

1 Introduction

In the past year at the City College of New York, the Computer Science program has become immensely crowded - particularly the introductory computer science courses. Overcrowding of introductory classes leads to less feedback from Professors at a particularly vulnerable time for new students in which feedback is crucial for understanding the material. On the other hand, in the current generation of software engineering, learning how to use Version Control Systems (VCS) has become an absolute and necessary skill for all students studying computer science to learn. With these two points in mind, how can we leverage VCS to help alleviate the problem of providing feedback to students while at the same time teaching students how to successfully collaborate together? We will explore both of these ideas in this paper and provide a technical solution to address these problems.

2 Version Control Systems

In Software Engineering, we are faced with one common problem - how does a group of individuals successfully collaborate on a project while sharing one codebase? Using a VCS, we can achieve precisely that. Currently, there are two well-known VCS: ¹ git and ² mercurial. These tools allow developers to collaborate on a group of files in a **project** while solving problems such as **communication** and **merge conflicts** between files. These projects are stored as **repositories** in the **cloud** through the use of **collaborative platforms** which are ³ and ⁴. These platforms provide a service which allows developers to communicate, collaborate and control their codebase history. In this paper we will focus exclusively on using git repositories on Bitbucket.

3 Collaboration

One common thing that students tend to do when working on software projects is to collaborate with other students. This allows them to learn from each other while at the same time having someone to bounce ideas off of in hopes of being able to solve their own problems. Since students already collaborate verbally with each other in class, one idea to handling overcrowding in classes would be to group students together. By having closely knit groups, students can learn from one another while at the same time being able to finish their numerous programming assignments.

This brings us to our next question, how exactly do a group of students collaborate on a programming assignment together? We explore several scenarios in the upcoming sections.

3.1 Project Skeletons

A common format for giving

¹Git: <https://git-scm.com/>

²Mercurial: <https://www.mercurial-scm.org/>

³Github: <https://github.com/>

⁴Bitbucket: <https://bitbucket.org/>

References

- [1] David Silver et. al. *Mastering the game of Go with deep neural networks and tree search*. (doi:10.1038/nature16961) January 28, 2016.
- [2] Campbell, M., Hoane, A, Hsu, F. *Deep Blue*. Artif. Intell. 134, 5783 (2002).
- [3] Google Research Blog *What we learned in Seoul with AlphaGo*. <https://goo.gl/bTT19m> March 16, 2016.